

SIMATIC

Programovatelný automat S7-200 Systémový manuál

Předmluva, Obsah	
Úvod do řídicích systémů SIMATIC S7–200	1
Úvod do programování	2
Instalace S7-200	3
Základní principy	4
Pojmy, konvence a možnosti programování	5
Instrukční soubor S7-200	6
Síťová komunikace	7
Návod pro odstraňování poruch hardwaru a ladění softwaru	8
Tvorba programu s polohovacím modulem	9
Tvorba programu pro modemový modul	10
Použití knihovny USS protokolu pro řízení pohonu MicroMaster	11
Použití knihovny protokolu Modbus	12
Technické specifikace	A
Výpočet energetické bilance	B
Chybové kódy	C
Speciální paměťové bity (SM)	D
Objednací čísla S7-200	E
Doby provádění instrukcí STL	F
Stručný přehled informací o S7-200	G

6ES7 298-8FA22-8BH0

Index

Bezpečnostní pokyny

Tento manuál obsahuje upozornění, která byste měli dodržovat, abyste zajistili svou vlastní bezpečnost, ochranu výrobku a k němu připojeného zařízení. Tato upozornění jsou v manuálu zvýrazněna výstražným trojúhelníkem. Podle úrovně nebezpečí jsou označena takto:



Nebezpečí

Nebezpečí znamená bezprostředně nebezpečnou situaci, která bez přijetí adekvátních opatření bude mít za následek smrt nebo vážný úraz.



Varování

Varování znamená potenciálně nebezpečnou situaci, která by bez přijetí adekvátních opatření mohla mít za následek smrt nebo vážný úraz.



Upozornění

Pozor! použito s výstražným bezpečnostním symbolem znamená potenciálně nebezpečnou situaci, která by bez přijetí adekvátních opatření mohla mít za následek lehký nebo střední úraz.

Upozornění

Upozornění! použito bez výstražného bezpečnostního symbolu znamená potenciálně nebezpečnou situaci, která by bez přijetí adekvátních opatření mohla mít za následek poškození majetku.

Poznámka

Znamená možnou situaci, která by bez přijetí adekvátních opatření mohla mít za následek nežádoucí výsledek nebo stav.

Kvalifikovaný personál

Montáž tohoto zařízení smí provádět pouze **kvalifikovaný personál**. Kvalifikované osoby jsou definovány jako ty osoby, které jsou oprávněny zprovozňovat, uzemňovat a opatřovat bezpečnostními tabulkami obvodu, zařízení a systémy v souladu se zavedenými bezpečnostními zvyklostmi a normami.

Správné užívání

Věnujte pozornost následujícímu:



Varování

Toto zařízení a jeho komponenty mohou být používány pouze pro aplikace popsané v katalogu nebo v technických popisech a pouze ve spojení s těmi zařízeními nebo komponentami jiných výrobců, které byly schváleny nebo doporučeny společností Siemens.

Tento výrobek bude fungovat správně a bezpečně pouze tehdy, je-li přepravován, skladován, připraven k provozu a instalován správně a je provozován a udržován podle doporučení.

Ochranné známky

SIMATIC®, SIMATIC HMI® a SIMATIC NET® jsou registrované ochranné známky společnosti SIEMENS AG.

Některé jiné názvy použité v těchto dokumentech jsou také registrované ochranné známky; jejich použití třetí stranou pro její vlastní účely může znamenat porušení vlastnických práv.

Copyright Siemens AG 2002 Všechna práva vyhrazena

Kopírování, rozšiřování nebo použití tohoto dokumentu nebo jeho obsahu bez výslovného písemného souhlasu autora není povoleno. Osoby, které toto poruší, jsou povinny nahradit škody. Všechna práva včetně práv tvořených udělením patentu nebo registrací uživatelského modelu nebo vzoru jsou vyhrazena.

Siemens AG
Automation and Drives (A&D)
Industrial Automation Systems (AS)
Postfach 4848, D- 90327 Nürnberg

Odmítnutí odpovědnosti

Ověřili jsme si shodu obsahu tohoto manuálu s popisovaným hardwarem a softwarem. Protože odchylky není možné úplně vyloučit, nemůžeme garantovat plnou shodu. Údaje v tomto manuálu jsou však pravidelně kontrolovány a eventuelní potřebné opravy jsou zahrnuty do následujících vydání. Návrhy na zlepšení jsou vítány.

© Siemens AG 2002
Změny technických údajů jsou vyhrazeny.

Předmluva

Série S7-200 je řada malých programovatelných automatů (mikro-PLC) určených k řízení v různých automatizačních aplikacích. Díky kompaktnímu designu, nízkým nákladům a výkonnému instrukčnímu souboru je S7-200 perfektním řešením pro řízení malých aplikací. Široká nabídka různých modelů S7-200 a programovací nástroj na bázi Windows pokrývají flexibilitu potřebnou pro řešení vašich automatizačních potřeb.

Okruh čtenářů

Tento manuál poskytuje informace o instalaci a programování mikro-PLC S7-200; je určen pro inženýry, programátory, montážní pracovníky a elektrikáře, kteří mají obecné znalosti o programovatelných automatech.

Rozsah manuálu

Informace, obsažené v tomto manuálu, se týkají zejména následujících výrobků:

- ☐ Modelů CPU S7-200: CPU 221, CPU 222, CPU 224, CPU 226 a CPU 226XM
- ☐ Rozšiřovacích modulů EM 22x pro S7-200
- ☐ STEP 7 – Micro/WIN, verze 3.2, 32bitového programovacího softwarového balíku pro S7-200
- ☐ Knihoven instrukcí STEP 7 – Micro/WIN a TP-Designer pro TP070, verze 1.0, sady softwarových nástrojů pro zákazníky, kteří používají S7-200 s jinými komponentami, jako je Touch Panel TP070, Modbus nebo pohon MicroMaster

Normy, kterým výrobek odpovídá

Řada SIMATIC S7-200 splňuje následující normy:

- ☐ Směrnice Evropského společenství (CE) pro nízké napětí 73/23/EEC
EN 61131 – 2: Programovatelné automaty – Požadavky na zařízení
- ☐ Směrnice Evropského společenství (CE) pro EMC (elektromagnetickou kompatibilitu) 89/336/EEC
Norma pro elektromagnetické emise
EN 50081 – 1: pro obyvatelstvo, komerční použití a lehký průmysl
EN 50081 – 2: průmyslové prostředí
Normy elektromagnetické odolnosti
EN 61000 – 6 – 2: průmyslové prostředí
- ☐ Underwriters Laboratories, Inc.
UL 508 Zahrnuto (Průmyslová řídicí zařízení) Registrační číslo E75310
- ☐ Canadian Standards Association (Kanadské sdružení pro normy): CSA C22.2 číslo 142 schváleno (Zařízení pro řízení procesu)
- ☐ Factory Mutual Research: FM třída I, oddělení 2, nebezpečná místa skupiny A, B, C a D, T4A a třída I, pásmo 2, IIC, T4

Informace o splnění norem jsou v příloze A.

Schválení pro námořní provoz

V době tisku tohoto manuálu odpovídala řada SIMATIC S7-200 požadavkům níže uvedených námořních agentur. Chcete-li zjistit nejnovější schválení výrobku, kontaktujte místního distributora nebo zastoupení Siemens.

Agentura	Číslo certifikátu
Lloyds Register of Shipping (LRS)	99 / 20018(E1)
American Bureau of Shipping (ABS)	01-HG20020-PDA
Germanischer Lloyd (GL)	12 045 - 98 HH
Det Norske Veritas (DNV)	A-8071
Bureau Veritas (BV)	09051 / A2 BV
Nippon Kaiji Kyokai (NK)	A-534

Jak používat tento manuál

Pokud - mikro PLC S7-200 používáte poprvé, měli byste si přečíst celý *Systémový manuál pro programovatelný automat S7-200*. Jste-li zkušený uživatel, stačí vám podle obsahu nebo rejstříku vyhledat konkrétní informace.

Systémový manuál pro programovatelný automat S7-200 je rozdělen do následujících oblastí:

- ☐ Kapitola 1 (Úvod do řídicích systémů SIMATIC S7–200) uvádí přehled některých vlastností řady malých programovatelných automatů S7-200 (micro PLC).
- ☐ Kapitola 2 (Úvod do programování) učí vytvořit vzorový řídicí program pro S7-200 a provést jeho download.
- ☐ Kapitola 3 (Instalace S7-200) obsahuje rozměry a základní návody pro instalaci modulů CPU S7-200 a rozšiřovacích vstup/výstupních modulů.
- ☐ Kapitola 4 (Základní principy) poskytuje informace o provozu S7-200.
- ☐ Kapitola 5 (Pojmy, konvence a možnosti programování) uvádí informace o funkcích STEP 7–Micro/WIN, programových editorech a typech instrukcí (IEC 1131-3 nebo SIMATIC), typech dat S7-200 a návodech pro tvorbu programů.
- ☐ Kapitola 6 (Instrukční soubor S7-200) uvádí popisy a příklady programovacích instrukcí podporovaných S7-200.
- ☐ Kapitola 7 (Síťová komunikace) poskytuje informace pro nastavení různých konfigurací sítě podporovaných automatem S7-200.
- ☐ Kapitola 8 (Návod pro odstraňování poruch hardwaru a softwaru) uvádí informace pro odstranění problémů hardwaru S7-200 a informace o funkcích STEP 7–Micro/WIN, které pomáhají při ladění uživatelského programu.
- ☐ Kapitola 9 (Tvorba programu s polohovacím modulem) uvádí informace o instrukcích a průvodci, které se používají při tvorbě programu pro polohovací modul EM 253.
- ☐ Kapitola 10 (Tvorba programu pro modemový modul) uvádí informace o instrukcích a průvodci, které se používají při tvorbě programu pro modemový modul EM 241.
- ☐ Kapitola 11 (Použití knihovny USS protokolu pro řízení pohonu MicroMaster) poskytuje informace o instrukcích používaných pro tvorbu a řízení programu pro pohon MicroMaster. Informuje také o tom, jak konfigurovat pohony MicroMaster 3 a MicroMaster 4.
- ☐ Kapitola 12 (Použití knihovny protokolu Modbus) uvádí informace o instrukcích používaných pro tvorbu programu, který pro komunikaci používá protokol Modbus.
- ☐ V příloze A (Technické specifikace) jsou technické informace a tabulky s údaji o hardwaru automatu S7-200.

Ostatní přílohy obsahují další rychle dostupné informace, jako jsou popis chybových kódů, popis oblasti speciální paměti (SM), čísla součástí pro objednávání zařízení S7-200 a doby provádění instrukcí STL.

Další informace a asistenční služby

Informace o S7-200 a STEP 7–Micro/WIN

Kromě tohoto manuálu je software STEP 7–Micro/WIN vybaven i rozsáhlou online nápovědou pro uvedení do programování automatu S7-200. Součástí koupě STEP 7–Micro/WIN je i bezplatné CD s dokumentací. Na něm najdete tipy pro použití, elektronickou verzi tohoto manuálu a jiné informace.

Online nápověda

Stačí stisknout klávesu a pomoc je tu! Stisknutím klávesy F1 získáte přístup k rozsáhlé online nápovědě pro STEP 7–Micro/WIN. Online nápověda obsahuje užitečné informace o tom, jak začít programovat S7-200, a mnoho dalších témat.

Elektronický manuál

Na dokumentačním CD je k dispozici elektronická verze tohoto Systémového manuálu pro S7-200. Můžete ji nahrát do svého počítače, čímž získáte tak snadný přístup k informacím obsaženým v manuálu při práci se softwarem STEP 7–Micro/WIN.

Tipy a triky

CD s dokumentací obsahuje Tipy a triky, sadu příkladů použití se vzorovými programy. Posouzení nebo úprava těchto příkladů může pomoci nalézt účinná nebo úplně nová řešení pro aplikaci. Nejaktuálnější verzi Tipů a triků také můžete najít na internetové stránce S7-200.

Internet: www.siemens.com/S7-200

Další informace o výrobcích a službách poskytovaných společností Siemens, technické podpore, často kladených dotazech (FAQ), aktualizaci výrobků nebo tipech pro používání najdete na následujících internetových adresách:

- ☐ www.ad.siemens.de *pro obecné informace společnosti Siemens*
Tato internetová stránka Siemens Automation & Drives (Automatizace a pohony Siemens) obsahuje informace o řadě výrobků SIMATIC i jiných výrobcích nabízených touto společností.
- ☐ www.siemens.com/S7-200 *informace o výrobku S7-200*
Internetová stránka S7-200 obsahuje často kladené dotazy (FAQ), Tipy a triky (příklady použití a vzorové programy), informace o nově nabízených výrobcích a aktualizace nebo download pro výrobky.
- ☐ www.siemens.cz/micro *česká stránka o mikrosystémech*

Technická pomoc a nákup výrobků S7-200

Místní zastoupení nebo distributor Siemens

Pro pomoc při řešení jakýchkoliv technických otázek, školení pro výrobky S7-200 a objednávky výrobků S7-200 kontaktujte místního distributora nebo zastoupení Siemens. Protože jsou obchodní zástupci technicky vyškolení a mají konkrétní znalosti o vašem provozu, technologii a průmyslovém odvětví i o jednotlivých výrobcích Siemens, které používáte, mohou vám dát nejrychlejší a nejefektivnější odpovědi na jakékoliv problémy, se kterými byste se mohli setkat.

Technické služby

K dispozici je také vysoce kvalifikovaný personál Centra technických služeb S7-200, který pomůže vyřešit jakýkoliv problém, se kterým byste se mohli setkat. Můžete volat celých 24 hodin, 7 dnů v týdnu:

- ☐ *Pro hovory ze Spojených států amerických*
Místní čas: Pondělí až pátek 08.00 až 19.00 východoamerického času
Telefon: +1 800 241–4453
Fax: +1 (0) 770 740–3699
E-mail: drives.support@sea.siemens.com
- ☐ *Pro hovory ze Severní a Jižní Ameriky mimo USA*
Místní čas: Pondělí až pátek 08.00 až 19.00 východoamerického času
Telefon: +1 (0) 770 740–3505
Fax: +1 (0) 770 740–3699
E-mail: drives.support@sea.siemens.com
- ☐ *Pro hovory z Evropy a Afriky*
Místní čas (Norimberk): Pondělí až pátek 07.00 až 17.00
Telefon: +49 (0) 180 5050–222
Fax: +49 (0) 180 5050–223
E-mail: techsupport@ad.siemens.de
- ☐ *Pro hovory z Asie a Austrálie*
Místní čas (Singapur): Pondělí až pátek 08.30 až 17.30
Telefon: +65 (0) 740–7000
Fax: +65 (0) 740–7001
E-mail: drives.support@sae.siemens.com.sg
- ☐ *Pro hovory z České Republiky*
Místní čas (ČR): Pondělí až pátek 08.30 až 18.00
Telefon: +420 541 191 742
Fax: +420 541 191 749
E-mail: micro@rg.siemens.cz

Obsah

1	Úvod do řídicích systémů SIMATIC S7–200	1
	CPU S7-200	2
	Rozšiřovací moduly S7-200	3
	Programovací balík STEP 7–Micro/WIN	3
	Možnosti komunikace	4
	Operátorské panely	4
2	Úvod do programování	5
	Připojení S7-200 CPU	6
	Vytvoření vzorového programu	8
	Download vzorového programu	11
	Uvedení S7-200 do režimu RUN	11
3	Instalace S7-200	13
	Návod pro instalaci zařízení S7-200	14
	Montáž a demontáž modulů S7-200	15
	Návod pro uzemnění a elektrické zapojení	18
4	Základní principy	21
	Jak S7-200 vykonává řídicí program	22
	Přístup k datům	24
	Ukládání a obnova dat	34
	Uložení programu do paměťového modulu	36
	Výběr pracovního režimu CPU S7-200	37
	Program pro uložení paměti V do EEPROM	38
	Vlastnosti automatů řady S7-200	39
5	Pojmy, konvence a možnosti programování	47
	Návod pro návrh mikro–PLC systému	48
	Základní prvky programu	49
	Použití STEP 7–Micro/WIN pro vytvoření programu	51
	Volba mezi instrukčním souborem SIMATIC a IEC 1131–3	53
	Vysvětlení konvencí použitých v programových editorech	54
	Použití průvodců jako pomůcky pro tvorbu uživatelského řídicího programu	56
	Ošetření chyb	56
	Přiřazení adres a počátečních hodnot v editoru datového bloku	58
	Použití tabulky symbolů pro symbolické adresování proměnných	58
	Použití lokálních proměnných	59
	Použití stavového diagramu pro monitorování uživatelského programu	59
	Vytvoření knihovny instrukcí	60
	Funkce pro ladění uživatelského programu	60

6	Instrukční soubor S7-200	61
	Konvence použité při popisu instrukcí	63
	Rozsahy a vlastnosti paměti S7-200	64
	Instrukce bitové logiky	66
	Kontakty	66
	Cívky	68
	Práce se zásobníkem	70
	Bistabilní klopný obvod	72
	Instrukce hodin	73
	Instrukce pro komunikaci	74
	Čtení a zápis do sítě	74
	Vysílání a příjem (Freeport)	79
	Zjištění a nastavení adresy portu	88
	Instrukce porovnávání	89
	Porovnání číselných hodnot	89
	Porovnání řetězce	91
	Instrukce pro konverzi	92
	Standardní konverze	92
	Konverze na ASCII	96
	Konverzi řetězců	100
	Zakódování a dekodování	105
	Instrukce čítače	106
	Čítač SIMATIC	106
	Čítač IEC	109
	Instrukce vysokorychlostního čítače	111
	Instrukce pulzního výstupu	125
	Matematické instrukce	140
	Sčítání, odčítání, násobení a dělení	140
	Násobení celých čísel s 32bitovým výsledkem a dělení celého čísla se zbytkem	142
	Numerických funkce	143
	Inkrementace a dekrementace	144
	Instrukce PID algoritmu	145
	Instrukce přerušení	155
	Instrukce logické operace	162
	Inverze	162
	AND, OR a XOR	163
	Instrukce přesunu	165
	Přesun bytu, word, double word nebo real	165
	Okamžitý přesun bytu (čtení a zápis)	166
	Přesun bloku	167
	Řízení programu	168
	Podmíněný konec programu	168
	Stop	168
	Reset Watchdogu	168
	Smyčka For–Next	170
	Skokové instrukce	172
	Sekvenčního řídicí relé (SCR)	173

Instrukce posunu a rotace	179
Posun vpravo a vlevo	179
Rotace vpravo a vlevo	179
Posunutí bitů registru	181
Výměna bytů	183
Instrukce pro řetězcce	184
Instrukce pro práci s tabulkou	189
Přidávání do tabulky	189
Instrukce FIFO a LIFO	190
Vyplnění paměti	192
Prohledávání tabulky	193
Instrukce časovače	196
Časovač SIMATIC	196
Časovač IEC	201
Instrukce podprogramů	203
7 Sítová komunikace	207
Vysvětlení základů sítové komunikace S7-200	208
Výběr sítového komunikačního protokolu	211
Instalace a odstranění komunikačních rozhraní	216
Návrh sítě	218
Vytvoření protokolů definovaných uživatelem v režimu Freeport	222
Použití modemu a STEP 7–Micro/WIN	224
Pro pokročilé	228
8 Návod pro odstraňování poruch hardwaru a nástroje pro ladění softwaru	235
Funkce pro ladění uživatelského programu	236
Zobrazení stavu programu	238
Použití stavové tabulky pro monitorování a modifikaci dat	239
Force (vnucení) specifických hodnot	240
Zadání běhu programu po určený počet programových cyklů	240
Návod pro odstraňování poruch hardwaru	241
9 Tvorba programu pro polohovací modul	243
Funkce polohovacího modulu	244
Konfigurace polohovacího modulu	246
Polohovací instrukce vytvořené pomocí "Průvodce polohováním"	257
Vzorové programy	269
Monitorování polohovacího modulu "Control panelem pro EM 253"	274
Chybové kódy pro polohovací modul a polohovací instrukce	276
Témata pro pokročilé	278

10	Tvorba programu pro modemový modul	287
	Funkce modemového modulu	288
	Použití průvodce pro konfiguraci modemového modulu	294
	Přehled instrukcí a omezení pro modem	298
	Instrukce pro modemový modul	299
	Vzorový program	303
	CPU S7-200 podporující inteligentní moduly	303
	Byty speciální paměti pro modemový modul	304
	Témata pro pokročilé	306
	Formát telefonního čísla pro zprávu	308
	Formát textové zprávy	309
	Formát zprávy pro přenos dat CPU	310
11	Použití knihovny knihovny USS protokolu pro řízení pohonu MicroMaster	311
	Požadavky použití USS protokolu	312
	Výpočet doby, potřebné pro komunikaci s pohonem	313
	Používání instrukcí USS	314
	Instrukce USS protokolu	315
	Vzorové programy USS protokolu	322
	Chybové kódy při běhu USS	323
	Připojení a nastavení pohonu MicroMaster řady 3	324
	Připojení a nastavení pohonu MicroMaster řady 4	327
12	Užívání knihovny protokolu Modbus	329
	Požadavky použití protokolu Modbus	330
	Inicializace a doba provádění protokolu Modbus	330
	Adresování Modbus	331
	Užívání instrukcí protokolu Modbus slave	332
	Instrukce protokolu Modbus slave	333
A	Technická specifikace	337
	Obecné technické specifikace	338
	Specifikace CPU	340
	Specifikace digitálních rozšiřovacích modulů	346
	Specifikace analogových rozšiřovacích modulů	351
	Specifikace termočlávkových a RTD rozšiřovacích modulů	361
	Specifikace modulu EM 277 PROFIBUS–DP	373
	Specifikace modemového modulu EM 241	385
	Specifikace polohovacího modulu EM 253	387
	Specifikace modulu AS–Interface (CP 243–2)	393
	Volitelné zásuvné moduly	395
	Kabel pro rozšiřovací moduly	395
	PC/PPI kabel	396
	Simulátory vstupu	398
B	Výpočet energetické bilance	399

C	Chybové kódy	403
	Kódy a hlášení kritických chyb	404
	Problémy programování při běhu programu	405
	Porušení pravidel kompilace	406
D	Speciální paměťové bity (SM)	407
	SMB0: Stavové bity	408
	SMB1: Stavové bity	408
	SMB2: Znak přijatý v režimu Freeport	409
	SMB3: Chyba parity v režimu Freeport	409
	SMB4: Přeplnění fronty	409
	SMB5: Stav vstupů a výstupů	410
	SMB6: Registr ID CPU	410
	SMB7: Rezervován	410
	SMB8 až SMB21: ID a chybové registry I/O modulu	411
	SMW22 až SMW26: Délka programového cyklu	412
	SMB28 a SMB29: Analogové potenciometry	412
	SMB30 a SMB130: Řídící registry pro Freeport	412
	SMB31 a SMW32: Řízení zápisu do trvalé paměti (EEPROM)	413
	SMB34 a SMB35: Registry časového intervalu pro časovaná přerušení	413
	SMB36 až SMB65: Registr HSC0, HSC1 a HSC2	413
	SMB66 až SMB85: Registry PTO/PWM	415
	SMB86 až SMB94 a SMB186 až SMB194: Řízení příjmu zpráv	416
	SMW98: Chyby na rozšiřovací I/O sběrnici	417
	SMB130: Řídící registr pro Freeport (viz SMB30)	417
	SMB131 až SMB165: Registr HSC3, HSC4 a HSC5	417
	SMB166 až SMB185: Tabulka definice profilu PTO0, PTO1	418
	SMB186 až SMB194: Řízení příjmu zpráv (viz SMB86 až SMB94)	418
	SMB200 až SMB549: Stav inteligentních modulů	419
E	Objednací čísla S7-200	421
F	Doby provádění instrukcí STL	425
G	Stručný přehled informací o S7-200	431
	Rejstřík	437

Úvod do řídicích systémů SIMATIC S7–200



SIMATIC S7-200 je řada malých programovatelných automatů (mikro–PLC), určených k řízení v různých automatizačních aplikacích.

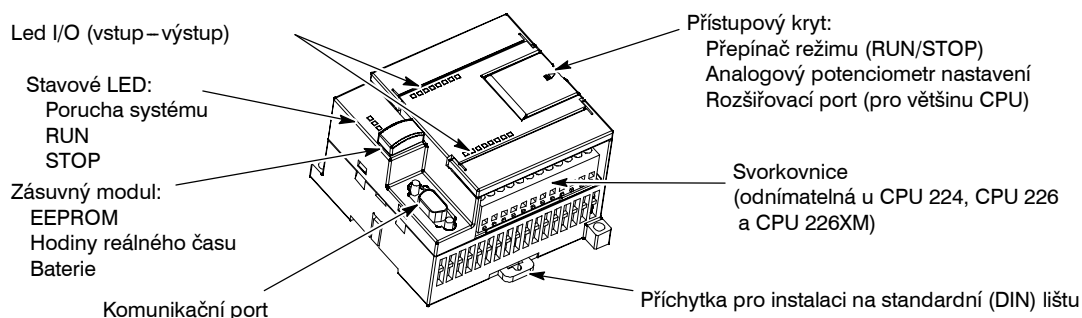
Zařízení S7-200 monitoruje vstupy a řídí výstupy pomocí uživatelského programu, který může obsahovat Booleovu logiku, počítání, časování, složité matematické operace a komunikaci s jinými inteligentními zařízeními. Kompaktní design, flexibilní konfigurace a výkonný instrukční soubor jsou důvody, proč je zařízení S7-200 optimálním řešením pro řízení široké škály aplikací.

Přehled kapitoly

CPU S7-200	2
Rozšiřovací moduly S7-200	3
Programovací balík STEP 7–Micro/WIN	3
Možnosti komunikace	4
Operátorské panely	4

CPU S7-200

S7-200 obsahuje mikroprocesor, integrovaný napájecí zdroj, vstupní a výstupní obvody v kompaktním pouzdru, které tak tvoří výkonný programovatelný automat (mikro-PLC). Viz obr. 1-1. Po downloadu vašeho programu bude S7-200 obsahovat logiku, potřebnou k monitorování a řízení vstupních a výstupních zařízení aplikace.



Obr. 1-1 S7-200 Mikro-PLC

Siemens dodává různé modely S7-200 s různými parametry a schopnostmi, které pomáhají vytvořit efektivní řešení pro vaše aplikace. Tabulka 1-1 stručně srovnává některé parametry CPU. Podrobné informace o konkrétním CPU jsou uvedeny v příloze A.

Tabulka 1-1 Porovnání modelů S7-200 CPU

Parametr	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Rozměry (mm)	90 x 80 x 62	90 x 80 x 62	120,5 x 80 x 62	190 x 80 x 62	190 x 80 x 62
Paměť pro program	2048 wordů	2048 wordů	4096 wordů	4096 wordů	8192 wordů
Paměť pro data	1024 wordů	1024 wordů	2560 wordů	2560 wordů	5120 wordů
Zálohování dat	standardně 50 hodin	standardně 50 hodin	standardně 190 hodin	standardně 190 hodin	standardně 190 hodin
Integrované I/O	6 vst/4 výst	8 vst/6 výst	14 vst/10 výst	24 vst/16 výst	24 vst/16 výst
Rozšiřovací moduly	0	2	7	7	7
Vysokorychlostní čítače					
Jednofázové	4 (30 kHz)	4 (30 kHz)	6 (30 kHz)	6 (30 kHz)	6 (30 kHz)
Dvoufázové	2 (20 kHz)	2 (20 kHz)	4 (20 kHz)	4 (20 kHz)	4 (20 kHz)
Pulzní výstupy (DC)	2 (20 kHz)	2 (20 kHz)	2 (20 kHz)	2 (20 kHz)	2 (20 kHz)
Analogové potenciometry	1	1	2	2	2
Hodiny reálného času	Zásuvný modul	Zásuvný modul	Zabudované	Zabudované	Zabudované
Komunikační porty	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
Matematika s pohyblivou řádovou čárkou	Ano				
Velikost registru obrazu digitálních I/O	256 (128 vst, 128 výst)				
Prováděcí rychlost Booleovských instrukcí	0.37 mikrosekund/instrukce				

Instalace STEP 7–Micro/WIN

Vložte CD STEP 7–Micro/WIN do mechaniky CD-ROM počítače. Automaticky se spustí pomocník instalace, který vás provede instalačním procesem. Další informace o instalaci STEP 7–Micro/WIN získáte v souboru "Readme".



Tip

K instalaci STEP 7–Micro/WIN na operační systém Windows NT nebo Windows 2000 se musíte přihlásit s oprávněním administrátora.

Možnosti komunikace

Siemens nabízí dvě programovací možnosti připojení počítače k S7-200: přímé propojení PC/PPI kabelem nebo kartou komunikačního procesoru (CP) s MPI kabelem pro síť MPI a PROFIBUS–DP.

PC/PPI programovací kabel je nejběžnější a nejekonomičtější způsob připojení počítače k S7-200. Tímto kabelem se propojí komunikační port S7-200 se sériovým portem počítače. Programovací kabel PC/PPI se může rovněž použít pro připojení dalších zařízení k S7-200.

Abyste mohli použít MPI kabel, musíte nainstalovat do počítače CP kartu. Ta poskytuje další hardware nutný pro připojení při vyšších přenosových rychlostech a ke zvládnutí vysokorychlostní síťové komunikace.

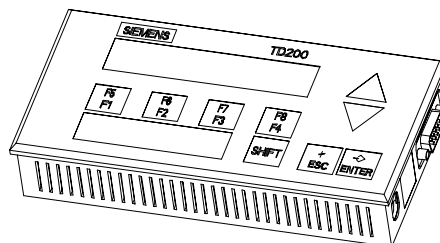
Operátorské panely

Textový displej TD 200

TD 200 je textové zobrazovací zařízení o 2 řádcích a 20 znacích, které může být připojeno k S7-200. Pomocí průvodce můžete TD 200 snadno programovat a zobrazovat textové zprávy a jiná data, týkající se aplikace.

TD 200 poskytuje základní rozhraní tím, že umožňuje sledovat a měnit proměnné procesu.

Samostatný manuál popisuje podrobnou funkci a technické specifikace TD 200.



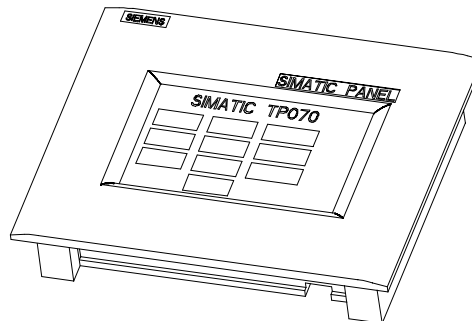
Obr. 1-3 Textový displej TD 200

Touch panel TP070

TP070 je dotykem ovládaná zobrazovací jednotka, která může být připojena k S7-200. Tato jednotka umožňuje vytvořit vlastní uživatelské prostředí.

TP070 může zobrazovat uživatelskou grafiku, sloupcové grafy, aplikační proměnné, uživatelská tlačítka a další prvky pomocí uživatelsky příjemného dotykového panelu.

Doplňkové CD TP–Designer pro TP070 obsahuje software TP Designer, který je nutný pro programování TP070.



Obr. 1-4 Touch panel TP070

Úvod do programování



STEP 7 – Micro/WIN umožňuje snadno naprogramovat systém S7-200. Pouze v několika krátkých krocích se můžete pomocí jednoduchého příkladu naučit, jak připojit, naprogramovat a provozovat S7-200.

Pro tento příklad potřebujete pouze PC/PPI kabel, CPU S7-200 a počítač se softwarem STEP 7 – Micro/WIN.

Přehled kapitoly

Připojení S7-200 CPU	6
Vytvoření vzorového programu	8
Download vzorového programu	11
Uvedení S7-200 do režimu RUN	11

Připojení CPU S7-200

Připojení S7-200 je snadné. Pro tento příklad je potřeba jen připojit napájení k CPU S7-200 a pak připojit komunikační kabel mezi programovací zařízení a S7-200.

Připojení napájení k S7-200 CPU

Prvním krokem je připojení S7-200 k napájecímu zdroji. Obrázek 2-1 ukazuje zapojení pro stejnosměrný nebo střídavý model S7-200.

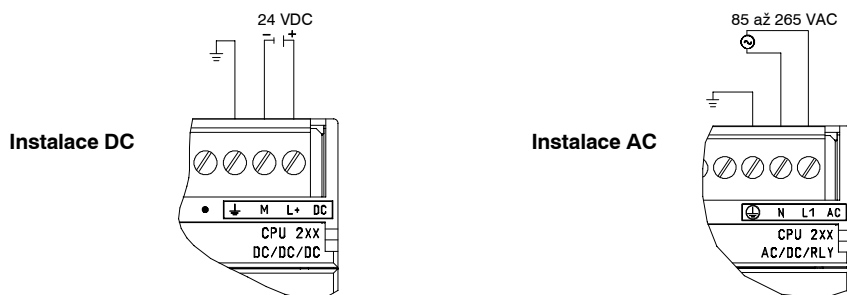
Před montáží nebo demontáží elektrického zařízení zajistěte, aby bylo odpojeno napájení. Vždy dodržujte příslušné bezpečnostní předpisy. Zajistěte, aby bylo napájení S7-200 vypnuto před pokusem instalovat nebo demontovat jednotku S7-200.



Varování

Pokusy instalovat nebo zapojovat S7-200 nebo připojená zařízení se zapnutým napájením by mohly způsobit úraz elektrickým proudem nebo chybný chod zařízení. Nevypnutí napájení systému S7-200 a připojených zařízení během instalace nebo demontáže by mohlo mít za následek smrt nebo vážné zranění osob a poškození zařízení.

Vždy dodržujte příslušné bezpečnostní předpisy a zajistěte, aby bylo napájení S7-200 vypnuto před pokusem instalovat nebo demontovat jednotku S7-200 nebo připojená zařízení.

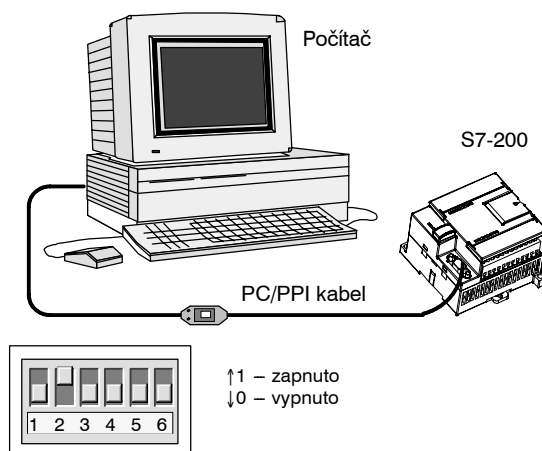


Obr. 2-1 Připojení napájení S7-200

Připojení PC/PPI kabelu

Na obr. 2-2 je kabel PC/PPI, kterým se připojí S7-200 k počítači. Připojení PC/PPI kabelu:

1. Připojte konektor RS-232 (označený "PC") PC/PPI kabelu ke komunikačnímu portu počítače. (Pro tento příklad jej připojte na COM 1.)
2. Připojte konektor RS-485 (označený "PPI") PC/PPI kabelu k portu 0 nebo k portu 1 jednotky S7-200.
3. Zajistěte, aby DIP přepínače PC/PPI kabelu byly nastaveny tak, jak je vyobrazeno na obr. 2-2.



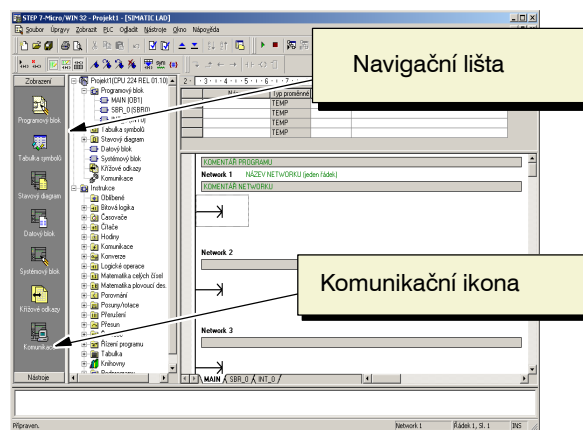
Obr. 2-2 Připojení PC/PPI kabelu

Spuštění STEP 7–Micro/WIN

Kliknutím na ikonu STEP 7–Micro/WIN otevřete nový projekt, který je zobrazen na obr. 2-3.

Všimněte si navigační lišty. K otevření prvků projektu STEP 7–Micro/WIN můžete použít ikony na navigační liště.

Klikněte na komunikační ikonu v navigační liště; zobrazí se tak komunikační dialogové okno. Pomocí tohoto dialogového okna můžete nastavit komunikaci pro STEP 7–Micro/WIN.



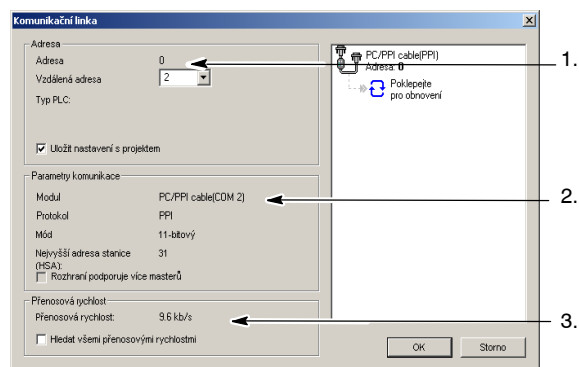
Obr. 2-3 Nový projekt STEP 7–Micro/WIN

Ověření parametrů komunikace pro STEP 7–Micro/WIN

Vzorový projekt používá implicitní nastavení pro STEP 7–Micro/WIN a PC/PPI kabel. Ověření těchto nastavení:

1. Ověřte, zda je v komunikačním dialogovém okně nastavena adresa PC/PPI kabelu na 0.
2. Ověřte, zda je síťové rozhraní nastaveno na PC/PPI kabel (COM1).
3. Ověřte, zda je přenosová rychlost nastavena na 9,6 kb/s.

Pokud potřebujete změnit nastavení parametrů komunikace, přejděte na Kapitulu 7.



Obr. 2-4 Ověření parametrů komunikace

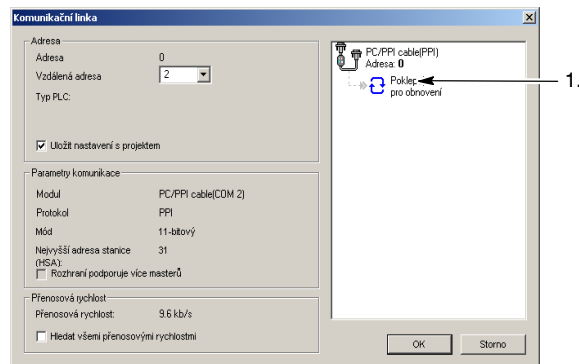
Navázání komunikace s S7-200

Pro připojení S7-200 použijte komunikační dialogové okno:

1. V komunikačním dialogovém okně klikněte dvakrát na obnovovací ikonu.
STEP 7–Micro/WIN vyhledá stanici S7-200 a zobrazí ikonu CPU pro připojenou stanici S7-200.
2. Zvolte S7-200 a klikněte na OK.

Pokud STEP 7–Micro/WIN nenajde jednotku S7-200, zkontrolujte nastavení parametrů komunikace a opakujte postup.

Po navázání komunikace s S7-200 je možné vytvořit a spustit download vzorového programu.

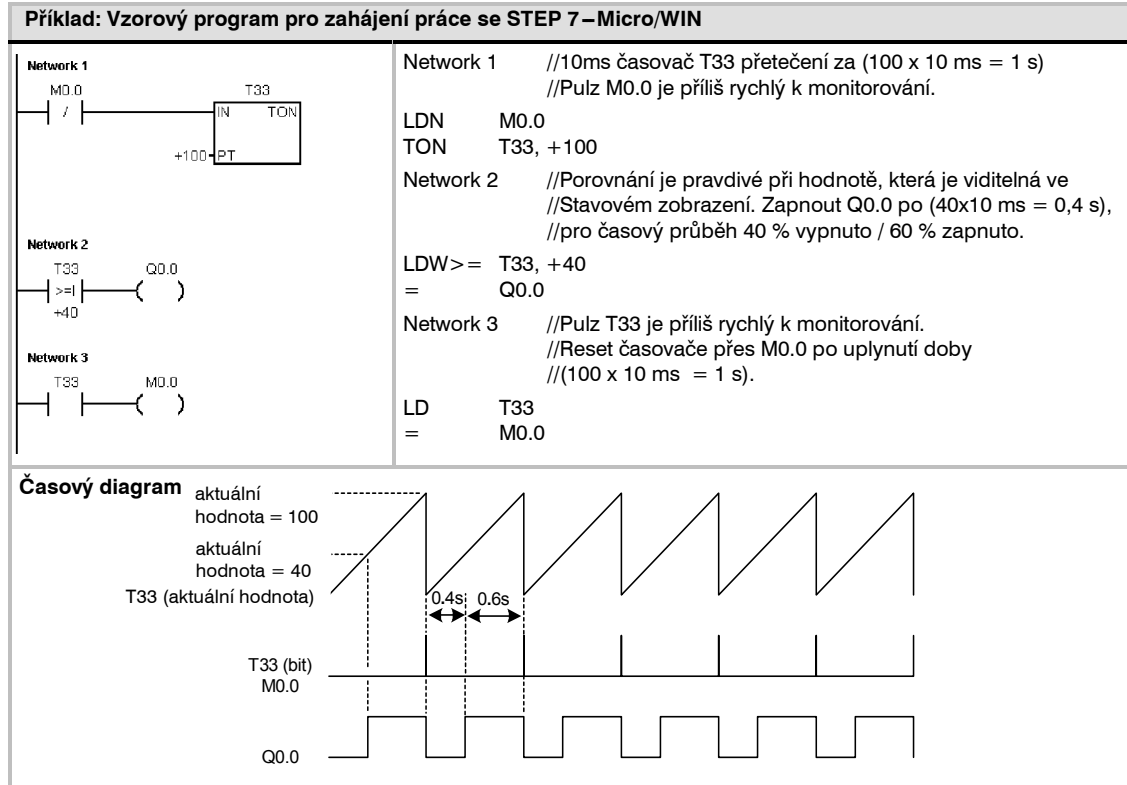


Obr. 2-5 Navázání komunikace s S7-200

Vytvoření vzorového programu

Tento příklad řídicího programu pomáhá pochopit, jak snadné je používat STEP 7–Micro/WIN. Tento program používá šest instrukcí ve třech networkách k vytvoření velmi jednoduchého samospouštěcího časovače, který se sám nuluje.

Pro tento příklad použijte editor LADDER (LAD) k zadání instrukcí k programu. Následující příklad zobrazuje úplný program v LAD i v STL. Komentář networku v programu STL vysvětluje logiku pro každý network. Diagram časování zobrazuje chod programu.



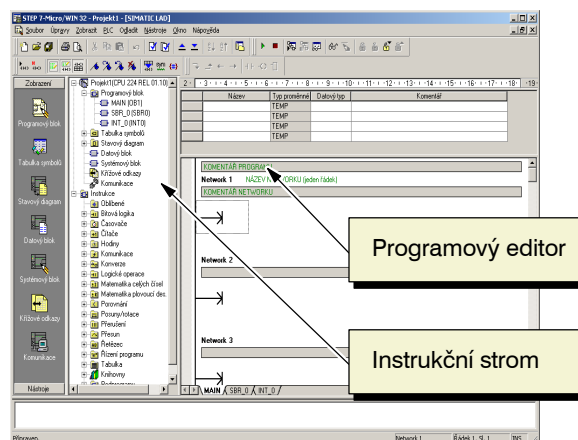
Otevření programového editoru

Kliknutím na ikonu programového bloku otevřete programový editor. Viz obr. 2-6.

Všimněte si instrukčního stromu a programového editoru. Instrukční strom se používá k vložení LAD instrukcí do networků programového editoru tažením a puštěním instrukcí z instrukčního stromu.

Ikony nástrojové lišty poskytují zkrácené povely k příkazům menu.

Po vložení a uložení programu můžete provést download programu do S7-200.

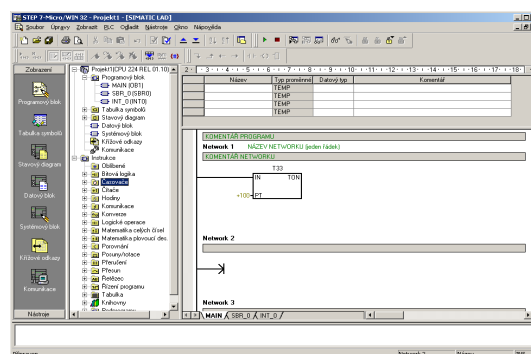


Obr. 2-6 Okno STEP 7–Micro/WIN

Zadání networku 1: spuštění časovače

Když je M0.0 vypnut (0), kontakt sepne a dá povel ke spuštění časovače. Zadání kontaktu pro M0.0:

1. Pro zobrazení instrukcí bitové logiky dvakrát klikněte na ikonu Bitová logika, nebo na znaménko plus (+) .
2. Zvolte v klidu zavřený kontakt.
3. Držte stisknuté levé tlačítko myši a přesuňte kontakt na první network.
4. Klikněte na “???” nad kontaktem a zadejte tuto adresu: M0.0
5. Stiskněte Enter pro uložení adresy kontaktu.



Obr. 2-7 Network 1

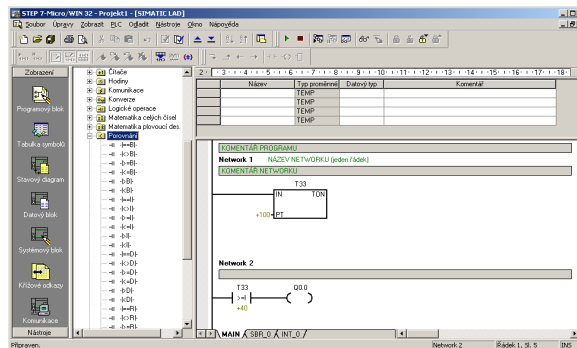
Zadání instrukce časovače T33:

1. Pro zobrazení instrukcí dvakrát klikněte na ikonu časovačů.
2. Zvolte TON (Časovač zpožděného zapnutí).
3. Držte stisknuté levé tlačítko myši a přesuňte časovač na první network.
4. Klikněte na “???” nad oknem časovače a zadejte číslo časovače: T33
5. Stiskněte Enter pro vložení čísla časovače a posuňte zobrazení na parametr nastaveného času (PT).
6. Zadejte hodnotu pro nastavený čas: 100
7. Stiskněte Enter pro uložení hodnoty.

Zadání networku 2: Zapnutí výstupu

Pokud je hodnota časovače T33 vyšší nebo rovna 40 (40krát 10 milisekund neboli 0,4 sekundy), kontakt dá povel pro sepnutí výstupu Q0.0 jednotky S7-200. Instrukce porovnání:

1. Dvakrát klikněte na ikonu pro zobrazení porovnávacích instrukcí. Zvolte instrukci $>=I$ (větší než nebo roven integer).
2. Držte stisknuté levé tlačítko myši a přesuňte instrukci na první network.
3. Klikněte na "???" nad kontaktem a zadejte adresu časovače: T33
4. Pro vložení adresy stiskněte Enter; posuňte zobrazení na druhou hodnotu, která má být srovnávána s hodnotou časovače.
5. Zadejte hodnotu srovnání s hodnotou časovače: 40
6. Stiskněte Enter pro uložení hodnoty.



Obr. 2-8 Network 2

Zadání instrukce pro zapnutí výstupu Q0.0:

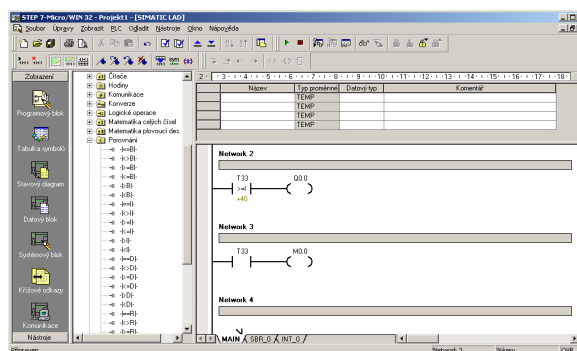
1. Pro zobrazení instrukcí bitové logiky dvakrát klikněte na ikonu Bitové logiky a zvolte výstup.
2. Držte stisknuté levé tlačítko myši a přesuňte výstup na druhý network.
3. Klikněte na "???" nad relé a zadejte adresu: Q0.0
4. Stiskněte Enter pro vložení adresy výstupního bitu.

Zadání networku 3: Nulování časovače

Jakmile časovač dosáhne přednastavenou hodnotu (100) a nastaví bit časovače, sepne se kontakt T33. Signál z tohoto kontaktu nastaví paměťové místo M0.0. Protože je časovač normálně zapnut při zavření kontaktu M0.0, vynuluje časovač změnu stavu M0.0 z vypnuto (0) na zapnuto (1).

Vložení kontaktu pro bit časovače T33:

1. Z instrukcí bitové logiky zvolte kontakt v klidu otevřen.
2. Držte stisknuté levé tlačítko myši a přesuňte kontakt na třetí network.
3. Klikněte na "???" nad kontaktem a zadejte adresu časovače: T33
4. Stiskněte Enter pro uložení hodnoty.



Obr. 2-9 Network 3

Vložení kontaktu pro nastavení M0.0:

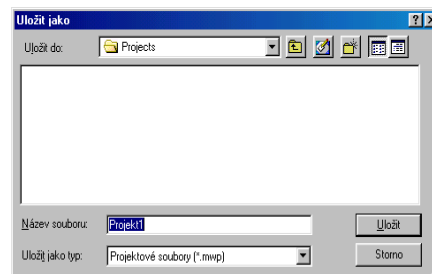
1. Zvolte výstupní kontakt z instrukcí bitové logiky.
2. Držte stisknuté levé tlačítko myši a přesuňte výstup na třetí network.
3. Dvakrát klikněte na "???" nad kontaktem a zadejte tuto adresu: M0.0
4. Stiskněte Enter pro uložení hodnoty.

Uložení vzorového projektu

Zadáním instrukcí tří networků jste ukončili zadávání programu. Uložení programu vytvoříte projekt, který obsahuje typ jednotky S7-200 a další parametry. Uložení projektu:

1. Z menu zvolte příkaz **Soubor > Uložit jako**.
2. Zadejte název projektu v dialogovém okně Uložit jako.
3. Kliknutím na Uložit projekt uložíte.

Po uložení projektu můžete provést download programu do S7-200.



Obr. 2-10 Uložení vzorového programu

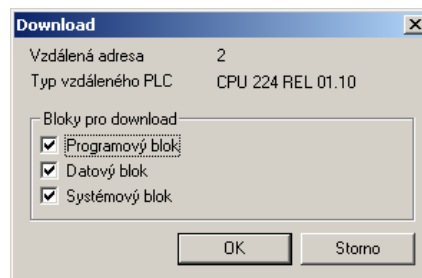
Download vzorového programu



Tip

Každý projekt STEP 7–Micro/WIN je spojen s typem CPU (CPU 221, CPU 222, CPU 224, CPU 226 nebo CPU 226XM). Pokud typ projektu nesouhlasí s CPU, STEP 7–Micro/WIN oznámí neshodu a žádá o zásah. Pokud se tak stane, zvolte "Pokračování downloadu".

1. Pro provedení downloadu programu klikněte na ikonu Download na nástrojové liště nebo zvolte příkaz menu **Soubor > Download**. Viz obr. 2-11.
2. Kliknutím na OK provedete download prvků programu do S7-200.



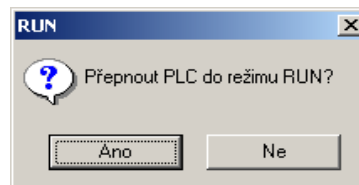
Obr. 2-11 Download programu

Pokud je S7-200 v režimu RUN, dialogové okno vás vyzve k přepnutí S7-200 do režimu STOP. Kliknutím na Ano přepnete S7-200 do režimu STOP.

Uvedení S7-200 do režimu RUN

Aby STEP 7–Micro/WIN uvedl S7-200 do režimu RUN, musí být přepínač režimu S7-200 nastaven na TERM nebo RUN. Jakmile uvedete S7-200 do režimu RUN, začne S7-200 vykonávat program:

1. Klikněte na ikonu RUN na nástrojové liště nebo zvolte příkaz menu **PLC > RUN**.
2. Kliknutím na OK změníte provozní režim S7-200.



Obr. 2-12 Uvedení S7-200 do režimu RUN

Když přejde S7-200 do režimu RUN, výstupní LED se bude pro Q0.0 zapínat a vypínat podle běhu programu v S7-200.

Blahopřejeme! Právě jste dokončili váš první program S7-200.

Program můžete sledovat volbou příkazu menu **Odladit > Stav programu**. STEP 7–Micro/WIN zobrazí hodnoty pro instrukce. K zastavení programu uveďte S7-200 do režimu STOP kliknutím na ikonu STOP nebo volbou příkazu menu **PLC > STOP**.

Instalace S7-200



Zařízení S7-200 je navrženo tak, aby byla jeho instalace snadná. K připevnění modulů na panel můžete použít montážní otvory nebo vestavěné klipsy k upevnění modulů na standardní (DIN) lištu. Malá velikost S7-200 umožňuje efektivně využít prostor.

Tato kapitola poskytuje návod pro instalaci a zapojení systému S7-200.

Přehled kapitoly

Návod pro instalaci zařízení S7-200	14
Montáž a demontáž modulů S7-200	15
Návod pro uzemnění a elektrické zapojení	18

Návod pro montáž zařízení S7-200

Zařízení S7-200 umístěte na panel nebo na standardní lištu a namontujte jej horizontálně nebo vertikálně.

Chraňte zařízení S7-200 před teplem, vysokým napětím a elektrickým rušením.

Pro umístění zařízení v systému platí všeobecné pravidlo: vždy je nutné oddělit zařízení, která vytvářejí vysoké napětí nebo velké elektrické rušení, od nízkonapěťových a logických zařízení, jako je S7-200.

Při instalaci systému S7-200 do panelu vezměte v úvahu zařízení vytvářející teplo; elektronická zařízení umístěte do chladnějších částí rozvaděče. Provoz jakéhokoli elektronického zařízení v prostředí s vysokou teplotou zkrátí dobu bezporuchového provozu.

Zvažte rovněž položení kabeláže pro zařízení umístěné v panelu. Neumísťujte nízkonapěťové signálové vodiče a komunikační kabely do stejné lišty s napájecími kabely střídavého napětí a vysokonapěťovými vodiči stejnosměrného napětí, které jsou rychle spínány.

Zajistěte přiměřený prostor pro chlazení a kabeláž

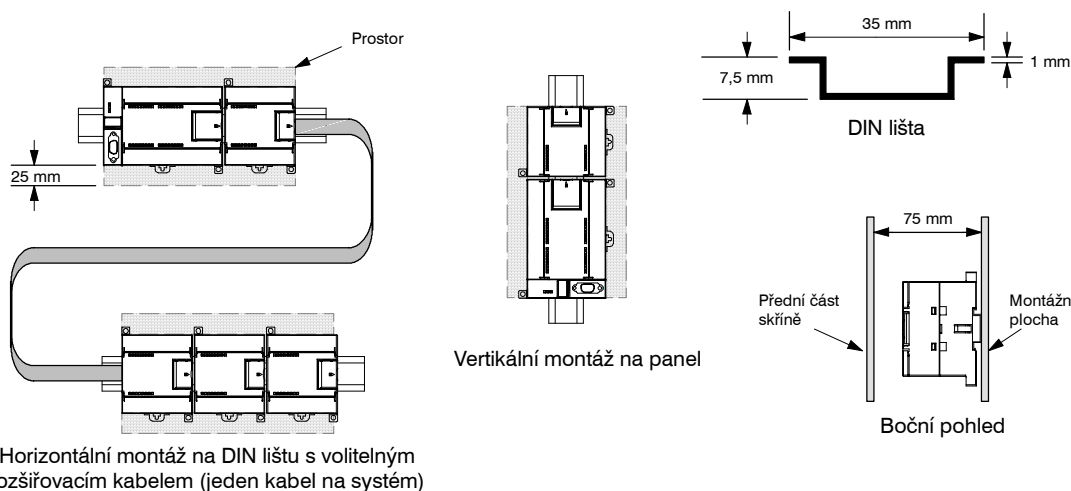
Zařízení S7-200 jsou určena pro přirozené konvekční chlazení. Kvůli řádnému chlazení musíte zajistit prostor asi 25 mm nad a pod zařízeními. Navíc ponechte montážní hloubku alespoň 75 mm.



Tip

Při vertikální montáži je maximální povolená teplota okolí snížena o 10 °C. CPU S7-200 namontujte pod rozšiřovací moduly.

Při plánování umístění systému ponechte dostatek prostoru pro kabeláž a konektory komunikačních kabelů. Pro flexibilní konfiguraci rozmístění systému S7-200 použijte rozšiřovací kabel.



Obr. 3-1 Montážní metody, orientace a prostor

Energetická bilance

Všechny CPU S7-200 mají vnitřní napájecí zdroje, jimiž jsou napájeny CPU, rozšiřovací moduly a další uživatelská zařízení pro napětí 24 VDC.

S7-200 poskytuje 5V stejnosměrné napětí, potřebné pro rozšíření systému. Věnujte pozornost konfiguraci systému, abyste zajistili, že CPU může dodávat 5V napětí požadované zvolenými rozšiřovacími moduly. Pokud konfigurace vyžaduje větší proud, než může CPU dodávat, musíte modul demontovat nebo zvolit CPU s vyšší proudovou kapacitou. Informace o 5V DC energetické bilanci, dodávané jednotkou S7-200, a o požadavcích na napětí 5 VDC u rozšiřovacích modulů jsou uvedeny v příloze A. Jako průvodce pro stanovení, jaký výkon (nebo proud) může CPU dodávat pro vaši konfiguraci, použijte přílohu B.

Všechny jednotky S7-200 také poskytují napájení snímačů 24 VDC, které může dodávat 24 VDC pro vstupy, pro napájení relé u rozšiřovacích modulů nebo pro jiná zařízení. Pokud požadavky na napájení překročí výkon zdroje, musíte přidat k systému externí zdroj napětí 24 VDC. V příloze A jsou uvedeny informace pro energetickou bilanci zdroje pro snímače 24 VDC pro konkrétní jednotku S7-200.

Pokud požadujete externí napájecí zdroj 24 VDC, zajistěte, aby nebyl zdroj paralelně propojen se zdrojem pro snímače S7-200. Pro zvýšení odolnosti proti elektrickému rušení se doporučuje, aby byly společné vodiče (M) různých napájecích zdrojů propojeny.



Varování

Připojení externího napájecího zdroje 24 VDC paralelně se zdrojem pro snímače S7-200 24 VDC může způsobit konflikt mezi těmito dvěma zdroji, protože každý se snaží nastavit svou vlastní úroveň výstupního napětí.

Následkem tohoto konfliktu může být zkrácena životnost zařízení nebo může dojít k okamžité poruše jednoho nebo obou napájecích zdrojů a k následnému nepředvídatelnému chování řídicího systému. Nepředvídatelné chování by mohlo mít za následek smrt nebo vážné poranění osob a poškození zařízení.

Stejnosměrný zdroj pro snímače S7-200 a externí zdroj by měly dodávat napětí do různých bodů.

Montáž a demontáž modulů S7-200

Zařízení S7-200 může být snadno nainstalováno na standardní DIN lištu nebo na panel.

Požadavky

Před montáží nebo demontáží elektrického zařízení zajistěte, aby bylo napájení odpojeno. Rovněž zajistěte, aby bylo vypnuto napájení jakéhokoli připojeného zařízení.



Varování

Pokusy instalovat nebo demontovat S7-200 nebo připojené zařízení se zapnutým napájením by mohly způsobit úraz elektrickým proudem nebo chybný chod zařízení.

Nevypnutí napájení systému S7-200 a připojených zařízení během instalace nebo demontáže by mohlo mít za následek smrt nebo vážné poranění osob a poškození zařízení.

Vždy dodržujte příslušné bezpečnostní předpisy a zajistěte, aby bylo vypnuto napájení S7-200 před pokusem instalovat nebo demontovat jednotky CPU nebo připojená zařízení.

Vždy dbejte na to, abyste při jakékoli výměně nebo instalaci zařízení S7-200 použili správný modul nebo odpovídající zařízení.



Varování

Pokud nainstalujete nesprávný modul, může program v S7-200 pracovat nepředvídatelně.

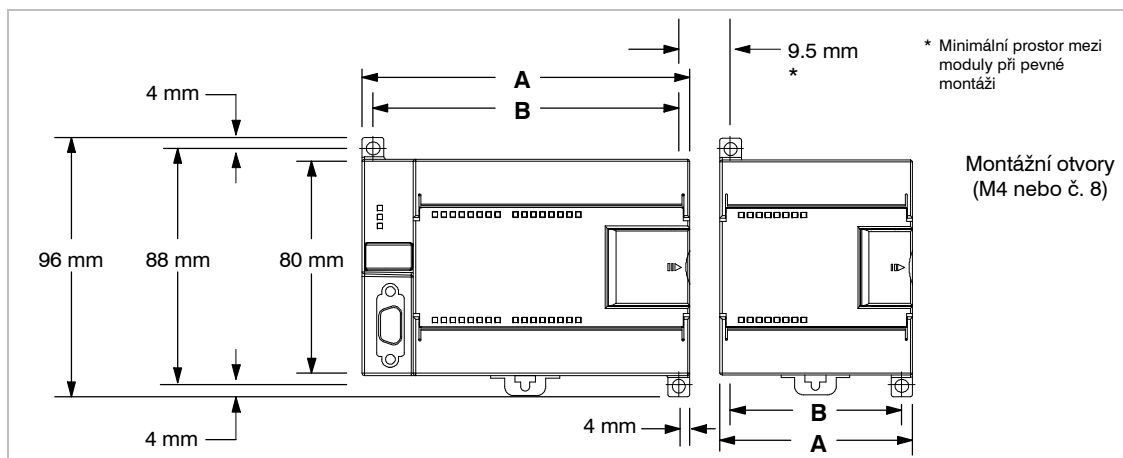
Pokud nenahradíte zařízení S7-200 stejným modelem se stejnou orientací nebo ve stejném pořadí, může to mít za následek smrt nebo vážné zranění osob a poškození zařízení.

Zařízení S7-200 nahraďte stejným modelem a zajistěte jeho správnou orientaci a polohu.

Montážní rozměry

Jednotky S7-200 a rozšiřovací moduly mají montážní otvory, které usnadňují instalaci na panel. Montážní rozměry jsou uvedeny v Tabulce 3-1.

Tabulka 3-1 Montážní rozměry



S7-200 modul	Šířka A	Šířka B
CPU 221 a CPU 222	90 mm	82 mm
CPU 224	120,5 mm	112,5 mm
CPU 226 a CPU 226XM	196 mm	188 mm
Rozšiřovací moduly: 8bodový DC a reléový vst./výst. (8I, 8Q, 4I/4Q, 2 AQ)	46 mm	38 mm
Rozšiřovací moduly: 16bodový digitální vst./výst. (8I/8Q), analogový vst./výst. (4AI, 4AQ/1AQ), RTD, termočlánek, PROFIBUS, AS rozhraní, 8bodový AC (8I a 8Q), polohovací a modemový	71,2 mm	63,2 mm
Rozšiřovací moduly: 32bodový digitální vst./výst. (16I/16Q)	137,3 mm	129,3 mm

Montáž CPU nebo rozšiřovacího modulu

Montáž S7-200 je snadná! Řiďte se těmito kroky.

Montáž na panel

1. Umístěte a vyvrtejte montážní otvory a vyřízněte do nich závit (M4 nebo podle americké normy č. 8) podle rozměrů uvedených v Tabulce 3-1.
2. Připevněte modul(y) na panel pomocí příslušných šroubů.
3. Pokud používáte rozšiřovací modul, připojte plochý kabel rozšiřovacího modulu ke konektoru rozšiřovacího portu pod přístupovým krytem.

Montáž na DIN lištu

1. Lištu připevňujte k montážnímu panelu vždy po 75 mm.
2. Otevřete zajišťovací uzávěr DIN (ve spodní části modulu) a nasadte modul zadní částí na DIN lištu.
3. Pokud používáte rozšiřovací modul, připojte plochý kabel rozšiřovacího modulu ke konektoru rozšiřovacího portu pod přístupovým krytem.
4. Otočte modul dolů k DIN liště a uzavřete uzávěr. Opatrně zkontrolujte, zda uzávěr pevně přichytil modul na lištu. Abyste zabránili poškození modulu, netlačte přímo na přední panel modulu, ale tlačte na úchytka montážního otvoru.

**Tip**

Použití zářezek DIN lišty by mohlo být užitečné, pokud je zařízení S7-200 v prostředí s velkými vibracemi nebo pokud bylo zařízení S7-200 namontováno vertikálně.

Pokud je váš systém v prostředí s velkými vibracemi, bude před nimi S7-200 při montáži na panel poskytovat vyšší úroveň ochrany.

Demontáž CPU nebo rozšiřovacího modulu

Pro demontáž S7-200 nebo rozšiřovacího modulu dodržujte tyto kroky:

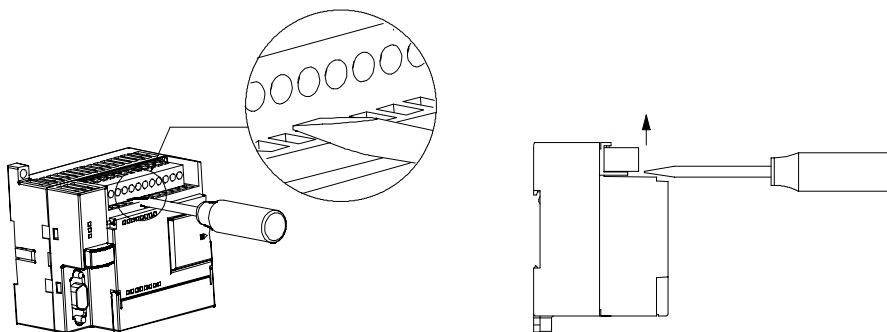
1. Odpojte napájení od S7-200.
2. Odpojte všechny vodiče a kabely, které jsou k modulu připojeny. Většina jednotek S7-200 a rozšiřovacích modulů má odnímatelné konektory, které tuto práci usnadňují.
3. Pokud máte rozšiřovací moduly připojeny k jednotce, kterou demontujete, otevřete přístupový kryt a odpojte plochý kabel rozšiřovacího modulu od sousedních modulů.
4. Odšroubujte montážní šrouby nebo otevřete DIN uzávěr.
5. Vyjměte modul.

Demontáž a montáž svorkovnice

Většina modulů S7-200 má odnímatelné svorkovnice, které usnadňují montáž a výměnu modulu. V příloze A zjistíte, zda má váš modul S7-200 odnímatelné svorkovnice. Pro moduly, které nemají odnímatelné svorkovnice, si můžete objednat doplňkový konektor. Objednací čísla – viz přílohu E.

Vyjmutí svorkovnice

1. Otevřete kryt svorkovnice, abyste k němu měli přístup.
2. Vsuňte malý šroubovák do zářezu uprostřed svorkovnice.
3. Použijte šroubovák jako páku a vysuňte svorkovnici z pouzdra S7-200. Viz obr. 3-2.



Obr. 3-2 Vyjmutí svorkovnice

Nasazení svorkovnice

1. Otevřete kryt svorkovnice.
2. Srovnejte svorkovnici s kolíky na jednotce a srovnejte okraj svorkovnice v rámu patky svorkovnice.
3. Pevně zatlačte dolů a otáčejte svorkovnici, dokud nezapadne na své místo. Opatrně zkontrolujte, zda je svorkovnice řádně srovnána a zcela zasunuta.

Návod pro uzemnění a elektrické zapojení

Správné uzemnění a zapojení všech elektrických zařízení je důležité pro zajištění optimálního provozu systému a pro další ochranu vaší aplikace a jednotky S7-200 před elektrickým rušením.

Požadavky

Před uzemněním nebo montáží kabeláže elektrického zařízení zajistěte, aby bylo napájení zařízení vypnuto. Rovněž zajistěte, aby bylo vypnuto napájení jakéhokoli připojeného zařízení.

Při zapojování S7-200 a připojených zařízení dodržujte veškeré předpisy platné pro elektroinstalaci. Instalujte a provozujte veškerá zařízení podle všech platných státních a místních norem. U místních úřadů zjistěte, jaké předpisy a normy jsou platné pro konkrétní účel použití.



Varování

Pokusy instalovat nebo zapojovat S7-200 nebo připojená zařízení se zapnutým napájením by mohly způsobit úraz elektrickým proudem nebo chybný chod zařízení. Nevypnutí napájení systému S7-200 a připojených zařízení během instalace nebo demontáže by mohlo mít za následek smrt nebo vážné poranění osob a poškození zařízení.

Vždy dodržujte příslušné bezpečnostní předpisy a zajistěte, aby napájení S7-200 bylo vypnuto před pokusem instalovat nebo demontovat jednotku CPU nebo připojená zařízení.

Při navrhování uzemnění a zapojení systému S7-200 mějte vždy na paměti bezpečnost. Elektronická řídicí zařízení, jako je S7-200, mohou selhat a způsobit nepředvídatelné chování zařízení, které je řízeno nebo monitorováno. Proto byste měli zařadit ochranné obvody nezávislé na S7-200 jako ochranu před zraněním osob nebo poškozením zařízení.



Varování

Řídicí zařízení mohou selhat a dostat se do nebezpečného stavu, který může mít za následek nepředvídatelné chování řízeného zařízení. Toto nepředvídatelné chování by mohlo mít za následek smrt nebo vážné zranění osob a poškození zařízení.

Použijte funkci nouzového zastavení, elektromechanické spínače nebo jiné doplňující ochranné obvody nezávislé na S7-200.

Návod pro izolaci

Limity střídavého napájecího napětí S7-200 a napětí na vstupech/výstupech jsou počítány na 1500 VAC. Tyto izolační limity byly vyzkoušeny a odsouhlaseny jako bezpečné oddělení střídavého napětí od nízkonapěťových obvodů.

Všechny nízkonapěťové obvody připojené k S7-200, jako je napětí 24 V, musí být napájeny ze schváleného zdroje, který poskytuje bezpečnou izolaci střídavého napětí a jiných vysokonapěťových okruhů. Tyto zdroje obsahují dvojitou ochranu, která je definována v mezinárodních normách bezpečnosti elektrických zapojení, a mají výstupy, které spadají do třídy SELV, PELV, třídy 2 nebo omezené napětí podle různých norem.



Varování

Použití neoddělených nebo nesprávně oddělených zdrojů pro napájení nízkonapěťových obvodů z vedení střídavého napětí může mít za následek riziková napětí na obvodech, která mají být bezpečná na dotek, jako jsou komunikační obvody a nízkonapěťové rozvody snímačů.

Toto nepředvídatelné vysoké napětí by mohlo mít za následek smrt nebo vážné zranění osob a poškození zařízení.

Používejte pouze takové měniče z vysokého napětí na nízké, které jsou schváleny jako zdroje pro obvody omezeného napětí bezpečného na dotek.

Návod pro uzemnění S7-200

Nejjlepší způsob, jak uzemnit zařízení, je zajistit, aby všechny společné vodiče systému S7-200 a připojených zařízení byly uzemněny v jednom bodě. Ten by měl být připojen přímo k uzemnění vašeho systému.

Ke zvýšení ochrany proti elektrickému rušení se doporučuje, aby všechny společné vratné vodiče stejnosměrného napětí byly připojeny k jedinému bodu uzemnění. Společný vodič (M) napájení snímačů 24 VDC připojte k uzemnění.

Všechny uzemňovací vodiče by měly být co nejkratší; měly by být použity vodiče s vyšším průřezem než 2 mm² (14 AWG).

Při umístění uzemnění nezapomeňte vzít v úvahu požadavky na bezpečné uzemnění a řádnou funkci ochranných jističů zařízení.

Návod pro zapojení S7-200

Při zapojování systému namontujte jeden odpojovací vypínač, který současně odpojí napájení od jednotky S7-200 a od všech vstupních a výstupních obvodů. Pro omezení poruchových proudů na napájecích vodičích nainstalujte nadproudovou ochranu, jako je pojistka nebo stykačový jistič. Další ochranu můžete přidat nainstalováním pojistky nebo jiného omezovače proudu do každého výstupního obvodu.

Namontujte příslušná zařízení na omezení proudových nárazů pro všechny okruhy, které by mohly být vystaveny výbojům blesku.

Neumísťujte nízkonapětové signálové vodiče a komunikační kabely do stejné lišty s napájecími kabely střídavého napětí a vysokonapětovými vodiči stejnosměrného napětí, které jsou rychle spínány. Vždy pokládejte vodiče v párech s neutrálním nebo společným vodičem spárovaným s živým nebo signálním vodičem.

Použijte co nejkratší vodiče, abyste zajistili, že jsou řádně dimenzovány pro přenos požadovaného proudu. Ke konektoru může být připojen vodič o průřezu od 2 mm² do 0,3 mm² (14 AWG až 22 AWG). Kvůli optimální ochraně proti elektrickému rušení použijte stíněné vodiče. Standardně uzemněné stínění dává u S7-200 nejlepší výsledky.

Při zapojování vstupních obvodů, které jsou napájeny externím napájecím zdrojem, nainstalujte do tohoto obvodu nadproudovou ochranu. Externí ochrana není nutná u obvodů, které jsou napájeny zdrojem pro snímače 24 VDC z jednotky S7-200, protože zdroj pro snímače už proudové omezení obsahuje.

Většina modulů S7-200 má odnímatelné svorkovnice kvůli snadnějšímu zapojování. (V příloze A zjistíte, zda má váš modul odnímatelné svorkovnice.) Studeným spojům předejdete tak, že zajistíte, aby byl konektor pevně usazen a aby k němu byl vodič pevně připojen. Neutahujte šrouby příliš, abyste nepoškodili konektor. Maximální utahovací moment pro šroub konektoru je 0,56 Nm.

Pro zabránění nežádoucích proudů v instalaci má S7-200 na určitých bodech napětová oddělení. Limity těchto oddělení byste měli brát v úvahu při plánování zapojení vašeho systému. V příloze A je uvedena hodnota a umístění oddělení. Limity s hodnotou menší než 1500 VAC nesmí být brány za bezpečnostní limity.



Tip

Pro komunikační síť je maximální délka komunikačního kabelu 50 m bez použití zesilovače. Komunikační port na S7-200 není napětově oddělen. Další informace jsou v kapitole 7.

Návod pro ochranné obvody

Indukční zátěže byste měli vybavit ochrannými obvody pro omezení špiček napětí při vypnutí výstupu. Ochranné obvody chrání výstupy před předčasnou poruchou, způsobenou velkými indukčními vypínacími proudy. Navíc omezují elektrické rušení, tvořené při vypnutí indukční zátěže.



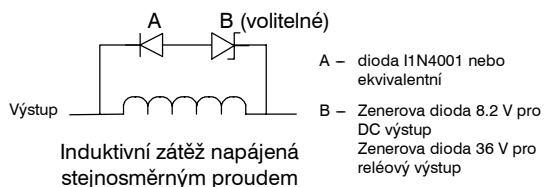
Tip

Účinnost ochranného obvodu závisí na použitém zařízení; musíte jej ověřit pro konkrétní použití. Vždy zajistěte, aby byly všechny komponenty použité v ochranném obvodu vypočítány pro použití v příslušném zařízení.

Stejnosemné a reléové výstupy, které napájejí indukční zátěž stejnosměrným proudem

Stejnosemné výstupy mají vnitřní ochranu, která je dostačující pro většinu aplikací. Protože reléové výstupy mohou být použity pro stejnosměrný i střídavý proud, není u nich vnitřní ochrana implementována.

Na obr. 3-3 je vzor přepětové ochrany pro indukční zátěž napájenou stejnosměrným proudem. U většiny aplikací je vhodné použít diodu (A) přes indukční zátěž, pokud ale vyžaduje vaše aplikace rychlejší vypínací časy, doporučuje se použít Zenerovu diodu (B). Zenerovu diodu správně dimenzujte pro proud ve vašem výstupním obvodu.

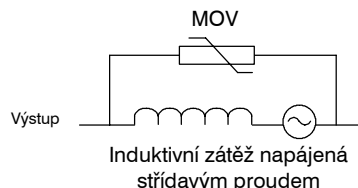


Obr. 3-3 Přepětová ochrana při napájení stejnosměrným proudem

Střídavé a reléové výstupy, které napájejí indukční zátěž střídavým proudem

Střídavé výstupy mají vnitřní ochranu, která je dostačující pro většinu aplikací. Protože reléové výstupy mohou být použity pro stejnosměrný i střídavý proud, není u nich vnitřní ochrana implementována.

Na obr. 3-4 je vzor přepětové ochrany pro indukční zátěž napájenou střídavým proudem. U většiny aplikací omezí přidání varistoru z oxidu kovu (MOV) špičkové napětí a poskytne ochranu pro vnitřní obvody S7-200. Zajistěte, aby pracovní napětí MOV bylo alespoň o 20 % vyšší než nominální napětí vedení.



Obr. 3-4 Přepětová ochrana při napájení střídavým proudem

Základní principy

4

Základní funkcí programovatelného automatu S7-200 je monitorovat vstupy a na základě řídicího programu zapínat nebo vypínat výstupní zařízení. Tato kapitola vysvětluje principy používané pro provádění programu, různé typy použitých pamětí a to, jak je program v dané paměti uchováván.

Přehled kapitoly

Jak S7-200 vykonává řídicí program	22
Přístup k datům	24
Ukládání a obnova dat	34
Uložení programu do paměťového modulu	36
Výběr pracovního režimu CPU S7-200	37
Program pro uložení paměti V do EEPROM	38
Vlastnosti automatů řady S7-200	39

Jak S7-200 vykonává řídicí program

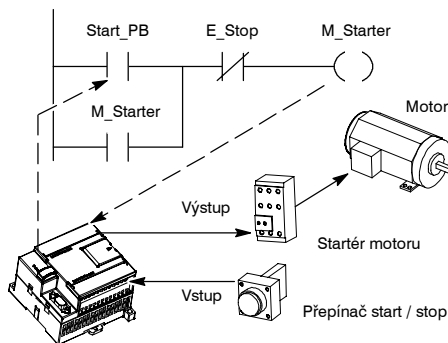
Programovatelný automat cyklicky provádí uložený řídicí program, čte a zapisuje data.

Jak S7-200 spojuje uložený program s fyzickými vstupy a výstupy

Základní provoz S7-200 je velmi jednoduchý:

- ☐ S7-200 přečte stav vstupů.
- ☐ Program uložený v automatu S7-200 použije tyto vstupy při zpracování řídicího programu. Při běhu programu S7-200 aktualizuje svoje data.
- ☐ Automat S7-200 zapíše data na výstup.

Obrázek 4-1 zobrazuje jednoduchý diagram vztahu kontaktního schématu a S7-200. V tomto příkladu je stav přepínače pro start motoru kombinován se stavem ostatních vstupů. Výpočtem těchto stavů pak S7-200 určí stav výstupu, který je přenesen na akční člen, jenž spustí motor.

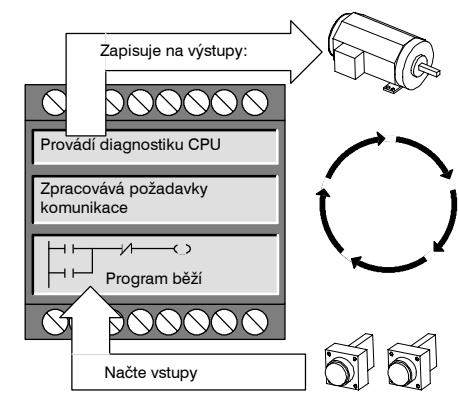


Obr. 4-1 Řízení vstupů a výstupů

S7-200 provádí své úkony v programovém cyklu

S7-200 cyklicky zpracovává řadu úloh. Toto cyklické provádění jednotlivých úloh se nazývá programový cyklus. Jak je znázorněno na obrázku 4-2, S7-200 provádí během programového cyklu všechny následující úlohy nebo jejich větší část:

- ☐ Čtení vstupů: S7-200 kopíruje stav fyzických vstupů do registru obrazu vstupů.
- ☐ Provádění řídicí logiky programu: S7-200 provede instrukce programu a hodnoty uloží do různých oblastí paměti.
- ☐ Zpracování požadavků komunikace: S7-200 provede všechny úlohy, požadované pro komunikaci.
- ☐ Provádění autodiagnostiky CPU: S7-200 kontroluje, zda firmware, paměť pro program a všechny rozšiřovací moduly pracují správně.
- ☐ Zapisování na výstupy: hodnoty uložené v registru obrazu výstupů jsou zapsány na fyzické výstupy.



Obr. 4-2 Programový cyklus S7-200

Provádění programového cyklu závisí na tom, zda je S7-200 v režimu STOP, nebo RUN. V režimu RUN program běží, v režimu STOP nikoliv.

Čtení vstupů:

Digitální vstupy: Každý programový cyklus začíná přečtením aktuálních hodnot na digitálních vstupech a zápisem do registru obrazu vstupů.

Analogové vstupy: Pokud není povoleno filtrování analogových vstupů, automat S7-200 neaktualizuje analogové vstupy při běžném programovém cyklu. Analogový filtr je instalován kvůli získání stabilnějšího signálu. Analogový filtr můžete nastavit pro každý analogový vstupní bod.

Pokud je filtrování analogového vstupu povoleno, S7-200 aktualizuje analogový vstup jednou za programový cyklus, provede funkci filtrování a interně uloží filtrovanou hodnotu. Tato hodnota je pak k dispozici pokaždé, když program přistupuje na tento analogový vstup.

Pokud analogový filtr povolený není, přečte S7-200 hodnotu analogového vstupu pokaždé, když program přistupuje na tento analogový vstup.

**Tip**

Filtrování analogového vstupu je instalováno pro získání stabilnější analogové hodnoty. Filtr použijte pro aplikace, kde se pomalu mění hodnota vstupního signálu. Pokud se signál mění rychle, analogový filtr by neměl být použit.

Nepoužívejte analogový filtr pro moduly, které předávají digitální informace nebo indikaci poruchy analogovým signálem. Analogovou filtraci vždy vypněte pro RTD moduly, termočláňkové moduly a moduly AS-Interface Master.

Provádění programu

Během programového cyklu S7-200 začíná program vykonávat instrukce od první až po poslední. Instrukce bezprostředního vstupu/výstupu poskytují v době běhu programu nebo během přerušení bezprostřední přístup na vstupy a výstupy.

Pokud v programu používáte přerušení, jsou podprogramy přerušení uloženy jako součást programu. Podprogramy přerušení se neprovádějí v programovém cyklu, ale provedou se ve chvíli, kdy je splněna podmínka pro přerušení (což může nastat v libovolném okamžiku programového cyklu).

Zpracování požadavků komunikace

Během této fáze programového cyklu zpracuje S7-200 všechny zprávy, které byly přijaty z komunikačního rozhraní nebo inteligentních vstupních/výstupních modulů.

Provádění autodiagnostiky CPU

Během této fáze programového cyklu zkontroluje S7-200 správný provoz CPU, paměťových oblastí a stav všech rozšiřovacích modulů.

Zapisování na digitální výstupy

Na konci každého programového cyklu zapisuje S7-200 hodnoty uložené v registru obrazu výstupů na digitální výstupy. (Analogové výstupy jsou aktualizovány okamžitě, nezávisle na programovém cyklu.)

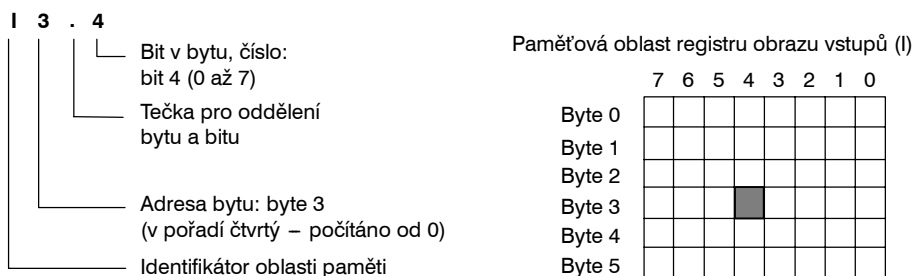
Přístup k datům

Programovatelný automat S7-200 uchovává informace v různých paměťových oblastech, které mají jedinečné adresy. Můžete explicitně určit konkrétní adresu paměti. To programu umožní přímý přístup k informacím. Tabulka 4-1 ukazuje rozsahy, které mohou být reprezentovány daty různých velikostí.

Tabulka 4-1 Desítkové a šestnáctkové rozsahy pro různé velikosti dat

Reprezentace	Byte (B)	Word (W)	Double word (D)
Unsigned integer	0 až 255 0 až FF	0 až 65 535 0 až FFFF	0 až 4 294 967 295 0 až FFFF FFFF
Signed integer	– 128 až +127 80 až 7F	– 32 768 až +32 767 8000 až 7FFF	– 2 147 483 648 až +2 147 483 647 8000 0000 až 7FFF FFFF
Real IEEE 32bitová pohyblivá řádová čárka	<i>Nelze</i>	<i>Nelze</i>	+ 1.175495E–38 až +3.402823E+38 (kladné) – 1.175495E–38 až –3.402823E+38 (záporné)

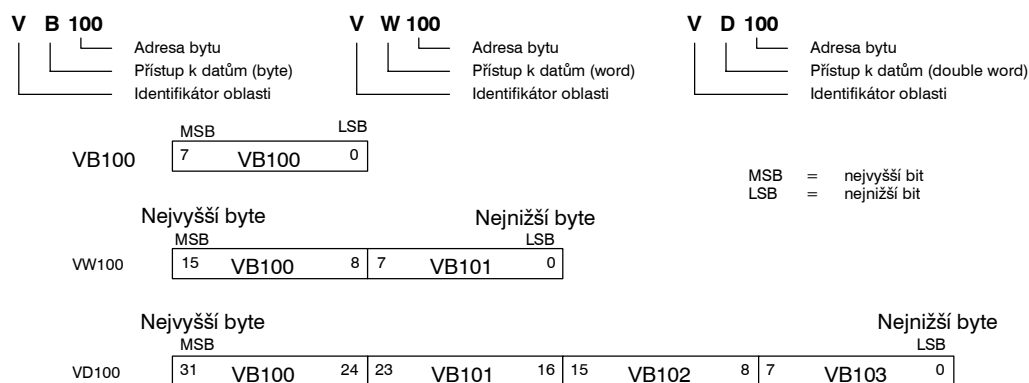
Když chcete přistoupit k nějakému bitu v paměťové oblasti, specifikujete adresu, která se skládá z identifikátoru paměťové oblasti, adresy bytu a čísla bitu. Obr. 4-3 ukazuje příklad přístupu k bitu (často se nazývá adresování "byte.bit"). V tomto příkladu za paměťovou oblastí a adresou bytu (I = vstup a 3 = byte 3) následuje tečka ("."), která odděluje bitovou adresu (bit 4).



Obr. 4-3 Adresování byte.bit

K datům ve většině paměťových oblastí (V, I, Q, M, S, L, a SM) typu byte, word nebo double word můžete přistupovat v tzv. bytovém adresním formátu. Pro přístup k bytu, word nebo double word v paměti je nutné vytvořit adresu podobně, jako byla vytvořena adresa pro přístup k bitu. Adresa se skládá z identifikátoru oblasti, z označení velikosti dat a z adresy počátečního bytu hodnoty bytu, word nebo double word, jak to ukazuje obr. 4-4.

K datům v jiných paměťových oblastech (jako jsou T, C, HC a v akumulátorech) se přistupuje s použitím formátu adresy s identifikátorem oblasti a číslem zařízení.



Obr. 4-4 Porovnání přístupu do stejné oblasti paměti k byte, word a double word

Přístup k datům v paměťových oblastech

Registr obrazu vstupů: I

Automat S7-200 na začátku každého programového cyklu načte aktuální hodnoty vstupních bodů a zapíše je do registru obrazu vstupů. Přístupovat k registru obrazu vstupů můžete v bitech, bytech, word nebo double word:

Bit: $I[\text{adresa bytu}].[adresa bitu]$ I0.1
 Byte, word, double word: $I[\text{velikost}][adresa počátečního bytu]$ IB4

Registr obrazu výstupů: Q

Na konci každého programového cyklu zapisuje S7-200 hodnoty uložené v registru obrazu výstupů na výstupy. K registru obrazu výstupů můžete přístupovat v bitech, bytech, word nebo double word:

Bit: $Q[\text{adresa bytu}].[adresa bitu]$ Q1.1
 Byte, word, double word: $Q[\text{velikost}][adresa počátečního bytu]$ QB5

Oblast proměnné paměti: V

Paměť V můžete používat pro uložení mezivýsledků operací prováděných řídicí logikou programu. Můžete ji také používat pro ukládání ostatních dat, souvisejících s procesem nebo úlohou. Do paměťové oblasti V můžete přístupovat v bitech, bytech, word nebo double word:

Bit: $V[\text{adresa bytu}].[adresa bitu]$ V10.2
 Byte, word, double word: $V[\text{velikost}][adresa počátečního bytu]$ VW100

Oblast bitové paměti: M

Oblast bitové paměti (paměť M) se využívá na řídicí kontakty pro uložení mezistavu procesu nebo mezistavu jiných řídicích informací. Do bitové paměťové oblasti můžete vstupovat v bitech, bytech, word nebo double word:

Bit: $M[\text{adresa bytu}].[adresa bitu]$ M26.7
 Byte, word, double word: $M[\text{velikost}][adresa počátečního bytu]$ MD20

Oblast paměti časovačů: T

Časovače mají časové rozlišení 1 ms, 10 ms nebo 100 ms. K časovači patří dvě proměnné:

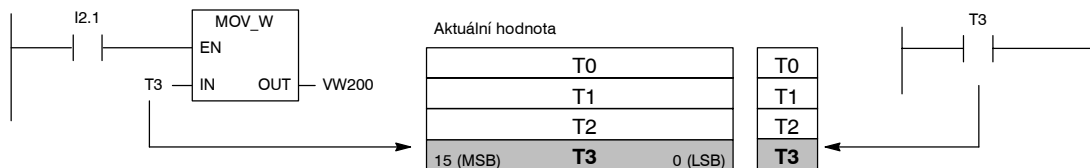
- ☐ Aktuální hodnota: 16bitové číslo, které reprezentuje aktuální čas napočítaný časovačem.
- ☐ Bit časovače: tento bit má hodnotu 0 nebo 1 podle výsledku porovnání aktuální a nastavené hodnoty. Nastavená hodnota je součástí instrukce časovače.

Přístup k proměnným získáte použitím adresy časovače (T + číslo časovače). Přístup k bitu časovače nebo k aktuální hodnotě závisí na použité instrukci: instrukce s bitovými operandy přistupují k bitu časovače, instrukce s operandy word přistupují k aktuální hodnotě. Jak je ukázáno na obrázku 4-5, instrukce kontaktu "V klidu otevřen" vyhodnotí bit časovače, instrukce "Přesunout word" umožní přístup k aktuální hodnotě časovače.

Formát:

T[číslo časovače]

T24



Obr. 4-5 Přístup k bitu časovače nebo k jeho aktuální hodnotě

Oblast paměti čítačů: C

S7-200 má tři typy čítačů, které počítají každou událost s přechodem nahoru nebo dolů na vstupech čítače: jeden typ počítá pouze nahoru, druhý pouze dolů a poslední nahoru i dolů. Každý čítač má dvě proměnné:

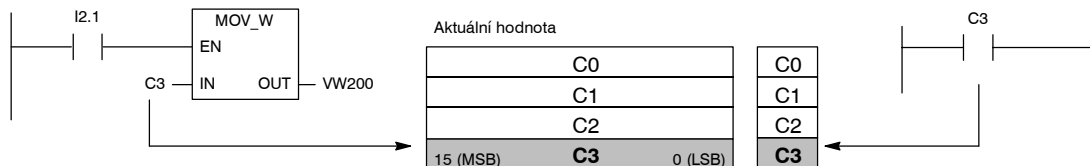
- ☐ Aktuální hodnota: 16bitové číslo, kde je uložen aktuální počet událostí.
- ☐ Bit čítače: tento bit má hodnotu 0 nebo 1 podle výsledku porovnání aktuální a nastavené hodnoty. Nastavená hodnota je součástí instrukce čítače.

Přístup k proměnným získáte použitím adresy čítače (C + číslo čítače). Přístup k bitu čítače nebo k aktuální hodnotě závisí na použité instrukci: instrukce s bitovými operandy přistupují k bitu čítače, instrukce s operandy word přistupují k aktuální hodnotě. Jak je vidět na obrázku 4-6, instrukce kontaktu "V klidu otevřen" vyhodnotí bit čítače, instrukce "Přesunout word" umožní přístup k aktuální hodnotě čítače.

Formát:

C[číslo čítače]

C24



Obr. 4-6 Přístup k bitu čítače nebo k jeho aktuální hodnotě

Vysokorychlostní čítače: HC

Vysokorychlostní čítače počítají události s vysokou frekvencí nezávisle na programovém cyklu. Hodnota čítače je vyjádřena 32bitovým celým číslem. Pro přístup k hodnotě vysokorychlostního čítače je třeba zadat adresu čítače tak, že zadáme typ paměti (HC) a číslo čítače (např. HC0). Aktuální hodnotu čítače je možné číst a je možné k ní přistupovat pouze jako k double word (32 bitů).

Formát: HC[číslo vysokorychlostního čítače]

HC1

Akumulátory: AC

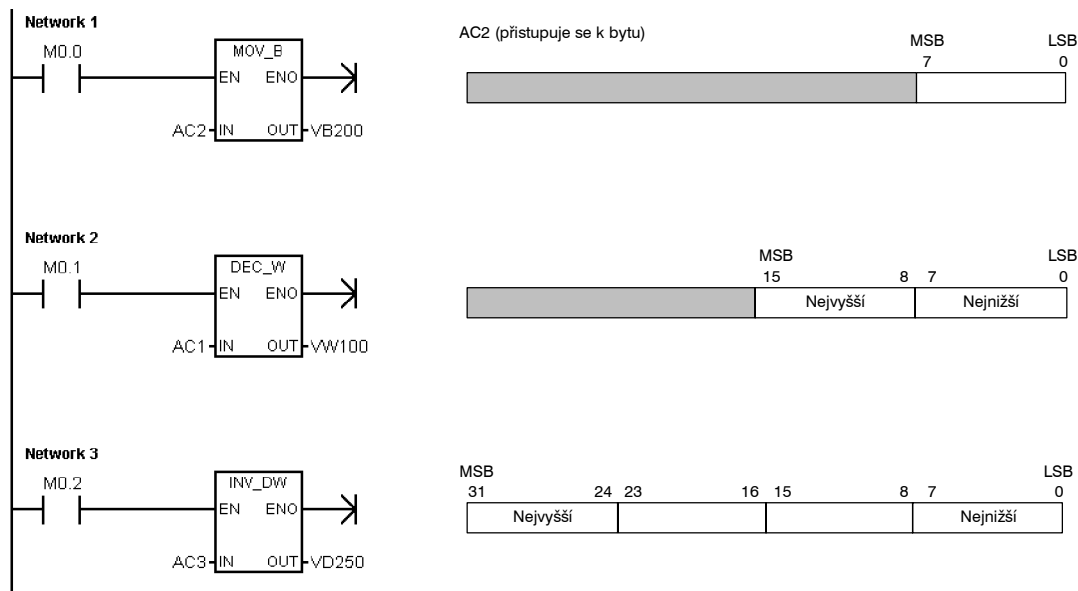
Akumulátory se používají na čtení a zápis podobně jako paměť. Akumulátory můžete například použít na předávání parametrů do podprogramů nebo pro uložení mezivýsledků výpočtů. Programovatelný automat S7-200 má čtyři 32bitové akumulátory (AC0, AC1, AC2 a AC3). K datům v akumulátorech můžete přistupovat po bytech, word nebo double word.

Velikost dat, kterou chcete číst nebo zapisovat, je určena typem instrukce pro práci s akumulátorem. Jak je patrné z obrázku 4-7, při přístupu k bytu přistupujete k 8 nejnižším bitům akumulátoru, při přístupu k word přistupujete k 16 nejnižším bitům akumulátoru. Při přístupu k double word využijete všech 32 bitů akumulátoru.

Informace o tom, jak používat akumulátory v přerušeních, najdete v kapitole 6 – Pokyny pro přerušení.

Formát: AC[číslo akumulátoru]

AC0



Obr. 4-7 Přístup k akumulátorům

Speciální paměť: SM

SM bity slouží pro výměnu informací mezi CPU a programem. Tyto bity se používají pro nastavení a řízení některých speciálních funkcí CPU S7-200, jako jsou: bit prvního programového cyklu; bit, který se mění s pevnou frekvencí; nebo bit stavu matematických nebo provozních instrukcí. (Podrobné informace o SM bitech jsou uvedeny v Příloze D.) K datům v této paměti můžete přistupovat po bitech, bytech, word nebo double word:

Bit:	SM[adresa bytu].[adresa bitu]	SM0.1
Byte, word, double word:	SM[velikost][adresa počátečního bytu]	SMB86

Oblast lokální paměti: L

Programovatelný automat S7-200 má 64 bytů lokální paměti, z nichž 60 bytů může být použito jako zásobníková paměť nebo pro předávání formálních parametrů podprogramům.

**Tip**

V programovacím jazyku LAD nebo FBD rezervuje STEP 7–Micro/WIN poslední čtyři byty lokální paměti pro vlastní použití. Pokud programujete v STL, můžete přistupovat ke všem 64 bytům paměti L; nedoporučuje se ale používat poslední čtyři byty.

Lokální paměť je podobná paměti V s jedinou výjimkou. Paměť V je globální, zatímco paměť L je lokální. Globální paměť je taková paměť, jejíž data je možné číst a zapisovat z libovolné části programu (hlavního programu, podprogramů nebo přerušení). Lokální paměť je taková paměť, která je určena pouze pro nějakou část programu. S7-200 přiděluje 64 bytů paměti L pro hlavní program, 64 bytů pro každou úroveň vnoření podprogramu a 64 bytů pro přerušení.

Paměť určená pro hlavní program se nedá adresovat (není možné číst nebo zapisovat data) z podprogramů nebo z přerušení. Podprogram nemůže adresovat část paměti L přidělenou hlavnímu programu, přerušení nebo jinému podprogramu. Podobně nemůže přerušení adresovat paměť L hlavního programu nebo jiného podprogramu.

Programovatelný automat S7-200 přiděluje paměť L podle potřeby. To znamená, že během provádění hlavního programu neexistuje paměť L pro podprogramy a pro přerušení. Až v okamžiku, kdy je vyvolán podprogram nebo přerušení, přidělí se pro něj lokální paměť podle potřeby. Nové přidělení paměti L může opět použít stejná paměťová místa jiného podprogramu nebo přerušení.

Při přidělení paměti L nevymaže automat její paměťová místa; tato místa mohou obsahovat libovolné hodnoty. Při odevzdávání formálních parametrů do programu vloží S7-200 parametry na odpovídající místo v paměti L volaného podprogramu. Místa paměti L, která neobdrží žádnou hodnotu kvůli tomu, že je vypuštěn krok formálního parametru, se neresetují a mohou v době přidělení obsahovat libovolnou hodnotu.

Bit:	L[adresa bytu].[adresa bitu]	L0.0
Byte, word, double word:	L[velikost] [adresa počátečního bytu]	LB33

Analogové vstupy: AI

Automat S7-200 převádí analogové hodnoty (jako je teplota nebo napětí) na digitální hodnoty o délce word (16 bitů). Tyto hodnoty se dají získat pomocí identifikátoru oblasti (AI), velikosti dat (W) a adresy počátečního bytu. Protože analogové vstupy mají velikost word a vždy začínají na adrese bytu se sudým číslem (jako je 0, 2 nebo 4), přistupuje se k nim vždy se sudou adresou bytu (například AIW0, AIW2 nebo AIW4). Analogové vstupní hodnoty se dají pouze číst.

Formát: AIW[adresa počátečního bytu] AIW4

Analogové výstupy: AQ

Automat S7-200 převádí 16bitovou digitální hodnotu velikosti word na proud nebo na napětí úměrné digitální hodnotě. Tyto hodnoty se dají zapsat pomocí identifikátoru oblasti (AQ), velikosti dat (W) a adresy počátečního bytu. Protože analogové výstupy mají velikost word a vždy začínají na adrese bytu se sudým číslem (jako je 0, 2 nebo 4), zapisují se vždy se sudou adresou bytu (například AQW0, AQW2 nebo AQW4). Hodnoty analogových výstupů se dají pouze zapisovat.

Formát: AQW[adresa počátečního bytu] AQW4

Paměťová oblast sekvenčních řídicích relé (SCR): S

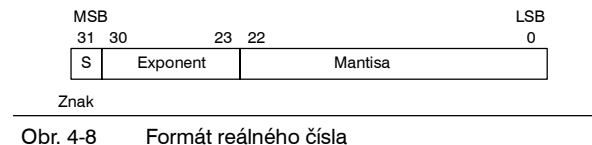
SCR neboli bity S se používají pro organizaci operací strojů nebo kroků v ekvivalentních segmentech programu. SCR umožňují logické členění řídicího programu. K datům v této paměti můžete přistupovat po bitech, bytech, word nebo double word.

Bit: S[adresa bytu].[adresa bitu] S3.1
 Byte, word, double word: S[velikost][adresa počátečního bytu] SB4

Formát pro reálná čísla

Reálná čísla jsou reprezentována 32bitovými čísly s jednoduchou přesností, jejichž formát popisuje norma ANSI/IEEE 754–1985. Viz obr. 4-8. Reálná čísla je možné číst jako data double word.

V automatu S7-200 jsou čísla s pohyblivou řádovou čárkou přesná na 6 desetinných míst. Proto je možné při vkládání konstanty s pohyblivou desetinnou čárkou specifikovat maximálně 6 desetinných míst.

**Přesnost při výpočtech s reálnými čísly**

Výpočty, do kterých patří dlouhé řady hodnot, obsahující velmi velká a velmi malá čísla, mohou dát nepřesné výsledky. K tomu může dojít, pokud se čísla liší o 10 na $x-tou$, kde $x > 6$.

Například: $100\,000\,000 + 1 = 100\,000\,000$

Formát pro řetězce

Řetězec je posloupnost znaků, kde je každý znak uložen jako byte. První byte řetězce určuje jeho délku, tzn. počet znaků. Obr. 4-9 ukazuje formát řetězce. Řetězec může mít délku 0 až 254 znaků plus byte udávající délku, takže maximální délka řetězce je 255 bytů.

Délka	Znak 1	Znak 2	Znak 3	Znak 4	— — — —	Znak 254

Obr. 4-9 Formát řetězce

Specifikace konstanty pro instrukce S7-200

V mnoha instrukcích automatu S7-200 můžete použít konstantu. Konstanty mohou být typu byte, word nebo double word. Programovatelný automat S7-200 ukládá všechny konstanty jako binární čísla reprezentovaná ve formátu desítkovém, šestnáctkovém, ASCII formátu nebo jako reálná čísla. Viz tabulku 4-2.

Tabulka 4-2 Reprezentace konstant

Reprezentace	Formát	Příklad
Desítková	[desítková hodnota]	20047
Šestnáctková	16#[šestnáctková hodnota]	16#4E4F
Binární	2#[binární číslo]	2#1010_0101_1010_0101
ASCII	'[ASCII text]'	'Text patří mezi apostrofy.'
Reálná	ANSI/IEEE 754 – 1985	+1.175495E–38 (kladné) –1.175495E–38 (záporné)



Tip

CPU S7-200 nepodporuje kontrolu typu dat (jako například specifikaci toho, že konstanta se ukládá jako integer, signed integer nebo jako double integer). Například instrukce "Sečíst" může použít hodnotu na adrese VW100 jako signed integer a instrukce XOR může tuto hodnotu použít jako binární hodnotu bez znaménka.

Adresování lokálních a rozšiřovacích I/O

Lokální vstupy a výstupy, které jsou integrované přímo v CPU, mají pevně stanovené adresy. Je možné přidat I/O body CPU S7-200 připojením rozšiřovacích modulů k pravé straně CPU, a zvýšit tak počet vstupů nebo výstupů. Adresy bodů v rozšiřovacích modulech jsou určeny typem vstupu nebo výstupu a umístěním modulu v řetězci s přihlédnutím k předcházejícímu vstupnímu nebo výstupnímu modulu stejného typu. Například výstupní modul neovlivňuje adresy bodů ve vstupním modulu a naopak. Obdobně analogové moduly neovlivňují adresování digitálních modulů a naopak.



Tip

Pro digitální rozšiřovací moduly je vždy rezervováno místo v registru obrazu s přírůstkem po osmi bitech (po jednom bytu). Pokud modul nevyužívá všechny bity z rezervovaného bytu, není možné je přiřadit dalším modulům v řetězci I/O modulů. Nevyužité bity vstupních modulů jsou v každém programovém cyklu nastaveny na nulu.

Analogové rozšiřovací moduly mají vždy vymezená místa s adresami po dvou bodech. Pokud modul nevyužívá všechny bity z rezervovaného slova, není možné je využít ani je není možné přiřadit dalším modulům v řetězci I/O modulů.

Na obrázku 4-10 je příklad číslování vstupů a výstupů pro konkrétní konfiguraci hardwaru. Mezery v adresách (zvýrazněné šedou kurzívou) není možné programem využít.

CPU 224	4 vst./4 výst.	8 vstupů	4 analogové vstupy 1 analogový výstup	8 výstupů	4 analogové vstupy 1 analogový výstup
I0.0 Q0.0 I0.1 Q0.1 I0.2 Q0.2 I0.3 Q0.3 I0.4 Q0.4 I0.5 Q0.5 I0.6 Q0.6 I0.7 Q0.7 I1.0 Q1.0 I1.1 Q1.1 I1.2 Q1.2 I1.3 Q1.3 I1.4 Q1.4 I1.5 Q1.5 I1.6 Q1.6 I1.7 Q1.7 Lokální vstup–výstup	Modul 0 I2.0 Q2.0 I2.1 Q2.1 I2.2 Q2.2 I2.3 Q2.3 I2.4 Q2.4 I2.5 Q2.5 I2.6 Q2.6 I2.7 Q2.7 Rozšiřovací vstup–výstup	Modul 1 I3.0 I3.1 I3.2 I3.3 I3.4 I3.5 I3.6 I3.7	Modul 2 AIW0 AQW0 AIW2 AQW2 AIW4 AIW6	Modul 3 Q3.0 Q3.1 Q3.2 Q3.3 Q3.4 Q3.5 Q3.6 Q3.7	Modul 4 AIW8 AQW4 AIW10 AQW6 AIW12 AIW14

Obr. 4-10 Příklad I/O adres pro lokální a rozšiřovací I/O (CPU 224)

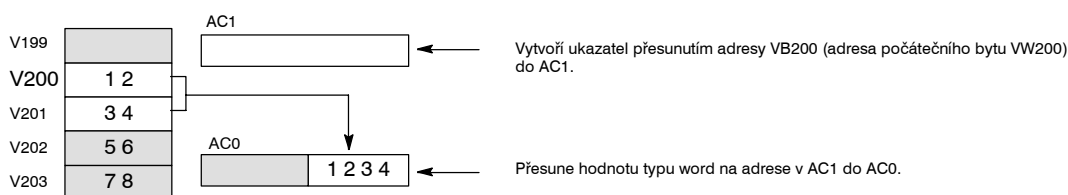
Použití ukazatelů pro nepřímé adresování paměťových oblastí S7-200

Nepřímé adresování používá pro přístup k datům v paměti ukazatel. Ukazatel je místo v paměti s velikostí double word, které obsahuje adresu jiného místa v paměti. Jako ukazatel je možné používat místa v paměti V, místa v paměti L nebo registry akumulátorů (AC1, AC2, AC3). Pro vytvoření ukazatele se používá instrukce "Přesunout double word", kterou se přesune adresa nepřímě adresovaného místa v paměti na adresu ukazatele. Ukazatel je také možné předat podprogramu jako parametr.

Automat S7-200 umožňuje ukazatelům přístup do následujících paměťových oblastí: I, Q, V, M, S, T (pouze aktuální hodnota) a C (pouze aktuální hodnota). Nepřímé adresování se nemůže použít pro přístup k jednotlivému bitu nebo pro přístup do paměťových oblastí AI, AQ, HC, SM nebo L.

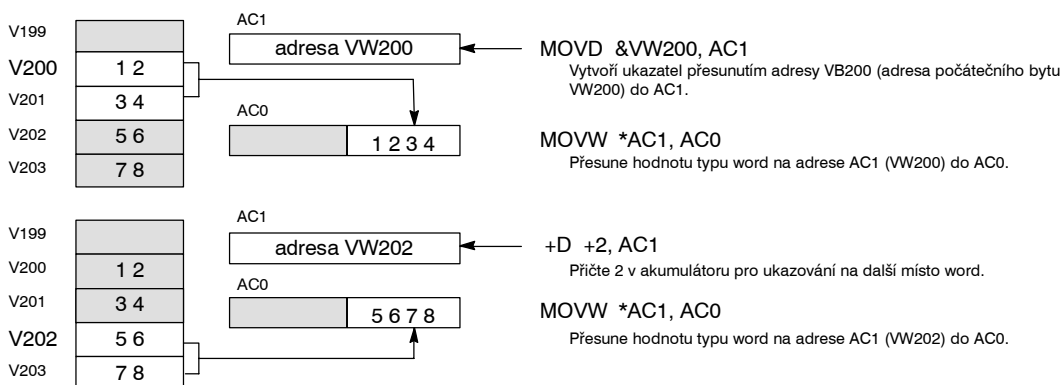
Pro nepřímý přístup k datům na dané paměťové adrese se vytvoří ukazatel na toto místo vložením ampersandu (&) a paměťového místa, které má být adresováno. Vstupní operand instrukce musí začínat ampersandem (&) označujícím, že se na místo určené výstupním operandem instrukce (ukazatel) má přesunout adresa paměťového místa, a ne jeho obsah.

Vložení hvězdičky (*) před operand instrukce znamená, že operand je ukazatel. Podle příkladu na obrázku 4-11 vložení *AC1 určuje, že AC1 je ukazatel na hodnotu typu word, na kterou odkazuje instrukce "Přesunout word". V tomto příkladu se hodnoty uložené ve VB200 i ve VB201 přesunou do akumulátoru AC0.



Obr. 4-11 Vytvoření a používání ukazatele

Jak je vidět na obrázku 4-12, hodnotu ukazatele je možné změnit. Protože ukazatele jsou 32bitové hodnoty, může být jeho hodnota změněna pomocí instrukcí pracujících s hodnotou double word. Pro změnu hodnot ukazatele se mohou použít jednoduché matematické operace, například přičtení nebo inkrementace hodnoty.



Obr. 4-12 Změna ukazatele

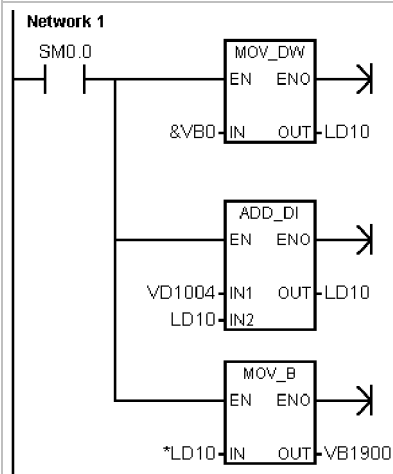


Tip

Je nutné provést přizpůsobení na typ dat, ke kterým se přistupuje: pro přístup k datům typu byte se hodnota ukazatele zvyšuje o 1; pro přístup k datům typu word nebo aktuální hodnotě časovače nebo čítače se hodnota ukazatele zvyšuje o 2 nebo se přičtou násobky dvou; pro přístup k datům typu double word se hodnota ukazatele zvyšuje o 4 nebo se přičtou násobky čtyř.

Ukázkový program použití offsetu pro přístup k datům v paměti V

Tento příklad používá LD10 jako ukazatel na adresu VB0. Ukazatel se pak zvětší o offset uložený ve VD1004. Pak LD10 ukazuje na jinou adresu v paměti V ($VB0 + \text{offset}$). Hodnota uložená na adrese v paměti V, na kterou ukazuje LD10, je pak zkopírována do VB1900. Změnou hodnoty ve VD1004 se získá přístup na libovolné místo paměti V.



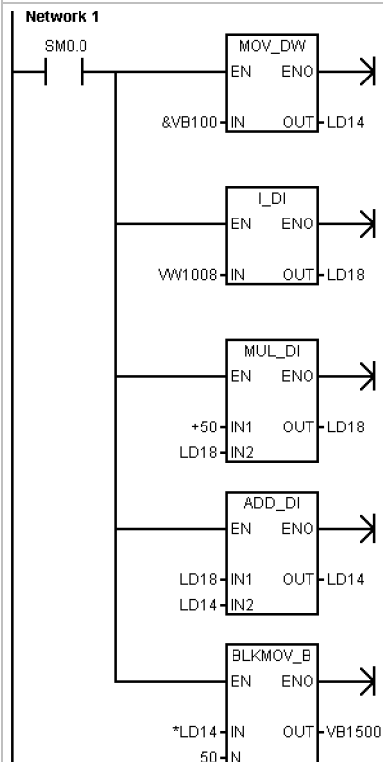
Network 1 //Použití offsetu pro čtení hodnoty jakéhokoliv místa VB:

```
//
//1. Počáteční adresa z paměti V se zavede do ukazatele.
//2. Hodnota offsetu se přičte k ukazateli.
//3. Hodnota z místa v paměti V (offset) se zkopíruje do VB1900.
```

```
LD    SM0.0
MOVD  &VB0, LD10
+D    VD1004, LD10
MOVB  *LD10, VB1900
```

Ukázkový program použití ukazatele pro přístup k datům v tabulce

V tomto příkladu se LD14 použije jako ukazatel na recepturu uloženou v tabulce receptur, která začíná na VB100. Ve VW1008 je uložen index konkrétní receptury z tabulky. Pokud má každá receptura v tabulce délku 50 bytů, index se vynásobí 50, aby se získal offset pro počáteční adresu konkrétní receptury. Přičtením offsetu k ukazateli se získá přístup k příslušné receptuře z tabulky. V tomto příkladu je receptura zkopírována do 50 bytů, které začínají na VB1500.



Network 1 //Jak přesunout recepturu z tabulky receptur:

```
// – Každá receptura má délku 50 bytů.
// – Index (VW1008) určuje recepturu,
//   která má být zavedena.
//
//1. Vytvoří se ukazatel na počáteční adresu tabulky receptur.
//2. Index receptury se převede na hodnotu double word.
//3. Offset se vynásobí, aby odpovídal velikosti receptury.
//4. Hodnota upraveného offsetu se přičte k ukazateli.
//5. Vybraná receptura se přesune na adresy VB1500 až VB1549.
```

```
LD    SM0.0
MOVD  &VB100, LD14
ITD   VW1008, LD18
*D    +50, LD18
+D    LD18, LD14
BMB   *LD14, VB1500, 50
```

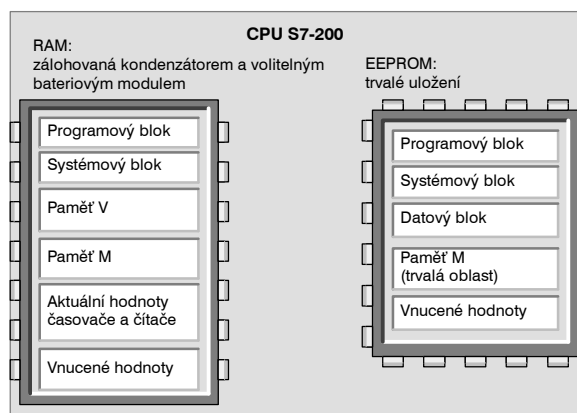
Ukládání a obnova dat

Programovatelný automat S7-200 poskytuje více způsobů bezpečného uchování programu, programových dat a konfiguračních dat.

Programovatelný automat S7-200 má zabudovaný velkokapacitní kondenzátor pro udržení informací v paměti RAM po odpojení napájení. V závislosti na typu automatu S7-200 mohou být tímto způsobem data v paměti RAM uchována i několik dnů.

S7-200 má také zabudovanou paměť EEPROM pro trvalé uložení celého programu, datových oblastí definovaných uživatelem a konfiguračních dat.

Automat může mít dodatečný bateriový modul, který prodlouží čas udržení informací v paměti RAM po vypnutí napájení. Bateriový modul napájí automat i po vybití kondenzátoru.



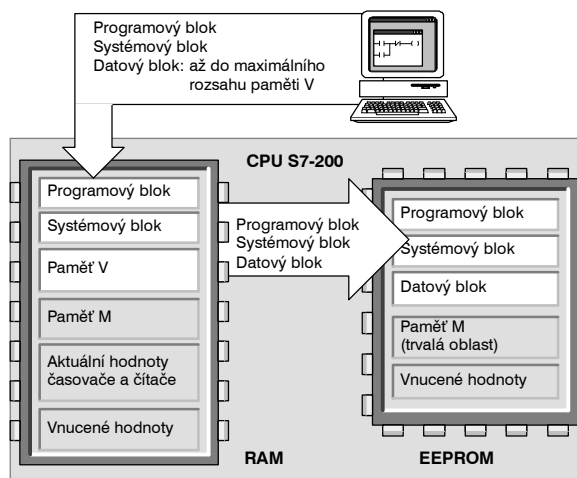
Obr. 4-13 Oblasti paměti CPU S7-200

Download a upload projektu

Každý projekt se skládá ze tří částí: programového bloku, datového bloku (volitelné) a systémového bloku (volitelné).

Obr. 4-14 ukazuje, jak probíhá download projektu do S7-200.

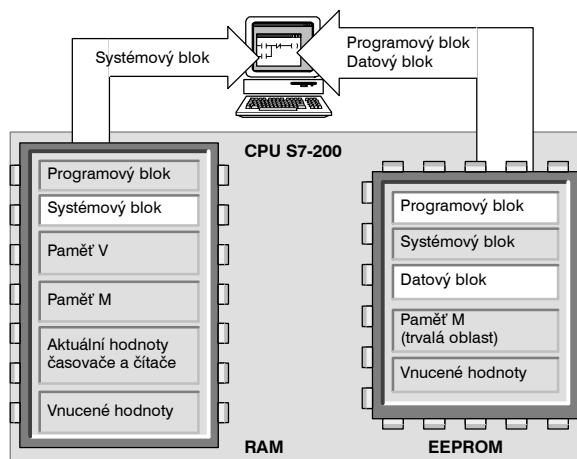
Při downloadu projektu jsou jeho prvky uloženy do oblasti paměti RAM. Automat S7-200 automaticky zkopíruje uživatelský program, datový blok a systémový blok také do paměti EEPROM pro trvalé uložení.



Obr. 4-14 Download projektu do S7-200

Obr. 4-15 ukazuje, jak probíhá upload projektu z S7-200.

Při nahrávání projektu z S7-200 do PC (upload) nahraje CPU systémový blok z paměti RAM a programový a datový blok nahraje z paměti EEPROM.



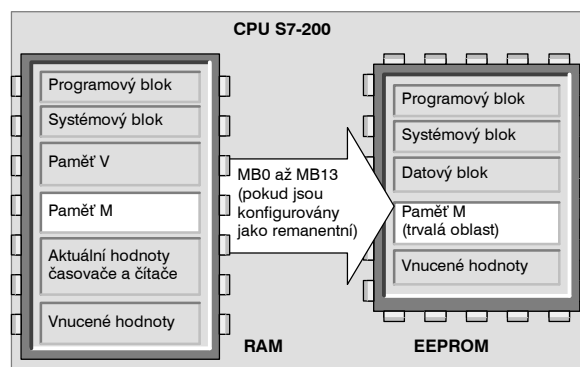
Obr. 4-15 Upload projektu ze S7-200

Automatické ukládání dat z bitové paměti (M) při výpadku napájení

Pokud je prvních 14 bytů bitové paměti (MB0 až MB13) nakonfigurováno jako remanentní (uchovávají svoji hodnotu), jsou tyto byty v případě výpadku napájení S7-200 trvale uloženy v EEPROM.

Jak je ukázáno na obrázku 4-16, S7-200 přesouvá remanentní oblasti paměti M do EEPROM.

Implicitní nastavení prvních 14 bytů paměti M je neremanentní. Implicitní nastavení blokuje uložení, ke kterému dochází při odpojení napájení S7-200.

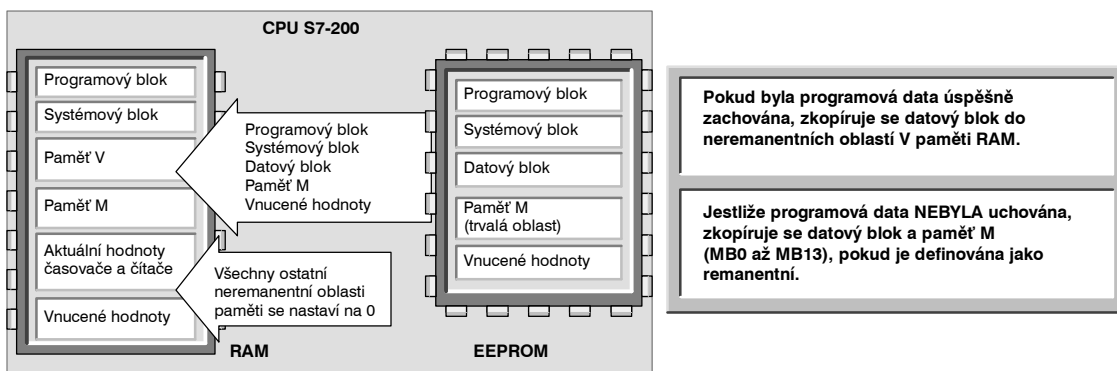


Obr. 4-16 Automatické ukládání dat z paměti M při výpadku napájení

Obnovení dat po zapnutí napájení

Při zapnutí napájení obnoví S7-200 programový a systémový blok z paměti EEPROM tak, jak je znázorněno na obrázku 4-17. Při zapnutí napájení S7-200 také zkontroluje stav paměti RAM, aby si ověřil, že kondenzátor s velkou kapacitou úspěšně zálohoval data uložená v paměti RAM. Pokud byla RAM úspěšně zálohována, ponechají se remanentní oblasti RAM nezměněné.

Remanentní a neremanentní oblasti paměti V jsou obnoveny z odpovídajícího datového bloku v EEPROM. Pokud si paměť RAM neudržela svůj obsah (po delším výpadku napájení), S7-200 RAM vymaže jak remanentní, tak neremanentní rozsahy; v prvním programovém cyklu po opětovném náběhu nastaví paměťový bit ztráty remanentních dat (SM0.2). Do RAM se také následně zkopírují data uložená v paměti EEPROM.



Obr. 4-17 Obnovení dat po zapnutí napájení

Uložení programu do paměťového modulu

K automatu S7-200 je možné připojit volitelný paměťový modul, který umožňuje uložit program do přenosné paměti EEPROM. Do paměťového modulu ukládá S7-200 následující data: programový blok, datový blok, systémový blok a vnucené (force) hodnoty.

Program se dá kopírovat z paměti RAM do paměťového modulu pouze tehdy, když je automat S7-200 napájený, je v režimu STOP a má zasunutý paměťový modul. Paměťový modul se dá instalovat nebo odstranit pouze tehdy, je-li automat S7-200 napájený.

Varování

Elektrostatický výboj může poškodit paměťový modul nebo svorkovnici na CPU S7-200.

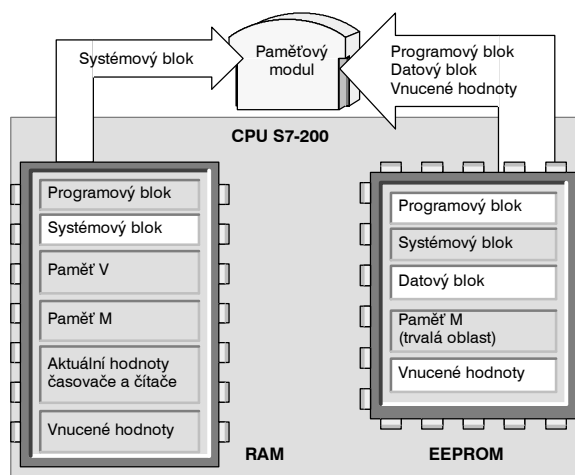
Při manipulaci s modulem je nutné být v kontaktu s uzemněnou vodivou podložkou nebo použít uzemněný vodivý náramek. Paměťový modul je třeba uchovávat ve vodivém obalu.

Při instalaci paměťového modulu je třeba nejdříve odstranit plastový kryt na CPU S7-200 a zasunout paměťový modul do konektoru. Tvar paměťového modulu je upraven tak, že může být zasunut pouze tehdy, je-li správně orientovaný.

Kopírování programu do paměťového modulu

Po vložení paměťového modulu použijte pro zkopírování programu následující postup:

1. Uved'te CPU S7-200 do režimu STOP.
2. Pokud ještě nebyl proveden download programu do S7-200, je nutné ho provést.
3. Pomocí příkazu z menu **PLC > Paměťový modul** se zkopíruje program do paměťového modulu. Obr. 4-18 ukazuje části paměti CPU, které jsou uloženy do paměťového modulu.
4. Volitelně: Vytáhněte paměťový modul a vraťte na místo kryt konektoru na S7-200.



Obr. 4-18 Kopírování do paměťového modulu

Obnovení programu z paměťového modulu

Pro nahrání programu z paměťového modulu do S7-200 je třeba nejdříve automat vypnout a se zasunutým modulem ho opět zapnout.

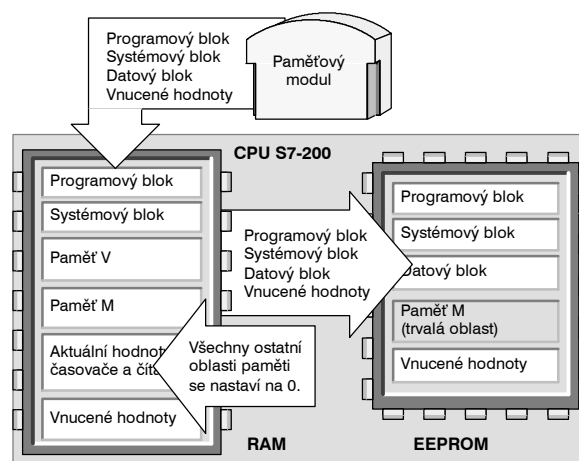
Poznámka

Při zapnutí CPU S7-200 s prázdným paměťovým modulem nebo s modulem naprogramovaným jiným typem CPU S7-200 může dojít k chybě. Paměťové moduly naprogramované v modelech CPU s nižším typovým číslem se v modelech s číslem vyšším dají přečíst. Opačně to však není možné. Například moduly naprogramované v CPU 221 nebo CPU 222 je možné přečíst v CPU 224, ale CPU 221 nebo CPU 222 nedokáže přečíst moduly naprogramované v CPU 224.

Je třeba odstranit paměťový modul a znovu zapnout S7-200. Teprve pak lze modul opět vložit a přeprogramovat podle potřeby.

Jak je vidět na obrázku 4-19, automat S7-200 provádí po vypnutí a opětném zapnutí se zasunutým paměťovým modulem následující úkony:

1. Pokud se obsah paměťového modulu liší od obsahu paměti EEPROM, vymaže S7-200 paměť RAM.
2. S7-200 zkopíruje obsah paměťového modulu do paměti RAM.
3. S7-200 zkopíruje programový blok, systémový blok a datový blok do EEPROM.



Obr. 4-19 Obnovení z paměťového modulu

Výběr pracovního režimu CPU systému S7-200

S7-200 má dva pracovní režimy: Režim STOP a režim RUN. Stavová LED dioda na předním panelu CPU signalizuje aktuální pracovní režim. V režimu STOP neprovádí S7-200 program a je možné nahrávat program (download) nebo konfigurovat CPU. V režimu RUN S7-200 program zpracovává.

- ❑ S7-200 je vybaven přepínačem pro změnu pracovního režimu. Přepínač (umístěný pod předními dvířky S7-200) je možné použít pro ruční výběr pracovního režimu: jeho přepnutí na režim STOP zastaví provádění programu, přepnutí na režim RUN spustí provádění programu. Přepnutím na režim TERM (terminál) se provozní režim nezmění.

Jestliže je při výpadku napájení S7-200 ve stavu STOP nebo TERM, bude po obnovení napájení ve stavu STOP. Jestliže je při výpadku napájení S7-200 ve stavu RUN, bude po obnovení napájení ve stavu RUN.

- ❑ Software STEP 7 – Micro/WIN umožňuje měnit pracovní režim S7-200 online. Aby mohl software změnit pracovní režim, je nutné ručně nastavit přepínač na S7-200 buď na TERM, nebo na RUN. Pro změnu pracovního režimu je možné použít příkazy menu **PLC > STOP** nebo **PLC > RUN** nebo příslušná tlačítka na panelu nástrojů.
- ❑ Vložením instrukce STOP do programu se S7-200 přepne do režimu STOP. Tím se umožňuje zastavit provádění programu na základě uživatelského programu. Podrobné informace o instrukci STOP jsou uvedeny v kapitole 6.

Program pro uložení paměti V do EEPROM

Hodnotu (byte, word nebo double word), uloženou kdekoli v oblasti paměti V, můžete uložit do paměti EEPROM. Proces uložení do paměti EEPROM prodlouží čas programového cyklu maximálně o 5 ms. Nově zapsaná hodnota přepíše jakoukoliv předchozí hodnotu, uloženou v oblasti paměti EEPROM.

Operace uložení do paměti EEPROM neaktualizuje data v paměťovém modulu.



Tip

Vzhledem k tomu, že je počet operací uložení do EEPROM omezený (minimálně 100.000 a obvykle 1.000.000), je třeba zabezpečit, aby byly ukládány pouze nezbytné hodnoty. Jinak může dojít k opotřebení EEPROM a CPU může selhat. Operace zápisu do EEPROM by se měly provádět při výskytu specifických událostí, ke kterým nedochází často.

Například při jednom ukládání za programový cyklus S7-200, který trvá 50 ms, bude mít EEPROM životnost 5,000 sekund, což je méně než hodina a půl. Jestliže se naproti tomu hodnota ukládá jednou za hodinu, má EEPROM životnost minimálně 11 let.

Zkopírování paměti V do paměti EEPROM

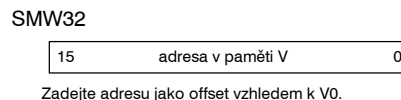
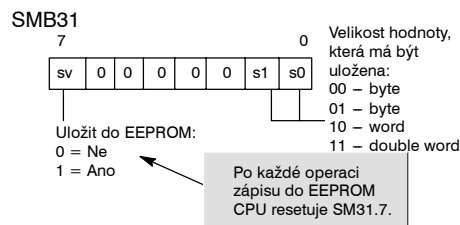
Byte 31 speciální paměti (SMB31) přikáže, aby automat zkopíroval hodnotu z paměti V do paměti EEPROM. V SMW32 je uložena adresa paměti, která má být zkopírována. Obr. 4-20 ukazuje formát SMB31 a SMW32.

Pro naprogramování S7-200 pro uložení nebo zápis konkrétní hodnoty použijte následující kroky:

1. Adresu paměti V, která má být uložena, nahrajte do SMW32.
2. Velikost dat uložte do SM31.0 a SM31.1, jak ukazuje Obr. 4-20.
3. SM31.7 se nastaví na 1.

Na konci každého programového cyklu S7-200 kontroluje SM31.7. Pokud se SM31.7 rovná 1, uloží se daná hodnota do EEPROM. Operace se ukončí tím, že S7-200 resetuje SM31.7 na 0.

Hodnotu v paměti V před ukončením operace neměňte.

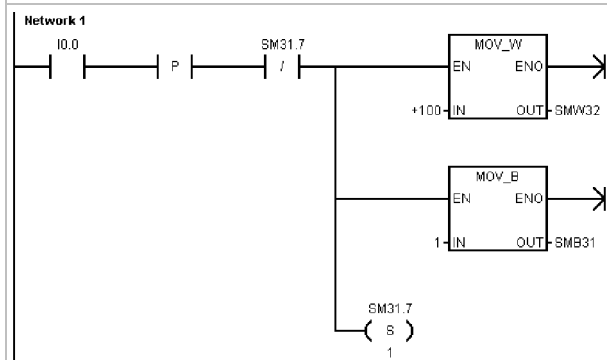


Obr. 4-20 SMB31 a SMW32

Vzorový program: Zkopírování paměti V do paměti EEPROM

V tomto příkladu se do EEPROM zkopíruje VB100. Při náběhové hraně na I0.0 (pokud neprobíhá jiný přesun) se nahraje adresa místa v paměti V, které má být zkopírováno, do SMW32. Vybere se velikost paměti V, která bude zkopírována (1=byte; 2=word; 3=double word nebo real). Pak se SM31.7 nastaví tak, aby automat S7-200 na konci cyklu data přesunul.

S7-200 automaticky zresetuje SM31.7, když je přesun dokončen.



Network 1 //Přesun obsahu paměti V (VB100)
//do paměti EEPROM

```

LD      I0.0
EU
AN      SM31.7
MOVW    +100, SMW32
MOVB    1, SMB31
S        SM31.7, 1

```

Vlastnosti programovatelného automatu S7-200

Automat S7-200 má několik speciálních možností, které uživateli dovolují upravit funkce S7-200 tak, aby lépe vyhovovaly aplikaci.

Okamžité čtení a zápis vstupů a výstupů

Instrukční sada automatu S7-200 obsahuje instrukce pro okamžité čtení a zápis fyzických vstupů a výstupů. Tyto instrukce bezprostředního I/O umožňují přímý přístup na vstupní nebo výstupní bod, i když se normálně jako zdroj nebo cíl I/O operací používá registr obrazu vstupů a výstupů.

Při použití instrukcí okamžitého vstupu se odpovídající bity registru obrazu vstupů nezmění. Při použití instrukcí okamžitého výstupu se odpovídající bity registru obrazu výstupů přepíšou na aktuální hodnotu.



Tip

Pokud není povoleno filtrování analogového vstupu, programovatelný automat S7-200 zachází s analogovými I/O jako s bezprostředními daty. Když se zapíše hodnota na analogový výstup, je tento výstup bezprostředně aktualizován.

Obvykle bývá výhodné používat registr obrazu vstupů a výstupů místo přímého čtení hodnot vstupů a výstupů během provádění programu. Pro použití registrů obrazu existují tři důvody:

- ☐ Vzorkování všech vstupů na začátku programového cyklu synchronizuje a uchovává hodnoty vstupů pro celý programový cyklus. Po provedení programu se výstupy aktualizují podle registru obrazu výstupů, což má stabilizační efekt.
- ☐ Přístup do registru je daleko rychlejší než přístup přímo ke vstupům a výstupům; důsledkem toho je zrychlení celého programu.
- ☐ Vstupy a výstupy jsou bitové hodnoty, a proto se k nim musí přistupovat jako k bitům nebo bytům; do registru obrazu je možné vstupovat po bitech, bytech, word nebo double word. Práce s registrem je proto efektivnější.

Přerušení programového cyklu

Pokud používáte přerušení, jsou podprogramy spojené s každou naprogramovanou událostí uloženy jako součást programu. Podprogramy přerušení se neprovádějí jako součást běžného programového cyklu, ale provedou se tehdy, je-li splněna podmínka pro přerušení (což může nastat v libovolném okamžiku programového cyklu).

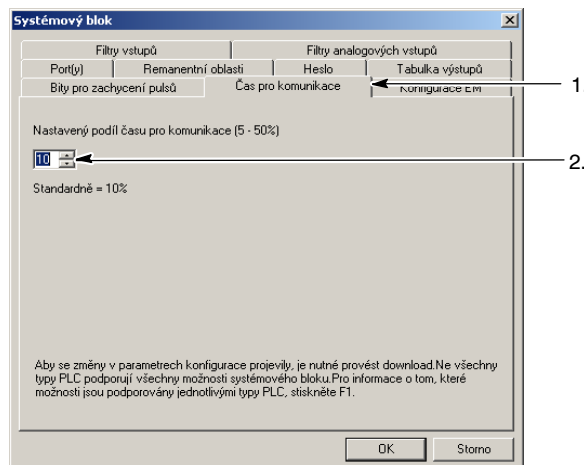
S7-200 obsluhuje přerušení na základě pravidla: kdo přijde první, je jako první obslužen, ale s přihlédnutím k prioritě. Další informace poskytnou instrukce pro přerušení v kapitole 6.

Přidělení doby zpracování pro komunikační úlohy

Je možné nakonfigurovat procento programového cyklu určené pro zpracování požadavků komunikace, které jsou spojeny s editací programu v režimu RUN nebo stavem programu. (Editace v režimu RUN a stav jsou funkce STEP 7 – Micro/WIN, které usnadňují ladění programu). Zvětšením procenta času, určeného pro zpracování požadavků komunikace, se prodlouží doba programového cyklu a řídicí proces probíhá pomaleji.

Implicitně je procento programového cyklu, určené pro zpracování požadavků komunikace, nastavené na 10 %. Toto nastavení bylo zvoleno jako rozumný kompromis mezi zpracováním kompilačních a stavových operací a minimalizuje dopad na řídicí proces. Tuto hodnotu můžete upravovat po 5 % až na maximálních 50 %. Nastavení časového úseku programového cyklu pro komunikaci na pozadí:

1. Vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok** a klikněte na záložku Čas pro komunikace.
2. Editujte vlastnosti času na pozadí pro komunikace a klikněte na OK.
3. Proveďte download modifikovaného systémového bloku do S7-200.

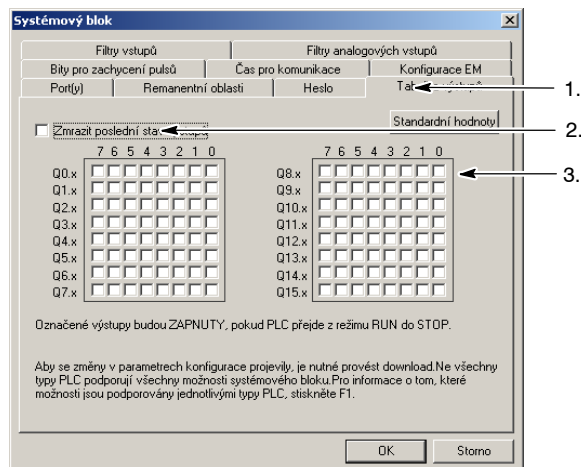


Obr. 4-21 Čas v pozadí pro komunikaci

Nastavení stavu digitálních výstupů v režimu STOP

Při přechodu do stavu STOP umožňuje tabulka výstupů automatu S7-200 nastavit stav digitálních výstupů na přednastavené hodnoty nebo výstupy ponechat ve stavu, ve kterém byly před tímto přechodem. Tabulka výstupů je součástí systémového bloku, který je nahrán a uložen v S7-200 a platí pouze pro digitální výstupy.

1. Vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok** a klikněte na záložku Tabulka výstupů.
2. Chcete-li výstupy ponechat v jejich posledním stavu, zaškrtněte volbu Zmrazit výstupy.
3. Chcete-li kopírovat hodnoty z tabulky na výstupy, vložte je kliknutím na zaškrťovací okénko pro každý výstupní bit, který chcete nastavit po přechodu RUN-STOP na log. 1. (Všechny implicitní hodnoty v tabulce jsou log. 0.)
4. Kliknutím na OK výběr uložíte.
5. Proveďte download modifikovaného systémového bloku do S7-200.



Obr. 4-22 Konfigurace tabulky výstupů

Definování remanentních oblastí paměti, které udržují svoji hodnotu i při výpadku napájení

Můžete definovat až šest remanentních oblastí paměti, které udržují svoji hodnotu při výpadku napájení. Remanentní oblasti se dají definovat v těchto částech paměti: V, M, C a T. Hodnoty si mohou uchovávat pouze remanentní časovače (TONR). Implicitní nastavení prvních 14 bytů paměti M je neremanentní.

Časovače a čítače mohou mít remanentní pouze aktuální hodnoty: bity časovače a čítače nejsou remanentní.

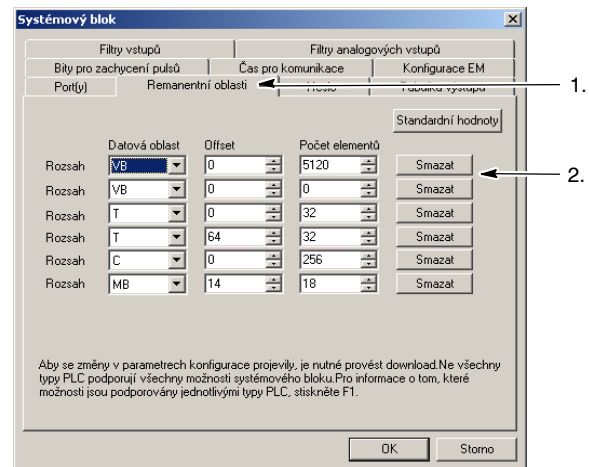


Tip

Změna oblasti MB0 až MB13 na remanentní spustí speciální funkci, která při výpadku napájení automaticky ukládá tuto oblast do EEPROM.

Chcete-li definovat remanentní paměť:

1. Vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok** a klikněte na záložku Remanentní oblasti.
2. Vyberte oblasti paměti, které si mají udržet svoji hodnotu při výpadku napájení, a klikněte na OK.
3. Proveďte download modifikovaného systémového bloku do S7-200.



Obr. 4-23 Remanentní paměť

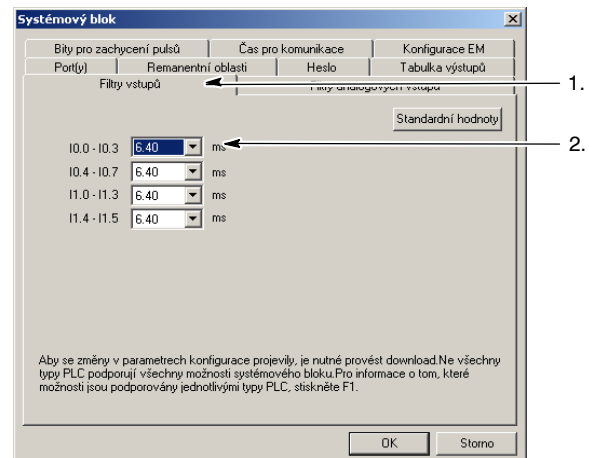
Filtrace digitálních vstupů

S7-200 umožňuje použít vstupní filtr, který definuje čas zpoždění (volitelný od 0,2 ms do 12,8 ms) pro některé nebo pro všechny lokální digitální vstupní body. Toto zpoždění pomáhá filtrovat šum kabeláže, který by mohl způsobit nepředpokládanou změnu stavu na vstupu.

Vstupní filtr je částí systémového bloku, který je nahrán a uložen v S7-200. Implicitní čas filtru je 6,4 ms. Jak je vidět na obrázku 4-24, každá hodnota zpoždění platí pro skupinu vstupů.

Chcete-li nakonfigurovat dobu zpoždění pro vstupní filtr:

1. Vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok** a klikněte na záložku Filtry vstupů.
2. Vložte dobu zpoždění pro každou skupinu vstupů a klikněte na OK.
3. Proveďte download modifikovaného systémového bloku do S7-200.



Obr. 4-24 Konfigurace vstupních filtrů



Tip

Filtr digitálních vstupů ovlivňuje vstupní hodnotu z hlediska použití přerušení od vstupu a zachycení pulzů. V závislosti na konfiguraci filtru by mohl program vynechat událost pro přerušení nebo zachycení pulzů. Vysokorychlostní čítače počítají události na nefiltrovaných vstupech.

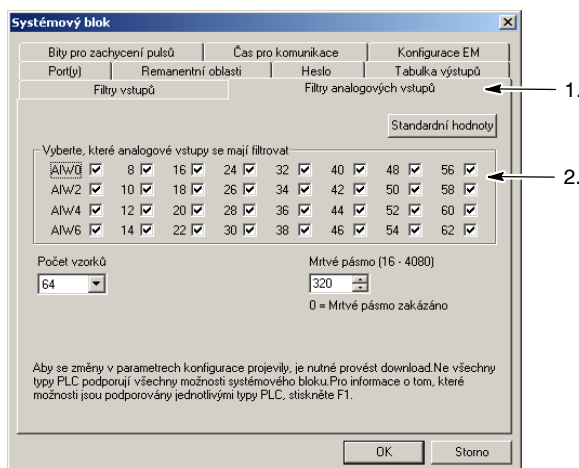
Filtrace analogových vstupů

Pro S7-200 je možné vybrat softwarový filtr pro jednotlivé analogové vstupy. Filtrovaná hodnota je průměr ze součtu zvoleného počtu vzorků analogového signálu. Specifikace filtru (počet vzorků a mrtvé pásmo) je stejná pro všechny filtrované analogové vstupy.

Filtr má velmi rychlou odezvu a umožňuje, aby se rychle promítly i velké změny ve vstupním signálu. Filtr provádí skokovou změnu na poslední hodnotu analogového vstupu, když vstup překročí specifikovanou změnu aktuální hodnoty. Tato změna se nazývá mrtvé pásmo a je dána počtem jednotek digitální hodnoty analogového vstupu.

Implicitně je analogový filtr povolený pro všechny vstupy.

1. Vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok** a klikněte na záložku Filtry analogových vstupů.
2. Vyberte analogové vstupy, které chcete filtrovat, počet vzorků a mrtvé pásmo.
3. Klikněte na OK.
4. Proveďte download modifikovaného systémového bloku do S7-200.



Obr. 4-25 Filtry analogových vstupů



Tip

Analogový filtr nepoužívejte pro moduly, které předávají digitální informace nebo zprávy o poruše analogovými signály. Analogovou filtraci vždy zablokujte pro RTD moduly, termočláňkové moduly a moduly AS-Interface Master.

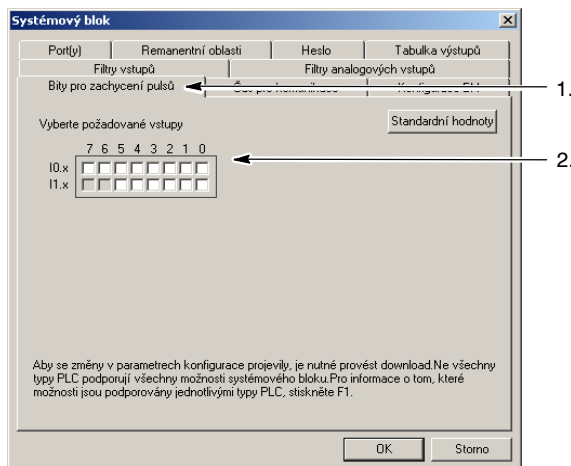
Zachycení krátkých pulzů

S7-200 nabízí pro některé nebo pro všechny lokální digitální vstupy funkci zachycení pulzů. Tato funkce umožňuje zachytit pulzy, které jsou natolik krátké, že je S7-200 na začátku programového cyklu při čtení digitálních vstupů nezaregistruje. Pokud je povolena funkce zachycení pulzů na vstupu, potom bude změna stavu na vstupu pozdržena až do příštího cyklu aktualizace. Tím je zajištěno, že pulz, který trvá pouze krátce, je zachycen a pozdržen, dokud S7-200 nepřečte vstupy.

Funkci zachycení pulzů je možné povolit pro každý lokální digitální vstup jednotlivě.

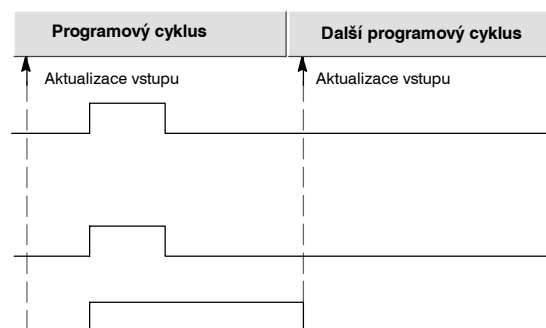
Pro zobrazení konfigurace zachycení pulzů:

1. Vyberte povel menu **Zobrazit > Komponenty > Systémový blok** a klikněte na záložku Bity pro zachycení pulzů.
2. Klikněte na odpovídající zaškrťovací okénko a potvrďte na OK.
3. Proveďte download modifikovaného systémového bloku do S7-200.



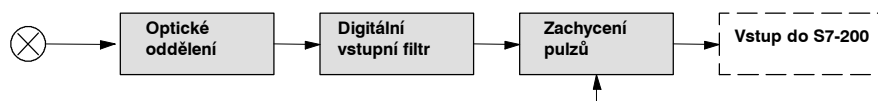
Obr. 4-26 Zachycení pulzů

Obr. 4-27 ukazuje základní princip práce S7-200 s povolenou a blokovanou funkcí zachycení pulzů.



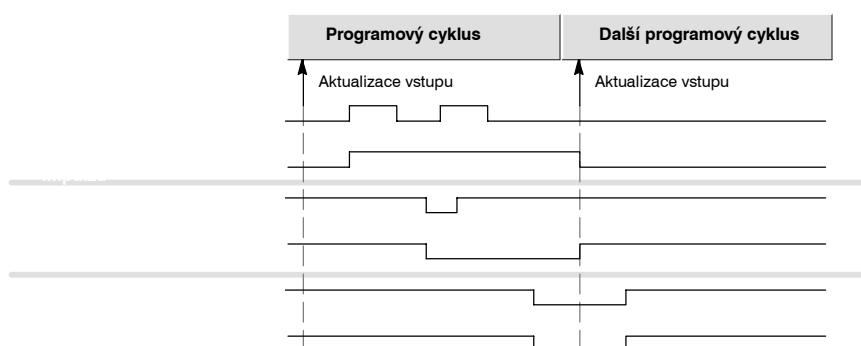
Obr. 4-27 Chod S7-200 s povolenou a blokovanou funkcí zachycení pulzů

Protože funkce zachycení pulzů následuje až po vstupní filtraci, je třeba přizpůsobit čas vstupního filtru tak, aby pulz nebyl filtrem odstraněn. Obr. 4-28 ukazuje blokové schéma vstupního digitálního obvodu.



Obr. 4-28 Digitální vstupní obvod

Obr. 4-29 znázorňuje odezvu povolené funkce zachycení pulzů na různé vstupní podmínky. Pokud se objeví v době programového cyklu více než jeden pulz, zachytí funkce zachycení pulzů jen pulz první. Pokud je v programovém cyklu více pulzů, je třeba naprogramovat obsluhu přerušení spouštěnou náběžnou/sestupnou hranou. (Seznam událostí přerušení je v tabulce 6-44.)



Obr. 4-29 Odezvy funkce zachycení pulzů na různé vstupní podmínky

Ochrana heslem

Všechny modely S7-200 mají ochranu heslem pro omezení přístupu ke konkrétním funkcím.

Heslo opravňuje k přístupu k funkcím a paměti: bez použití hesla je přístup do S7-200 neomezený. Když se použije ochrana heslem, jsou omezeny všechny operace podle konfigurace při programování hesla.

Heslo nerozlišuje malá a velká písmena.

Jak je patrné z tabulky 4-3, S7-200 má tři úrovně omezení přístupu. Každá úroveň umožňuje přístup k určitým funkcím i bez hesla. Pro všechny tři úrovně platí, že po zadání správného hesla je přístup neomezený. Implicitně je S7-200 nastaven na úroveň 1 (bez omezení).

Zadáním hesla se ochrana S7-200 heslem neodstraní.

Tabulka 4-3 Omezení přístupu do S7-200

Funkce CPU	Úroveň 1	Úroveň 2	Úroveň 3
Čtení a zápis uživatelských dat	Přístup povolen	Přístup povolen	Přístup povolen
Spuštění, zastavení a restart CPU			
Čtení a zápis hodin reálného času			
Aktualizace uživatelského programu, dat a konfigurace CPU	Přístup povolen	Přístup povolen	Požadováno heslo
Nahrávání programu do CPU (download)	Přístup povolen	Požadováno heslo	
Stav STL			
Vymazání programového, datového nebo systémového bloku			
Vnucení hodnoty (force) nebo provedení jediného či více programových cyklů			
Kopírování do paměťového modulu			
Zápis výstupů v režimu STOP			

Oprávnění přístupu daného uživatele k omezeným funkcím neopravňuje jiné uživatele přistupovat k těmto funkcím. V daném čase může mít neomezený přístup k S7-200 pouze jeden uživatel.



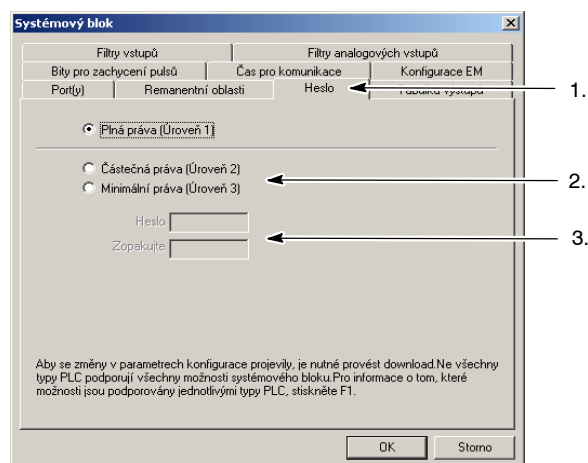
Tip

Po zadání hesla zůstává úroveň oprávnění pro toto heslo v platnosti ještě jednu minutu po odpojení počítače od S7-200. Vždy ukončete program STEP 7-Micro/WIN před odpojením kabelu, abyste zabránili jinému uživateli v přístupu k programu.

Naprogramování hesla pro S7-200

Dialogové okno Systémový blok (Obr. 4-30) umožňuje naprogramovat heslo pro S7-200:

1. Vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok**, zobrazí se dialogové okno Systémový blok; pak klikněte na Heslo.
2. Vyberte příslušnou úroveň přístupu k S7-200.
3. Zadejte dvakrát heslo.
4. Klikněte na OK.
5. Proveďte download modifikovaného systémového bloku do S7-200.



Obr. 4-30 Naprogramování hesla

Obnovení ztraceného hesla

Pokud zapomenete heslo, je třeba smazat paměť S7-200 a opět do ní nahrát program. Smazáním paměti se S7-200 dostane do režimu STOP, je nastaven na své implicitní hodnoty, nastavené ve výrobě, s výjimkou síťové adresy, přenosové rychlosti a hodin reálného času. Chcete-li vymazat program v automatu S7-200:

1. Vyberte příkaz menu **PLC > Smazat** – objeví se dialogové okno Smazat.
2. Vyberte všechny tři bloky a svoji činnost potvrďte kliknutím na OK.
3. Pokud bylo naprogramováno heslo, STEP 7 – Micro/WIN zobrazí dialogové okno hesla pro oprávnění k přístupu. Pro vymazání hesla zadejte CLEARPLC v dialogovém okně pro heslo a operace smazání pokračuje. (Heslo CLEARPLC nerozlišuje malá a velká písmena.)

Operace smazání neodstraní program z paměťového modulu. Protože se program do paměťového modulu ukládá spolu s heslem, je nutné přeprogramovat i paměťový modul, a odstranit tak heslo i z něj.



Varování

Po vymazání paměti S7-200 se výstupy nastaví na hodnotu log. 0 (nebo v případě analogových výstupů zůstanou zmrazené na zadané hodnotě).

Pokud je automat S7-200 připojený k řízenému zařízení v době mazání paměti, změny stavu výstupů mohou být přeneseny na řízené zařízení. Pokud jste nakonfigurovali "bezpečný stav" výstupů jinak, než bylo zadáno ve výrobě, změny na výstupech mohou zapříčinit nepředvídatelný stav zařízení, což může způsobit smrt nebo vážné zranění osob a poškození zařízení.

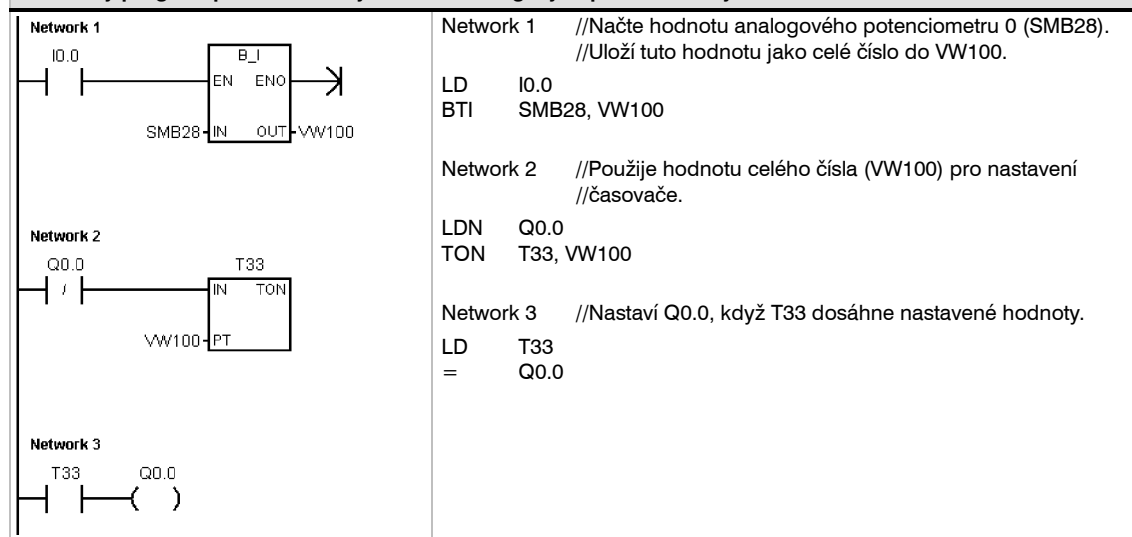
Vždy dodržujte příslušná bezpečnostní opatření. Před vymazáním paměti S7-200 zajistěte, aby byl celý proces v bezpečném stavu.

Analogové potenciometry

Analogové potenciometry jsou umístěny pod předním krytem modulu. Těmito potenciometry můžete zvýšit nebo snížit hodnoty uložené v bytech speciální paměti (SMB). Hodnoty se dají pouze číst, v programu mohou být použity pro různé funkce, např. pro nastavení hodnot časovače nebo čítače, pro zadávání nebo změnu přednastavených hodnot nebo pro nastavování limitů. Pro nastavení potenciometru použijte malý šroubovák: pro zvýšení hodnoty otáčejte potenciometrem ve směru hodinových ručiček (doprava), pro snížení hodnoty proti směru hodinových ručiček (doleva).

SMB28 obsahuje číselnou hodnotu, která reprezentuje polohu analogového potenciometru 0. SMB29 obsahuje číselnou hodnotu, která reprezentuje polohu analogového potenciometru 1. Analogová hodnota má nominální rozsah 0 až 255 a opakovatelnost ± 2 .

Ukázkový program použití hodnoty vložené analogovými potenciometry



Vysokorychlostní vstupy a výstupy

Vysokorychlostní čítače

Automat S7-200 má integrované vysokorychlostní čítače, které dokáží počítat externí události s vysokou rychlostí bez omezení činnosti S7-200. Rychlosti jednotlivých typů CPU najdete v příloze A. Každý čítač má vstupy vyhrazené pro hodinové pulzy, řízení směru, reset a start, pokud tyto funkce podporuje. Pro změnu rychlosti čítače je možné vybrat různé kvadrurní režimy. Více informací o vysokorychlostních čítačích najdete v kapitole 6.

Vysokorychlostní pulzní výstup

Programovatelný automat S7-200 podporuje vysokorychlostní pulzní výstupy Q0.0 a Q0.1, které mohou generovat sérii rychlých výstupních pulzů (PTO) nebo pulzní šířkovou modulaci (PWM).

Funkce PTO vytváří na výstupu obdélníkové pulzy (střída 1:1) s definovaným počtem pulzů (od 1 do 4.294.967.295 pulzů) a definovanou periodou (v mikrosekundách od 50 μ s do 65.535 μ s, nebo v milisekundách od 2 ms do 65.535 ms). Funkce PTO může být naprogramována tak, aby generovala buď jednu sérii pulzů, nebo pulzní profil s více sériemi pulzů. Pulzní profil je například možné použít pro řízení krokového motoru prostřednictvím jednoduché posloupnosti rozběh, chod, doběh nebo prostřednictvím složitějších posloupností. Pulzní profil se může skládat až z 225 segmentů, kde segment odpovídá rozběhu nebo chodu nebo doběhu.

Funkce PWM má pevnou frekvenci pulzů s různou šířkou pulzu a mezery; perioda a šířka pulzu se zadává v mikrosekundách nebo milisekundách. Perioda se může pohybovat v rozsahu od 50 μ s do 65.535 μ s nebo od 2 ms do 65.535 ms. Šířka pulzu má rozsah od 0 μ s do 65.535 μ s nebo od 0 ms do 65.535 ms. Jestliže se šířka pulzu rovná délce trvání jedné periody, tvoří pulz 100 % periody a výstup je stále zapnutý. Jestliže je šířka pulzu nulová, tvoří pulz 0 % periody a výstup je stále vypnutý.

Více informací o vysokorychlostních výstupech najdete v kapitole 6.

Pojmy, konvence a možnosti programování



Programovatelný automat S7-200 nepřetržitě vykonává uživatelský program, aby řídil úlohu nebo proces. Pro vytvoření programu a jeho downloadu do S7-200 se používá STEP 7–Micro/WIN, který obsahuje mnoho různých nástrojů a funkcí pro návrh, implementaci a ladění programu.

Přehled kapitoly

Návod pro návrh systému	48
Základní prvky programu	49
Použití STEP 7–Micro/WIN pro vytvoření programu	51
Volba mezi instrukčním souborem SIMATIC a IEC 1131–3	53
Vysvětlení konvencí použitých v programových editorech	54
Použití průvodců jako pomůcky pro tvorbu řídicího programu	56
Ošetření chyb	56
Přiřazení adres a počátečních hodnot v editoru datového bloku	58
Použití tabulky symbolů pro symbolické adresování proměnných	58
Použití lokálních proměnných	59
Použití stavového diagramu pro monitorování uživatelského programu	59
Vytvoření knihovny instrukcí	60
Funkce pro ladění uživatelského programu	60

Návod pro návrh systému

Existuje mnoho způsobů, jak navrhout mikro–PLC systém. Následující obecné pokyny se dají aplikovat na mnoho konstrukčních projektů. Musíte se pochopitelně řídit směrnicemi vaší společnosti a zavedenou praxí, danou vašimi zkušenostmi a působišťem.

Rozdělení procesu nebo stroje

Rozdělte proces nebo stroj, který chcete řídit, na části s jistou úrovní vzájemné nezávislosti. Toto rozdělení určí hranice mezi řídicími systémy a ovlivní funkční specifikaci a přiřazení prostředků.

Vytvoření funkční specifikace

Popište operace každé části procesu nebo stroje. Zahrňte následující témata: vstupní a výstupní body; funkční popis provozu; stavy, kterých musí být dosaženo, aby byla povolena činnost každého akčního členu (např. solenoidy, motory a pohony); popis operátorského rozhraní a dalších rozhraní s ostatními částmi procesu nebo stroje.

Návrh bezpečnostních obvodů

Specifikujte zařízení, které kvůli bezpečnosti vyžaduje pevně zapojenou logiku. Řídicí zařízení mohou selhat a způsobit nečekané spuštění nebo změnu stavu řízených strojů. Tam, kde by neočekávaný nebo nesprávný provoz strojů mohl mít za následek zranění obsluhy nebo významné majetkové škody, by se měly použít elektromechanické ochrany, pracující nezávisle na S7-200; zabrání se tak nebezpečným operacím. Do návrhu bezpečnostních obvodů by měly být zahrnuty následující úlohy:

- ☐ Specifikujte nesprávné nebo neočekávané operace akčních členů, které by mohly být nebezpečné.
- ☐ Specifikujte podmínky, které vás ujistí o tom, že provoz není nebezpečný, a určete, jak tyto podmínky detekovat nezávisle na S7-200.
- ☐ Specifikujte, jak při připojení nebo odpojení napájení a při zjištění chyb ovlivní CPU S7-200 a jeho vstupy a výstupy proces. Tyto informace by se měly použít pouze při projektování pro normální a předpokládaný mimořádný provoz; nemělo by se na ně spoléhat pro bezpečnostní účely.
- ☐ Navrhněte manuální nebo elektromechanické ochrany, které zablokuje nebezpečné operace nezávisle na S7-200.
- ☐ Pro S7-200 zajistěte příslušné stavové informace z nezávislých obvodů tak, aby program a kterékoliv operátorské rozhraní byly dostatečně informovány.
- ☐ Specifikujte jakékoliv jiné požadavky, vztahující se k bezpečnosti procesu.

Specifikace operátorských stanic

Na základě požadavků funkční specifikace nakreslete výkresy operátorských stanic. Zahrňte následující položky:

- ☐ Přehled umístění každé operátorské stanice ve vztahu k procesu nebo stroji
- ☐ Mechanické uspořádání zařízení na operátorské stanici, jako jsou displeje, přepínače a signálky
- ☐ Elektrotechnická schémata s příslušnými vstupy a výstupy CPU S7-200 a rozšiřovacími moduly

Tvorba schémat konfigurace

Na základě požadavků funkční specifikace nakreslete schémata konfigurace řídicího zařízení. Zahrňte následující položky:

- ☐ Přehled umístění každého automatu ve vztahu k procesu nebo stroji
- ☐ Mechanické uspořádání jednotek S7–200 a rozšiřujících I/O modulů (včetně rozvaděčů a dalšího zařízení)
- ☐ Elektrotechnická schémata pro každý automat a rozšiřující I/O modul (včetně čísla modelu přístroje, komunikační adresy a adres vstupů a výstupů).

Vytvoření seznamu symbolických názvů (volitelné)

Pokud chcete používat pro adresování symbolické názvy, vypracujte jejich seznam pro absolutní adresy. Zahrňte nejen fyzické I/O signálky, ale i jiné prvky, které mají být použity v uživatelském programu.

Základní prvky programu

Programový blok se skládá ze spustitelného kódu a komentáře. Spustitelný kód se skládá z hlavního programu a dalších podprogramů nebo přerušení. Kód je kompilován a nahrán do S7-200; komentář k programu nikoliv. Pro zpřehlednění řídicího programu můžete použít organizační prvky (hlavní program, podprogramy a přerušení).

Následující příklad ukazuje program, který obsahuje hlavní program a podprogram přerušení. Tento vzorový program používá časové přerušení pro odečítání hodnoty analogového vstupu každých 100 ms.

Příklad: Základní prvky programu		
M A I N		Network 1 //V prvním cyklu volá podprogram 0. LD SM0.1 CALL SBR_0
S B R 0		Network 1 //Nastaví interval na 100 ms //pro časové přerušení. //Povolí přerušení 0. LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI
I N T 0		Network 1 //Vzorkuje analogový vstup 4. LD SM0.0 MOVW AIW4, VW100

Hlavní program

Hlavní program obsahuje instrukce, které řídí aplikaci. S7-200 vykonává tyto instrukce postupně za sebou v časovém sledu jednou za programový cyklus. Hlavní program se označuje jako OB1.

Podprogramy

Tyto volitelné prvky uživatelského programu se provádějí pouze tehdy, jsou-li volány: hlavním programem, podprogramem přerušení nebo jiným podprogramem. Podprogramy jsou užitečné v případě, kdy chcete nějakou funkci provádět opakovaně. Pohodlnější než přepisovat logiku pro každé místo v programu, kde chcete, aby se daná funkce vyskytla, je zapsat logiku jednou do podprogramu a volat podprogram tolikrát, kolikrát je to během hlavního programu potřeba. Podprogramy mají několik výhod:

- ☐ Použití podprogramů zmenšuje celkovou velikost uživatelského programu.
- ☐ Použití podprogramu zkracuje dobu trvání programového cyklu, protože se tak přesune kód mimo hlavní program. S7-200 vyhodnocuje kód v hlavním programu v každém programovém cyklu, ať se kód provádí, nebo ne. V podprogramu ale vyhodnocuje S7-200 kód pouze tehdy, je-li tento podprogram volán; v programových cyklech, ve kterých volán není, ho nevyhodnocuje.
- ☐ Použití podprogramů vytváří kód, který je přenositelný. Kód funkce je možné uložit jako podprogram a ten pak zkopírovat do jiných programů pouze s minimálními úpravami.



Tip

Použití adres v paměti V může omezit přenositelnost podprogramu, protože je možné, že přiřazení adres paměti V jednoho programu může kolidovat s přiřazením v jiném programu. Naproti tomu podprogramy, které pro všechna přiřazení adres používají tabulku lokálních proměnných (L paměť), jsou snadno přenositelné, protože při použití lokálních proměnných nevzniknou konflikty adres mezi podprogramem a jinou částí programu.

Podprogramy přerušení

Tyto volitelné prvky uživatelského programu reagují na konkrétní události přerušení. Program navrhnete tak, aby ošetřil předdefinovanou událost přerušení. Kdykoliv ke specifikované události dojde, S7-200 provede podprogram přerušení.

Podprogramy přerušení nejsou volány hlavním uživatelským programem. Vy pouze přiřadíte k definované události podprogram přerušení a S7-200 provede instrukce podprogramu pouze při výskytu této události.



Tip

Protože není možné předvídat, kdy automat generuje přerušení, je žádoucí omezit počet proměnných, které se používají jak v podprogramu přerušení, tak kdekoli v hlavním programu.

Abyste zajistili, že přerušení používá pouze dočasnou paměť a nepřepisuje data používaná jinde v programu, používejte v podprogramu přerušení tabulku lokálních proměnných.

Existuje mnoho programovacích technik, které můžete použít, abyste zajistili správné sdílení dat mezi hlavním programem a podprogramy přerušení. Tyto techniky jsou popsány v kapitole 6 spolu s instrukcemi přerušení.

Ostatní prvky programu

Ostatní bloky obsahují informace pro S7-200. Download těchto bloků můžete provést spolu s downloadem uživatelského programu.



Systémový blok

Systémový blok

Systémový blok umožňuje konfigurovat různé volitelné části hardwaru pro S7-200.



Datový blok

Datový blok

Datový blok uchovává hodnoty proměnných (paměť V) používaných uživatelským programem. Datový blok můžete použít pro vložení počátečních hodnot dat.

Použití STEP 7–Micro/WIN pro vytvoření programu

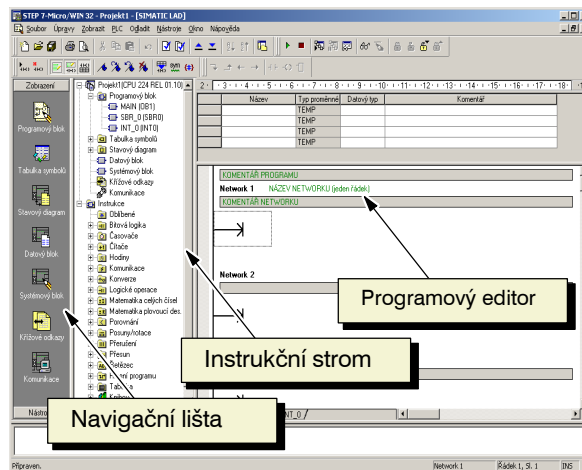
Chcete-li otevřít STEP 7–Micro/WIN, dvakrát klikněte na ikonu STEP 7–Micro/WIN nebo vyberte příkaz menu **Start > SIMATIC > STEP 7 MicroWIN 3.2**. Jak je vidět na obrázku 5-1, okno projektu STEP 7–Micro/WIN nabízí přehledné pracovní prostředí pro vytváření řídicího programu.

Nástrojové lišty obsahují tlačítka pro zkratky k často používaným příkazům menu. Všechny nástrojové lišty můžete zobrazit nebo skryt.

Navigační lišta nabízí ikony pro přístup k různým programovacím prvkům STEP 7–Micro/WIN.

Strom s instrukcemi zobrazuje všechny objekty projektu a instrukce pro tvorbu řídicího programu. Jednotlivé instrukce můžete do programu vložit metodou drag&drop nebo můžete na instrukci dvakrát kliknout, čímž ji vložíte na současnou pozici kurzoru v programovém editoru.

Programový editor obsahuje program a tabulku lokálních proměnných, ve které můžete přiřadit symbolické názvy dočasným lokálním proměnným. Podprogramy a přerušení jsou zobrazeny jako záložky ve spodní části okna programového editoru. Chcete-li přecházet mezi podprogramy, přerušeními a hlavním programem, klikněte na příslušnou záložku.



Obr. 5-1 STEP 7–Micro/WIN



Programový editor

STEP 7–Micro/WIN obsahuje tři editory pro vytváření uživatelského programu: kontaktní schémata (LAD), výpis příkazů (STL) a funkční bloky (FBD). S určitými omezeními mohou být programy, psané v kterémkoliv z těchto programových editorů, prohlíženy a editovány ostatními editory.

Vlastnosti editoru STL

Editor STL zobrazuje program jako znakově orientovaný programovací jazyk. Umožňuje vytvářet řídicí programy vkládáním textových instrukcí. Editor STL také umožňuje tvorbu programů, které by pomocí editorů LAD nebo FBD nešlo vytvořit. Je to proto, že v STL programujete v jazyku S7-200, a nikoli v jazyku grafického editoru, kde platí určitá omezení, aby byly diagramy správně nakresleny. Jak je vidět na obrázku 5-2, je tato znakově orientovaná koncepce velmi podobná programování ve strojovém kódu.

S7-200 provádí každou instrukci v pořadí určeném programem shora dolů; pak začne opět odshora.

Pro řešení řídicí logiky používá STL logický zásobník. Vy vkládáte instrukce STL pro manipulaci se zásobníkem.

LD	I0.0	//Načtení jednoho vstupu
A	I0.1	//Logický součin s jiným vstupem
=	Q1.0	//Zapsání hodnoty na výstup 1

Obr. 5-2 Příklad programu STL

Jestliže pro programování volíte editor STL, vezměte v úvahu tyto hlavní body:

- ☐ STL je nejvhodnější pro zkušené programátory.
- ☐ STL někdy umožní řešit problémy, které se nedají snadno řešit pomocí editorů LAD a FBD.
- ☐ Editor STL můžete používat pouze s instrukčním souborem SIMATIC.
- ☐ Editor STL můžete vždy použít na prohlížení nebo editaci programu, který byl vytvořen pomocí editorů LAD nebo FBD; naopak to ale není vždy možné. Nemůžete vždy použít editory LAD nebo FBD pro zobrazení programu, který byl napsán v editoru STL.

Vlastnosti editoru LAD

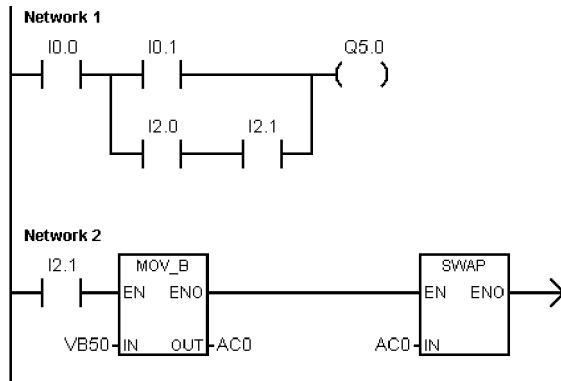
Editor LAD zobrazuje program v grafické formě podobné schémátům. Programy v kontaktním schématu umožňují simulovat tok elektrického proudu z napájecího zdroje přes řadu logických vstupních podmínek, které následně aktivují výstupní logické podmínky. Program LAD obsahuje levou napájecí lištu, která je pod napětím. Kontakty, které jsou sepnuté, umožňují tok energie do dalšího prvku; kontakty, které jsou rozepnuté, tok energie blokují.

Logika je dělena do spojitých sítí (network). CPU provádí vždy jeden network zleva doprava a pak shora dolů tak, jak je to určeno programem. Obrázek 5-3 ukazuje příklad programu LAD. Instrukce jsou představovány grafickými symboly a mají tři základní formy.

Kontakty představují logické vstupní podmínky, jako jsou spínače, tlačítka nebo vnitřní podmínky.

Cívky obvykle představují logické výstupní obvody, jako jsou žárovky, startéry motorů, přechodová relé nebo vnitřní výstupní podmínky.

Bloky představují doplňující instrukce, jako jsou časovače, čítače nebo matematické instrukce.



Obr. 5-3 Příklad programu LAD

Jestliže zvolíte pro programování editor LAD, vezměte v úvahu tyto hlavní body:

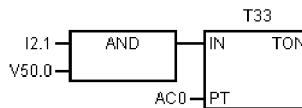
- ☐ Kontaktní schémata snadno používají i programátoři začátečníci.
- ☐ Grafické zobrazení se snadno chápe a je populární na celém světě.
- ☐ Editor LAD je možné použít s instrukčním souborem SIMATIC i IEC 1131–3.
- ☐ Pro zobrazení programu vytvořeného editorem SIMATIC LAD můžete vždy použít editor STL.

Vlastnosti editoru FBD

Editor FBD zobrazuje program v grafické formě, která připomíná běžná logická schémata. Neobsahuje kontakty ani cívky, které se nalézají v editoru LAD, ale ekvivalentní instrukce, které se objevují jako blokové instrukce.

Obrázek 5-4 ukazuje příklad programu FBD.

FBD nepoužívá pojem levých a pravých napájecích lišt, proto se termín “signálový tok” používá pro vyjádření analogického pojmu toku řízení přes logické bloky FBD.



Obr. 5-4 Příklad programu FBD

Cesta logické “1” přes prvky FBD se nazývá signálový tok. Vstup a výstup signálového toku je možné přímo přiřadit operandu.

Logika programu je odvozena od spojení těchto bloků. To znamená, že výstup jedné instrukce (např. blok AND) je možné použít pro aktivaci jiné instrukce (např. časovače). Takto vytvoříte potřebnou logiku řízení. Tato koncepce spojování umožní vyřešit mnoho různých logických problémů.

Jestliže zvolíte pro programování editor FBD, vezměte v úvahu tyto hlavní body:

- ☐ Styl zobrazení pomocí grafických logických členů umožňuje dobře sledovat tok programu.
- ☐ Editor FBD je možné použít s instrukčním souborem SIMATIC i IEC 1131–3.
- ☐ Pro zobrazení programu vytvořeného editorem SIMATIC FBD můžete vždy použít editor STL.

Volba mezi instrukčním souborem SIMATIC a IEC 1131–3

Většina programovatelných automatů nabízí podobné základní instrukce; obvykle jsou ale mezi jednotlivými výrobci drobné rozdíly ve vzhledu, provozu apod. V průběhu posledních několika let vypracovala Mezinárodní elektrotechnická asociace (IEC) globální normu, která se speciálně vztahuje na mnoho aspektů programování automatů. Tato norma doporučuje různým výrobcům programovatelných automatů, aby nabízeli instrukce, které jsou stejné jak svým vzhledem, tak funkcí.

Automat S7-200 obsahuje dva instrukční soubory, které umožní řešit velké množství různých automatizačních úloh. Instrukční soubor IEC odpovídá normě IEC 1131–3 pro programování PLC a instrukční soubor SIMATIC je navržen speciálně pro automat S7-200.



Tip

Pokud je program STEP 7–Micro/WIN nastaven do režimu IEC, zobrazí se v Instrukčním stromu vedle instrukcí, které nejsou definovány normou IEC 1131–3, červený kosočtverec (♦).

Mezi instrukčními soubory SIMATIC a IEC existuje jen několik klíčových rozdílů:

- ☐ Instrukční soubor IEC se omezuje na ty instrukce, které jsou mezi prodejci programovatelných automatů standardní. Některé instrukce, které jsou běžně zahrnuty do souboru SIMATIC, nejsou standardními instrukcemi ze specifikace IEC 1131–3. Jsou stále k dispozici pro užívání jako nestandardní instrukce; pokud je ale použijete, není program nadále kompatibilní s IEC 1131–3.
- ☐ Některé blokové instrukce IEC akceptují více datových formátů. Této praxi se často říká vícenásobná definice (overloading). Například místo matematických bloků `ADD_I` (přičtení celého čísla) a `ADD_R` (přičtení reálného čísla) přezkoumá IEC instrukce `ADD` formát přičítaných dat a automaticky zvolí správnou instrukci S7-200. Tím se dá ušetřit cenný čas při navrhování programu.
- ☐ Pokud použijete instrukce IEC, jsou parametry instrukcí automaticky kontrolovány na správný datový formát, jako například celé číslo se znaménkem (signed integer) nebo celé číslo bez znaménka (unsigned integer). Jestliže se například pokusíte provést s hodnotou integer instrukci, která očekává bitovou hodnotu (zapnuto/vypnuto), je výsledkem chybové hlášení. Tato vlastnost pomáhá minimalizovat syntaktické chyby programu.

Jestliže volíte mezi instrukčním souborem SIMATIC a IEC, vezměte v úvahu následující body:

- ☐ Instrukce SIMATIC mají obvykle kratší dobu zpracování. Některé instrukce IEC mohou mít delší dobu zpracování.
- ☐ Některé instrukce IEC, například instrukce časovačů, čítačů, násobení a dělení, se chovají jinak než jejich ekvivalenty SIMATIC.
- ☐ S instrukčním souborem SIMATIC můžete použít všechny tři programové editory (LAD, STL, FBD). Pro instrukce IEC můžete použít pouze programové editory LAD a FBD.
- ☐ Funkce instrukcí IEC je standardní pro různé značky programovatelných automatů a znalost tvorby programu odpovídajícího IEC je podporována platformami různých programovatelných automatů.
- ☐ Norma IEC sice definuje méně instrukcí, než nabízí instrukční soubor SIMATIC, do programu IEC je ale vždy možné zahrnout instrukce SIMATIC.
- ☐ Norma IEC 1131–3 vyžaduje přiřazení typů k proměnným a podporuje systémovou kontrolu datového typu.

Vysvětlení konvencí použitých v programových editorech

Ve všech programových editorech používá STEP 7–Micro/WIN následující konvence:

- ☐ # před jménem symbolu (#var1) znamená, že symbol má lokální platnost.
- ☐ V instrukcích IEC označuje symbol % přímou adresu.
- ☐ Symbol operandu “?.?” nebo “????” znamená, že se požaduje zadání operandu.

Programy LAD jsou rozděleny do segmentů, nazývaných networky. Network je uspořádaná sestava kontaktů, cívek a bloků, které jsou propojeny tak, že tvoří úplný obvod: neexistují v něm zkraty, přerušení obvodu ani podmínky zpětného signálového toku. STEP 7–Micro/WIN umožňuje tvorbu komentáře pro program LAD systémem jeden network za druhým. Při programování editorem FBD se pojem networku používá pro rozdělení programu na segmenty a vypracování komentáře.

Programy STL nepoužívají networky; můžete ale použít klíčové slovo NETWORK pro rozdělení programu na segmenty.

Konvence charakteristické pro editor LAD

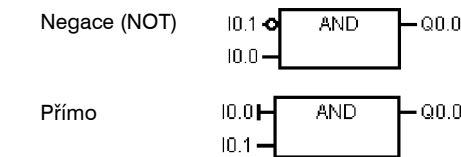
V editoru LAD můžete použít klávesy F4, F6 a F9 klávesnice pro přístup k instrukcím kontaktů, bloků a cívek. Editor LAD používá následující konvence:

- ☐ Symbol “--->” znamená přerušený obvod nebo požadovaný přípoj signálového toku.
- ☐ Symbol “→” znamená, že výstup je volitelnou součástí signálového toku pro instrukci, která může být uspořádána kaskádovitě nebo zapojena sériově.
- ☐ Symbol “>>” znamená, že můžete použít signálový tok.

Konvence charakteristické pro editor FBD

V editoru FBD můžete použít klávesy F4, F6 a F9 pro přístup k instrukcím AND, OR a blokovým instrukcím. Editor FBD používá následující konvence:

- ☐ Symbol “--->” na operandu EN je signálový tok nebo indikátor operandu. Může také znamenat přerušený obvod nebo požadovaný přípoj signálového toku.
- ☐ Symbol “→” znamená, že výstup je volitelná součást signálového toku pro instrukci, která může být uspořádána kaskádovitě nebo zapojena sériově.
- ☐ Symboly “<<” a “>>” znamenají, že můžete použít buď hodnotu, nebo signálový tok.
- ☐ Negace: Negace nebo inverze operandu nebo signálového toku se označují malým kroužkem u vstupu. Na obrázku 5-5 se Q0.0 rovná negaci logického součinu I0.0 a I0.1 (NOT I0.0 AND I0.1). Negace platí pouze pro booleovské signály, které mohou být specifikovány jako parametry nebo signálový tok.
- ☐ Indikátory okamžitého stavu: Jak je vidět na obrázku 5-5, editor FBD zobrazuje okamžitý stav boolovského operandu svislou čarou na vstupu pro instrukci FBD. Indikátor okamžitého stavu způsobí okamžité načtení ze specifikovaného fyzického vstupu. Okamžité operátory platí pouze pro fyzické vstupy.
- ☐ Blok bez vstupu nebo výstupu: Blok bez vstupu označuje instrukci, která je nezávislá na signálovém toku.



Obr. 5-5 Konvence FBD



Tip

Počet operandů pro instrukce AND a OR je možné rozšířit až na 32 vstupů. Pro zvětšení nebo zmenšení počtu operandů použijte klávesy “+” a “-” na klávesnici.

Obecné konvence pro programování automatu S7-200

Definování EN/ENO

EN (Enable IN) je booleovský vstup bloků v LAD a FBD. Aby byla bloková instrukce provedena, musí být na tomto vstupu přítomen signálový tok. V editoru STL nemají instrukce vstup EN; aby byla odpovídající instrukce provedena, musí být hodnota vrcholu zásobníku logická 1.

ENO (Enable Out) je booleovský výstup bloků v LAD a FBD. Pokud má blok signálový tok na vstupu EN a provede svou funkci bez chyby, výstup ENO předá signálový tok dalšímu prvku. Pokud je při provádění bloku zjištěna chyba, je signálový tok přerušen na bloku, který chybu generoval.

V editoru STL výstup ENO neexistuje, ale instrukce v STL, které odpovídají instrukcím LAD a FBD s výstupy ENO, nastaví speciální bit ENO. K tomuto bitu se přistupuje pomocí instrukce AND ENO (AENO) a je možné ho použít stejně jako výstup bloků ENO.



Tip

Operandy a datové typy EN/ENO nejsou uvedeny v tabulce platných operandů u každé instrukce, protože jsou tyto operandy pro všechny instrukce LAD a FBD stejné. Tabulka 5-1 uvádí tyto operandy a datové typy pro LAD a FBD. Tyto operandy platí pro všechny instrukce LAD a FBD, uvedené v této příručce.

Tabulka 5-1 Operandy a datové typy EN/ENO pro LAD a FBD

Programový editor	Vstupy/Výstupy	Operandy	Typy dat
LAD	EN, ENO	Signálový tok	BOOL
FBD	EN, ENO	I, Q, V, M, SM, S, T, C, L,	BOOL

Podmíněné/nepodmíněné vstupy

V LAD a FBD jsou bloky a cívky, které jsou závislé na signálovém toku, spojené s libovolným prvkem na levé straně. Cívky a bloky, které na signálovém toku nezávisí, jsou připojené přímo k levé napájecí liště. Tabulka 5-2 ukazuje příklad podmíněného i nepodmíněného vstupu.

Tabulka 5-2 Zobrazení podmíněných a nepodmíněných vstupů

Signálový tok	LAD	FBD
Instrukce závislejší na signálovém toku (podmíněná)		
Instrukce nezávislejší na signálovém toku (nepodmíněná)		

Instrukce bez výstupů

Bloky, které nemohou být řazeny do kaskády, se zobrazují bez booleovských výstupů. Patří mezi ně instrukce "Volání podprogramu", "Skok" a "Podmíněné ukončení programu". Existují také instrukce, které je možné umístit pouze na levou napájecí lištu. Patří mezi ně instrukce "Návěští", "Next", "Načíst SCR", "Podmíněný konec SCR" a "Konec SCR". Ty jsou v FBD zobrazeny jako bloky; poznají se podle neoznačených výkonových vstupů a podle toho, že nemají výstupy.

Instrukce porovnání

Instrukce porovnání se provádí bez ohledu na stav signálového toku. Pokud není signálový tok, má výstup hodnotu logická 0. Pokud je signálový tok přítomen, je výstup nastaven podle výsledku porovnání. Instrukce porovnání SIMATIC FBD, IEC Ladder a IEC FBD jsou zobrazovány jako bloky, přestože se operace provádějí jako kontakt.

Použití průvodců jako pomůcky pro tvorbu řídicího programu

STEP 7 – Micro/WIN obsahuje průvodce, které usnadňují jednotlivé aspekty programování a automatizují je. V kapitole 6 jsou instrukce, které mají svého průvodce, označeny ikonou:



Průvodce

Ošetření chyb S7-200

Automat S7-200 dělí chyby na chyby kritické a nekritické. Chybové kódy, generované vzniklou chybou, si můžete prohlédnout pomocí příkazu menu **PLC > Informace**.

Obrázek 5-6 ukazuje dialogové okno "PLC Information", kde je zobrazen chybový kód a popis chyby.

Políčko "Poslední kritická" ukazuje poslední kód kritické chyby, generovaný S7-200. Tato hodnota se uchová i po výpadku napájení, pokud je uchována paměť RAM. Je však vymazána při smazání celé paměti S7-200 nebo když se RAM vymaže v důsledku dlouhodobého výpadku napájení.

Políčko "Celkem kritických" obsahuje počet kritických chyb, generovaných od posledního smazání všech paměťových oblastí S7-200. Tato hodnota se uchovává při výpadku a obnovení napájení, pokud je uchována paměť RAM. Je však vymazána při smazání celé paměti S7-200 nebo když se RAM vymaže v důsledku dlouhodobého výpadku napájení.

V příloze C jsou uvedeny chybové kódy S7-200; v příloze D jsou popsány bity speciální paměti (SM), které je možné použít pro monitorování chyb.

Modul	Typ	V...	Start	V...	Start	Stav
PLC	Digitální I/O	16	I0.0	16	Q0.0	Bez chyby
0	EM253 Position	0	I0.0	8	Q2.0	Konfigurační blok není přít...
1						Není přítomen
2						Není přítomen
3						Není přítomen
4						Není přítomen
5						Není přítomen
6						Není přítomen

Obr. 5-6 Dialogové okno "Informace o PLC"

Nekritické chyby

Nekritické chyby indikují problémy v konstrukci uživatelského programu, problémy s prováděním instrukcí v uživatelském programu a problémy s rozšiřovacími vstupními/výstupními moduly. Pro prohlédnutí chybových kódů, které byly generovány nekritickou chybou, můžete použít STEP 7 – Micro/WIN. Existují tři základní kategorie nekritických chyb.

Chyby vzniklé při překladu programu

Automat S7-200 překládá program při jeho downloadu. Pokud S7-200 zjistí, že program porušuje pravidla překladu, je download přerušen a generuje se chybový kód. (Program, který byl nahrán již v S7-200, bude nadále uložen v EEPROM a nebude ztracen.) Program nejprve opravte; pak můžete opět provést jeho download. V příloze C najdete seznam porušení pravidel překladu.

Chyby vstupů a výstupů

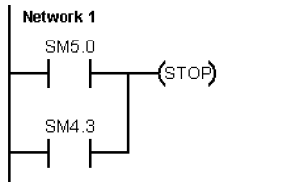
Při prvním spuštění načte S7-200 z každého modulu konfiguraci vstupů a výstupů. Během normálního provozu periodicky kontroluje S7-200 stav každého modulu a porovnává ho s konfigurací, získanou při prvním spuštění. Pokud automat S7-200 zjistí rozdíl, nastaví chybový bit konfigurace v chybovém registru modulu. S7-200 z tohoto modulu nenačítá vstupní data ani do něj nezapisuje výstupní data, dokud konfigurace modulu opět neodpovídá konfiguraci, získané při prvním spuštění.

Informace o stavu modulu jsou uloženy ve speciálních paměťových bitech (SM). Uživatelský program dokáže tyto bity monitorovat a vyhodnocovat. Více informací o bitech SM, používaných pro hlášení chyb vstupů a výstupů, najdete v příloze D. SM5.0 je globální bit pro chyby vstupů a výstupů a zůstává nastaven, dokud chybová podmínka v rozšiřovacím modulu trvá.

Chyby provádění programu

Uživatelský program dokáže během svého provádění vytvořit chybové podmínky. Takové chyby mohou být důsledkem nesprávného použití instrukce nebo toho, že instrukce zpracovává neplatná data. Například ukazatel na nepřímou adresu, který byl platný v době překladu programu, může být během provádění programu modifikován, takže ukazuje na adresu mimo rozsah. To je příklad problému programování při běhu programu. SM4.3 je nastaven při výskytu problému programování v době běhu programu a zůstává nastaven, dokud je S7-200 v režimu RUN. (Seznam problémů programování při běhu programu najdete v příloze C.) Informace o chybách provádění programu jsou uloženy ve speciálních paměťových bitech (SM). Uživatelský program dokáže tyto bity monitorovat a vyhodnocovat. Více informací o bitech SM, používaných pro hlášení chyb provádění programu, najdete v příloze D.

Jestliže automat S7-200 zjistí nekritickou chybu, nepřejde do režimu STOP. Pouze tuto událost zaznamená do paměti SM a pokračuje v provádění uživatelského programu. Uživatelský program můžete ale navrhnout tak, aby byl S7-200 nucen přejít do režimu STOP, když je zjištěna nekritická chyba. Následující vzorový program ukazuje network programu, který monitoruje dva globální nekritické chybové bity a uvede S7-200 do režimu STOP, kdykoliv bude mít některý z těchto bitů hodnotu logická 1.

Vzorový program: Logika pro zjištění nekritické chybové podmínky	
	Network 1 //Jestliže dojde k chybě vstupu/výstupu nebo k chybě v době běhu programu, přejde do režimu STOP <pre> LD SM5.0 O SM4.3 STOP </pre>

Kritické chyby

Kritické chyby způsobí, že automat zastaví provádění uživatelského programu. Kritická chyba může podle své závažnosti způsobit, že S7-200 nebude moci provádět některé nebo všechny své funkce. Cílem ošetření kritických chyb je přivést automat S7-200 do bezpečného stavu, ve kterém může odpovídat na dotazy o stávajících chybových podmínkách. Při zjištění kritické chyby přejde S7-200 do režimu STOP, rozsvítí LED diodu "Systémová chyba" a LED diodu STOP, zablokuje tabulku výstupů a vypne výstupy. V tomto stavu S7-200 zůstane, dokud není odstraněna kritická chyba.

Po provedení změn, nutných pro nápravu kritické chyby, použijte pro restart S7-200 některý z následujících způsobů:

- ☐ Vypněte a znovu zapněte napájení.
- ☐ Přepněte přepínač režimu z polohy RUN nebo TERM do polohy STOP.
- ☐ Vyberte příkaz menu **PLC > Softwarový reset** v STEP 7 – Micro/WIN, který provede restart S7-200 a vymaže všechny kritické chyby.

Restart S7-200 se vymaže kritická chyba a provede se diagnostický test, který ověří, že byla kritická chyba odstraněna. Pokud by byla nalezena jiná kritická chyba, S7-200 opět rozsvítí chybovou LED diodu, která indikuje, že chyba nadále přetrvává. S7-200 jinak zahájí normální provoz.

Některé chyby mohou způsobit nefunkčnost komunikace s automatem S7-200. V těchto případech si v S7-200 nemůžete prohlédnout chybový kód. Takové typy chyby indikují poruchy hardwaru, které vyžadují opravu S7-200; není možné je opravit změnou programu nebo vymazáním paměti S7-200.

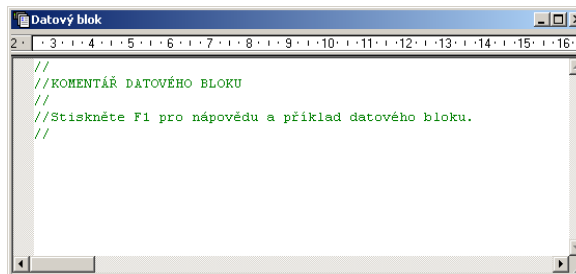
Přřazení adres a počátečních hodnot v editoru datového bloku



Datový blok

Editor datového bloku umožňuje provést přiřazení počátečních hodnot pouze pro paměť V. Přiřazení můžete provést k hodnotám byte, word nebo double word paměti V. Komentáře jsou nepovinné.

Editor datového bloku je textový editor volného formátu; to znamená, že pro konkrétní informace nejsou definována žádná specifická pole. Když dokončíte psaní řádku a stisknete klávesu Enter, editor datového bloku řádek zformátuje (zarovná sloupce adres, dat a komentářů, napíše adresy paměti V velkými písmeny) a znovu ho zobrazí. Editor datového bloku přiřadí příslušnou část paměti V podle toho, jak jste přiřadili adresy a typy dat (byte, word nebo double word).



Obr. 5-7 Editor datového bloku

První řádek datového bloku musí mít přiřazenu absolutní adresu. Následující řádky mohou mít přiřazeny absolutní nebo implicitní adresu. Přiřazení implicitní adresy editor provede, když po přiřazení jediné adresy napíšete více datových hodnot nebo napíšete řádek, který obsahuje pouze datové hodnoty.

Editor datového bloku akceptuje velká nebo malá písmena. Pro oddělení adres a datových hodnot jsou povoleny čárky, tabulátory nebo mezery.

Použití tabulky symbolů pro symbolické adresování proměnných



Tabulka symbolů

Tabulka symbolů umožňuje definovat a editovat symboly, ke kterým je odkudkoliv z programu možné přistupovat pomocí symbolických názvů. Tabulek symbolů můžete vytvořit několik. V tabulce symbolů je také záložka pro symboly, definované systémem, které je možné použít v uživatelském programu. Tabulka symbolů se také označuje jako tabulka globálních proměnných.

Operandy instrukcí uživatelského programu můžete označit absolutně, nebo symbolicky. Absolutní odkaz používá pro označení adresy paměťový bit nebo byte. Symbolický odkaz používá pro označení adresy kombinaci alfanumerických znaků.

V programech SIMATIC se provádí přiřazení globálních symbolů pomocí tabulky symbolů. V programech IEC se provádí přiřazení globálních symbolů pomocí tabulky globálních proměnných.

Chcete-li přiřadit symbol k adrese:

1. Klikněte na ikonu "Tabulka symbolů" v navigační liště; tím otevřete tabulku symbolů.
2. Vložte jméno symbolu (například Input1) do sloupce "Název". Maximální délka symbolu je 23 znaků.
3. Vložte adresu (například I0.0) do sloupce "Adresa".
4. V tabulce globálních proměnných IEC vložte hodnotu do sloupce "Typ dat" nebo ji vyberte ze seznamu.

	Název	Adresa	Komentář
1	USS_LOW_V		
2	USS_HIGH_V		
3	USS_TIME_OUT		
4	USS_I_LUNE		
5	USS_DAT_PTBO		
6	USS_ACT_DRV0		
7	USS_CUB_DRV0		
8	USS_PO		
9	USS_REQ_SNT0		

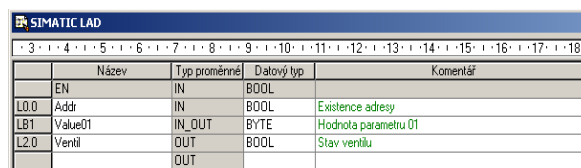
Obr. 5-8 Tabulka symbolů

Můžete vytvořit více tabulek symbolů; nemůžete však použít stejný řetězec pro přiřazení globálního symbolu více než jednou, a to ani v rámci jedné nebo více tabulek.

Použití lokálních proměnných

Tabulku lokálních proměnných, vytvořenou v programovém editoru, můžete použít pro přiřazení proměnných, které jsou jedinečné pro konkrétní podprogram nebo podprogram přerušení. Viz obr. 5-9.

Lokální proměnné se mohou použít jako parametry, které jsou předávány podprogramu; zvyšují tak jeho přenositelnost nebo opakované použití.



	Název	Typ proměnné	Datový typ	Komentář
	EN	IN	BOOL	
L0.0	Add	IN	BOOL	Existence adresy
LB1	Value01	IN_OUT	BYTE	Hodnota parametru 01
L2.0	Ventil	OUT	BOOL	Stav ventilu
		OUT		

Obr. 5-9 Tabulka lokálních proměnných

Použití stavového diagramu pro monitorování uživatelského programu



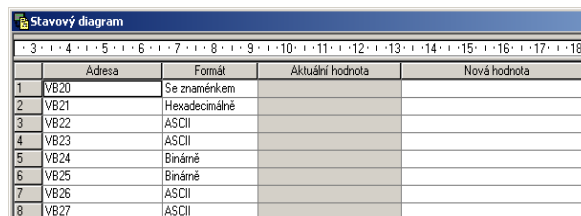
Stavová tabulka

Stavový diagram umožňuje monitorovat nebo modifikovat hodnoty procesních proměnných při běhu řídicího programu S7-200. Můžete sledovat stav programových vstupů, výstupů nebo proměnných tím, že si zobrazíte jejich aktuální hodnoty. Stavový diagram také umožňuje vnutit nebo změnit hodnoty procesních proměnných.

Můžete vytvořit více stavových diagramů, abyste mohli prohlížet různé části programu.

Chcete-li otevřít stavový diagram, vyberte příkaz menu **Zobrazit > Komponenty > Stavový diagram** nebo klikněte na ikonu "Stavový diagram" na navigační liště.

Když vytváříte stavový diagram, zadáváte adresy procesních proměnných pro monitorování. Není možné zobrazit stav konstant, akumulátorů nebo lokálních proměnných. Hodnotu časovače nebo čítače můžete zobrazit buď jako bit, nebo jako word. Při zobrazení hodnoty jako bitu uvidíte stav časovače nebo čítače; zobrazení hodnoty jako word ukáže hodnotu časovače nebo čítače.



	Adresa	Formát	Aktuální hodnota	Nová hodnota
1	VB20	Se znaménkem		
2	VB21	Hexadecimálně		
3	VB22	ASCII		
4	VB23	ASCII		
5	VB24	Binárně		
6	VB25	Binárně		
7	VB26	ASCII		
8	VB27	ASCII		

Obr. 5-10 Stavová tabulka

Chcete-li vytvořit stavový diagram a monitorovat proměnné:

1. Vložte adresy všech požadovaných hodnot do políčka "Adresa".
2. Vyberte typ dat ve sloupci "Formát".
3. Chcete-li zobrazit stav procesních proměnných v automatu S7-200, vyberte příkaz menu **Odladit > Stavový diagram**.
4. Chcete-li nepřetržitě vzorkovat hodnoty nebo jednorázově odečíst stav, klikněte na tlačítko na nástrojové liště. Stavový diagram také umožňuje modifikovat nebo vnutit hodnoty různých procesních proměnných.

Další řádky stavového diagramu můžete vložit pomocí příkazu menu **Úpravy > Vložit > Řádek**.



Tip

Můžete vytvořit více stavových diagramů tak, aby byly proměnné rozděleny do logických skupin, z nichž lze každou prohlížet v kratší, samostatné formě.

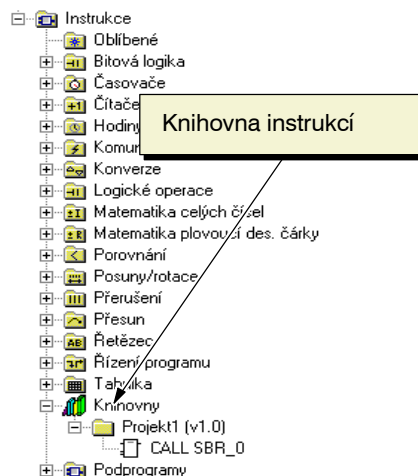
Vytvoření knihovny instrukcí

STEP 7–Micro/WIN umožňuje vytvoření své vlastní knihovny instrukcí nebo použití knihovny, vytvořené někým jiným. Viz obr. 5-11.

Chcete-li vytvořit knihovnu instrukcí, vytvořte standardní podprogram STEP 7–Micro/WIN a podprogram přerušení a seskupte je. Abyste zabránili náhodným změnám nebo z důvodu ochrany know-how autora, můžete v těchto podprogramech skrýt kód.

Chcete-li vytvořit knihovnu instrukcí, proveďte následující kroky:

1. Napište program jako standardní projekt STEP 7–Micro/WIN a funkci, která má být zahrnuta do knihovny, vložte do podprogramů nebo přerušení.
2. Zkontrolujte, že všechna místa paměti V v podprogramech nebo přerušeních mají přiřazen symbolický název. Abyste minimalizovali velikost paměti V, kterou knihovna potřebuje, použijte za sebou jdoucí místa paměti V.
3. Přejmenujte podprogramy a přerušení tak, aby měly názvy, které chcete mít v knihovně instrukcí.
4. Pro kompilování nové knihovny instrukcí použijte příkaz menu **Soubor > Vytvořit knihovnu**.



Více informací o tvorbě knihoven naleznete v online nápovědě programu STEP 7–Micro/WIN.

Pro přístup k instrukci v knihovně instrukcí použijte následující postup:

1. Pomocí příkazu menu **Soubor > Přidat knihovnu** přidejte adresář "Knihovny" do instrukčního stromu.
2. Vyberte příslušnou instrukci a vložte ji do programu (stejně jako jakoukoliv standardní instrukci).

Jestliže podprogram pro knihovnu potřebuje část paměti V, STEP 7–Micro/WIN vás po kompilaci projektu vyzve k přiřazení paměťového bloku. Pro přiřazení paměťových bloků použijte dialogové okno "Přidělení paměti pro knihovnu".

Obr. 5-11 Instrukční strom s knihovnami

Funkce pro ladění uživatelského programu

STEP 7–Micro/WIN má následující funkce, které vám pomohou při ladění uživatelského programu:

- ☐ Záložky, které u dlouhého programu usnadňují přechody dopředu a zpět mezi řádky.
- ☐ Tabulku křížových odkazů, která umožňuje kontrolu odkazů použitých v programu.
- ☐ Editování v režimu RUN umožňuje provádět malé změny programu s minimálním narušením řízeného procesu. Při editaci v režimu RUN také můžete provést download programového bloku.

Více informací o ladění programu najdete v kapitole 8.

Instrukční soubor S7-200

6

Tato kapitola popisuje soubor instrukcí SIMATIC a IEC 1131 pro mikro-PLC S7-200.

Přehled kapitoly

Konvence použité při popisu instrukcí	63
Rozsahy pamětí	64
Instrukce bitové logiky	66
Kontakty	66
Cívky	68
Práce se zásobníkem	70
Bistabilní klopný obvod	72
Instrukce hodin	73
Instrukce pro komunikace	74
Čtení a zápis do sítě	74
Vysílání a příjem (Freeport)	79
Zjištění a nastavení adresy portu	88
Instrukce porovnání	89
Porovnání číselných hodnot	89
Porovnání řetězců	91
Instrukce pro konverzi	92
Standardní konverze	92
Konverze na ASCII	96
Konverze řetězců	100
Zakódování a dekodování	105
Instrukce čítače	106
Čítač SIMATIC	106
Čítač IEC	109
Instrukce vysokorychlostního čítače	111
Instrukce pulzního výstupu	125
Instrukce matematiky	140
Sčítání, odčítání, násobení a dělení	140
Násobení celých čísel s 32bitovým výsledkem a dělení celého čísla se zbytkem	142
Numerické funkce	143
Inkrementace a dekrementace	144
Instrukce PID algoritmu	145
Instrukce přerušení	155
Instrukce logických operací	162
Inverze	162
AND, OR a XOR	163
Instrukce přesunu	165
Přesun byte, word, double word nebo real	165
Okamžitý přesun bytu (čtení a zápis)	166
Přesun bloku	167

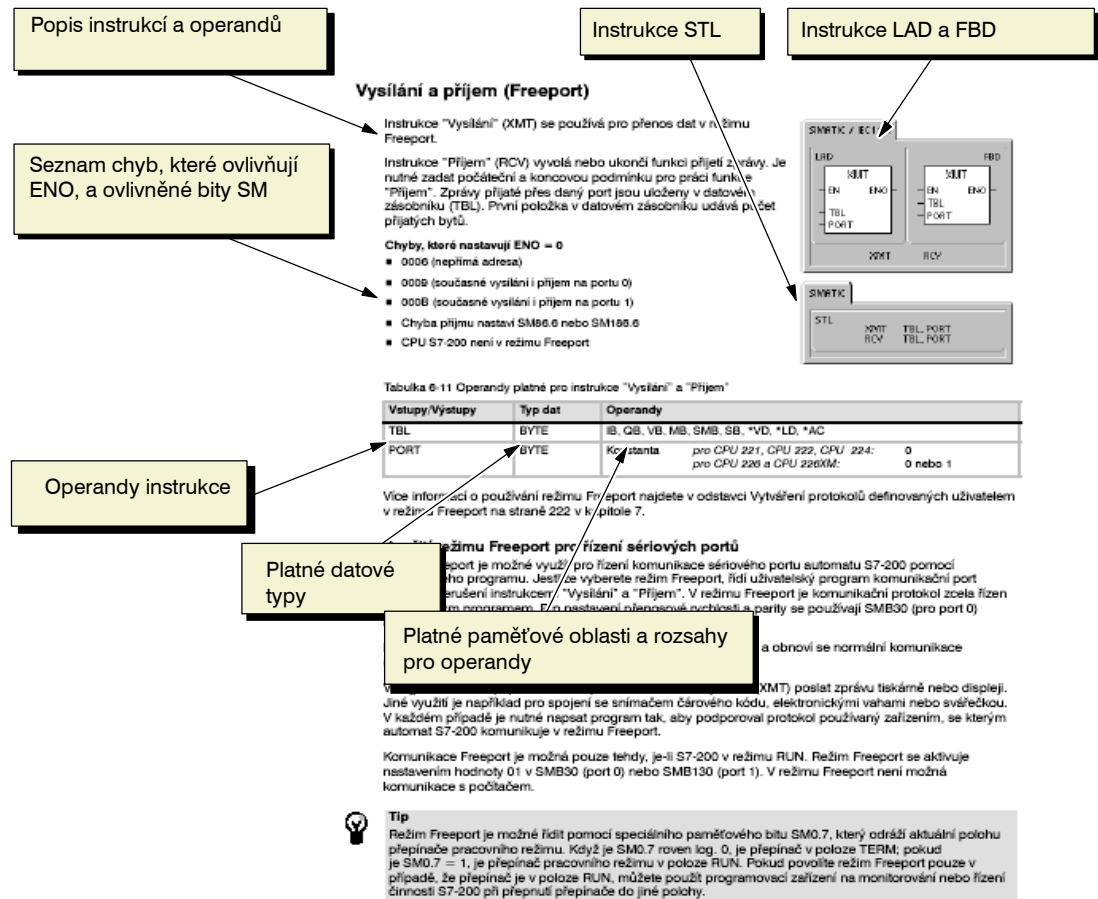
Řízení programu	168
Podmíněný konec programu	168
Stop	168
Reset Watchdogu	168
Smyčka For–Next	170
Skokové instrukce	172
Sekvenční řídicí relé (SCR)	173
Instrukce posunu a rotace	179
Posun vpravo a vlevo	179
Rotace vpravo a vlevo	179
Posun bitů registru	181
Výměna bytů	183
Instrukce pro řetězce	184
Instrukce pro práci s tabulkou	189
Přidávání do tabulky	189
Instrukce FIFO a LIFO	190
Vyplnění paměti	192
Prohledávání tabulky	193
Instrukce časovače	196
Časovač SIMATIC	196
Časovač IEC	201
Instrukce podprogramů	203

Konvence použité při popisu instrukcí

Obrázek 6-1 znázorňuje typický popis instrukce a ukazuje na různé oblasti používané pro popis instrukce a její vykonání. Instrukce jsou ve formátu LAD, FBD a STL. Tabulka operandů uvádí seznam operandů instrukce a ukazuje platné datové typy paměťové oblasti a velikost každého operandu.

Operandy a datové typy EN/ENO nejsou uvedeny v tabulce operandů, protože tyto operandy jsou pro všechny instrukce LAD a FBD stejné.

- ☐ Pro LAD: EN (Enable IN) a ENO (Enable Output) představují signálový tok; jde o booleovský typ dat.
- ☐ Pro FBD: EN a ENO představují I, Q, V, M, SM, S, T, C, L nebo signálový tok; jde o booleovský typ dat.



Obr. 6-1 Popisy instrukcí

Rozsahy pamětí

Tabulka 6-1 Rozsahy pamětí

Popis	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Velikost uživatelského programu	2 Kword	2 Kword	4 Kword	4 Kword	8 Kword
Velikost uživatelských dat	1 Kword	1 Kword	2,5 Kword	2,5 Kword	5 Kword
Registr obrazu vstupů	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7
Registr obrazu výstupů	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7
Analogové vstupy (čtení)	--	AIW0 až AIW30	AIW0 až AIW62	AIW0 až AIW62	AIW0 až AIW62
Analogové výstupy (zápis)	--	AQW0 až AQW30	AQW0 až AQW62	AQW0 až AQW62	AQW0 až AQW62
Paměť proměnných (V)	VB0 až VB2047	VB0 až VB2047	VB0 až VB5119	VB0 až VB5119	VB0 až VB10239
Lokální paměť (L) ¹	LB0 až LB63	LB0 až LB63	LB0 až LB63	LB0 až LB63	LB0 až LB63
Bitová paměť (M)	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7
Speciální paměť (SM) Pouze pro čtení	SM0.0 až SM179.7 SM0.0 až SM29.7	SM0.0 až SM299.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7
Časovače	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)
Remanentní zpožděné zapnutí	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
1 ms	T1 až T4, a	T1 až T4, a	T1 až T4, a	T1 až T4, a	T1 až T4, a
10 ms	T65 až T68	T65 až T68	T65 až T68	T65 až T68	T65 až T68
100 ms	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95
Zpožděné zapnutí/vypnutí	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
1 ms	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100
10 ms	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255
100 ms					
Čítače	C0 až C255	C0 až C255	C0 až C255	C0 až C255	C0 až C255
Vysokorychlostní čítače	HC0, HC3, HC4 a HC5	HC0, HC3, HC4 a HC5	HC0 až HC5	HC0 až HC5	HC0 až HC5
Sekvenční řídicí relé (S)	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7
Registry akumulátoru	AC0 až AC3	AC0 až AC3	AC0 až AC3	AC0 až AC3	AC0 až AC3
Skoky/návěští	0 až 255	0 až 255	0 až 255	0 až 255	0 až 255
Volání/podprogramy	0 až 63	0 až 63	0 až 63	0 až 63	0 až 127
Podprogram přerušení	0 až 127	0 až 127	0 až 127	0 až 127	0 až 127
Pozitivní/negativní přechody	256	256	256	256	256
PID algoritmus	0 až 7	0 až 7	0 až 7	0 až 7	0 až 7
Porty	Port 0	Port 0	Port 0	Port 0, Port 1	Port 0, Port 1

¹ LB60 až LB63 jsou ve STEP 7 – Micro/WIN verze 3.0 nebo pozdější rezervovány.

Tabulka 6-2 Rozsahy operandů CPU S7-200

Způsob přístupu		CPU 221	CPU 222	CPU 224, CPU 226	CPU 226XM
Přístup po bitech (byte.bit)	I	0,0 až 15,7	0,0 až 15,7	0,0 až 15,7	0,0 až 15,7
	Q	0,0 až 15,7	0,0 až 15,7	0,0 až 15,7	0,0 až 15,7
	V	0,0 až 2047,7	0,0 až 2047,7	0,0 až 5119,7	0,0 až 10239,7
	M	0,0 až 31,7	0,0 až 31,7	0,0 až 31,7	0,0 až 31,7
	SM	0,0 až 179,7	0,0 až 299,7	0,0 až 549,7	0,0 až 549,7
	S	0,0 až 31,7	0,0 až 31,7	0,0 až 31,7	0,0 až 31,7
	T	0 až 255	0 až 255	0 až 255	0 až 255
	C	0 až 255	0 až 255	0 až 255	0 až 255
	L	0,0 až 59,7	0,0 až 59,7	0,0 až 59,7	0,0 až 59,7
Přístup po bytech	IB	0 až 15	0 až 15	0 až 15	0 až 15
	QB	0 až 15	0 až 15	0 až 15	0 až 15
	VB	0 až 2047	0 až 2047	0 až 5119	0 až 10239
	MB	0 až 31	0 až 31	0 až 31	0 až 31
	SMB	0 až 179	0 až 299	0 až 549	0 až 549
	SB	0 až 31	0 až 31	0 až 31	0 až 31
	L	0 až 63	0 až 63	0 až 63	0 až 255
	AC	0 až 3	0 až 3	0 až 3	0 až 255
Přístup po word	IW	0 až 14	0 až 14	0 až 14	0 až 14
	QW	0 až 14	0 až 14	0 až 14	0 až 14
	VW	0 až 2046	0 až 2046	0 až 5118	0 až 10238
	MW	0 až 30	0 až 30	0 až 30	0 až 30
	SMW	0 až 178	0 až 298	0 až 548	0 až 548
	SW	0 až 30	0 až 30	0 až 30	0 až 30
	T	0 až 255	0 až 255	0 až 255	0 až 255
	C	0 až 255	0 až 255	0 až 255	0 až 255
	LW	0 až 58	0 až 58	0 až 58	0 až 58
	AC	0 až 3	0 až 3	0 až 3	0 až 3
	AIW	Není	0 až 30	0 až 62	0 až 62
	AQW	Není	0 až 30	0 až 62	0 až 62
Přístup po double word	ID	0 až 12	0 až 12	0 až 12	0 až 12
	QD	0 až 12	0 až 12	0 až 12	0 až 12
	VD	0 až 2044	0 až 2044	0 až 5116	0 až 10236
	MD	0 až 28	0 až 28	0 až 28	0 až 28
	SMD	0 až 176	0 až 296	0 až 546	0 až 546
	SD	0 až 28	0 až 28	0 až 28	0 až 28
	LD	0 až 56	0 až 56	0 až 56	0 až 56
	AC	0 až 3	0 až 3	0 až 3	0 až 3
	HC	0, 3, 4, 5	0, 3, 4, 5	0 až 5	0 až 5

Instrukce bitové logiky

Kontakty

Standardní kontakty (Standard Contacts)

Instrukce kontaktu "V klidu otevřen" (LD, A a O) a "V klidu zavřen" (LDN, AN, ON) získávají hodnotu z paměti nebo z registru obrazu. Instrukce standardního kontaktu získávají hodnotu z paměti (nebo z registru obrazu procesu, pokud je typ dat I nebo Q).

Kontakt "V klidu otevřen" je sepnutý, když se bit rovná 1, kontakt "V klidu zavřen" je sepnutý, když se bit rovná 0. V editoru FBD je možné vstupy do bloků AND a OR rozšířit maximálně na 32 vstupů. V editoru STL instrukce kontaktu "V klidu otevřen", "Načíst", "AND" nebo "OR" vykonávají funkce Načíst, AND nebo OR daného bitu s horním bitem zásobníku, instrukce kontaktu "V klidu zavřen", "Načíst negované", "NAND" nebo "NOR" vykonávají funkce negace Načíst, AND nebo OR daného bitu s horním bitem zásobníku.

Okamžité kontakty

Přímý kontakt se aktualizuje okamžitě, nezávisí na programovém cyklu. Instrukce kontaktu "V klidu otevřen, okamžitý" (LDI, AI a OI) a instrukce kontaktu "V klidu zavřen, okamžitý" (LDNI, ANI a ONI) získají hodnotu z fyzického vstupu v době vykonávání instrukce; registr obrazu se ale neaktualizuje.

Kontakt "V klidu otevřen, okamžitý" je sepnutý, když je na vstupu logická 1, kontakt "V klidu zavřen, okamžitý" je sepnutý, když je na vstupu logická 0. Instrukce "V klidu otevřen" okamžitě provedou operaci Načíst, AND nebo OR s hodnotou na vstupu a horní hodnotou zásobníku; instrukce "V klidu zavřen" okamžitě provedou operaci Načíst negované, NAND nebo NOR negace hodnoty fyzického vstupu s horní hodnotou zásobníku.

Negace (NOT)

Instrukce "Negace" (NOT) mění stav vstupu signálového toku (tzn. že mění horní hodnotu zásobníku z 0 na 1 nebo z 1 na 0).

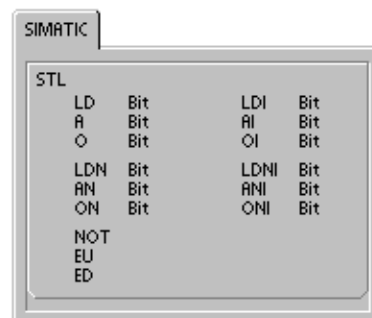
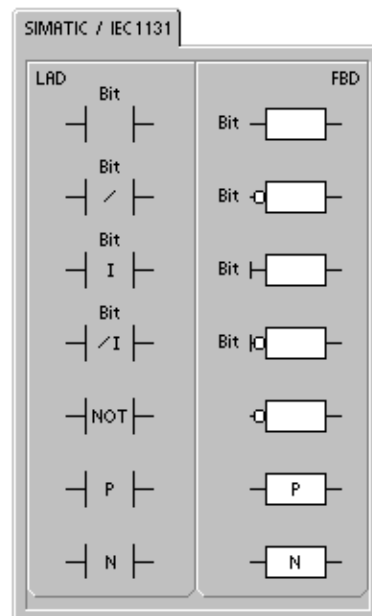
Náběžná a sestupná hrana

Instrukce "Náběžná hrana" (EU) aktivuje signálový tok pouze pro jeden programový cyklus, pokud signál na vstupu změní stav z log. 0 na log. 1. Instrukce "Sestupná hrana" (ED) aktivuje signálový tok pouze pro jeden programový cyklus, pokud signál na vstupu změní stav z log. 1 na log. 0. U instrukce "Náběžná hrana" nastaví detekce přechodu 0 na 1 horní hodnotu zásobníku na 1; jinak je nastavena na 0. U instrukce "Sestupná hrana" detekce přechodu 1 na 0 nastaví horní hodnotu zásobníku na 1; jinak je nastavena na 0.

Při editaci v režimu RUN musíte vložit parametr pro instrukce náběžná a sestupná hrana. Více informací o editaci v režimu RUN najdete v kapitole 5.

Tabulka 6-3 Operandy platné pro vstupy instrukcí bitové logiky

Vstupy/Výstupy	Typ dat	Operandy
Bit	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Bit (okamžitý)	BOOL	I



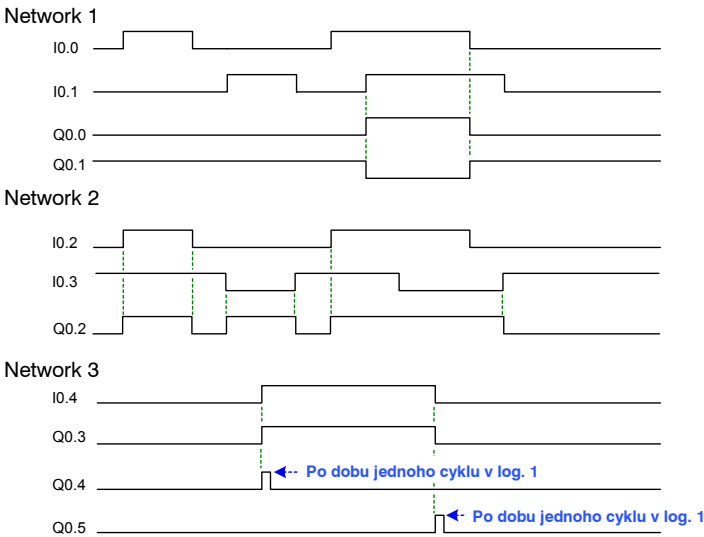


Tip
Protože instrukce náběžná a sestupná hrana vyžadují přechod ze zapnutého do vypnutého stavu nebo naopak, není možné zjistit náběžnou nebo sestupnou hranu v prvním programovém cyklu. Během prvního programového cyklu nastaví S7-200 stav bitu specifikovaného těmito instrukcemi. V příštích cyklech již mohou instrukce zjišťovat přechody u specifikovaného bitu.

Příklad: Instrukce kontaktů

Network 1	Network 1 //Pro aktivaci Q0.0 musí být sepnuté kontakty I0.0 a I0.1 //Instrukce NOT působí jako invertor. //V režimu RUN mají Q0.0 a Q0.1 opačné logické stavy. LD I0.0 A I0.1 = Q0.0 NOT = Q0.1
Network 2	Network 2 //Aby se aktivoval Q0.2, musí být sepnutý kontakt I0.2 nebo //rozepnutý I0.3. //Aby se aktivoval výstup, musí být aktivní jedna nebo více //paralelních větví LAD (logické vstupy OR). LD I0.2 ON I0.3 = Q0.2
Network 3	Network 3 //Náběžná hrana na kontaktu P nebo sestupná hrana //na kontaktu N dá pulz s trváním 1 programového //cyklu. V režimu RUN jsou změny stavu na Q0.4 a Q0.5 //příliš rychlé, aby byly patrné ve stavovém zobrazení. //Výstupy nastavení (Set) a vynulování (Reset) zachytí pulz na //Q0.3 a zobrazí změnu stavu ve stavovém zobrazení. LD I0.4 LPS EU S Q0.3, 1 = Q0.4 LPP ED R Q0.3, 1 = Q0.5

Časový diagram



Cívky

Výstup

Instrukce "Výstup" (=) запише novou hodnotu výstupního bitu do registru obrazu procesu. Když je provedena instrukce výstupu, S7-200 nastaví nebo vynuluje výstupní bit registru obrazu procesu. Pro LAD a FBD je tento bit nastaven tak, že se rovná signálovému toku. V STL je do specifikovaného bitu zkopírována horní hodnota zásobníku.

Okamžitý výstup

Instrukce "Okamžitý výstup" (=I) okamžitě запише při svém provádění novou hodnotu na fyzický výstup a také na odpovídající místo registru obrazu procesu.

Při provádění instrukce okamžitého výstupu je fyzický výstup (bit) okamžitě nastaven tak, že se rovná signálovému toku. Instrukce okamžitě zkopíruje horní hodnotu zásobníku do specifikovaného bitu fyzického výstupu (STL). "I" označuje okamžitou instrukci; při provádění instrukce je nová hodnota zapsána na fyzický výstup i na odpovídající místo registru obrazu procesu. Tím se liší od nepřímých instrukcí, které zapisují novou hodnotu pouze do registru obrazu procesu.

Set (nastavení) a Reset (nulování)

Instrukce Set (S) a Reset (R) nastaví nebo vynuluje specifikovaný počet výstupů (N), přičemž začíná na specifikované adrese (bitu). Můžete nastavovat nebo vynulovat 1 až 255 výstupních bodů.

Pokud se instrukce nulování vztahuje na bit časovače (T) nebo na bit čítače (C), vynuluje tato instrukce bit časovače nebo čítače a vymaže jejich aktuální hodnotu.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)

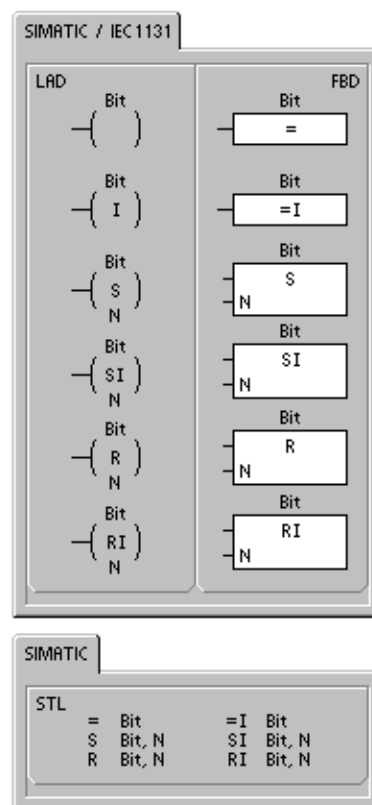
Set okamžitě a Reset okamžitě

Instrukce okamžitě nastaví nebo vynuluje specifikovaný počet výstupů (N), přičemž začíná na specifikované adrese (bitu). Můžete okamžitě nastavovat nebo vynulovat 1 až 128 bodů.

"I" označuje okamžitou instrukci; při provádění instrukce je nová hodnota zapsána na fyzický výstup stejně jako na odpovídající místo registru obrazu procesu. Tím se liší od nepřímých instrukcí, které zapisují novou hodnotu pouze do registru obrazu procesu.

Chyby, které nastavují ENO = 0

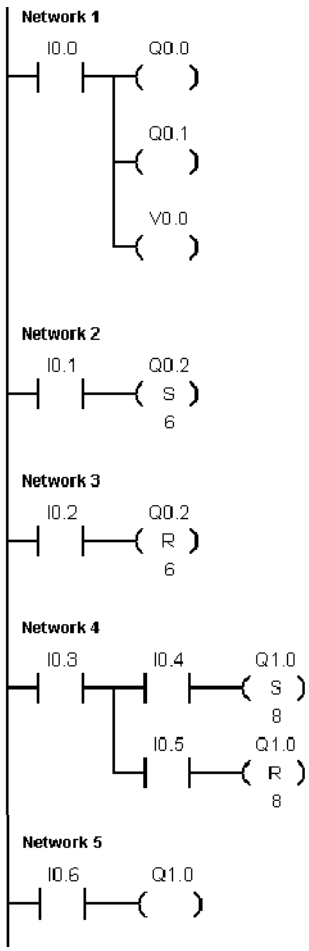
- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)



Tabulka 6-4 Operandy platné pro výstupy instrukcí bitové logiky

Vstupy/Výstupy	Typ dat	Operandy
Bit	BOOL	I, Q, V, M, SM, S, T, C, L,
Bit (okamžitý)	BOOL	Q
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta

Příklad: Instrukce cívky



Network 1 //Instrukce výstupu přidělí bitové hodnoty vstupům a výstupům (I, Q)
//a interní paměti (M, SM, T, C, V, S, L).
LD IO.0
= Q0,0
= Q0,1
= V0.0

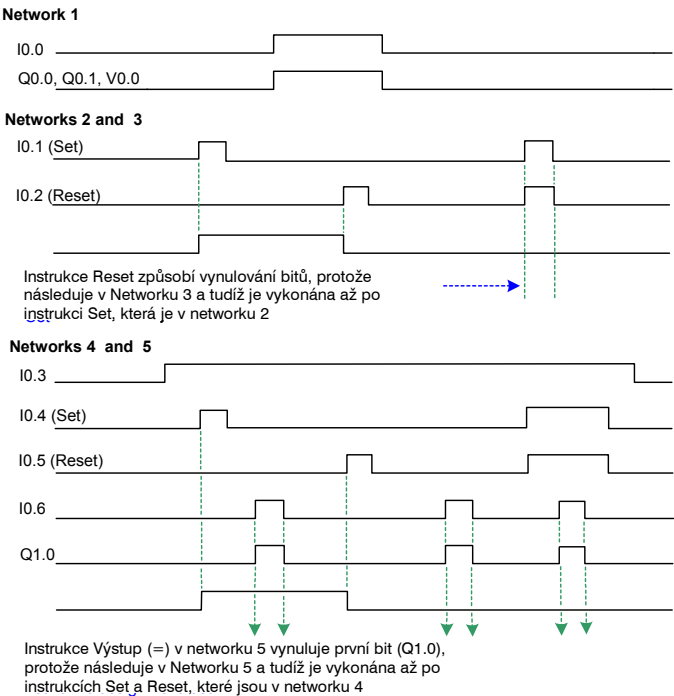
Network 2 //Nastaví 6 následujících bitů na hodnotu log. 1.
//Specifikuje adresu počátečního bitu a počet nastavovaných bitů.
//Indikátor stavu programu pro Set je ON (zapnutý), když hodnota
//prvního bitu (Q0.2) je log. 1.
LD IO.1
S Q0,2, 6

Network 3 //Nastaví 6 následujících bitů na hodnotu log. 0.
//Specifikuje adresu počátečního bitu a počet nastavovaných bitů.
//Indikátor stavu programu pro Reset je ON (zapnutý), když hodnota
//prvního bitu (Q0.2) je 0.
LD IO.2
R Q0,2, 6

Network 4 //Nastaví a vynuluje 8 výstupních bitů (Q1.0 až Q1.7) jako skupinu.
LD IO.3
LPS
A IO.4
S Q1,0, 8
LPP
A IO.5
R Q1,0, 8

Network 5 //Instrukce Set a Reset provádějí funkci RS klopného obvodu.
//Pro oddělení bitů Set/Reset se ujistěte, že nejsou přepsány
//jinou instrukcí. V tomto příkladu Network 4 nastavuje a
//nuluje osm výstupních bitů (Q1.0 až Q1.7) jako skupinu.
//V režimu RUN může Network 5 přepsat hodnotu bitu Q1.0 a
//řídí programové indikátory stavu Set/Reset v Networku 4.
LD IO.6
= Q1,0

Časový diagram



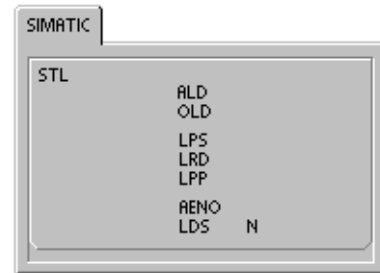
Práce se zásobníkem

Načíst AND

Instrukce "Načíst AND" (ALD) kombinuje hodnoty na první a druhé úrovni zásobníku pomocí logické operace AND. Výsledek se uloží na horní místo v zásobníku. Po vykonání instrukce ALD se velikost zásobníku zmenší o jedničku.

Načíst OR

Instrukce "Načíst OR" (OLD) kombinuje hodnoty na první a druhé úrovni zásobníku pomocí logické operace OR. Výsledek se uloží na horní místo v zásobníku. Po vykonání instrukce OLD se velikost zásobníku zmenší o jedničku.



Logické Push

Instrukce "Logické Push" (LPS) duplikuje horní hodnotu zásobníku. Spodní hodnota zásobníku je odsunuta a ztratí se.

Logické čtení

Instrukce "Logické čtení" (LRD) zkopíruje druhou hodnotu v zásobníku na vrchol zásobníku. Hodnoty v zásobníku se žádným směrem neposouvají, ztratí se však stará hodnota vrcholu zásobníku.

Logické pop

Instrukce "Logické pop" (LPP) odstraní jednu hodnotu ven ze zásobníku. Druhá hodnota v zásobníku se stane novou hodnotou vrcholu zásobníku.

AND ENO

Instrukce "AND ENO" (AENO) provede logický AND bitu ENO s vrcholem zásobníku a vytvoří stejný efekt jako bit ENO bloku v LAD nebo FBD. Výsledkem operace AND je nová hodnota na vrcholu zásobníku.

ENO je booleovský výstup bloků v LAD a FBD. Pokud má blok signálový tok na vstupu EN a je proveden bez chyby, výstup ENO předá signálový tok dalšímu elementu. ENO můžete použít jako bit pro povolení, který ukazuje úspěšné dokončení instrukce. Bit ENO se používá s horním bitem zásobníku pro ovládání signálového toku při provádění dalších instrukcí. Instrukce STL nemají vstup EN, aby byly provedeny podmíněné instrukce. Hodnota nejvyššího bitu zásobníku musí být logická 1. Instrukce STL nemají ani výstup ENO. Nicméně instrukce STL, které odpovídají instrukcím LAD a FBD s výstupy ENO, nastavují speciální bit ENO. K tomuto bitu se dá přistupovat pomocí AENO.

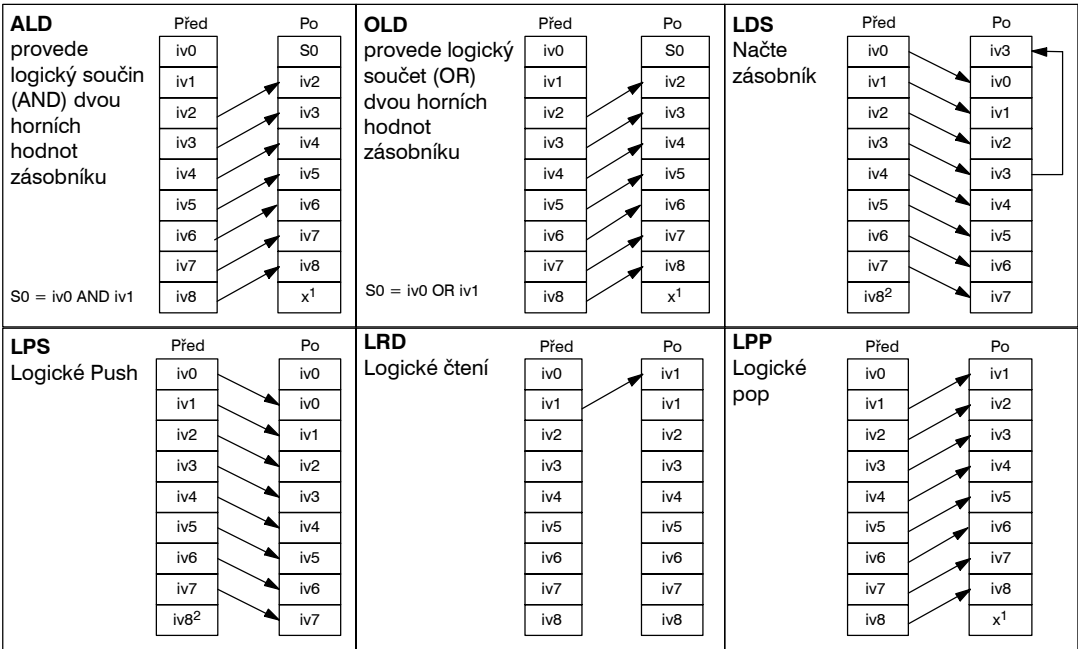
Načíst zásobník

Instrukce "Načíst zásobník" (LDS) duplikuje bit zásobníku (N) a hodnotu nahraje na vrchol zásobníku. Spodní hodnota zásobníku je odsunuta a ztratí se.

Tabulka 6-5 Operandy platné pro instrukci "Načíst zásobník"

Vstupy/Výstupy	Typy dat	Operandy
N	BYTE	Konstanta (0 až 8)

Automat S7-200 používá při vykonávání řídicího programu zásobník – viz obrázek 6-2. V těchto příkladech označují “iv0” až “iv7” počáteční hodnoty zásobníku, “nv” označuje novou hodnotu danou instrukcí a “S0” označuje vypočítanou hodnotu, která je uložena do zásobníku.



1 Hodnota je neznámá (může to být 0 nebo 1).
2 Po provedení instrukcí "Logické Push" nebo "Načíst zásobník" se hodnota iv8 ztratí.

Obr. 6-2 Operace logických instrukcí pro práci se zásobníkem

Příklad: Instrukce pro práci se zásobníkem	
Network 1 Network 2	Network 1 <pre>LD I0.0 LD I0.1 LD I2.0 A I2.1 OLD ALD = Q5,0</pre> Network 2 <pre>LD I0.0 LPS LD I0.5 O I0.6 ALD = Q7,0 LRD LD I2.1 O I1.3 ALD = Q6,0 LPP A I1.0 = Q3,0</pre>

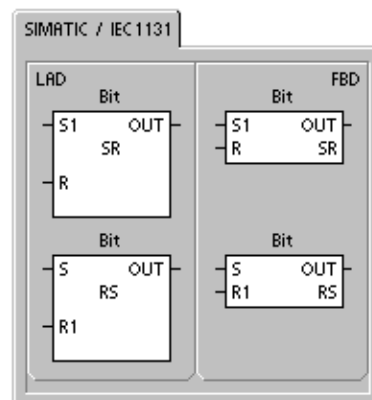
Bistabilní klopný obvod

Instrukce "SR klopný obvod" je funkce RS klopného obvodu, ve které je dominantní Set. Pokud mají signály Set (S1) a Reset (R) současně hodnotu 1, bude mít hodnotu 1 i výstup (OUT).

Instrukce "RS klopný obvod" je blokovácí funkce RS klopného obvodu, ve které je dominantní Reset. Pokud mají signály Set (S1) a Reset (R) současně hodnotu 1, bude mít výstup (OUT) hodnotu 0.

Booleovský parametr definuje, který bit se nastavuje nebo nuluje. Volitelný výstup odráží stav signálu parametru Bit.

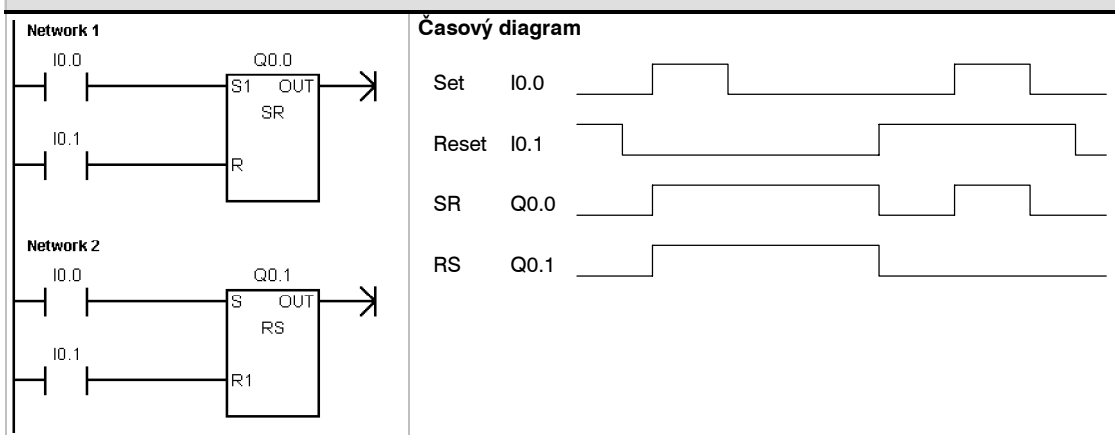
Tabulka 6-7 je pravdivostní tabulka pro vzorový program.



Tabulka 6-6 Operandy platné pro instrukce "SR klopný obvod" a "RS klopný obvod"

Vstupy/Výstupy	Typy dat	Operandy
S1, R	BOOL	I, Q, V, M, SM, S, T, C, signálový tok
S, R1, OUT	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Bit	BOOL	I, Q, V, M, S

Příklad: Instrukce Bistabilní KO Set/Reset



Tabulka 6-7 Pravdivostní tabulka pro instrukce "SR a RS klopný obvod"

Instrukce	S1	R	Out (Bit)
Instrukce "Bistabilní KO s převládajícím Set" (SR)	0	0	Předchozí stav
	0	1	0
	1	0	1
	1	1	1
Instrukce	S	R1	Out (Bit)
Instrukce "Bistabilní KO s převládajícím Reset" (RS)	0	0	Předchozí stav
	0	1	0
	1	0	1
	1	1	0

Instrukce hodin

Čtení a nastavení hodin reálného času

Instrukce "Čtení hodin reálného času" (TODR) přečte aktuální čas a datum z hardwarových hodin a zapíše ho do 8bytového zásobníku se začátkem na adrese T. Instrukce "Nastavení hodin reálného času" (TODW) zapíše do hodin aktuální čas a datum se začátkem na 8bytové adrese zásobníku dané parametrem T.

Všechny hodnoty data a času musí být v BCD formátu (například 16#97 pro rok 1997). Obrázek 6-3 ukazuje formát časového zásobníku (T).

Po déletrvajícím výpadku napájení nebo při ztrátě paměti inicializují hodiny reálného času (TOD) následující datum a čas:

Datum: 01 – Jan – 90
Čas: 00:00:00
Den v týdnu: neděle

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0007 (chyba dat TOD) pouze "Čtení hodin reálného času"
- 000C (hodiny nejsou připojeny)

Tabulka 6-8 Operandy platné pro instrukce hodin

Vstupy/Výstupy	Typy dat	Operandy
T	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
Rok: 00 až 99	Měsíc: 01 až 12	Den: 01 až 31	Hodiny: 00 až 23	Minuty: 00 až 59	Sekundy: 00 až 59	0	Den v týdnu: 0 až 7*

*T+7 1 = neděle, 7 = sobota
0 znamená den v týdnu vypnutý.

Obr. 6-3 Formát 8bytového časového zásobníku (T)



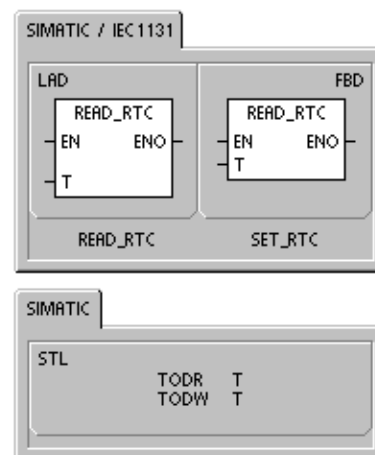
Tip

CPU S7-200 nekontroluje správné nastavení dne v týdnu podle data. Mohou být přijata i nesprávná data, jako například 30. únor. Proto byste zadání správného data měli kontrolovat.

Nepoužívejte instrukci TODR/TODW zároveň v hlavním programu i v přerušení. Instrukce TODR/TODW v podprogramu přerušení, které se pokouší o činnost během činnosti jiné instrukce TODR/TODW, nemůže být provedena. SM4.3 se nastaví tak, že označuje pokus o dva současně přístupy na hodiny (nekritická chyba 0007).

Hodiny reálného času v S7-200 používají pro rok pouze dvě poslední číslice, takže např. rok 2000 je reprezentován jako 00. Automat S7-200 informaci o roku žádným způsobem nevyužívá. Ale uživatelské programy, které používají aritmetiku nebo porovnání hodnoty roků, musí počítat s reprezentací roku dvěma číslicemi a s přechodem století.

Přestupný rok je správně zpracováván až do roku 2096.



Instrukce pro komunikaci

Čtení a zápis do sítě

Instrukce "Čtení ze sítě" (NETR) inicializuje čtení dat ze vzdáleného zařízení přes daný port (PORT) podle tabulky (TBL). Instrukce "Zápis do sítě" (NETW) inicializuje zápis dat na vzdálené zařízení přes daný port (PORT) podle tabulky (TBL).

Chyby, které nastavují ENO = 0:

- 0006 (nepřímá adresa)
- Pokud má funkce za následek chybu, je nastaven bit E stavového bytu tabulky (viz obrázek 6-4)

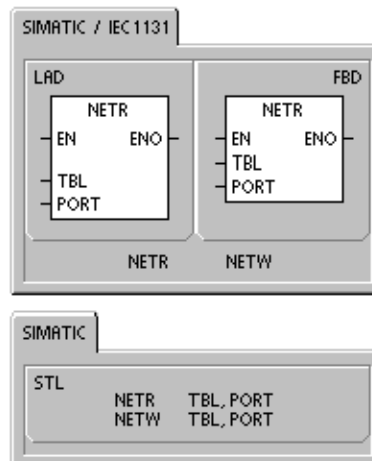
Instrukce "Čtení ze sítě" může přečíst až 16 bytů ze vzdálené stanice, instrukce "Zápis do sítě" může zapsat až 16 bytů do vzdálené stanice.

V programu může být libovolný počet instrukcí "Čtení ze sítě" a "Zápis do sítě"; současně však může být aktivováno maximálně osm instrukcí "Čtení ze sítě" a "Zápis do sítě". Ve stejnou dobu mohou být v daném S7-200 aktivovány 4 instrukce "Čtení ze sítě" a 4 instrukce "Zápis do sítě" nebo 2 instrukce "Čtení ze sítě" a 6 instrukcí "Zápis do sítě".



Průvodce

Pro konfiguraci komunikačních instrukcí můžete použít "Průvodce instrukcemi". Chcete-li spustit průvodce pro "Čtení ze sítě"/"Zápis do sítě", vyberte příkaz menu **Nástroje > Průvodce instrukcemi** a pak vyberte "NETR/NETW" z okna průvodce.



Tabulka 6-9 Operandy platné pro instrukce "Čtení ze sítě" a "Zápis do sítě"

Vstupy/Výstupy	Typ dat	Operandy
TBL	BYTE	VB, MB, *VD, *LD, *AC
PORT	BYTE	Konstanta <i>pro CPU 221, CPU 222, CPU 224:</i> 0 <i>pro CPU 226 a CPU 226XM:</i> 0 nebo 1

Obrázek 6-4 ukazuje tabulku pro parametr TBL; v tabulce 6-10 je seznam chybových kódů.

Offset bytu	7	0			
0	D	A	E	0	Kód chyby
1	Adresa vzdálené stanice				
2	Ukazatel na data				
3	oblast ve				
4	vzdálené stanici				
5	(I, Q, M nebo V)				
6	Délka dat				
7	Datový byte 0				
8	Datový byte 1				
	:				
22	Datový byte 15				

D Provedeno (funkce byla ukončena): 0 = nedokončena 1 = ukončena
A Aktivní (funkce byla zařazena do fronty): 0 = není aktivní 1 = aktivní
E Chyba (funkce byla ukončena s chybou): 0 = bez chyby 1 = chyba

Adresa vzdálené stanice: adresa PLC, k jejímž datům se má přistupovat.

Ukazatel na datovou oblast ve vzdálené stanici: nepřímý ukazatel na přistupovaná data.

Délka dat: počet bytů, ke kterým se přistupuje na vzdálené stanici (1 až 16 bytů).

Datová oblast pro vysílání nebo příjem. 1 až 16 bytů pro data.

U instrukce "Čtení ze sítě" ukládá hodnoty, které byly při provádění instrukce přečteny ze vzdálené stanice.

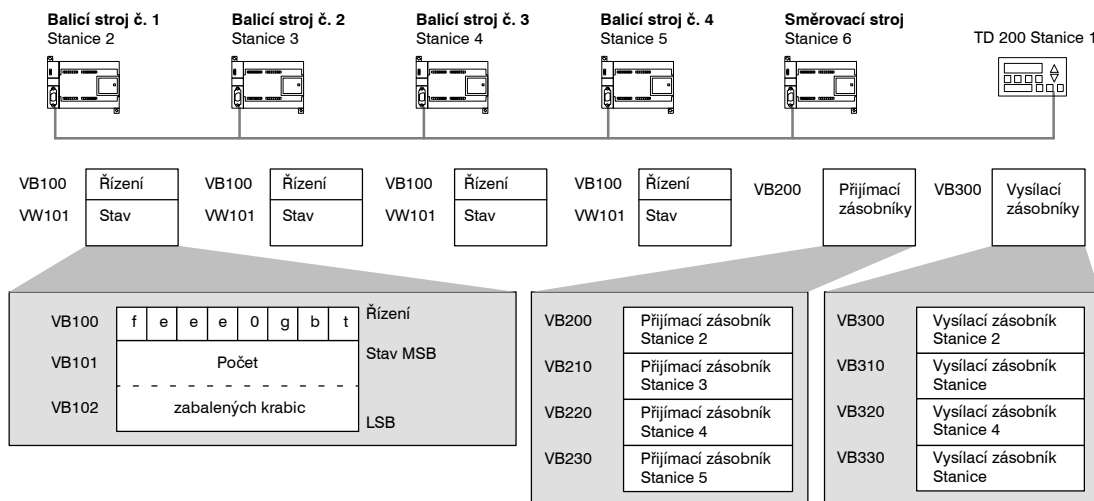
U instrukce "Zápis do sítě" ukládá hodnoty, které mají být při provádění instrukce odeslány na vzdálenou stanici.

Obr. 6-4 Parametr TBL pro instrukce "Čtení ze sítě" a "Zápis do sítě"

Tabulka 6-10 Chybové kódy pro parametr TBL

Kód	Definice
0	Bez chyby.
1	Překročení časového limitu: Vzdálená stanice neodpovídá.
2	Chyba příjmu: Chyba parity, rámce nebo kontrolního součtu v odpovědi.
3	Chyba offline: Kolize způsobené zdvojenou adresou stanice nebo chybou hardwaru.
4	Chyba přeplnění fronty: Bylo aktivováno více než 8 instrukcí "Čtení ze sítě" nebo "Zápis do sítě".
5	Porušení protokolu: Pokus o provedení instrukce "Čtení ze sítě" nebo "Zápis do sítě" bez povolení PPI Master Mode v SMB30 nebo SMB130.
6	Nepřípustný parametr: Parametr TBL obsahuje nepřípustnou nebo neplatnou hodnotu.
7	Chybí zdroj: Vzdálená stanice je obsazená. (Probíhá upload nebo download.)
8	Chyba úrovně: Porušení protokolu aplikace
9	Chyba zprávy: Chybná adresa dat nebo nesprávná délka dat
A až F	Nepoužito. (Rezervováno)

Obrázek 6-5 ukazuje příklad použití instrukcí "Čtení ze sítě" a "Zápis do sítě". V tomto příkladu jde o výrobní linku, kde se plní vaničky máslem a posílají se do jednoho ze čtyř balicích strojů (do krabic). Balicí stroj zabalí osm vaniček másla do jedné lepenkové krabice. Směrovací stroj řídí přísun vaniček másla do každého balicího stroje. Balicí stroje řídí čtyři programovatelné automaty S7-200, jeden S7-200 s textovým displejem TD 200 řídí směrovač.



- t Došly vaničky s máslem pro balicí stroj; t=1, došly vaničky na máslo
 b Docházejí krabice; b=1, nutné doplnit krabice v příštích 30 minutách
 g Dochází lepidlo; g=1, nutné doplnit lepidlo v příštích 30 minutách
 eee chybový kód, který určuje kód chyby, k níž došlo
 f Indikátor chyby; f=1, balicí stroj zaznamenal chybu

Obr. 6-5 Příklad instrukcí "Čtení ze sítě" a "Zápis do sítě"

Obrázek 6-6 ukazuje přijímací zásobník (VB200) a vysílací zásobník (VB300) pro přístup k datům ve stanici 2. Automat S7-200 používá instrukci "Čtení ze sítě" pro trvalé čtení řídicích a stavových informací z každého balicího stroje. Směrovač zaznamená každé zabalení 100 krabic balicím strojem a pošle zprávu s žádostí o vyčištění stavového slova pomocí instrukce "Zápis do sítě".

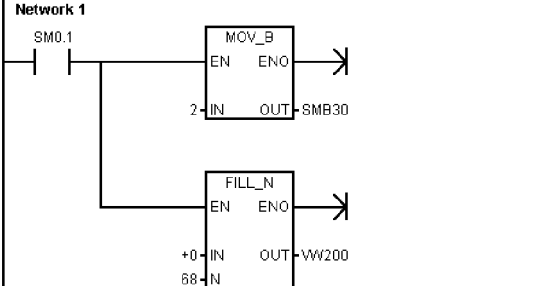
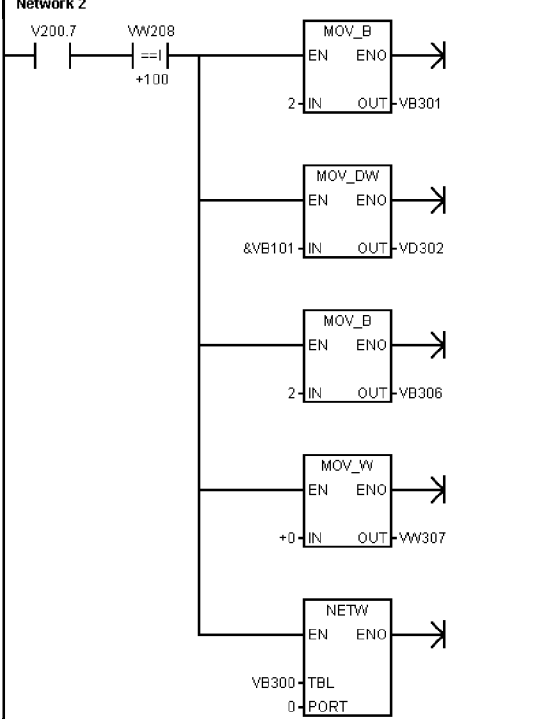
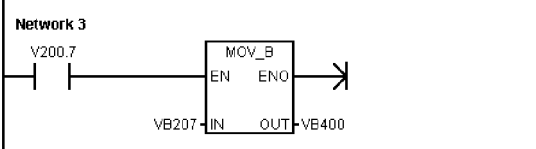
Přijímací zásobník pro čtení z balicího stroje č. 1

	7				0
VB200	D	A	E	0	Kód chyby
VB201	Adresa vzdálené stanice = 2				
VB202	Ukazatel na				
VB203	datovou oblast				
VB204	ve				
VB205	Vzdálená stanice = (&VB100)				
VB206	Délka dat = 3 byty				
VB207	Řízení				
VB208	Stav (MSB)				
VB209	Stav (LSB)				

Vysílací zásobník pro vymazání počtu v balicím stroji č. 1

	7	6	5	4	3	2	1	0
VB300	D	A	E	0	Kód chyby			
VB301	Adresa vzdálené stanice = 2							
VB302	Ukazatel na							
VB303	datovou oblast							
VB304	ve							
VB305	Vzdálená stanice = (&VB101)							
VB306	Délka dat = 2 byty							
VB307	0							
VB308	0							

Obr. 6-6 Vzorová data TBL pro uvedený příklad "Čtení a zápis do sítě"

Příklad: Instrukce čtení a zápisu do sítě	
Network 1  Network 2  Network 3 	Network 1 //V prvním programovém cyklu povolí PPI protokol //a vymaže všechny přijímací a vysílací zásobníky. LD SM0.1 MOVB 2, SMB30 FILL +0, VW200, 68 Network 2 //Pokud je nastaven bit Provedeno NETR (V200.7) //a bylo zabaleno 100 krabic: //1. Nahraje adresu stanice balicího stroje č. 1. //2. Nahraje ukazatel na data ve vzdálené stanici. //3. Nahraje délku vysílaných dat. //4. Nahraje vysílaná data. //5. Vynuluje počet krabic zabalených // balicím strojem č. 1 LD V200.7 AW= VW208, +100 MOVB 2, VB301 MOVD &VB101, VD302 MOVB 2, VB306 MOVW +0, VW307 NETW VB300, 0 Network 3 //Pokud je nastaven bit Provedeno NETR, uloží řídicí //data z balicího stroje č. 1. LD V200.7 MOVB VB207, VB400

6

Network 4

SM0.1

V200.6

V200.5

MOV_B

EN

ENO

2

IN

OUT

VB201

MOV_DW

EN

ENO

&VB100

IN

OUT

VD202

MOV_B

EN

ENO

3

IN

OUT

VB206

NETR

EN

ENO

VB200

TBL

0

PORT

Network 4

//Pokud se nevyskytla chyba a neběží první

//programový cyklus:

//1. Nahraje adresu stanice balicího stroje č. 1.

//2. Nahraje ukazatel na data ve vzdálené stanici.

//3. Nahraje délku přijímaných dat.

//4. Čte řídicí a stavová data

// v balicím stroji č. 1

LDN

AN

AN

AN

MOVB

MOVD

MOVB

NETR

SM0.1

V200.6

V200.5

2, VB201

&VB100, VD202

3, VB206

VB200, 0

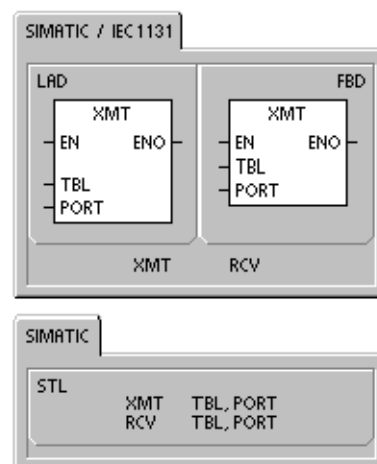
Vysílání a příjem (Freeport)

Instrukce "Vysílání" (XMT) se používá pro přenos dat v režimu Freeport.

Instrukce "Příjem" (RCV) vyvolá nebo ukončí funkci přijetí zprávy. Je nutné zadat počáteční a koncovou podmínku pro práci funkce "Příjem". Zprávy přijaté přes daný port jsou uloženy v datovém zásobníku (TBL). První položka v datovém zásobníku udává počet přijatých bytů.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0009 (současné vysílání i příjem na portu 0)
- 000B (současné vysílání i příjem na portu 1)
- Chyba příjmu nastaví SM86.6 nebo SM186.6
- CPU S7-200 není v režimu Freeport



Tabulka 6-11 Operandy platné pro instrukce "Vysílání" a "Příjem"

Vstupy/Výstupy	Typ dat	Operandy
TBL	BYTE	IB, QB, VB, MB, SMB, SB, *VD, *LD, *AC
PORT	BYTE	Konstanta <i>pro CPU 221, CPU 222, CPU 224:</i> 0 <i>pro CPU 226 a CPU 226XM:</i> 0 nebo 1

Více informací o používání režimu Freeport najdete v odstavci Vytváření protokolů definovaných uživatelem v režimu Freeport na straně 222 v kapitole 7.

Použití režimu Freeport pro řízení sériových portů

Režim Freeport je možné využít pro řízení komunikace sériového portu automatu S7-200 pomocí uživatelského programu. Jestliže vyberete režim Freeport, řídí uživatelský program komunikační port pomocí přerušení instrukcemi "Vysílání" a "Příjem". V režimu Freeport je komunikační protokol zcela řízen uživatelským programem. Pro nastavení přenosové rychlosti a parity se používají SMB30 (pro port 0) a SMB130 (pro port 1, pokud má automat S7-200 dva porty).

Při přechodu S7-200 do režimu STOP se režim Freeport zastaví a obnoví se normální komunikace (např. pro spojení s počítačem).

V nejjednodušším případě můžete pomocí instrukce "Vysílání" (XMT) poslat zprávu tiskárně nebo displeji. Jiné využití je například pro spojení se snímačem čárového kódu, elektronickými vahami nebo svářečkou. V každém případě je nutné napsat program tak, aby podporoval protokol používaný zařízením, se kterým automat S7-200 komunikuje v režimu Freeport.

Komunikace Freeport je možná pouze tehdy, je-li S7-200 v režimu RUN. Režim Freeport se aktivuje nastavením hodnoty 01 v SMB30 (port 0) nebo SMB130 (port 1). V režimu Freeport není možná komunikace s počítačem.

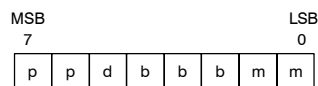


Tip

Režim Freeport je možné řídit pomocí speciálního paměťového bitu SM0.7, který odráží aktuální polohu přepínače pracovního režimu. Když je SM0.7 roven log. 0, je přepínač v poloze TERM; pokud je SM0.7 = 1, je přepínač pracovního režimu v poloze RUN. Pokud povolíte režim Freeport pouze v případě, že přepínač je v poloze RUN, můžete použít programovací zařízení na monitorování nebo řízení činnosti S7-200 při přepnutí přepínače do jiné polohy.

Změna komunikace PPI na režim Freeport

SMB30 a SMB130 konfiguruji komunikační porty 0 a 1 pro režim Freeport a nastavují přenosovou rychlost, paritu a počet datových bitů. Obrázek 6-7 ukazuje řídicí byte pro Freeport. Pro všechny konfigurace je přidán jeden stop bit.



SMB30 = Port 0
SMB130 = Port 1

pp: Výběr parity
00 = bez parity
01 = sudá parita
10 = bez parity
11 = lichá parita

d: Počet datových bitů na znak
0 = 8 bitů na znak
1 = 7 bitů na znak

bbb: přenosová rychlost pro Freeport

000 = 38 400 baud
001 = 19 200 baud
010 = 9 600 baud
011 = 4 800 baud
100 = 2 400 baud
101 = 1 200 baud
110 = 115,2 kbaud¹
111 = 57,6 kbaud¹

¹ CPU S7-200 verze 1.2 nebo vyšší podporuje rychlost přenosu 57,6 kbaud a 115,2 kbaud.

mm: Výběr protokolu

00 = režim PPI/slave
01 = protokol Freeport
10 = režim PPI/master
11 = Rezervováno (implicitně režim PPI/slave)

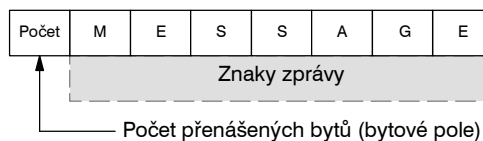
Obr. 6-7 Řídicí byte SM pro režim Freeport (SMB30 nebo SMB130)

Vysílání dat

Instrukce "Vysílání" umožňuje přenos jednoho nebo více znaků (maximálně 255).

Obrázek 6-8 ukazuje formát přenosového zásobníku.

Pokud je ukončení přenosu řízeno podprogramem přerušení, generuje S7-200 přerušení po vyslání posledního znaku ze zásobníku (přerušení 23 pro port 0 a přerušení 26 pro port 1).



Obr. 6-8 Formát vysílacího zásobníku

Přenos je možné uskutečnit bez použití přerušení (např. odeslání zprávy tiskárně) monitorováním SM4.5 nebo SM4.6 pro ukončení přenosu.

Instrukci "Vysílání" je možné použít pro generování podmínky BREAK nastavením počtu znaků na nulu a následným provedením instrukce. To generuje podmínku BREAK po dobu 16bitů při aktuální přenosové rychlosti. BREAK se vysílá stejně jako libovolná jiná zpráva, po ukončení se generuje přerušení vysílání a SM4.5 nebo SM4.6 signalizuje aktuální stav operace vysílání.

Příjem dat

Instrukce "Příjem" umožňuje příjem zásobníku složeného z jednoho nebo více znaků až do 255.

Obrázek 6-9 ukazuje formát přijímacího zásobníku.

Pokud je ukončení přenosu řízeno podprogramem přerušení, generuje S7-200 přerušení po přijetí posledního znaku do zásobníku (přerušení 23 pro port 0 a přerušení 26 pro port 1).



Obr. 6-9 Formát přijímacího zásobníku

Zprávy je možné přijímat bez použití přerušení monitorováním SMB86 (port 0) nebo SMB186 (port 1). Pokud je instrukce "Příjem" neaktivní nebo byla ukončena, je tento byte roven log. 1. Nulový je v případě, že přijímání pokračuje.

Jak je vidět v tabulce 6-12, instrukce "Příjem" umožňuje zvolit podmínku začátku zprávy a konce zprávy použitím SMB86 až SMB94 pro port 0 a SMB186 až SMB194 pro port 1.



Tip

Funkce příjmu zprávy se automaticky ukončí při přetečení nebo chybě parity. Aby byla funkční, je třeba definovat počáteční podmínku a ukončovací podmínku (maximální počet znaků).

Tabulka 6-12 Byty přijímacího zásobníku (SMB86 až SMB94 a SMB186 až SMB194)

Port 0	Port 1	Popis								
SMB86	SMB186	Stavový byte pro příjem zprávy MSB7 LSB0 <table><tr><td>n</td><td>r</td><td>e</td><td>0</td><td>0</td><td>t</td><td>c</td><td>p</td></tr></table> n: 1 = Funkce příjmu zprávy ukončena: uživatel vydal příkaz k odpojení. r: 1 = Funkce příjmu zprávy ukončena: chyba vstupních parametrů nebo chybí startovací či ukončovací podmínka. e: 1 = Přijat ukončovací znak. t: 1 = Funkce příjmu zprávy ukončena: vypršel čas. c: 1 = Funkce příjmu zprávy ukončena: dosaženo maximálního počtu znaků. p: 1 = Funkce příjmu zprávy ukončena: chyba parity.	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			
SMB87	SMB187	Řídící byte pro příjem zprávy MSB7 LSB0 <table><tr><td>en</td><td>sc</td><td>ec</td><td>il</td><td>c/m</td><td>tmr</td><td>bk</td><td>0</td></tr></table> en: 0 =Funkce příjmu zpráv není povolena. 1 =Funkce příjmu zpráv je povolena. Povolovací bit příjmu zpráv je kontrolován při každém provedení instrukce RCV. sc: 0 =Ignorovat SMB88 nebo SMB188. 1 =Použít hodnotu SMB88 nebo SMB188 pro rozpoznání začátku zprávy. ec: 0 =Ignorovat SMB89 nebo SMB189. 1 =Použít hodnotu SMB89 nebo SMB189 pro rozpoznání konce zprávy. il: 0 =Ignorovat SMW90 nebo SMW190. 1 =Použít hodnotu SMW90 nebo SMW190 pro rozpoznání prázdného úseku. c/m: 0 =Časovač je použit na časování mezi znaky. 1 =Časovač je použit na časování zpráv. tmr: 0 =Ignorovat SMW92 nebo SMW192. 1 =Ukončit příjem, pokud byl čas v SMW92 nebo SMW192 překročený. bk: 0 =Ignorovat BREAK (přestávku). 1 =Použít BREAK (přestávku) pro detekci začátku zprávy.	en	sc	ec	il	c/m	tmr	bk	0
en	sc	ec	il	c/m	tmr	bk	0			
SMB88	SMB188	Znak začátku zprávy.								
SMB89	SMB189	Znak konce zprávy.								
SMW90	SMW190	Doba prázdného úseku uvedená v milisekundách. První znak přijatý po skončení prázdného úseku je startovací znak.								
SMW92	SMW192	Hraniční hodnota pro časovač (v ms) odměřující čas mezi znaky/zprávami. Pokud je hodnota překročena, je funkce příjmu zpráv ukončena.								
SMB94	SMB194	Maximální počet přijatých znaků (1 až 255 bytů). Tato hodnota musí být nastavena na předpokládanou maximální velikost zásobníku i v případě, že není použito ukončení příjmu zprávy při dosažení maximálního počtu znaků.								

Podmínky začátku a ukončení pro instrukci "Příjem"

Instrukce "Příjem" používá pro definování podmínek začátku a ukončení zprávy bity řídicího bytu příjmu zprávy (SMB87 nebo SMB187).



Tip

Pokud na portu probíhá komunikace s jinými zařízeními, když se provádí instrukce "Příjem", může funkce příjmu zprávy začít přijímat znak odprostřed, což má za následek eventuální chybu parity a ukončení příjmu zprávy. Pokud není povolena parita, může přijatá zpráva obsahovat nesprávné znaky. K této situaci může dojít, když je podmínka začátku zprávy určena jako konkrétní startovací znak nebo libovolný znak popsany v bodu 2. a bodu 6. níže.

Instrukce "Příjem" podporuje několik podmínek začátku zprávy. Problém specifikace začátku zprávy, který zahrnuje rozpoznání BREAK (přestávky) nebo prázdného úseku řešíme tak, že přinutíme synchronizovat začátek zprávy se začátkem znaku dříve, než se znaky umístí do zásobníku.

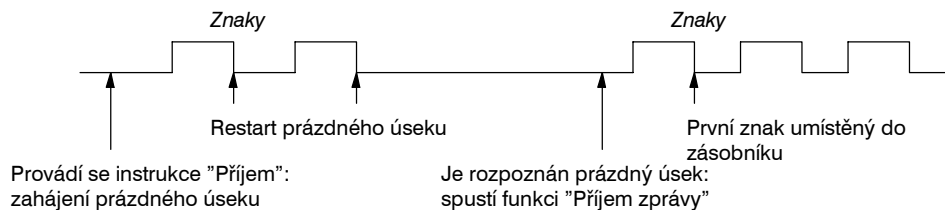
Instrukce "Příjem" podporuje několik podmínek začátku zprávy:

1. **Rozpoznání prázdného úseku:** Prázdný úsek je definován jako doba "ticha" nebo nečinnosti přenosového vedení. Příjem je zahájen, v případě, že komunikační linka nebyla v činnosti určitý počet milisekund, specifikovaných v SMW90 nebo SMW190. Když se provádí instrukce "Příjem", iniciuje funkce příjmu zprávy vyhledávání podmínky prázdného úseku. Pokud jsou přijaty nějaké znaky dříve, než vyprší doba prázdného úseku, funkce příjmu zprávy tyto znaky ignoruje a znovu spustí časovač prázdného úseku s časem z SMW90 nebo SMW190. Viz obr. 6-10. Po vypršení času prázdného úseku uloží funkce příjmu zprávy všechny následující přijaté znaky do zásobníku.

Doba prázdného úseku by měla být vždy delší než doba přenosu jednoho znaku (start bit, datové bity, parita a stop bity) při dané přenosové rychlosti. Běžná hodnota doby prázdného úseku je trojnásobek doby přenosu znaku při dané přenosové rychlosti.

Rozeznání prázdného úseku se používá jako podmínka pro zahájení binárních protokolů, kde neexistuje konkrétní startovací znak, nebo tam, kde protokol specifikuje minimální čas mezi zprávami.

Nastavení: $il = 1$, $sc = 0$, $bk = 0$, SMW90/SMW190 = časová prodleva prázdného úseku v milisekundách



Obr. 6-10 Použití rozeznání prázdného úseku pro spuštění instrukce "Příjem"

2. **Rozpoznání startovacího znaku:** Startovací znak je libovolný znak použitý jako první znak zprávy. Zpráva začíná, když je přijat startovací znak specifikovaný v SMB88 nebo SMB188. Funkce příjmu zprávy uloží startovací znak v příjmovém zásobníku jako první znak zprávy. Tato funkce ignoruje jakékoliv znaky přijaté před startovacím znakem. Startovací znak a všechny znaky přijaté po něm se uloží do zásobníku.

Rozpoznání startovacího znaku se běžně používá pro protokoly ASCII, ve kterých všechny zprávy začínají stejným znakem.

Nastavení: $il = 0$, $sc = 1$, $bk = 0$, SMW90/SMW190 = není podstatný, SMB88/SMB188 = startovací znak

3. *Prázdný úsek a startovací znak:* Při instrukci "Příjem" může zpráva začínat kombinací prázdného úseku a startovacího znaku. Když se provádí instrukce "Příjem", vyhledává funkce příjmu zprávy podmínku prázdného úseku. Po nalezení prázdného úseku hledá funkce příjmu zprávy specifikovaný startovací znak. Jestliže je nalezen jakýkoliv jiný znak než startovací, znovu se zahájí vyhledávání podmínky prázdného úseku. Všechny znaky přijaté před tím, než je splněna podmínka prázdného úseku a než je přijat startovací znak, jsou ignorovány. Startovací znak je umístěn do zásobníku spolu se všemi následujícími znaky.

Doba prázdného úseku by vždy měla být delší než doba přenosu jednoho znaku (start bit, datové bity, parita a stop bity) při dané přenosové rychlosti. Běžná hodnota doby prázdného úseku je trojnásobek doby přenosu znaku při dané přenosové rychlosti.

Běžně se tento typ startovací podmínky užívá u protokolu, který specifikuje minimální čas mezi zprávami a první znak je adresa nebo něco, co specifikuje konkrétní zařízení. Nejužitečnější je to při implementaci protokolu, kdy je na komunikační spoj připojeno více zařízení. V tomto případě instrukce "Příjem" spustí přerušování pouze tehdy, je-li přijata zpráva pro konkrétní adresu nebo zařízení specifikované startovacím znakem.

Nastavení:

il = 1, sc = 1, bk = 0, SMW90/SMW190 > 0, SMB88/SMB188 = startovací znak

4. *Rozpoznání BREAK (přestávky):* Přestávka je indikována tehdy, když přijímaná data setrvají na nulové hodnotě po dobu delší, než je doba přenosu úplného znaku. Doba přenosu úplného znaku se definuje jako celková doba start bitu, datových bitů, parity a stop bitů. Pokud je instrukce "Příjem" konfigurována tak, aby zahájila zprávu při příjmu podmínky přestávky, jsou jakékoliv znaky přijaté po podmínce umístěny do zásobníku. Jakékoliv znaky přijaté před přestávkou jsou ignorovány.

Běžně se rozpoznání přestávky používá jako podmínka pro začátek zprávy pouze tehdy, když to vyžaduje protokol.

Nastavení:

il = 0, sc = 0, bk = 1, SMW90/SMW190 = není podstatný, SMB88/SMB188 = není podstatný

5. *Znak BREAK (přestávky) a startovací znak:* Instrukci "Příjem" je možné konfigurovat tak, aby začala přijímat znaky po přijetí sledu přestávky a následujícího konkrétního startovacího znaku. Po podmínce přestávky se hledá specifikovaný startovací znak. Jestliže je nalezen jakýkoliv jiný znak než startovací, znovu se zahájí vyhledávání podmínky přestávky. Všechny znaky přijaté před splněním podmínky přestávky a před příjmem startovacího znaku jsou ignorovány. Startovací znak je umístěn do zásobníku spolu se všemi následujícími znaky.

Nastavení:

il = 0, sc = 1, bk = 1, SMW90/SMW190 = není podstatný,
SMB88/SMB188 = startovací znak

6. *Libovolný znak:* Instrukci "Příjem" je možné nakonfigurovat tak, aby zahájila okamžitý příjem jakýchkoliv znaků a umístila je do zásobníku. Jde o speciální případ rozpoznání prázdného úseku. V tomto případě je čas prázdného úseku (SMW90 nebo SMW190) nastaven na nulu. To nutí instrukci "Příjem" zahájit příjem znaků okamžitě po provedení.

Nastavení:

il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 = není podstatný

To, že zpráva začíná libovolným znakem, umožňuje použití časovače pro načasování příjmu zprávy. To je užitečné v případech, kdy se režim Freeport použije pro implementaci části protokolu "master" nebo "host" a je zapotřebí prodleva, pokud ze slave nepřijde ve specifikovaném časovém úseku odezva. Časovač zahájí činnost při provádění instrukce "Příjem", protože doba prázdného úseku byla nastavena na nulu. Časovač pozdrží a ukončí funkci příjmu, pokud není splněna žádná jiná podmínka pro její ukončení.

Nastavení:

il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 = není podstatný
c/m = 1, tmr = 1, SMW92 = prodleva (timeout) v milisekundách

Instrukce "Příjem" podporuje několik způsobů ukončení zprávy. Zprávu může ukončit jedno nebo kombinace následujících kritérií:

1. *Rozpoznání ukončovacího znaku:* Ukončovací znak je libovolný znak použitý pro označení konce. Po nalezení podmínky pro začátek kontroluje instrukce "Příjem" každý další přijatý znak, zda odpovídá ukončovacímu znaku. Když je přijat ukončovací znak, je uložen do zásobníku a příjem je ukončen.

Běžně se rozpoznání ukončovacího znaku používá v protokolech ASCII, kde každá zpráva končí specifickým znakem. Pro ukončení můžete rozpoznání ukončovacího znaku použít v kombinaci s časovačem měřícím čas mezi znaky, časovačem zprávy nebo maximálním počtem přijatých znaků.

Nastavení:

ec = 1, SMB89/SMB189 = ukončovací znak

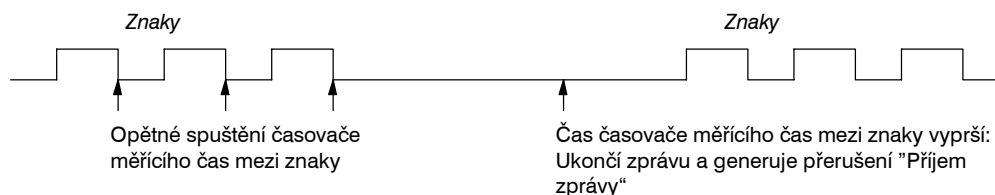
2. *Časovač měřící čas mezi znaky:* Čas mezi znaky je doba naměřená od konce jednoho znaku (stop bit) do konce následujícího znaku (stop bit). Pokud čas mezi znaky (včetně druhého znaku) překročí počet milisekund specifikovaných v SMW92 nebo SMW192, je funkce příjmu zprávy ukončena. Časovač měřící čas mezi znaky je znovu spuštěn po každém přijatém znaku. Viz obr. 6-11.

Časovač měřící čas mezi znaky je možné použít pro ukončení zprávy v protokolech, které nemají konkrétní ukončovací znak. Tento časovač musí být nastaven na hodnotu vyšší, než je doba jednoho znaku při zvolené přenosové rychlosti, protože časovač vždy zahrnuje čas pro příjem jednoho celého znaku (start bit, datové bity, paritní bit a stop bity).

Pro ukončení zprávy můžete použít časovač měřící čas mezi znaky v kombinaci s rozpoznáním ukončovacího znaku a s maximálním počtem přijatých znaků.

Nastavení:

c/m = 0, tmr = 1, SMW92/SMW192 = prodleva (timeout) zprávy v milisekundách



Obr. 6-11 Použití časovače měřícího čas mezi znaky pro ukončení instrukce "Příjem"

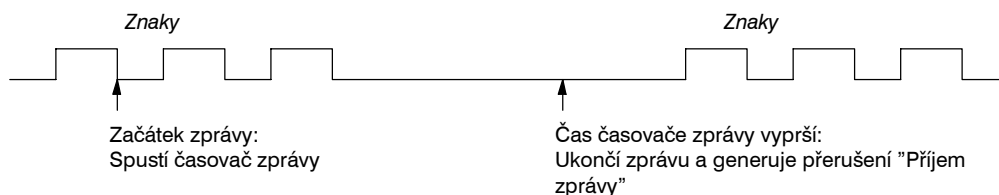
3. *Časovač zprávy:* Časovač zprávy ukončí zprávu ve specifikovaný čas po začátku. Časovač zprávy zahájí činnost, jakmile byla splněna/byly splněny podmínka/podmínky pro začátek příjmu. Doba časovače zprávy vyprší po uplynutí určitého počtu milisekund, specifikovaného v SMW92 nebo SMW192. Viz obr. 6-12.

Časovač zprávy se běžně používá tehdy, nemůže-li komunikační zařízení zaručit, že nenastanou časové mezery mezi znaky, nebo při použití modemů. U modemů můžete časovač zprávy využít pro zadání maximální doby povolené pro příjem po jejím zahájení. Typická hodnota časovače zprávy bude asi 1,5násobek doby potřebné pro příjem nejdelší možné zprávy při dané přenosové rychlosti.

Pro ukončení zprávy můžete použít časovač zprávy v kombinaci s rozpoznáním ukončovacího znaku a s maximálním počtem přijatých znaků.

Nastavení:

c/m = 1, tmr = 1, SMW92/SMW192 = prodleva (timeout) v milisekundách



Obr. 6-12 Použití časovače zprávy pro ukončení instrukce "Příjem"

4. *Maximální počet přijatých znaků:* V instrukci "Příjem" musí být zadán maximální počet přijatých znaků (SMB94 nebo SMB194). Když je této hodnoty dosaženo nebo když je překročena, je funkce příjmu ukončena. Instrukce "Příjem" vyžaduje, aby uživatel specifikoval maximální počet přijatých znaků, i když to není přímo podmínka pro ukončení zprávy. Důvodem je, že instrukce "Příjem" potřebuje znát maximální velikost přijímané zprávy, aby nebyla přepsána uživatelská data umístěná za zásobníkem.

Maximální počet přijatých znaků se používá pro ukončení zpráv v protokolech, kde je délka zprávy známá a vždy stejná. Maximální počet přijatých znaků se vždy používá v kombinaci s rozpoznáním ukončovacího znaku, časovačem měřícím čas mezi znaky nebo s časovačem zprávy.

5. *Chyby parity:* Instrukce "Příjem" je automaticky ukončena, když hardware signalizuje chybu parity u přijatého znaku. Chyby parity jsou možné pouze tehdy, jestliže je parita povolena v SMB30 nebo SMB130. Neexistuje způsob, jak tuto funkci zablokovat.
6. *Ukončení uživatelem:* Uživatelský program může ukončit příjem zprávy provedením jiné instrukce "Příjem" s povolovacím bitem (EN) v SMB87 nebo SMB187 nastaveným na nulu. Tím je příjem zprávy okamžitě ukončen.

Použití přerušení pro řízení příjmu dat

Pro umožnění úplné flexibility v podporovaných protokolech je možné přijímat data pomocí přerušení. Každý přijatý znak generuje přerušení. Přijatý znak se uloží do SMB2 a parita (pokud je povolena) se uloží do SMB3.0 bezprostředně před provedením podprogramu přerušení přiřazeného k události přijetí znaku. SMB2 je zásobník pro znak přijatý z Freeportu. Každý znak přijatý během režimu Freeport je umístěn na tuto adresu kvůli jednoduchému přístupu z uživatelského programu. SMB3 se používá pro režim Freeport a obsahuje bit chyby parity, který se aktivuje při detekci chyby parity na přijatém znaku. Všechny ostatní bity tohoto bytu jsou rezervované. Paritní bit se používá pro vyřazení zprávy nebo pro generování negativního potvrzení zprávy.

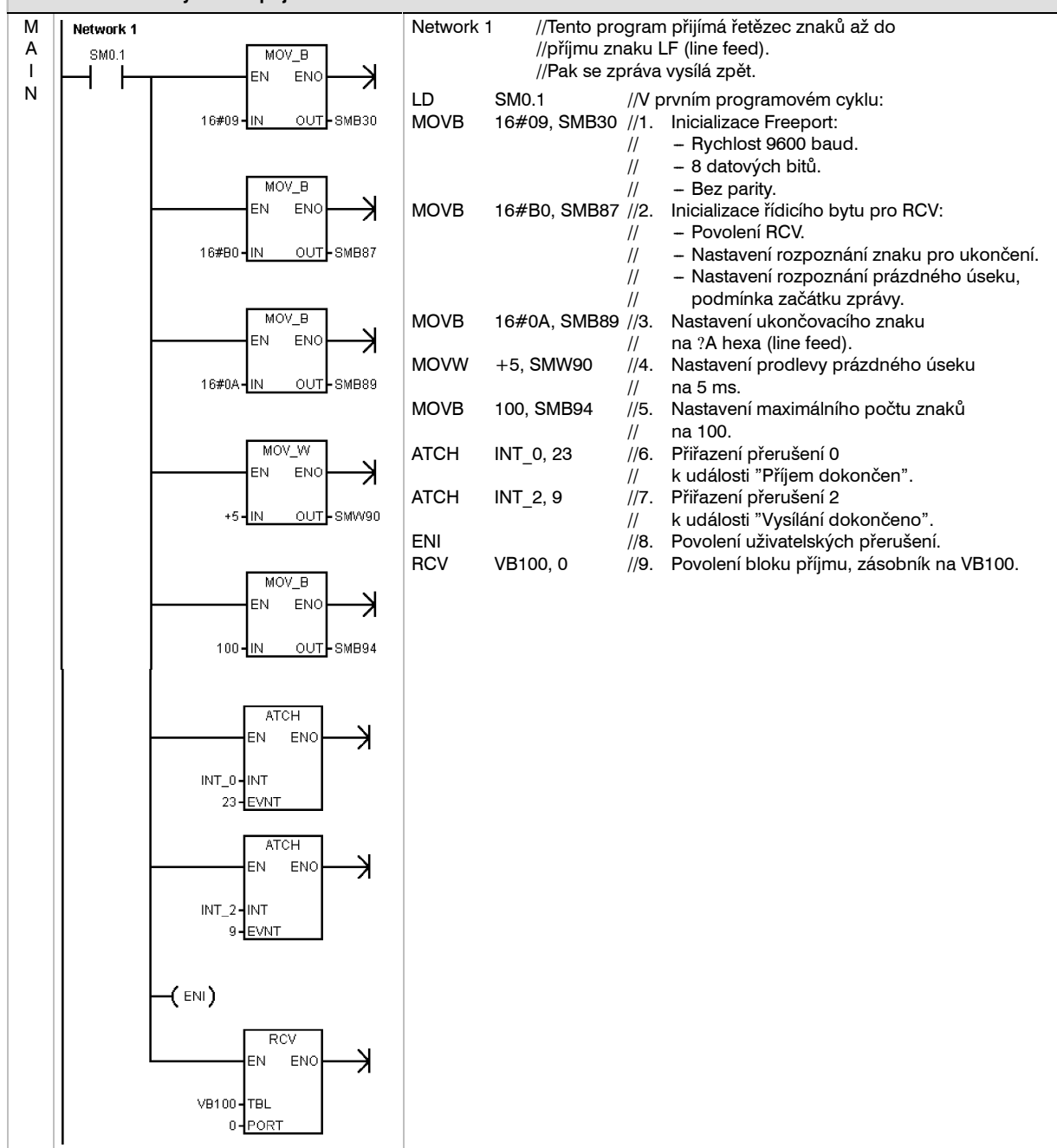
Jestliže se přerušení použije při vysokých přenosových rychlostech (38,4 kbaud až 115,2 kbaud), je doba mezi přerušeními velmi krátká. Například přerušení pro 38,4 kbaud je 260 mikrosekund, pro 57,6 kbaud je 173 mikrosekund a pro 115,2 kbaud je 86 mikrosekund. Abyste zabránili vynechání znaků, zajistěte, aby byly podprogramy přerušení vždy velmi krátké, nebo použijte instrukci "Příjem".



Tip

SMB2 a SMB3 jsou sdíleny mezi portem 0 a portem 1. Po přijetí znaku se na portu 0 provede podprogram přerušení přiřazený k této události (událost 8), SMB2 obsahuje znak přijatý na portu 0 a SMB3 obsahuje stav parity přijatého znaku. Pokud po přijetí znaku na portu 1 provede podprogram přerušení přiřazený k této události (událost 25), SMB2 obsahuje znak přijatý na portu 1 a SMB3 obsahuje stav parity přijatého znaku.

Příklad: Instrukce vysílání a příjmu



Příklad: Instrukce vysílání a příjmu, pokračování		
INT 0	Network 1 	Network 1 //Podprogram přerušení "Příjem dokončen": //1. Pokud stav příjmu signalizuje příjem ukončovacího znaku, // přiřadí 10ms časovač na spuštění vysílání a návratu. //2. Pokud byl příjem ukončen z jakéhokoliv jiného důvodu, // spustí nový příjem. LDB= SMB86, 16#20 MOVB 10, SMB34 ATCH INT_1, 10 CRETI NOT RCV VB100, 0
INT 1	Network 1 	Network 1 //Přerušení 10ms časovače: //1. Odpojí časové přerušení. //2. Vyšle zprávu zpět uživateli na portu. LD SM0.0 DTCH 10 XMT VB100, 0
INT 2	Network 1 	Network 1 //Přerušení "Vysílání dokončeno": //Povolí další příjem. LD SM0.0 RCV VB100, 0

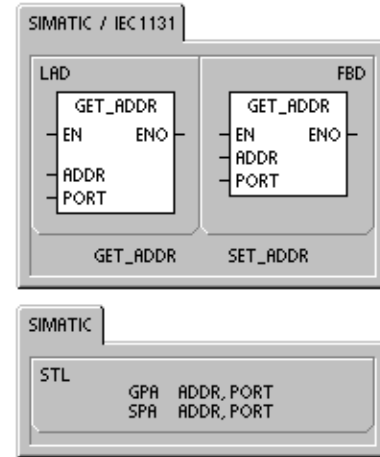
Zjištění a nastavení adresy portu

Instrukce "Zjistit adresu portu" (GPA) přečte adresu stanice z portu CPU S7-200 daného v PORT a hodnotu zapíše na adresu ADDR.

Instrukce "Nastavit adresu portu" (SPA) nastavuje adresu portu stanice (PORT) na hodnotu danou v ADDR. Nová adresa není uložena trvale. Po výpadku a obnovení napájení se daný port vrátí k poslední adrese (adrese, která byla nahrána společně se systémovým blokem).

Chyby, které nastavují ENO = 0:

- 0006 (nepřímá adresa)
- 0004 (pokus provést instrukci "Nastavit adresu portu" v přerušovacím podprogramu)



Tabulka 6-13 Operandy platné pro instrukce "Zjistit adresu portu" a "Nastavit adresu portu"

Vstupy/Výstupy	Typ dat	Operandy
ADDR	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta (Konstantní hodnota platí pouze pro instrukci "Nastavit adresu portu".)
PORT	BYTE	Konstanta pro CPU 221, CPU 222, CPU 224: 0 pro CPU 226 a CPU 226XM: 0 nebo 1

Instrukce porovnání

Porovnání číselných hodnot

Instrukce porovnání se používají pro porovnání dvou hodnot:

IN1 = IN2 IN1 >= IN2 IN1 <= IN2
IN1 > IN2 IN1 < IN2 IN1 <> IN2

Operace porovnání bytu jsou bez znaménka.
Operace porovnání celých čísel (integer) jsou se znaménkem.
Operace porovnání double word jsou se znaménkem.
Operace porovnání reálných čísel jsou se znaménkem.

V LAD a FBD: Je-li výsledek porovnání pravdivý, instrukce nastaví kontakt (LAD) nebo výstup (FBD) na hodnotu log. 1.

V STL: Je-li výsledek porovnání pravdivý, provedou se instrukce Načtení, AND nebo OR s jedničkou a vrcholem zásobníku (STL).

Použijete-li instrukce porovnávání IEC, můžete pro vstupy použít různé typy dat. Obě vstupní hodnoty však musí být stejného typu.

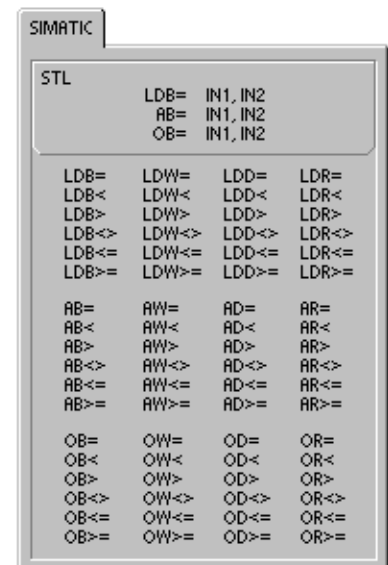
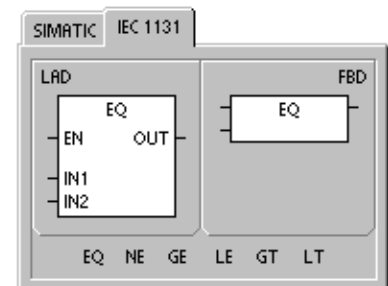
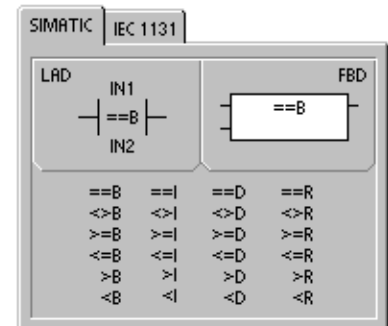
Poznámka

Následující podmínky jsou kritické chyby, které způsobí, že programovatelný automat S7-200 okamžitě zastaví provádění programu:

- Nepřípustná nepřímá adresa (kterákoliv instrukce porovnání)
- Nepřípustné reálné číslo (například NAN) (instrukce porovnání reálných čísel)

Abyste zabránili vzniku těchto chyb, zjistěte správnou inicializaci ukazatele a hodnoty, které obsahují reálná čísla, dříve než se vykoná instrukce porovnání.

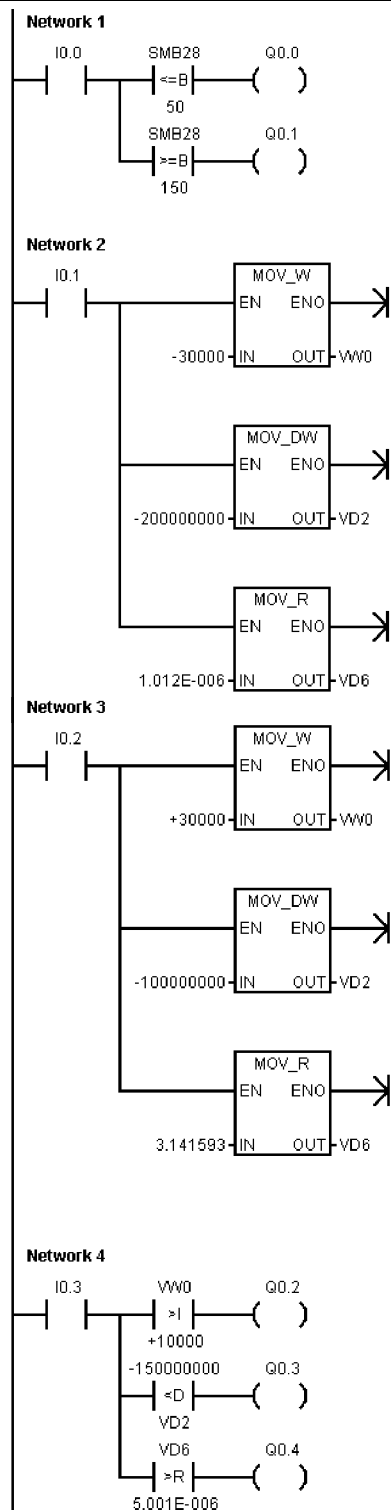
Instrukce porovnání se provádějí bez ohledu na stav signálového toku.



Tabulka 6-14 Operandy platné pro instrukce porovnání

Vstupy/Výstupy	Typy dat	Operandy
IN1, IN2	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
Výstup (OUT)	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok

Příklad: Porovnávání



Network 1 //Otáčejte potenciometrem 0, abyste změnili
//hodnotu bytu SMB28.
//Q0.0 je aktivní, když je hodnota SMB28 menší
//nebo rovna 50.
//Q0.1 je aktivní, když je hodnota SMB28 větší
//nebo rovna 150.
//Stavový indikátor je zapnutý, pokud platí
//výsledek porovnávání.

```

LD      I0.0
LPS
AB<=   SMB28, 50
=       Q0,0
LPP
AB>=   SMB28, 150
=       Q0,1

```

Network 2 //Nahraje adresy paměti V s nízkými hodnotami,
//které činí porovnání neplatnými a které vypínají
//stavové indikátory.

```

LD      I0.1
MOVW    -30000, VW0
MOVD    -200000000, VD2
MOVR    1.012E-006, VD6

```

Network 3 //Nahraje adresy paměti V s vysokými hodnotami,
//které činí porovnání platnými a které zapínají
//stavové indikátory.

```

LD      I0.2
MOVW    +30000, VW0
MOVD    -100000000, VD2
MOVR    3,141593, VD6

```

Network 4 //Porovnání "Integer Word" zjistí, zda
//VW0 > +10000 platí.
//Užívá konstanty programu pro zobrazení různých
//typů dat. Je také možné porovnat dvě hodnoty
//uložené v programovatelné paměti:
//VW0 > VW100

```

LD      I0.3
LPS
AW>     VW0, +10000
=       Q0,2
LRD
AD<     -150000000, VD2
=       Q0,3
LPP
AR>     VD6, 5.001E-006
=       Q0,4

```

Porovnání řetězců

Instrukce "Porovnání řetězců" porovnává dva řetězce znaků v ASCII kódu:

IN1 = IN2 IN1 <> IN2

Když je výsledek porovnání pravdivý, instrukce porovnání zapne kontakt (LAD) nebo výstup (FBD) nebo provede instrukce Načtení, AND nebo OR s jedničkou a vrcholem zásobníku (STL).

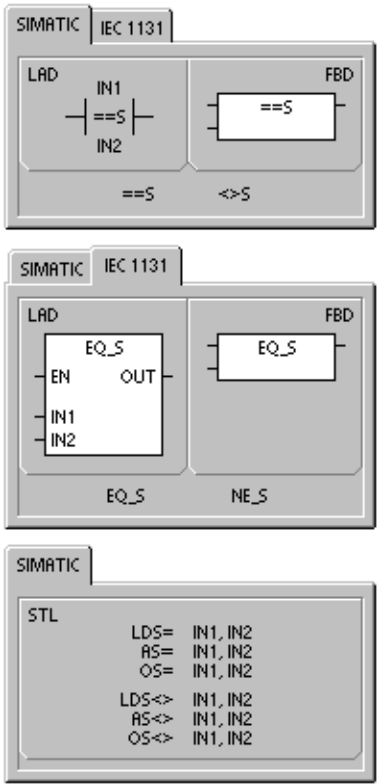
Poznámka

Následující podmínky jsou kritické chyby, které způsobí, že programovatelný automat S7-200 okamžitě zastaví provádění programu:

- Nepřípustná nepřímá adresa (kterákoliv instrukce porovnání)
- Řetězec delší než 254 znaků (Instrukce "Porovnání řetězců")
- Řetězec s takovou počáteční adresou a délkou, které se nevejdou do dané paměťové oblasti (Instrukce "Porovnání řetězců")

Aby k tomu nedošlo, zkontrolujte před prováděním instrukcí porovnání, jaké hodnoty používají a zda jste správně inicializovali ukazatele a místa paměti, které jsou určeny pro řetězce ASCII. Zkontrolujte, zda se zásobník, rezervovaný pro řetězec ASCII, celý vejde do dané paměťové oblasti.

Instrukce porovnání se provádějí bez ohledu na stav signálového toku.



Tabulka 6-15 Operandy platné pro instrukce porovnávání řetězce

Vstupy/Výstupy	Typ	Operandy
IN1, IN2	BYTE (Řetězec)	VB, LB, *VD, *LD, *AC
Výstup (OUT)	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok

Instrukce pro konverzi

Standardní konverze

Číselné konverze

Instrukce "Byte na integer" (BTI), "Integer na byte" (ITB), "Integer na double integer" (ITD), "Double integer na integer" (DTI), "Double integer na real" (DTR), "BCD na integer" (BCDI) a "Integer na BCD" (IBCD) konvertují vstupní hodnotu IN na specifikovaný formát a ukládají výstupní hodnotu v místě paměti specifikovaném v OUT. Můžete například konvertovat double integer na reálné číslo. Můžete také provádět konverzi mezi formáty integer a BCD.

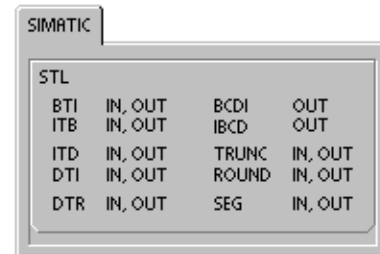
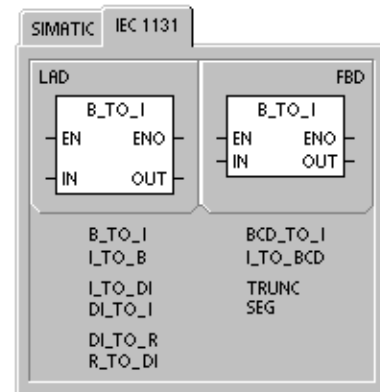
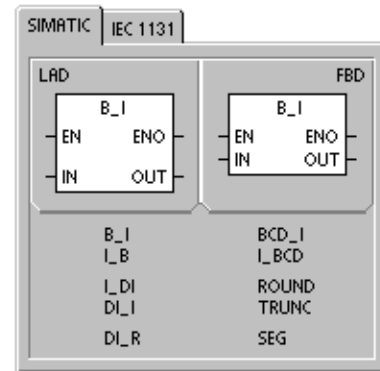
Zaokrouhlení a oříznutí

Instrukce "Zaokrouhlit" (ROUND) konvertuje reálnou hodnotu IN na double integer a zaokrouhlený výsledek uloží do proměnné specifikované v OUT.

Instrukce "Oříznout" (TRUNC) konvertuje reálné číslo IN na double integer a celočíselnou část výsledku zapíše do proměnné dané v OUT.

Segment

Instrukce "Segment" (SEG) generuje bitový vzor, který můžete použít pro řízení sedmisegmentového displeje.



Tabulka 6-16 Operandy platné pro instrukce standardní konverze

Vstupy/Výstupy	Typ dat	Operandy
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, AC, *VD, *LD, *AC, konstanta
	DINT	ID, QD, VD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC, konstanta
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Činnost instrukce pro konverzi BCD na Integer a Integer na BCD

Instrukce "BCD na integer" (BCDI) konvertuje vstupní hodnotu v BCD kódu (IN) na celočíselnou hodnotu a výsledek nahraje do proměnné specifikované OUT. Platný rozsah pro IN je v BCD kódu 0 až 9999.

Chyby, které nastavují ENO = 0

- SM1.6 (BCD chyba)
- 0006 (nepřímá adresa)

Instrukce "Integer na BCD" (IBCD) konvertuje vstupní hodnotu integer (IN) na hodnotu v BCD kódu a výsledek запиše do proměnné dané v OUT. Platný rozsah pro IN je od 0 do 9999 v celých číslech.

Ovlivněné bity SM:

- SM1.6 (BCD chyba)

Činnost instrukce pro konverzi double integer na reálné číslo

Instrukce "Double integer na real" (DTR) konvertuje 32bitové číslo integer se znaménkem (IN) na 32bitové reálné číslo a výsledek запиše do proměnné dané v OUT.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Činnost instrukce pro konverzi double integer na integer

Instrukce "Double integer na integer" (DTI) konvertuje double integer (IN) na integer a výsledek запиše do proměnné dané v OUT.

Chyby, které nastavují ENO = 0

- SM1.1 (přetečení)
- 0006 (nepřímá adresa)

Pokud je konvertovaná hodnota příliš velká pro reprezentaci na výstupu, nastaví se bit přeplnění a výstup se nezmění.

Ovlivněné bity SM:

- SM1.1 (přetečení)

Činnost instrukce pro konverzi integer na double integer

Instrukce "Integer na double integer" (ITD) konvertuje integer (IN) na double integer a výsledek запиše do proměnné dané v OUT. Přidá se znaménkový bit.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Činnost instrukce pro konverzi byte na integer

Instrukce "Byte na integer" (BTI) konvertuje byte IN na integer a výsledek запиše do proměnné dané v OUT. Byte nemá znaménko, proto se nepoužívá znaménkové rozšíření.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Činnost instrukce pro konverzi integer na byte

Instrukce "Integer na Byte" (ITB) konvertuje hodnotu word IN na byte a výsledek запиše do proměnné dané v OUT. Konvertují se hodnoty 0 až 255. Všechny ostatní hodnoty mají za následek přeplnění a výstup není ovlivněn.

Chyby, které nastavují ENO = 0

- SM1.1 (přetečení)
- 0006 (nepřímá adresa)

Ovlivněné bity SM:

- SM1.1 (přetečení)



Tip

Chcete-li změnit integer na reálné číslo, použijte instrukci "Integer na double integer" a potom instrukci "Double integer na real".

Činnost instrukcí "Zaokrouhlit" a "Oříznout"

Instrukce "Zaokrouhlit" (ROUND) konvertuje hodnotu reálného čísla IN na double integer a výsledek uloží do proměnné dané v OUT. Pokud je za desetinnou čárkou 0,5 nebo větší, je číslo zaokrouhleno nahoru.

Instrukce "Oříznout" (TRUNC) konvertuje hodnotu reálného čísla IN na double integer a výsledek uloží do proměnné dané v OUT. Je konvertována pouze celočíselná část reálného čísla, za desetinnou tečkou se část vynechá.

Pokud konvertovaná hodnota není platné reálné číslo nebo je příliš velká pro reprezentaci na výstupu, nastaví se bit přetečení a výstup se nezmění.

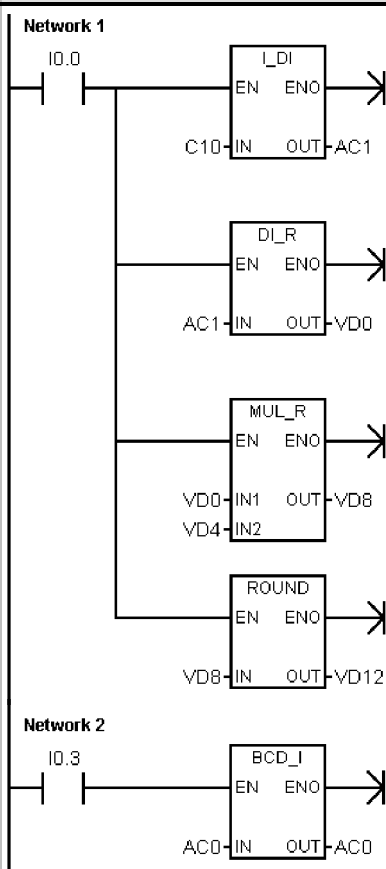
Chyby, které nastavují ENO = 0

- SM1.1 (přetečení)
- 0006 (nepřímá adresa)

Ovlivněné bity SM:

- SM1.1 (přetečení)

Příklad: Instrukce pro standardní konverzi



Network 1 //Konvertuje palce na centimetry:
 //1. Nahraje hodnotu z čítače (palce) do AC1.
 //2. Konvertuje hodnotu na reálné číslo.
 //3. Vynásobí 2,54 (převod na centimetry).
 //4. Konvertuje hodnotu zpět na integer.

```
LD      I0.0
ITD     C10, AC1
DTR     AC1, VD0
MOVR    VD0, VD8
*R       VD4, VD8
ROUND   VD8, VD12
```

Network 2 //Konvertuje hodnotu v BCD na integer

```
LD      I0.3
BCDI    AC0
```

Double integer na reálné číslo a zaokrouhlení

C10	101	Počet = 101 palců
VD0	101.0	Počet (jako reálné číslo)
VD4	2.54	konstanta 2,54 (palce na centimetry)
VD8	256.54	256,54 centimetru jako reálné číslo
VD12	257	257 centimetrů jako double integer

BCD na celé číslo

AC0	1234
BCDI	
AC0	04D2

Činnost instrukce "Segment"

Instrukce "Segment" (SEG) konvertuje segmenty sedmisegmentového displeje tak, že znak specifikovaný v IN generuje bitový vzorec (byte) na místě daném v OUT.

Označené segmenty reprezentují znak na nejnižší platné číslici vstupního bytu. Obrázek 6-13 ukazuje kódování sedmisegmentového displeje s instrukcí "Segment".

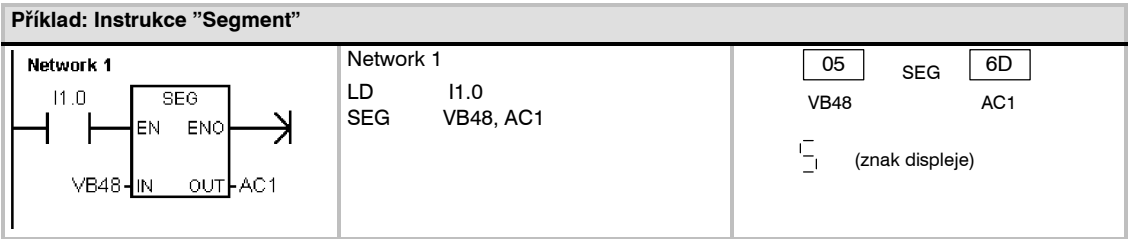
Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

(IN) LSD	Segmentový displej	(OUT) - g f e d c b a
0		0 0 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0
2		0 1 0 1 1 0 1 1
3		0 1 0 0 1 1 1 1
4		0 1 1 0 0 1 1 0
5		0 1 1 0 1 1 0 1
6		0 1 1 1 1 1 0 1
7		0 0 0 0 0 1 1 1

(IN) LSD	Segmentový displej	(OUT) - g f e d c b a
8		0 1 1 1 1 1 1 1
9		0 1 1 0 0 1 1 1
A		0 1 1 1 0 1 1 1
B		0 1 1 1 1 1 0 0
C		0 0 1 1 1 0 0 1
D		0 1 0 1 1 1 1 0
E		0 1 1 1 1 0 0 1
F		0 1 1 1 0 0 0 1

Obr. 6-13 Kódování sedmisegmentového displeje



Konverze na ASCII

Platné znaky ASCII jsou hexadecimální hodnoty 30 až 39 a 41 až 46.

Konverze mezi ASCII a hexadecimálními hodnotami

Instrukce "ASCII na hexadecimální" (ATH) konvertuje počet (LEN) ASCII znaků začínající na IN, na hexadecimální číslice začínající v OUT. Instrukce "Hexadecimální na ASCII" (HTA) konvertuje hexadecimální číslice začínající vstupním bytem IN na znaky ASCII začínající v OUT. Počet konvertovaných hexadecimálních číslic je dán délkou LEN.

Maximální počet znaků ASCII nebo hexadecimálních číslic, které je možno konvertovat, je 255.

Chyby, které nastavují ENO = 0

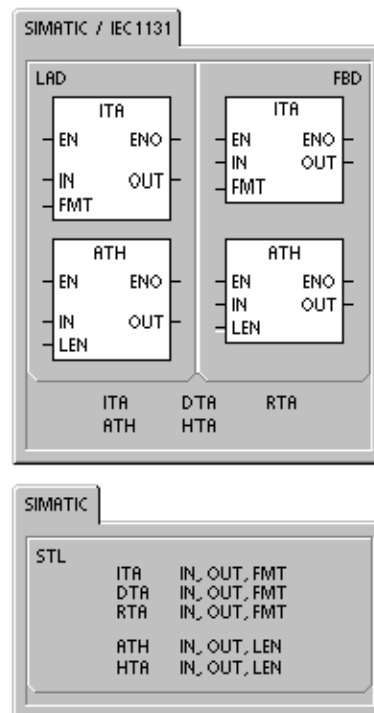
- SM1.7 (neplatné ASCII) pouze u "ASCII na hexadecimální"
- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)

Ovlivněné bity SM:

- SM1.7 (neplatné ASCII)

Konverze číselných hodnot na ASCII

Instrukce "Integer na ASCII" (ITA), "Double integer na ASCII" (DTA) a "Real na ASCII" (RTA) konvertují číselné hodnoty integer, double integer nebo reálného čísla na znaky ASCII.



Tabulka 6-17 Operandy platné pro instrukce konverze ASCII

Vstupy/Výstupy	Typ dat	Operandy
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
LEN, FMT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

Činnost instrukce pro konverzi integer na ASCII

Instrukce "Integer na ASCII" (ITA) konvertuje integer word IN na řetězec znaků ASCII. Formát FMT udává přesnost konverze napravo od desetinné čárky a to, má-li se zobrazovat desetinná čárka, nebo tečka. Výsledná konverze je zapsaná do 8 za sebou jdoucích bytů počínaje OUT.

Řetězec ASCII má vždy 8 znaků.

Obrázek 6-14 ukazuje formát operandu instrukce "Integer na ASCII". Výstupní zásobník má vždy velikost 8 bytů. Počet číslic vpravo od desetinné čárky ve výstupním zásobníku je dán v poli *nnn*. Platný rozsah pole *nnn* je 0 až 5. Zadáním počtu číslic 0 bude hodnota zobrazena bez desetinné čárky. Pro hodnoty *nnn* vyšší než 5 je výstupní zásobník naplněn ASCII mezerami. V bitu *c* je dáno, zda má být pro oddělení celého čísla a desetinné části použita desetinná čárka (*c*=1), nebo desetinná tečka (*c*=0). Horní 4 bity musí být nulové.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- Neplatný formát
- *nnn* > 5

Obrázek 6-14 ukazuje příklady hodnot zformátovaných s desetinnou tečkou (c=0) a třemi číslicemi vpravo od desetinné tečky (nnn=011). Výstupní zásobník je formátován podle následujících pravidel:

- ☐ Kladné hodnoty se do výstupního zásobníku zapisují bez znaménka.
- ☐ Záporné hodnoty se do výstupního zásobníku zapisují se znaménkem minus (–) před nimi.
- ☐ Nuly vlevo od desetinné čárky (s výjimkou nuly těsně vedle desetinné čárky) se nezobrazují.
- ☐ Ve výstupním zásobníku jsou hodnoty zarovnané vpravo.

FMT

MSB	7	6	5	4	3	2	1	0	LSB
	0	0	0	0	c	n	n	n	

c = desetinná čárka (1) nebo desetinná tečka (0)
nnn = počet číslic vpravo od desetinné čárky

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7
in=12				0	.	0	1	2
in=-123			–	0	.	1	2	3
in=1234				1	.	2	3	4
in=-12345		–	1	2	.	3	4	5

Obr. 6-14 Operand FMT pro instrukci "Integer na ASCII" (ITA)

6

Činnost instrukce pro konverzi double integer na ASCII

Instrukce "Double integer na ASCII" (DTA) konvertuje double word (IN) na řetězec ASCII znaků. Operand formátu FMT udává přesnost konverze za desetinnou čárkou. Výsledná konverze je zapsaná do 12 za sebou jdoucích bytů počínaje OUT.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- Neplatný formát
- $nnn > 5$

Výstupní zásobník má vždy velikost 12 bytů.

Obrázek 6-15 ukazuje formát operandu v instrukci "Double integer na ASCII". Počet číslic vpravo od desetinné čárky ve výstupním zásobníku je dán v poli *nnn*. Platný rozsah pole *nnn* je 0 až 5. Zadáním počtu číslic 0 bude hodnota zobrazena bez desetinné čárky. Pro hodnoty *nnn* vyšší než 5 je výstupní zásobník naplněn ASCII mezery. V bitu *c* je dáno, zda má být pro oddělení celého čísla a desetinné části použita desetinná čárka (c=1), nebo desetinná tečka (c=0). Horní 4 bity musí být nulové.

Obrázek 6-15 ukazuje příklady hodnot zformátovaných s desetinnou tečkou (c=0) a čtyřmi číslicemi vpravo od desetinné tečky (nnn=100). Výstupní zásobník je formátován podle následujících pravidel:

- ☐ Kladné hodnoty se do výstupního zásobníku zapisují bez znaménka.
- ☐ Záporné hodnoty se do výstupního zásobníku zapisují se znaménkem minus (–) před nimi.
- ☐ Nuly vlevo od desetinné čárky (s výjimkou nuly těsně vedle desetinné čárky) se nezobrazují.
- ☐ Ve výstupním zásobníku jsou hodnoty zarovnané vpravo.

FMT

MSB	7	6	5	4	3	2	1	0	LSB
	0	0	0	0	c	n	n	n	

c = desetinná čárka (1) nebo desetinná tečka (0)
nnn = počet číslic vpravo od desetinné čárky

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7	Out +8	Out +9	Out +10	Out +11
in=-12						–	0	.	0	0	1	2
in=1234567					1	2	3	.	4	5	6	7

Obr. 6-15 Operand FMT pro instrukci "Double integer na ASCII" (DTA)

Činnost instrukce pro konverzi reálného čísla na ASCII

Instrukce "Real na ASCII" (RTA) konvertuje hodnotu reálného čísla IN na řetězec ASCII. Formát FMT udává přesnost konverze napravo od desetinné čárky, zda se má zobrazovat desetinná čárka nebo tečka a velikost výstupního zásobníku.

Výsledná konverze je zapsaná do výstupního zásobníku začínajícího v OUT.

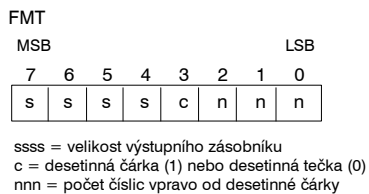
Počet znaků (nebo délka) výsledného ASCII řetězce je shodný s velikostí výstupního zásobníku; může to být 3 až 15 bytů.

Formát reálného čísla použitý v S7-200 podporuje nejvíce 7 platných číslic. Pokus o zobrazení více než 7 platných číslic způsobí chybu zaokrouhlení.

Obrázek 6-16 ukazuje formát operandu (FMT) pro instrukci RTA. Velikost výstupního zásobníku je dána polem ssss. Velikosti 0, 1 nebo 2 byty jsou neplatné. Počet číslic vpravo od desetinné čárky ve výstupním zásobníku je dán v poli nnn. Platný rozsah pole nnn je 0 až 5. Zadáním počtu číslic 0 bude hodnota zobrazena bez desetinné čárky. Výstupní zásobník je naplněn ASCII mezerami pro hodnoty nnn větší než 5 nebo pokud je zadaná velikost výstupního zásobníku příliš malá na vložení konvertované hodnoty. V bitu c je dáno, zda má být pro oddělení celého čísla a zlomkové části použita desetinná čárka (c=1), nebo desetinná tečka (c=0).

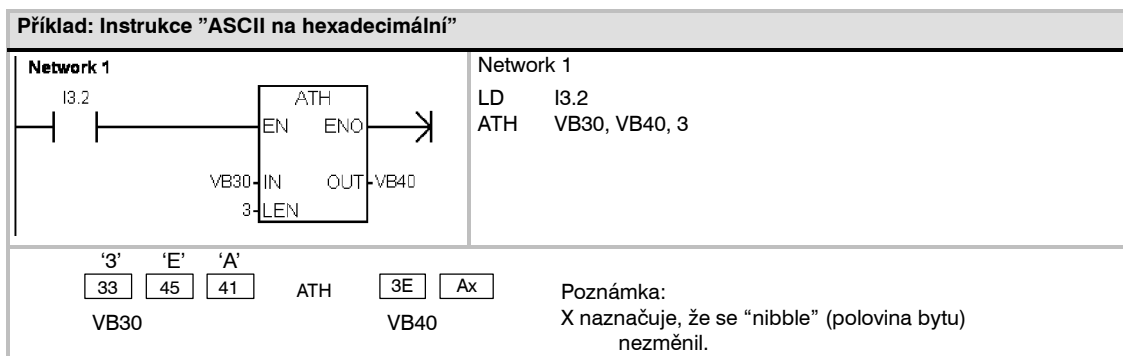
Obrázek 6-16 také ukazuje příklady hodnot zformátovaných s desetinnou tečkou (c=0), jednou číslicí vpravo od desetinné tečky (nnn=001) a velikostí zásobníku šest bytů (ssss=0110). Výstupní zásobník je formátován podle následujících pravidel:

- ☐ Kladné hodnoty se do výstupního zásobníku zapisují bez znaménka.
- ☐ Záporné hodnoty se do výstupního zásobníku zapisují se znaménkem minus (–) před nimi.
- ☐ Nuly vlevo od desetinné čárky (s výjimkou nuly těsně vedle desetinné čárky) se nezobrazují.
- ☐ Hodnoty napravo od desetinné čárky jsou zaokrouhleny tak, aby měly daný počet číslic vpravo od desetinné čárky.
- ☐ Velikost výstupního zásobníku musí být alespoň o tři byty větší, než je počet číslic vpravo od desetinné čárky.
- ☐ Ve výstupním zásobníku jsou hodnoty zarovnané vpravo.



	Out	Out	Out	Out	Out	Out
		+1	+2	+3	+4	+5
in = 1234,5	1	2	3	4	.	5
in = –0,0004				0	.	0
in = –3,67526			–	3	.	7
in = 1,95				2	.	0

Obr. 6-16 Operand FMT pro instrukci "Real na ASCII" (RTA)



Příklad: Instrukce "Integer na ASCII"

Network 1

LD I2.3
ITA VW2, VB10, 16#0B

Network 1 //Konvertuje hodnotu integer ve VW2
//na 8 ASCII znaků začínajících VB10,
//s použitím formátu 16#0B (desetinná
//čárka, po níž následují 3 číslice).

LD I2.3
ITA VW2, VB10, 16#0B

		' '	' '	'1'	'2'	' '	'3'	'4'	'5'
12345	ITA	20	20	31	32	2C	33	34	35
VW2		VB10	VB11	...					

Příklad: Instrukce "Real na ASCII"

Network 1

LD I2.3
RTA VD2, VB10, 16#A3

Network 1 //Konvertuje hodnotu reálného čísla ve VD2
//na 10 ASCII znaků začínajících VB10,
//s použitím formátu 16#A3 (desetinná
//tečka, po níž následují 3 číslice).

LD I2.3
RTA VD2, VB10, 16#A3

		' '	' '	' '	'1'	'2'	'3'	'.'	'4'	'5'	'0'
123.45	RTA	20	20	20	31	32	33	2E	34	35	30
VD2		VB10	VB11	...							

Konverze řetězců

Konverze číselných hodnot na řetězec

Instrukce "Integer na řetězec" (ITS), "Double integer na řetězec" (DTS) a "Real na řetězec" (RTS) konvertují hodnoty integer, double integer nebo reálného čísla na řetězec znaků ASCII (OUT).

Činnost "Integer na řetězec"

Instrukce "Integer na řetězec" (ITS) konvertuje integer o velikosti word (IN) na řetězec ASCII o délce 8 znaků. Formát (FMT) udává přesnost konverze napravo od desetinné čárky a to, má-li se zobrazovat desetinná čárka, nebo tečka. Výsledný řetězec je zapsán do 9 za sebou jdoucích bytů s počátkem v OUT. Více informací najdete v kapitole 4. v odstavci o formátu řetězců.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)
- Neplatný formát (nnn > 5)

Obrázek 6-17 ukazuje formát operandu v instrukci "Integer na řetězec". Výstupní řetězec má vždy velikost 8 znaků. Počet číslic vpravo od desetinné čárky ve výstupním zásobníku je dán v poli nnn. Platný rozsah pole nnn je 0 až 5. Zadáním počtu číslic 0 bude hodnota zobrazena bez desetinné čárky. Pro hodnoty nnn větší než 5 je výstupem řetězec 8 ASCII mezer. V bitu c je dáno, zda má být pro oddělení celého čísla a desetinné části použita desetinná čárka (c=1), nebo desetinná tečka (c=0). Horní 4 bity formátu musí být nulové.

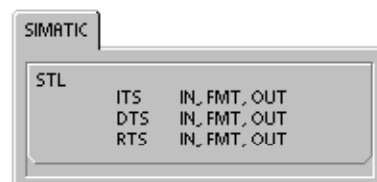
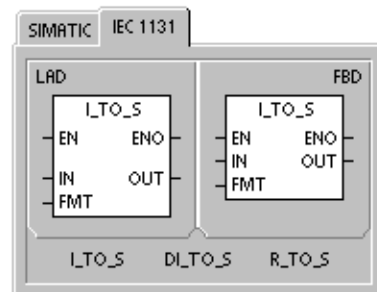
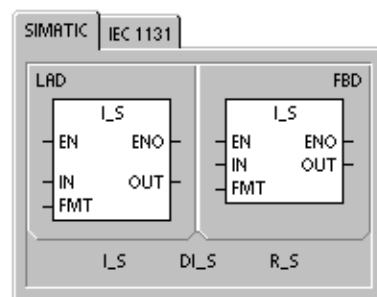
Obrázek 6-17 také ukazuje příklady hodnot zformátovaných s desetinnou tečkou (c=0) a třemi číslicemi vpravo od desetinné tečky (nnn = 011). Hodnota v OUT udává délku řetězce.

Výstupní řetězec je formátován podle následujících pravidel:

- ☐ Kladné hodnoty se do výstupního zásobníku zapisují bez znaménka.
- ☐ Záporné hodnoty se do výstupního zásobníku zapisují se znaménkem minus (–) před nimi.
- ☐ Nuly vlevo od desetinné čárky (s výjimkou nuly těsně vedle desetinné čárky) se nezobrazují.
- ☐ Ve výstupním řetězci - jsou hodnoty zarovnané vpravo.

Tabulka 6-18 Operandy platné pro instrukce, které konvertují číselné hodnoty na řetězce

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE (Řetězec) INT DINT REAL	VB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC, konstanta ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
INDX, FMT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
OUT	BYTE (Řetězec) INT DINT, REAL	VB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC



FMT

MSB				LSB			
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c = desetinná čárka (1), nebo desetinná tečka (0)
 nnn = počet číslic vpravo od desetinné čárky

	Out	Out	Out	Out	Out	Out	Out	Out	Out
	+	+	+	+	+	+	+	+	+
in=12	8	1	2	3	4	0	5	6	7
in = -123	8				0	.	1	2	3
in=1234	8				1	.	2	3	4
in = -12345	8		-	1	2	.	3	4	5

Obr. 6-17 Operand FMT pro instrukci "Integer na řetězec"

Činnost instrukce "Double integer na řetězec"

Instrukce "Double integer na řetězec" (DTS) konvertuje double integer (IN) na řetězec ASCII o délce 12 znaků. Formát (FMT) udává přesnost konverze vpravo od desetinné čárky a to, má-li se zobrazovat desetinná čárka, nebo tečka. Výsledný řetězec je zapsán do 13 za sebou jdoucích bytů s počátkem v OUT. Více informací najdete v kapitole 4 v odstavci, který popisuje formát řetězců.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)
- Neplatný formát (nnn > 5)

Obrázek 6-18 ukazuje formát operandu v instrukci "Double integer na řetězec". Výstupní řetězec má vždy velikost 8 znaků. Počet číslic vpravo od desetinné čárky ve výstupním zásobníku je dán v poli nnn. Platný rozsah pole nnn je 0 až 5. Zadáním počtu číslic 0 bude hodnota zobrazena bez desetinné čárky. Pro hodnoty nnn větší než 5 je výstupem řetězec 12 ASCII mezer. V bitu c je dáno, zda má být pro oddělení celého čísla a desetinné části použita desetinná čárka (c=1), nebo desetinná tečka (c=0). Horní 4 bity formátu musí být nulové.

Obrázek 6-18 ukazuje příklady hodnot zformátovaných s desetinnou tečkou (c = 0) a čtyřmi číslicemi vpravo od desetinné tečky (nnn = 100). Hodnota v OUT udává délku řetězce. Výstupní řetězec je formátován podle následujících pravidel:

- ☐ Kladné hodnoty se do výstupního zásobníku zapisují bez znaménka.
- ☐ Záporné hodnoty se do výstupního zásobníku zapisují se znaménkem minus (–) před nimi.
- ☐ Nuly vlevo od desetinné čárky (s výjimkou nuly těsně vedle desetinné čárky) se nezobrazují.
- ☐ Ve výstupním řetězci jsou hodnoty zarovnané vpravo.

FMT

MSB				LSB			
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c = desetinná čárka (1), nebo desetinná tečka (0)
 nnn = počet číslic vpravo od desetinné čárky

Out	Out	Out	Out	Out	Out	Out	Out	Out	Out	Out	Out	Out
	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12
12					.	-	0	.	0	0	1	2
12					1	2	3	.	4	5	6	7

Obr. 6-18 Operand FMT pro instrukci "Double integer na řetězec"

Činnost instrukce "Real na řetězec"

Instrukce "Real na řetězec" (RTS) konvertuje reálnou hodnotu (IN) na řetězec ASCII. Formát (FMT) udává přesnost konverze napravo od desetinné čárky, délku výstupního řetězce a to, má-li se zobrazovat desetinná čárka, nebo tečka.

Výsledná konverze je zapsaná do řetězce začínajícího v OUT. Délka výsledného řetězce je specifikována ve formátu – může být 3 až 15 znaků. Více informací najdete v kapitole 4 v odstavci, který popisuje formát řetězců.

Formát reálného čísla použitý v S7-200 podporuje maximálně 7 platných číslic. Pokus o zobrazení více než 7 platných číslic způsobí chybu zaokrouhlení.

Obrázek 6-19 ukazuje formát operandu v instrukci "Real na řetězec". Délka výstupního řetězce je dána polem ssss. Velikosti 0, 1 nebo 2 byty jsou neplatné. Počet číslic vpravo od desetinné čárky ve výstupním zásobníku je dán v poli nnn. Platný rozsah pole nnn je 0 až 5. Zadáním počtu číslic 0 bude hodnota zobrazena bez desetinné čárky. Pro hodnoty nnn větší než 5 nebo je-li zadána délka výstupního řetězce kratší na vložení konvertované hodnoty je naplněn výstupní řetězec ASCII mezerami. V bitu c je dáno, zda má být pro oddělení celého čísla a desetinné části použita desetinná čárka (c=1), nebo desetinná tečka (c=0).

Obrázek 6-19 také ukazuje příklady hodnot zformátovaných s desetinnou tečkou (c = 0), jednou číslicí vpravo od desetinné tečky (nnn = 001) a délkou výstupního řetězce 6 znaků (ssss = 0110). Hodnota v OUT udává délku řetězce. Výstupní řetězec je formátován podle následujících pravidel:

- ☐ Kladné hodnoty se do výstupního zásobníku zapisují bez znaménka.
- ☐ Záporné hodnoty se do výstupního zásobníku zapisují se znaménkem minus (–) před nimi.
- ☐ Nuly vlevo od desetinné čárky (s výjimkou nuly těsně vedle desetinné čárky) se nezobrazují.
- ☐ Hodnoty napravo od desetinné čárky jsou zaokrouhleny tak, aby měly daný počet číslic vpravo od desetinné čárky.
- ☐ Velikost výstupního řetězce musí být alespoň o tři byty větší, než je počet číslic vpravo od desetinné čárky.
- ☐ Ve výstupním řetězci jsou hodnoty zarovnané vpravo.

FMT

MSB							LSB
7	6	5	4	3	2	1	0
s	s	s	s	c	n	n	n

ssss = délka výstupního řetězce
c = desetinná čárka (1), nebo desetinná tečka (0)
nnn = počet číslic vpravo od desetinné čárky

	Out	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6
in=1234.5	6	1	2	3	4	.	5
in = -0,0004	6				0	.	0
in = -3,67526	6			-	3	.	7
in = 1.95	6				2	.	0

Obr. 6-19 Operand FMT pro instrukci "Real na řetězec"

Konverze řetězců na číselné hodnoty

Instrukce "Řetězec na integer" (STI), "Řetězec na double integer" (STD) a "Řetězec na real" (STR) konvertují hodnotu řetězce IN začínající s offsetem INDX na hodnotu integer, double integer nebo reálné číslo v OUT.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)
- 009B (index = 0)
- SM1.1 (přetečení)

Instrukce "Řetězec na integer" a "Řetězec na double integer" konvertují řetězec s následujícím tvarem:
[mezery] [+ nebo –] [čísllice 0 – 9]

Instrukce "Řetězec na real" konvertuje řetězec s následujícím tvarem:
[mezery] [+ nebo –] [čísllice 0 – 9] [. nebo ,] [čísllice 0 – 9]

Hodnota INDX je normálně nastavena na 1, což zahájí konverzi od prvního znaku řetězce. Je možné nastavit hodnotu INDX na jiné hodnoty pro zahájení konverze na různých místech uvnitř řetězce. Toho lze využít, jestliže vstupní řetězec obsahuje text, který není součástí konvertovaného čísla. Například, pokud je vstupní řetězec "Teplota: 77.8", nastavíte INDX na hodnotu 13, aby se přeskočilo slovo "Teplota: " na začátku řetězce.

Instrukce "Řetězec na real" nekonvertuje řetězce, které používají zobrazení s pohyblivou řádovou čárkou nebo exponenciální tvary reálných čísel. Instrukce nedává chybu přeplnění (SM1.1), ale konvertuje řetězec na reálné číslo až k exponenciálnímu tvaru a pak konverzi ukončí. Například řetězec '1.234E6' je konvertován bez ohlášení chyby na reálnou hodnotu 1,234.

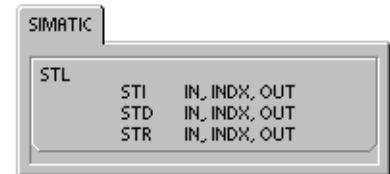
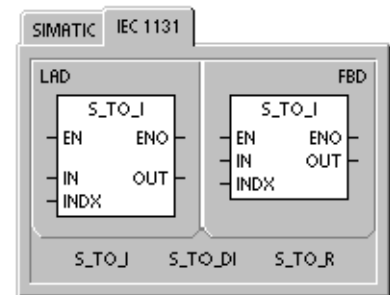
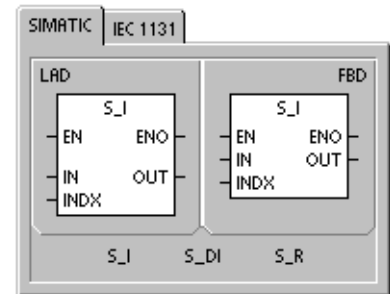
Konverze se ukončí, když je dosaženo konce řetězce nebo když je zjištěn první neplatný znak. Neplatný je kterýkoliv znak, který není číslice (0 – 9).

Chyba přetečení (SM1.1) je ohlášena, kdykoliv je výsledkem konverze hodnota integer, která je příliš velká pro výstupní hodnotu. Například instrukce "Řetězec na integer" nastaví chybu přetečení, pokud je výsledkem vstupního řetězce hodnota větší než 32767 nebo menší než –32768.

Chyba přetečení (SM1.1) je také nastavena, pokud není konverze možná, protože vstupní řetězec neobsahuje platnou hodnotu. Jestliže například vstupní řetězec obsahuje 'A123', nastaví instrukce pro konverzi SM1.1 (přetečení) a výstupní hodnota zůstává nezměněna.

Tabulka 6-19 Operandy platné pro instrukce, které konvertují řetězce na číselné hodnoty

Vstupy/Výstupy	Typ dat	Operandy
IN	BYTE (řetězec)	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC, konstanta
INDX	BYTE	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
OUT	BYTE (řetězec) INT DINT, REAL	VB, IB, QB, MB, SMB, SB, LB, *VD, *LD, *AC, konstanta VW, IW, QW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC



Platné vstupní řetězce pro integer a double integer		Platné vstupní řetězce pro reálná čísla		Neplatné vstupní řetězce
Vstupní řetězec	Výstupní integer	Vstupní řetězec	Výstupní reálné číslo	Vstupní řetězec
'123'	123	'123'	123.0	'A123'
'-00456'	-456	'-00456'	-456.0	' '
'123.45'	123	'123.45'	123.45	'++123'
'+2345'	2345	'+2345'	2345.0	'+-123'
'000000123ABCD'	123	'00.000000123'	0.000000123	'+ 123'

Obr. 6-20 Příklady platných a neplatných vstupních řetězců

Příklad: Konverze řetězce: Subřetězec na integer, double integer a reálné číslo

Network 1

LD I0.0
STI VB0,7,VW100
STD VB0,7,VD200
STR VB0,7,VD300

VB0

11	'T'	'e'	'm'	'p'	' '	'g'	'8'	'.'	'6'	'F'

Po provedení networku:
VW100 (integer) = 98
VD200 (double integer) = 98
VD300 (reálné číslo) = 98,6

Zakódování a dekódování

Kódování

Instrukce "Zakódovat" (ENCO) zapíše číslo bitu v nejnižším nastaveném bitu vstupního slova IN do nejnižšího "půlbytu" (4 bity) výstupního bytu OUT.

Dekódování

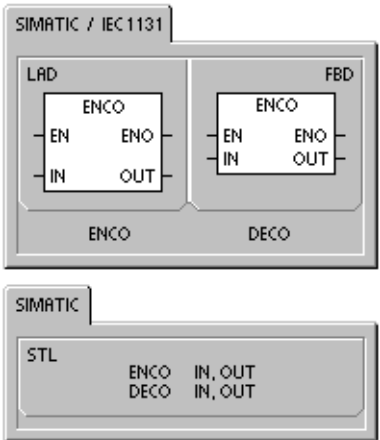
Instrukce "Dekódovat" (DECO) nastaví bit výstupního slova OUT, který odpovídá číslu bitu danému v nejnižším "půlbytu" (4 bity) vstupního bytu IN. Všechny ostatní bity výstupního slova jsou nastaveny na 0.

Bitsy SM a ENO

Jak v instrukci "Zakódovat", tak v instrukci "Dekódovat" ovlivňují ENO následující podmínky.

Chyby, které nastavují ENO = 0

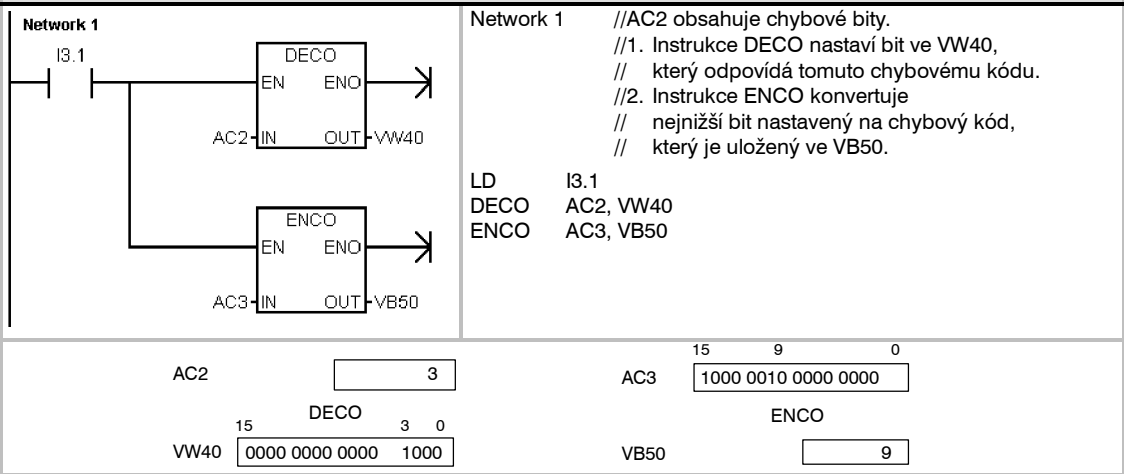
- 0006 (nepřímá adresa)



Tabulka 6-20 Operandy platné pro instrukce "Zakódovat" a "Dekódovat"

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
OUT	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC

Příklad: Instrukce "Zakódovat" a "Dekódovat"



Instrukce čítače

Čítač SIMATIC

Čítač nahoru

Instrukce "Čítač nahoru" (CTU) zvýší aktuální hodnotu o jedničku pokaždé, když vstup pro čítání (CU) přejde z vypnutého do zapnutého stavu. Když je aktuální hodnota Cxx větší nebo rovna nastavené hodnotě PV, nastaví se bit čítače Cxx. Čítač se vynuluje při zapnutí vstupu R (Reset) nebo při provedení instrukce "Reset". Čítač přestane čítat při dosažení maximální hodnoty (32.767).

V STL :

- Vstup "R": Vrchol zásobníku
- Vstup "CU": Hodnota uložená na druhém místě zásobníku

Čítač dolů

Instrukce "Čítač dolů" (CTD) sníží aktuální hodnotu čítače o jedničku pokaždé, když vstup pro čítání (CD) přejde z vypnutého do zapnutého stavu. Když je aktuální hodnota Cxx rovna 0, zapne se bit čítače Cxx. Čítač vynuluje bit Cxx a načte do aktuální hodnoty nastavenou hodnotu PV při zapnutí vstupu LD (Load). Čítač se zastaví při dosažení nuly a bit čítače Cxx se zapne.

V STL :

- Vstup "LD": Vrchol zásobníku
- Vstup "CD": Hodnota uložená na druhém místě zásobníku.

Čítač nahoru/dolů

Instrukce "Čítač nahoru/dolů" (CTUD) zvýší aktuální hodnotu o jedničku, když vstup pro čítání (CU) přejde ze stavu log. 0 do log. 1, a sníží aktuální hodnotu o jedničku, když vstup pro čítání (CD) přejde ze stavu log. 0 do log. 1. Aktuální hodnota čítače Cxx uchovává aktuální počet. Nastavená hodnota PV je srovnávána s aktuální hodnotou pokaždé, když je provedena instrukce čítače.

Jestliže aktuální hodnota čítače dosáhne maxima (32.767), další náběžná hrana na vstupu pro čítání způsobí nastavení aktuální hodnoty čítače na minimum (−32.768). Jestliže aktuální hodnota čítače dosáhne minima (−32.767), další náběžná hrana na vstupu pro čítání způsobí nastavení aktuální hodnoty čítače na maximum (32.768).

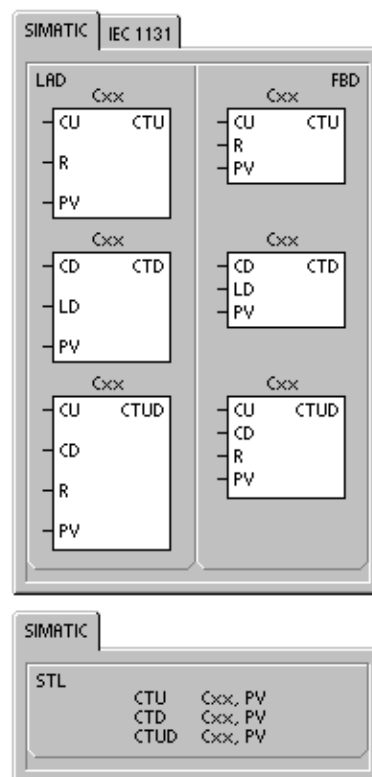
Když je aktuální hodnota Cxx větší nebo rovna nastavené hodnotě PV, nastaví se bit čítače Cxx. Jinak se bit čítače nastaví na log. 0. Čítač se vynuluje při zapnutí vstupu Reset (R) nebo při provedení instrukce "Reset". Při CTUD se čítač zastaví, když dosáhne PV.

V STL :

- Vstup "R": Vrchol zásobníku
- Vstup "CD": Hodnota uložená na druhém místě zásobníku
- Vstup "CU": Hodnota uložená na třetím místě zásobníku

Tabulka 6-21 Operandy platné pro SIMATIC instrukce čítače

Vstup/Výstupy	Typy dat	Operandy
Cxx	WORD	Konstanta (C0 až C255)
CU, CD, LD, R	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
PV	INT	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, AIW, *VD, *LD, *AC, konstanta





Tip
Protože má každý čítač jen jednu aktuální hodnotu, nepřirazujte stejné číslo více než jednomu čítači. (Čítače nahoru, dolů, nahoru/dolů se stejným číslem přistupují na stejnou aktuální hodnotu.)
Při vynulování čítače pomocí instrukce "Reset" se vynuluje bit čítače a aktuální hodnota čítače se nastaví na nulu. Číslo čítače použijte pro přístup k aktuální hodnotě i k bitu daného čítače.

Tabulka 6-22 Operace instrukcí čítače

Typ	Operace	Bit čítače	Výpadek a obnovení napájení/První programový cyklus
CTU	CU zvyšuje aktuální hodnotu. Aktuální hodnota se zvyšuje, dokud nedosáhne 32.767.	Bit čítače se nastaví na log. 1, když: Aktuální hodnota >= Nastavená	Bit čítače je nastaven na log. 0. Aktuální hodnotu je možné uchovat. ¹
CTUD	CU zvyšuje aktuální hodnotu. CD snižuje aktuální hodnotu. Aktuální hodnota se stále zvyšuje nebo snižuje, dokud není čítač vynulován.	Bit čítače se nastaví na log. 1, když: Aktuální hodnota >= Nastavená	Bit čítače je nastaven na log. 0. Aktuální hodnotu je možné uchovat. ¹
CTD	CD snižuje aktuální hodnotu, dokud aktuální hodnota nedosáhne 0.	Bit čítače se nastaví na log. 1, když: Aktuální hodnota = 0	Bit čítače je nastaven na log. 0. Aktuální hodnotu je možné uchovat. ¹

¹ Můžete nastavit, aby aktuální hodnota čítače byla remanentní. Více informací o remanentní paměti CPU S7-200 najdete v kapitole 4.

Příklad: Instrukce "Čítač dolů"

Network 1

Network 2

Časový diagram

Network 1 //Aktuální hodnota čítače dolů C1 čítá od 3 do 0
//a I0.1 je nastaven na log. 0,
//I0.0 "Off-on" snižuje aktuální hodnotu C1
//I0.1 "On" obnoví nastavenou hodnotu odpočtu 3

LD I0.0
LD I0.1
CTD C1, +3

Network 2 //Bit C1 je nastaven na log. 1, když se aktuální hodnota
//čítače C1 = 0

LD C1
= Q0.0

Příklad: Instrukce "Čítač nahoru/dolů"

Network 1

Network 2

Network 1 //I0.0 čítá nahoru
 //I0.1 čítá dolů
 //I0.2 vynuluje aktuální hodnotu

LD I0.0
LD I0.1
LD I0.2
CTUD C48, +4

Network 2 //Čítač nahoru/dolů C48 zapne bit C48,
 //když aktuální hodnota >= 4

LD C48
= Q0,0

Časový diagram

Čítač IEC

Čítač nahoru

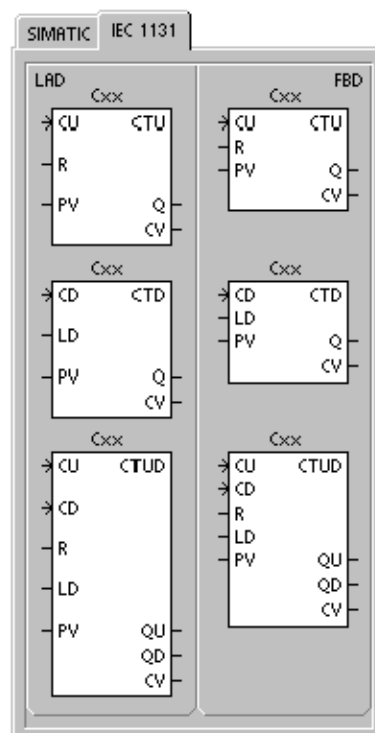
Pokud je na vstupu "čítač nahoru" (CU) náběžná hrana, instrukce "Čítač nahoru" (CTU) zvyšuje aktuální hodnotu o jedničku (až do nastavené hodnoty PV). Je-li aktuální hodnota (CV) větší nebo rovna nastavené hodnotě, nastaví se výstupní bit čítače (Q). Čítač se vynuluje, když je na resetovací vstup (R) přivedena log. 1. "Čítač nahoru" přestane čítat při dosažení nastavené hodnoty.

Čítač dolů

Instrukce "Čítač dolů" (CTD) snižují nastavenou hodnotu (PV) o jedničku při výskytu náběžné hrany na vstupu "čítač dolů" (CD). Když je aktuální hodnota (CV) rovna nule, nastaví se výstupní bit čítače (Q) na hodnotu log. 1. Když se povolí vstup (LD), čítač se vynuluje a obnoví aktuální hodnotu na nastavenou. Čítač dolů se zastaví, když dosáhne nuly.

Čítač nahoru/dolů

Instrukce "Čítač nahoru/dolů" (CTUD) čítá nahoru nebo dolů od aktuální hodnoty (CV) při výskytu náběžných hran na vstupu "čítač nahoru" (CU) nebo "čítač dolů" (CD). Rovná-li se aktuální hodnota nastavené, výstup nahoru (QU) se nastaví na hodnotu log. 1. Rovná-li se aktuální hodnota nule, výstup dolů (QD) se nastaví na hodnotu log. 1. Čítač obnoví aktuální hodnotu na nastavenou (PV), pokud se na vstup (LD) přivede log. 1. Obdobně se čítač vynuluje a aktuální hodnota nastaví na 0, když se na vstup "Reset" (R) přivede log. 1. Čítač přestane čítat, když dosáhne nastavené hodnoty nebo 0.



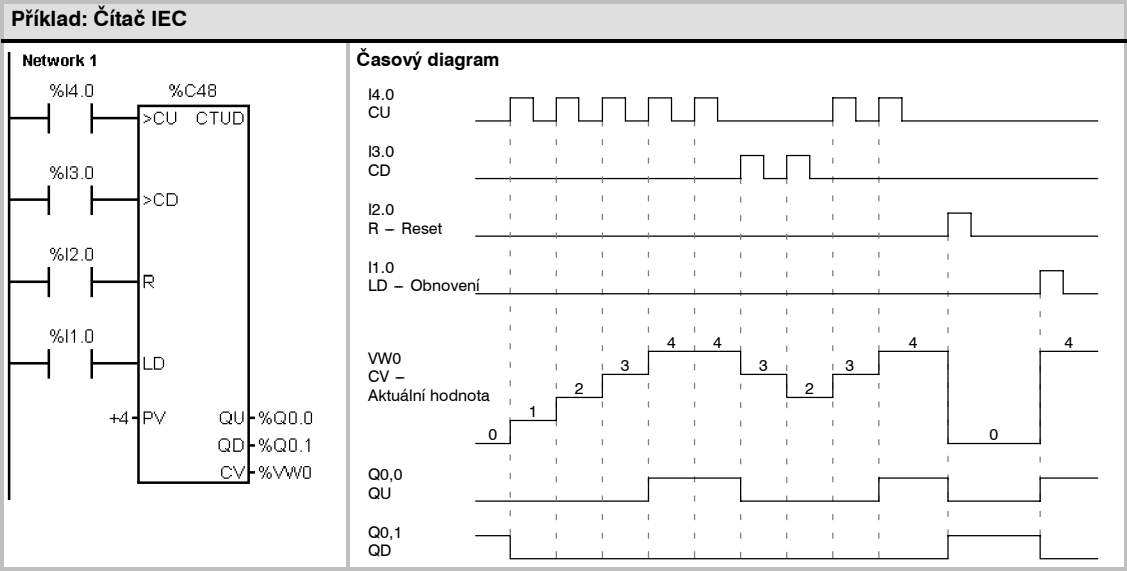
Tabulka 6-23 Operandy platné pro IEC instrukce čítače

Vstupy/Výstupy	Typy dat	Operandy
Cxx	CTU, CTD, CTUD	Konstanta (C0 až C255)
CU, CD, LD, R	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
PV	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, konstanta
Q, QU, QD	BOOL	I, Q, V, M, SM, S, L
CV	INT	IW, QW, VW, MW, SW, LW, AC, *VD, *LD, *AC



Tip

Protože má každý čítač jen jednu aktuální hodnotu, nepřizpůsobte stejné číslo více než jednomu čítači. (Čítače nahoru, čítače dolů a čítače nahoru/dolů se stejným číslem přistupují na stejnou aktuální hodnotu.)



Vysokorychlostní čítače

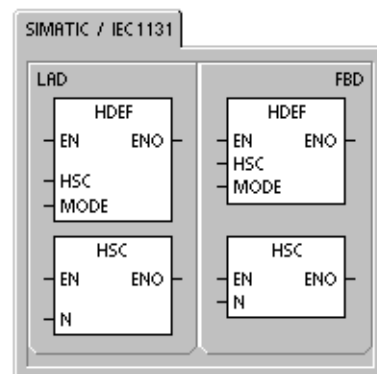
Definice vysokorychlostního čítače

Instrukce "Definice vysokorychlostního čítače" (HDEF) přiřadí provozní režim příslušnému vysokorychlostnímu čítači (HSCx). Režim definuje pro vysokorychlostní čítač hodiny řízení směru, start a nulování.

Pro každý vysokorychlostní čítač použijte jednu instrukci "Definice vysokorychlostního čítače".

Chyby, které nastavují ENO = 0

- 0003 (konflikt vstupů)
- 0004 (nepovolená instrukce v přerušení)
- 000A (předdefinování HSC)



Vysokorychlostní čítač

Instrukce "Vysokorychlostní čítač" (HSC) konfiguruje a řídí vysokorychlostní čítač na základě stavu bitů speciální paměti HSC. Parametr N udává číslo čítače.



Vysokorychlostní čítače mohou být konfigurovány až pro dvanáct různých pracovních režimů. Viz tabulku 6-25.

Každý čítač má vstupy vyhrazené pro hodinové pulzy, řízení směru, nulování a start, pokud tyto funkce podporuje. Ve dvofázových čítačích mohou obojí hodiny běžet na nejvyšší frekvenci. V kvadrantních režimech je možné zvolit jednonásobnou (1x) nebo čtyřnásobnou (4x) maximální rychlost čítače. Všechny čítače mohou běžet s maximální rychlostí bez vzájemného rušení.

Chyby, které nastavují ENO = 0

- 0001 (HSC před HDEF)
- 0005 (současné HSC/PLS)

Tabulka 6-24 Operandy platné pro instrukce vysokorychlostních čítačů

Vstupy/Výstupy	Typy dat	Operandy
HSC, MODE	BYTE	Konstanta
N	WORD	Konstanta



Tipy a triky

Na CD s dokumentací naleznete tipy a triky pro programy, které používají vysokorychlostní čítače. Viz tip 4 a tip 29.

Vysokorychlostní čítače čítají události, které se jinak nedají při dané rychlosti programového cyklu S7-200 zachytit. Maximální frekvence čítání vysokorychlostního čítače závisí na typu CPU S7-200. Další informace jsou v příloze A.



Tip

CPU 221 a CPU 222 podporují čtyři vysokorychlostní čítače: HSC0, HSC3, HSC4 a HSC5. Tyto CPU nepodporují HSC1 a HSC2.

CPU 224, CPU 226 a CPU 226XM podporují šest vysokorychlostních čítačů: HSC0 až HSC5.

Běžně se vysokorychlostní čítač používá jako zdroj pulzů pro krokový motor, kde je hřídel rotující konstantní rychlostí zapojena na inkrementální snímač. Kodér hřídele dává určité množství pulzů na otáčku a nulovací pulz, který vzniká jednou za otáčku. Hodiny a nulovací pulz z kodéru hřídele jsou vstupy pro vysokorychlostní čítač.

Dále se do vysokorychlostního čítače nahraje první z více požadovaných hodnot a aktivují se požadované výstupy na časový interval, dokud je aktuální počet menší než počet požadovaný. Čítač je nastaven tak, že vyvolá přerušení, když se aktuální počet rovná požadovanému a když dojde k vynulování.

Při každém výskytu události, která vyvolá přerušení, kdy "aktuální hodnota je rovna požadované", se nastaví nová požadovaná hodnota a pro výstup se nastaví další stav. Když se objeví událost přerušení od nulování, nastaví se první požadovaná hodnota a počáteční stavy výstupu a cyklus se opakuje.

Protože se přerušení objevují méně často, než je frekvence vysokorychlostních čítačů, je možné implementovat přesnou kontrolu vysokorychlostních operací s relativně malým vlivem na celkový programový cyklus PLC. Metoda přiřazení přerušení umožňuje, aby se každé nové nahrání požadované hodnoty uskutečnilo v samostatném podprogramu přerušení, čímž se ulehčuje stavová kontrola. (Alternativně je možné zpracovávat všechny události přerušení jedním podprogramem přerušení.)

Vysvětlení různých vysokorychlostních čítačů

Všechny čítače pracují stejným způsobem, pokud jsou ve stejném režimu. Existují čtyři základní typy čítačů: jednofázový čítač s vnitřním řízením směru, jednofázový čítač s vnějším řízením směru, dvufázový čítač se 2 hodinovými vstupy a A/B fázový kvadrurní čítač. Všimněte si, že všechny režimy nejsou podporovány všemi čítači. Každý typ můžete používat: bez nulovacího a startovacího vstupu, s nulovacím vstupem a bez startovacího nebo se startovacím i nulovacím vstupem.

- ☐ Když aktivujete nulovací vstup, vymaže se aktuální hodnota čítače, a zůstane tak až do deaktivace nulovacího vstupu.
- ☐ Když aktivujete startovací vstup, začne čítač počítat. Když je startovací vstup deaktivován, je aktuální hodnota čítače konstantní a hodinové pulzy jsou ignorovány.
- ☐ Při aktivaci nulovacího vstupu v době, kdy je startovací vstup neaktivní, je nulování ignorováno a aktuální hodnota se nemění. Při aktivaci startovacího vstupu a aktivním nulovacím vstupem je aktuální hodnota vymazána.

Před použitím vysokorychlostního čítače použijte pro nastavení jeho režimu instrukci HDEF (definice vysokorychlostního čítače). Použijte paměťový bit prvního programového cyklu SM0.1 (tento bit je zapnutý jen při prvním programovém cyklu, potom se vypne), který vyvolá podprogram s instrukcí HDEF.

Programování vysokorychlostního čítače



Průvodce
instrukcemi

Pro konfigurování čítače můžete použít průvodce pro instrukci HSC. Průvodce používá následující informace: typ a režim čítače, nastavená hodnota čítače, aktuální hodnota čítače a výchozí směr počítání. Chcete-li spustit průvodce pro HSC vyberte příkaz menu **Nástroje > Průvodce instrukcemi** a pak vyberte z okna průvodce HSC.

Abyste naprogramovali vysokorychlostní čítač, musíte provést následující základní kroky:

- ☐ Definování čítače a režimu.
- ☐ Nastavení řídicího bytu.
- ☐ Nastavení aktuální hodnoty (počáteční hodnota).
- ☐ Nastavení požadované hodnoty (cílová hodnota).
- ☐ Přiřazení a povolení přerušovacího programu.
- ☐ Aktivace vysokorychlostního čítače.

Definice režimů a vstupů čítače

Pro definování režimů a vstupů čítače použijte instrukci "Definice vysokorychlostního čítače".

Tabulka 6-25 ukazuje vstupy použité pro funkce hodin, řízení směru, nulování a spuštění pro vysokorychlostní čítače. Stejný vstup nemůže být použit pro dvě různé funkce, ale libovolný vstup, který není využit současným režimem vysokorychlostního čítače, může být použit pro jiný účel. Jestliže například HSC0 pracuje v režimu 1, který využívá I0.0 a I0.2, může být I0.1 použit pro přerušení od hrany signálu nebo pro HSC3.



Tip

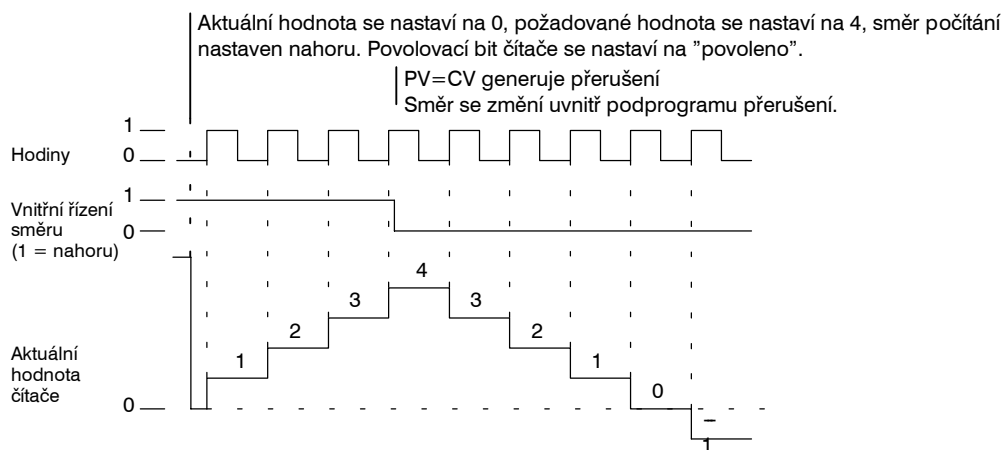
Všimněte si, že všechny režimy HSC0 vždy používají I0.0 a všechny režimy HSC4 vždy používají I0.3, takže při použití uvedených čítačů nejsou tyto vstupy nikdy k dispozici pro jiné použití.

Tabulka 6-25 Vstupy pro vysokorychlostní čítače

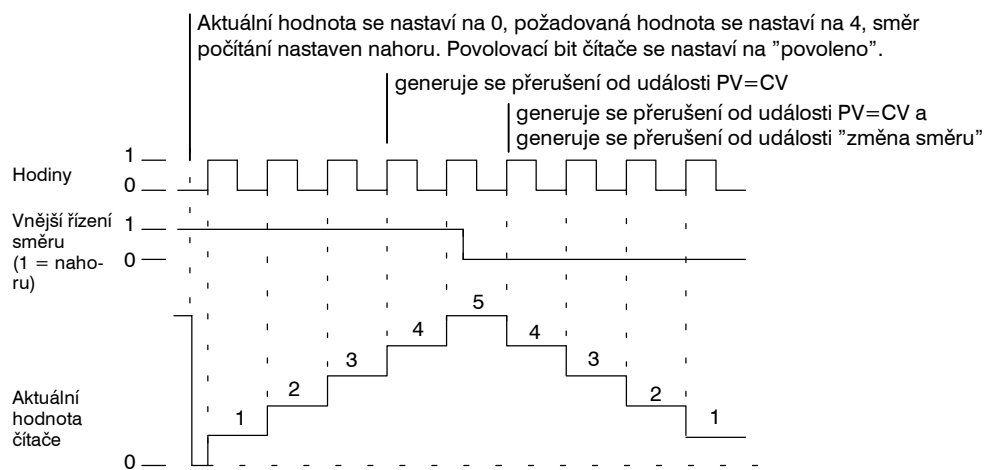
Režim	Popis	Vstupy			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I0.2	I1.1
	HSC2	I1.2	I1.3	I1.1	I1.2
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	Jednofázový čítač s vnitřním řízením směru	Hodiny			
1		Hodiny		Nulování	
2		Hodiny		Nulování	Start
3	Jednofázový čítač s vnějším řízením směru	Hodiny	Směr		
4		Hodiny	Směr	Nulování	
5		Hodiny	Směr	Nulování	Start
6	Dvojfázový čítač se 2 hodinovými vstupy	"Hodiny nahoru"	"Hodiny dolů"		
7		"Hodiny nahoru"	"Hodiny dolů"	Nulování	
8		"Hodiny nahoru"	"Hodiny dolů"	Nulování	Start
9	A/B fázový kvadrurní čítač	Hodiny A	Hodiny B		
10		Hodiny A	Hodiny B	Nulování	
11		Hodiny A	Hodiny B	Nulování	Start

Příklady režimů HSC

Časové diagramy na obrázcích 6-21 až 6-25 ukazují funkci jednotlivých čítačů podle nastaveného režimu.

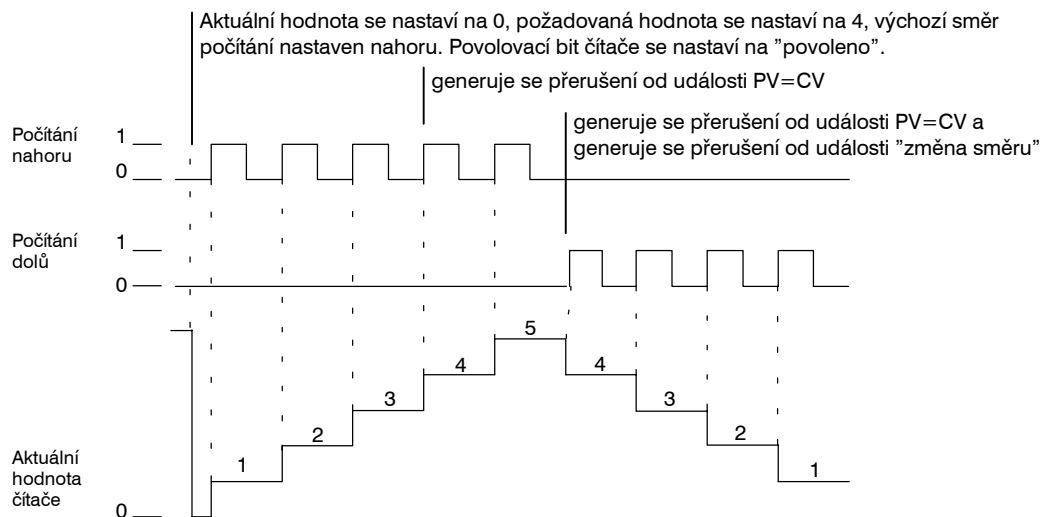


Obr. 6-21 Příklad funkce režimů 0, 1 nebo 2

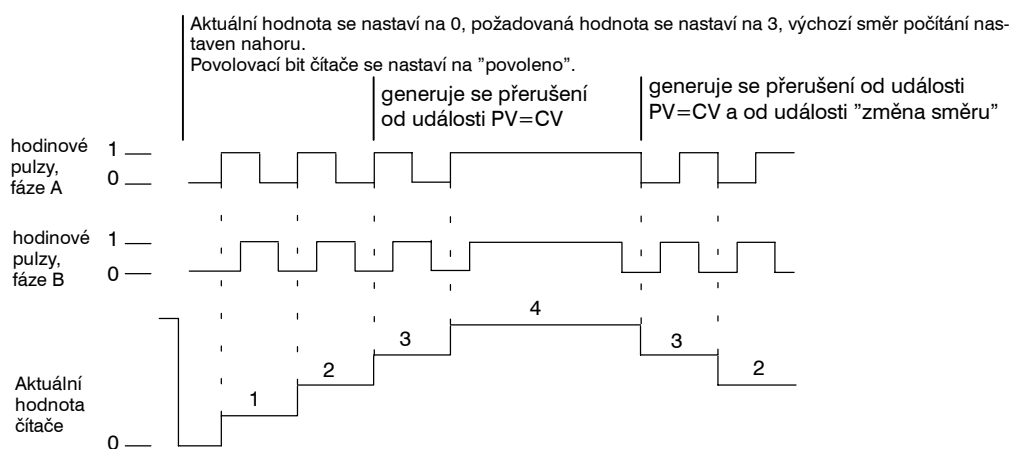


Obr. 6-22 Příklad funkce režimu 3, 4 nebo 5

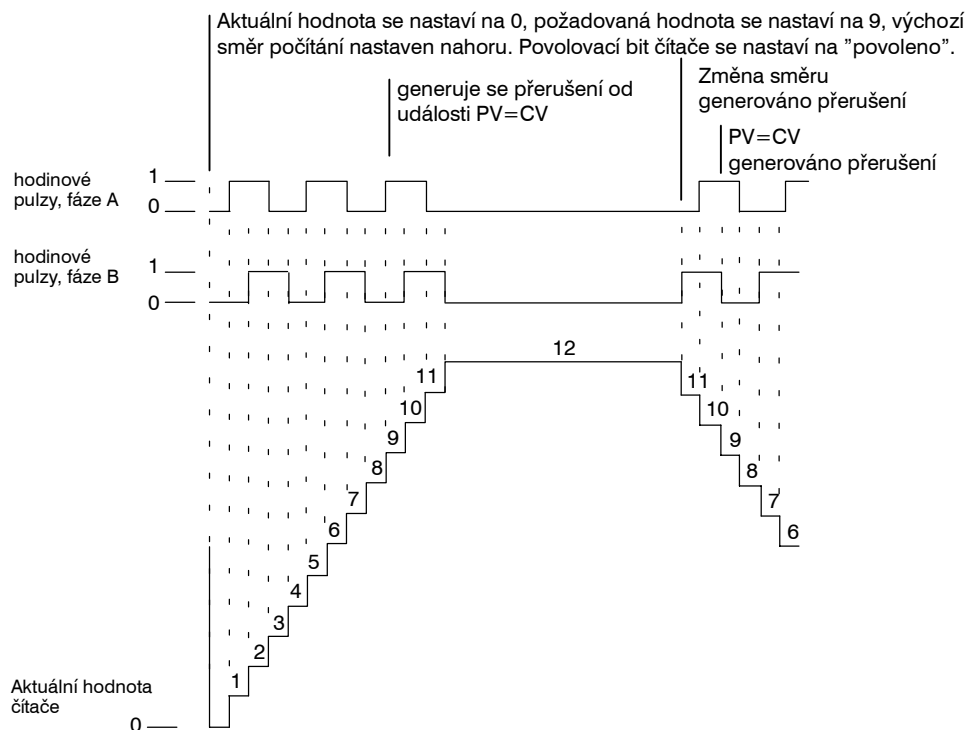
Pokud se v režimech 6, 7 nebo 8 vzestupné hrany na vstupech "hodiny nahoru" a "hodiny dolů" objeví v rozmezí 0,3 mikrosekundy, může vysokorychlostní čítač považovat tyto dvě události za současné. V takovém případě se aktuální hodnota nezmění a neindikuje se ani změna směru počítání. Pokud je časový odstup vzestupných hran vstupů hodin nahoru a dolů větší než tento časový úsek, zachytí vysokorychlostní čítač každou událost zvlášť. V obou případech se negeneruje chyba a čítač si udržuje správnou aktuální hodnotu.



Obr. 6-23 Příklad funkce režimu 6, 7 nebo 8



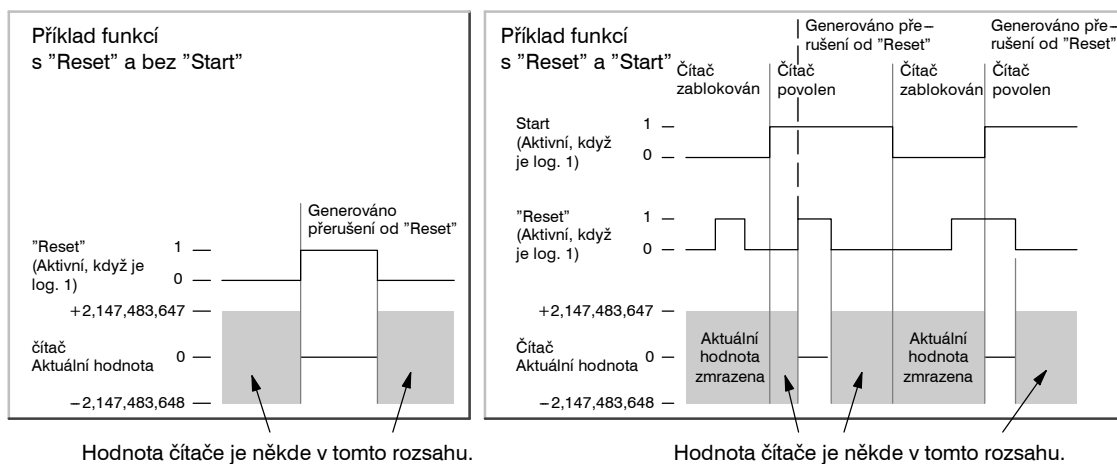
Obr. 6-24 Příklad funkce režimu 9, 10 nebo 11 (kvadrurní 1x režim)



Obr. 6-25 Příklad funkce režimu 9, 10 nebo 11 (kvadrurní 4x režim)

Vstupy "Reset" a "Start"

Funkce vstupů "Reset" a "Start" znázorněná na obrázku 6-26 platí pro všechny režimy, které používají tyto vstupy. V diagramech vstupů "Reset" a "Start" jsou nulování i start naprogramovány jako aktivní v úrovni log. 1.



Obr. 6-26 Příklady funkcí s "Reset" a se "Start" a bez něj

Čtyři čítače mají tři řídicí bity, které se používají pro konfiguraci aktivního stavu nulovacích a startovacích vstupů a pro zvolení 1x nebo 4x režimů (pouze kvadraturní čítače). Tyto bity jsou umístěny v řídicím bytu příslušného čítače a jsou použity pouze při provádění instrukce HDEF. Uvedené bity jsou definovány v tabulce 6-26.



Tip
Před provedením instrukce HDEF musíte nastavit tyto tři řídicí bity na požadované hodnoty. Jinak se čítač nastaví na implicitní konfiguraci daného režimu.
Po provedení instrukce HDEF se nastavení čítače nedá měnit, pokud nejprve nepřepnete automat S7-200 do režimu STOP.

Tabulka 6-26 Aktivní úroveň pro vstupy "Reset", "Start" a 1x/4x řídicí bity

HSC0	HSC1	HSC2	HSC4	Popis (použito pouze po provedení HDEF)
SM37.0	SM47.0	SM57.0	SM147.0	Aktivní úroveň řídicího bitu pro "Reset" ¹ : 0 = "Reset" je aktivní v úrovni log. 1 1 = "Reset" je aktivní v úrovni log. 0
---	SM47.1	SM57.1	---	Aktivní úroveň řídicího bitu pro "Start" ¹ : 0 = "Start" je aktivní v úrovni log. 1 1 = "Start" je aktivní v úrovni log. 0
SM37.2	SM47.2	SM57.2	SM147.2	Výběr rychlosti počítání pro kvadraturní čítače: 0 = 4X rychlost počítání 1 = 1X rychlost počítání

1 Implicitní nastavení vstupů "Reset" a "Start" je aktivní stav v úrovni log. 1 a kvadraturní rychlost počítání 4x (neboli čtyřikrát vstupní frekvence hodin).

Příklad: Instrukce "Definice vysokorychlostního čítače"

M
A
I
N

Network 1

Network 1 //V prvním programovém cyklu:
 //1. Naprogramuje vstupy "Start" a "Reset"
 // s aktivním stavem v úrovni log. 1 a 4x
 //2. Konfiguruje HSC1 na kvadraturní režim
 // se vstupy "Reset" a "Start"

LD SM0.1
MOVB 16#F8, SMB47
HDEF 1, 11

Nastavení řídicího bytu

Po definici čítače a jeho režimu můžete naprogramovat dynamické parametry čítače. Každý vysokorychlostní čítač má řídicí byte, který umožňuje následující činnosti:

- ☐ Povolení nebo blokování čítače
- ☐ Řízení směru (pouze režimy 0, 1 a 2) nebo počáteční směr čítání pro všechny ostatní režimy
- ☐ Nahrání aktuální hodnoty
- ☐ Nahrání nastavené hodnoty

Provedením instrukce HSC je možné prohlédnout řídicí byte a příslušné aktuální a nastavené hodnoty. Tabulka 6-27 popisuje každý z těchto řídicích bitů.

Tabulka 6-27 Řídicí bity pro HSC0, HSC1, HSC2, HSC3, HSC4 a HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Popis
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	Řídicí bit pro směr čítání: 0 = čítání nahoru 1 = čítání dolů
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	Zapsání směru počítání do HSC: 0 = Beze změny 1 = Aktualizuje směr
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	Zapsání nové požadované hodnoty do HSC: 0 = Beze změny 1 = Aktualizuje požadované hodnoty
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	Zapsání nové aktuální hodnoty do HSC: 0 = Beze změny 1 = Aktualizuje aktuální hodnotu
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	Povolení HSC: 0 = blokování HSC 1 = Povolení HSC

Nastavení aktuálních hodnot a požadovaných hodnot

Každý vysokorychlostní čítač má 32bitovou aktuální hodnotu a 32bitovou požadovanou hodnotu. Obě jsou integer se znaménkem. Pro nahrání nové aktuální nebo požadované hodnoty do vysokorychlostního čítače je třeba nastavit řídicí byte a byty speciální paměti, které obsahují dané hodnoty, a také provést instrukci HSC, aby se nové hodnoty přesunuly do vysokorychlostního čítače. V tabulce 6-28 jsou byty speciální paměti používané pro uchování aktuální a požadované hodnoty.

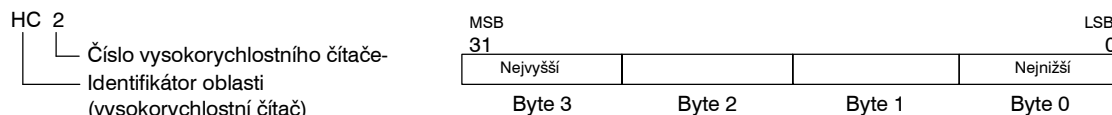
Kromě řídicích bytů a bytů obsahujících nové požadované a aktuální hodnoty je možné také přečíst aktuální hodnotu každého vysokorychlostního čítače pomocí datového typu HC, po kterém následuje číslo čítače (0, 1, 2, 3, 4 nebo 5). Aktuální hodnota je tedy přímo přístupná pro operace čtení, ale zapisovat se dá pouze instrukcí HSC.

Tabulka 6-28 Aktuální a požadované hodnoty HSC0, HSC1, HSC2, HSC3, HSC4 a HSC5

Nahrávaná hodnota	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Nová aktuální	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Nová požadovaná	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

Adresování vysokorychlostních čítačů (HC)

Pro přístup k hodnotě vysokorychlostního čítače je třeba zadat adresu čítače zadáním typu paměti (HC) a čísla čítače (jako HC0). Aktuální hodnotu čítače je možné pouze číst a je možné k ní přistupovat pouze jako double word (32 bitů), jak ukazuje obrázek 6-27.



Obr. 6-27 Přístup k aktuálním hodnotám vysokorychlostních čítačů

Přiřazení přerušení

Všechny režimy podporují přerušení v případě rovnosti aktuální a požadované hodnoty. Režimy, které využívají externí nulovací vstup, podporují přerušení při aktivaci externího nulování. Všechny režimy čítače s výjimkou režimů 0, 1 a 2 podporují přerušení při změně směru čítání. Každá z těchto podmínek přerušení může být samostatně povolena nebo blokována. Použití přerušení je podrobně popsáno v odstavci o přerušení a komunikaci.

Poznámka

Při použití přerušení od externího nulovacího vstupu se nepokoušejte nahrát novou aktuální hodnotu nebo blokovat a následně povolit vysokorychlostní čítač z přerušovacího podprogramu přiřazeného k dané události. Výsledkem může být výskyt kritické chyby.

Stavový byte

Stavový byte pro každý vysokorychlostní čítač má stavové paměťové bity indikující aktuální směr čítání a obsahující informaci o tom, zda je aktuální hodnota větší nebo rovna požadované hodnotě. Tabulka 6-29 definuje tyto stavové bity pro každý vysokorychlostní čítač.



Tip

Stavové bity jsou platné pouze v době provádění podprogramu přerušení pro vysokorychlostní čítač. Účelem monitorování stavu vysokorychlostního čítače je povolit přerušení pro události, které jsou pro právě prováděnou operaci důležité.

Tabulka 6-29 Stavové bity pro HSC0, HSC1, HSC2, HSC3, HSC4 a HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Popis
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	Nevyužito
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	Nevyužito
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	Nevyužito
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	Nevyužito
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	Nevyužito
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	Stavový bit aktuálního směru čítání: 0 = čítání dolů 1 = čítání nahoru
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	Stavový bit rovnosti požadované a aktuální hodnoty: 0 = Nejsou si rovny 1 = Rovny
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	Stavový bit příznaku aktuální hodnota větší než požadovaná: 0 = Menší nebo rovna 1 = Větší než

Příklad inicializačních sekvencí pro vysokorychlostní čítače

V následujících popisech inicializačních a provozních sekvencí je jako modelový čítač použit HSC1. Při popisech inicializace se předpokládá, že byl do režimu RUN právě přepnut automat S7-200, a že tedy platí paměťový bit prvního programového cyklu. Pokud není tato podmínka splněna, je třeba si uvědomit, že instrukce HDEF může být pro každý vysokorychlostní čítač provedena po přepnutí do režimu RUN pouze jednou. Provedení instrukce HDEF pro vysokorychlostní čítač podruhé generuje při běhu programu chybu a nezmění nastavení čítače, které bylo zadáno v první HDEF.



Tip

Přestože následující sekvence ukazují, jak změnit směr, aktuální hodnotu a požadovanou hodnotu jednotlivě, je možné změnit všechny tyto prvky nebo jejich libovolnou kombinaci v jediné sekvenci příslušným nastavením hodnoty SMB47 a následným provedením instrukce HSC.

Inicializace režimů 0, 1 nebo 2

Následující kroky popisují inicializaci HSC1 pro jednofázový čítač nahoru/dolů s vnitřním ovládáním směru (režimy 0, 1 nebo 2).

1. Na vyvolání inicializačního podprogramu použijte paměťový bit prvního programového cyklu. Protože používáte volání podprogramu, následující programové cykly už podprogram nevolají, čímž se zkracuje čas provádění programového cyklu a zlepšuje se strukturovanost programu.
2. V inicializačním podprogramu naplňte SMB47 podle požadované řídicí operace. Například:

SMB47 = 16#F8 *má za následek:*

Povolí čítač

Zapíše novou aktuální hodnotu

Zapíše novou požadovanou hodnotu

Nastaví směr čítání nahoru

Nastaví startovací a nulovací vstup na aktivní v úrovni log. 1.
3. Proved'te instrukci HDEF se vstupem HSC nastaveným na 1 a vstupem MODE nastaveným na jedno z následujících: 0 bez externího nulování nebo startu, 1 pro externí nulování a bez startu nebo 2 pro externí nulování i start.
4. Uložte do SMD48 (hodnota double word) požadovanou aktuální hodnotu (pro vymazání nahrajte 0).
5. Uložte do SMD52 (hodnota double word) požadovanou nastavenou hodnotu.
6. Abyste zachytili událost, kdy se aktuální hodnota rovná požadované, naprogramujte přerušení přidáním přerušovací události CV = PV (událost 13) do podprogramu přerušení. Úplné podrobné informace o zpracování přerušení najdete v části Instrukce přerušení.
7. Abyste zachytili externí nulovací událost, naprogramujte přerušení přidáním přerušovací události externího nulování (událost 15) do podprogramu přerušení.
8. Proved'te instrukci globálního povolení přerušení (ENI) pro povolení přerušení.
9. Proved'te instrukci HSC, aby automat S7-200 naprogramoval HSC1.
10. Ukončete podprogram.

Inicializace režimů 3, 4 nebo 5

Následující kroky popisují inicializaci HSC1 pro jednofázový čítač nahoru/dolů s externím ovládáním směru (režimy 3, 4 nebo 5).

1. Na vyvolání inicializačního podprogramu použijte paměťový bit prvního programového cyklu. Protože používáte volání podprogramu, následující programové cykly už podprogram nevolají, čímž se zkracuje čas programového cyklu a zlepšuje se strukturovanost programu.
2. V inicializačním podprogramu naplňte SMB47 podle požadované řídicí operace. Například:

SMB47 = 16#F8	<i>má za následek:</i>
	Povolí čítač
	Zapíše novou aktuální hodnotu
	Zapíše novou požadovanou hodnotu
	Nastaví počáteční směr čítání HSC nahoru
	Nastaví startovací a nulovací vstup na aktivní v úrovni log. 1.
3. Proveďte instrukci HDEF se vstupem HSC nastaveným na 1 a vstupem MODE nastaveným na jedno z následujících: 3 bez externího nulování nebo startu, 4 pro externí nulování a bez startu nebo 5 pro externí nulování i start.
4. Uložte do SMD48 (hodnota double word) požadovanou aktuální hodnotu (pro vymazání nahraďte 0).
5. Uložte do SMD52 (hodnota double word) požadovanou nastavenou hodnotu.
6. Abyste zachytili událost, kdy se aktuální hodnota rovná požadované, naprogramujte přerušení přidáním události přerušení CV = PV (událost 13) do podprogramu přerušení. Úplné podrobné informace o zpracování přerušení najdete v části Instrukce přerušení.
7. Abyste zachytili změny směru, naprogramujte přerušení přidáním přerušovací události změna směru (událost 14) do podprogramu přerušení.
8. Abyste zachytili externí nulovací událost, naprogramujte přerušení přidáním přerušovací události externího nulování (událost 15) do podprogramu přerušení.
9. Proveďte instrukci globálního povolení přerušení (ENI) pro povolení přerušení.
10. Proveďte instrukci HSC, aby automat S7-200 naprogramoval HSC1.
11. Ukončete podprogram.

Inicializace režimů 6, 7 nebo 8

Následující kroky popisují inicializaci HSC1 pro dvoufázový čítač nahoru/dolů se vstupy pro hodinové pulzy ve směru nahoru/dolů (režimy 6, 7 nebo 8):

1. Na vyvolání inicializačního podprogramu použijte paměťový bit prvního programového cyklu. Protože používáte volání podprogramu, následující programové cykly už podprogram nevolají, čímž se zkracuje čas programového cyklu a zlepšuje se strukturovanost programu.
2. V inicializačním podprogramu naplňte SMB47 podle požadované řídicí operace. Například:

SMB47 = 16#F8	<i>má za následek:</i>
	Povolí čítač
	Zapíše novou aktuální hodnotu
	Zapíše novou požadovanou hodnotu
	Nastaví počáteční směr čítání HSC nahoru
	Nastaví startovací a nulovací vstup na aktivní vysokou úroveň
3. Proveďte instrukci HDEF se vstupem HSC nastaveným na 1 a vstupem MODE nastaveným na jedno z následujících: 6 bez externího nulování nebo startu, 7 pro externí nulování a bez startu nebo 8 pro externí nulování i start.
4. Uložte do SMD48 (hodnota double word) požadovanou aktuální hodnotu (pro vymazání nahraďte 0).
5. Uložte do SMD52 (hodnota double word) požadovanou nastavenou hodnotu.
6. Abyste zachytili událost, kdy se aktuální hodnota rovná požadované, naprogramujte přerušení přidáním události přerušení CV = PV (událost 13) do podprogramu přerušení. Viz část o přerušeních.

7. Abyste zachytili změny směru, naprogramujte přerušení přidáním přerušovací události změny směru (událost 14) do podprogramu přerušení.
8. Abyste zachytili externí nulovací událost, naprogramujte přerušení přidáním přerušovací události externího nulování (událost 15) do podprogramu přerušení.
9. Proved'te instrukci globálního povolení přerušení (ENI) pro povolení přerušení.
10. Proved'te instrukci HSC, aby automat S7-200 naprogramoval HSC1.
11. Ukončete podprogram.

Inicializace režimů 9, 10 nebo 11

Následující kroky popisují inicializaci HSC1 pro A/B fázový kvadrurní čítač (pro režimy 9, 10 nebo 11):

1. Na vyvolání inicializačního podprogramu použijte paměťový bit prvního programového cyklu. Protože používáte volání podprogramu, následující programové cykly už podprogram nevolají, čímž se zkracuje čas provádění programového cyklu a zlepšuje se strukturovanost programu.
2. V inicializačním podprogramu naplňte SMB47 podle požadované řídicí operace.

Například (při režimu 1x):

SMB47 = 16#FC *má za následek:*
 Povolí čítač
 Zapiše novou aktuální hodnotu
 Zapiše novou požadovanou hodnotu
 Nastaví počáteční směr čítání HSC nahoru
 Nastaví startovací a nulovací vstup na aktivní v úrovni log. 1.

Například (při režimu 4x):

SMB47 = 16#F8 *má za následek:*
 Povolí čítač
 Zapiše novou aktuální hodnotu
 Zapiše novou požadovanou hodnotu
 Nastaví počáteční směr čítání HSC nahoru
 Nastaví startovací a nulovací vstup na aktivní v úrovni log. 1.

3. Proved'te instrukci HDEF se vstupem HSC nastaveným na 1 a vstupem MODE nastaveným na jedno z následujících: 9 bez externího nulování nebo startu, 10 pro externí nulování a bez startu nebo 11 pro externí nulování i start.
4. Uložte do SMD48 (hodnota double word) požadovanou aktuální hodnotu (pro vymazání nahrajte 0).
5. Uložte do SMD52 (hodnota double word) požadovanou nastavenou hodnotu.
6. Abyste zachytili událost, kdy se aktuální hodnota rovná požadované, naprogramujte přerušení přidáním události přerušení CV = PV (událost 13) do podprogramu přerušení. Úplné podrobné informace o zpracování přerušení najdete v části o povolování přerušení (ENI).
7. Abyste zachytili změny směru, naprogramujte přerušení přidáním přerušovací události změny směru (událost 14) do podprogramu přerušení.
8. Abyste zachytili externí nulovací událost, naprogramujte přerušení přidáním přerušovací události externího nulování (událost 15) do podprogramu přerušení.
9. Proved'te instrukci globálního povolení přerušení (ENI) pro povolení přerušení.
10. Proved'te instrukci HSC, aby automat S7-200 naprogramoval HSC1.
11. Ukončete podprogram.

Změna směru v režimech 0, 1 nebo 2

Následující kroky popisují nastavení HSC1 pro změnu směru pro jednofázový čítač s vnitřním ovládáním směru (režimy 0, 1 nebo 2):

- Pro zápis v požadovaném směru uložte do SMB47:

SMB47 = 16#90	Povolí čítač Nastaví směr čítání HSC dolů
SMB47 = 16#98	Povolí čítač Nastaví směr čítání HSC nahoru
- Proved'te instrukci HSC, aby automat S7-200 naprogramoval HSC1.

Zadání nové aktuální hodnoty (libovolný režim)

Během změny aktuální hodnoty je čítač blokován. Během blokování čítače se negenerují přerušení a nepokračuje čítání.

Následující kroky popisují změnu aktuální hodnoty čítače HSC1 (platí pro všechny režimy):

- Pro zápis požadované aktuální hodnoty uložte do SMB47:

SMB47 = 16#C0	Povolí čítač Zapíše novou aktuální hodnotu
- Uložte do SMD48 (hodnota double word) požadovanou aktuální hodnotu (pro vymazání nahrajte 0).
- Proved'te instrukci HSC, aby automat S7-200 naprogramoval HSC1.

Zadání nové požadované hodnoty (libovolný režim)

Následující kroky popisují změnu požadované hodnoty čítače HSC1 (platí pro všechny režimy):

- Pro zápis nové požadované hodnoty uložte do SMB47:

SMB47 = 16#A0	Povolí čítač Zapíše novou požadovanou hodnotu
- Uložte do SMD52 (hodnota double word) požadovanou nastavenou hodnotu.
- Proved'te instrukci HSC, aby automat S7-200 naprogramoval HSC1.

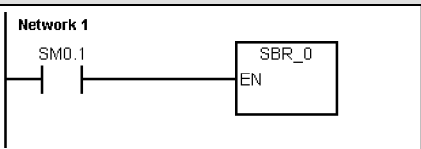
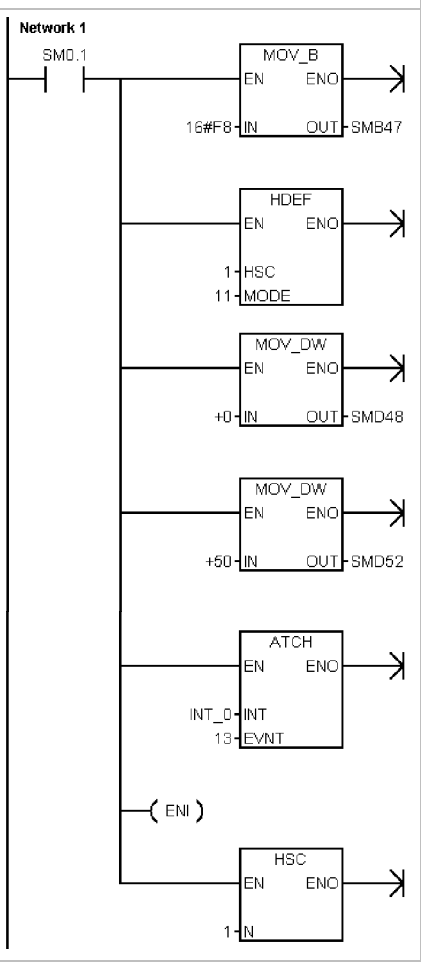
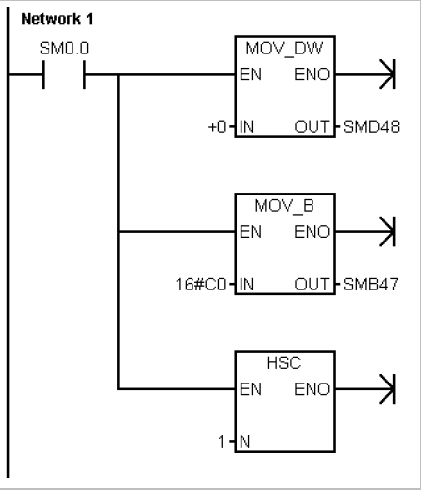
Pozastavení vysokorychlostního - čítače (libovolný režim)

Následující kroky popisují způsob, jak pozastavit vysokorychlostní čítač HSC1 (platí pro všechny režimy):

- Uložte do SMB47 pro pozastavení čítače:

SMB47 = 16#00	Pozastaví čítač
- Proved'te instrukci HSC, abyste čítač pozastavili.

Příklad: Instrukce vysokorychlostního čítače

MAIN	Network 1 	Network 1 //V prvním programovém cyklu volejte SBR_0. <pre>LD SM0.1 CALL SBR_0</pre>
SBR 0	Network 1 	Network 1 //V prvním programovém cyklu konfiguruje HSC1: //1. Povolí čítač. // – Zapiše novou aktuální hodnotu. // – Zapiše novou požadovanou hodnotu. // – Nastaví počáteční směr čítání nahoru. // – Nastaví startovací a nulovací vstup aktivní v úrovni log. 1. // – Nastaví 4x režim. //2. Konfiguruje HSC1 pro kvadrurní režim // se vstupy "Reset" a "Start". //3. Vymaže aktuální hodnotu HSC1. //4. Nastaví požadovanou hodnotu HSC1 na 50. //5. Pokud aktuální hodnota HSC1 = požadované hodnotě, // přiřadí událost 13 k přerušovacímu podprogramu INT_0. //6. Povolí globálního přerušení. //7. Naprogramuje HSC1. <pre>LD SM0.1 MOVB 16#F8, SMB47 HDEF 1, 11 MOVD +0, SMD48 MOVD +50, SMD52 ATCH INT_0, 13 ENI HSC 1</pre>
INT 0	Network 1 	Network 1 //Naprogramuje HSC1: //1. Vymaže aktuální hodnotu HSC1. //2. Vybere zapisování pouze nové aktuální hodnoty // a ponechá HSC1 povolený. <pre>LD SM0.0 MOVD +0, SMD48 MOVB 16#C0, SMB47 HSC 1</pre>

Pulzní výstupy



Řízení polohy

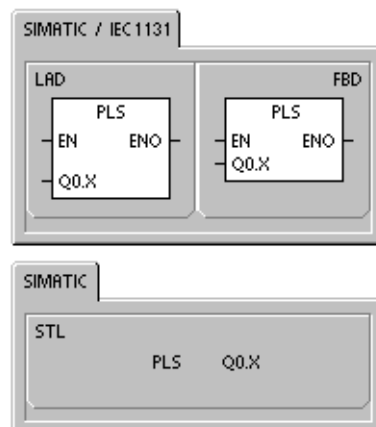
Instrukce "Pulzní výstup" (PLS) se používá pro řízení funkce pulzních výstupů (PTO) a výstupů s pulzní šířkovou modulací (PWM), dostupných na vysokorychlostních výstupech (Q0.0 a Q0.1). Na konfiguraci pulzních výstupů je možné použít "Průvodce polohováním".

PTO generuje obdélníkový signál (50% střída) s možností řízení periody a počtu pulzů.

PWM generuje kontinuální výstup s proměnnou střídou s možností řízení periody a počtu pulzů.

Programovatelný automat S7-200 má dva PTO/PWM generátory pro generování sledu vysokorychlostních pulzů a šířkově modulovaných signálů. Jeden generátor má přiřazen digitální výstup Q0.0 a druhý generátor Q0.1. Určená oblast speciální paměti (SM) uchovává pro každý generátor následující data: řídicí byte (8bitová hodnota), hodnotu počtu pulzů (32bitová hodnota bez znaménka) a hodnotu periody a šířky pulzu (16bitová hodnota bez znaménka).

Generátory PTO/PWM a registr obrazu procesu se dělí o použití Q0.0 a Q0.1. Když je PTO nebo PWM aktivní na Q0.0 nebo Q0.1, je výstup řízen generátorem PTO/PWM a normální použití výstupu není povoleno. Tvar výstupního signálu není ovlivněn stavem registru obrazu procesu, vnucenou (forced) hodnotou výstupu nebo provedením instrukcí přímého výstupu. Když je generátor PTO/PWM neaktivní, řízení výstupu přebírá zpět registr obrazu procesu. Registr obrazu procesu určuje stav výstupního signálu a řídí, zda je signál v log. 1 nebo log. 0.



Tabulka 6-30 Operandy platné pro instrukci pulzního výstupu

Vstupy/Výstupy	Typy dat	Operandy
Q0.X	WORD	Konstanta: 0 (= Q0.0) nebo 1 (= Q0.1)



Tip

Před povolením PTO nebo PWM nastavte hodnotu registru obrazu procesu pro Q0.0 a Q0.1 na 0. Implicitní hodnoty pro všechny řídicí bity, čas cyklu, šířku pulzu a počet pulzů jsou 0.

Výstupy PTO/PWM musí být zatíženy minimálně 10 % nominální zátěží, aby generovaly dostatečně ostré přechody z vypnutého do zapnutého stavu a ze zapnutého do vypnutého stavu.



Tipy a triky

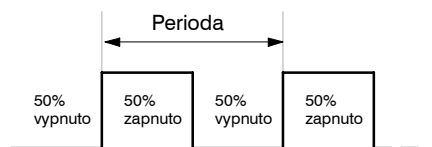
Tipy a triky pro programy, které používají instrukci PLS pro operaci PTO/PWM, najdete na CD s dokumentací. Viz tip 7, tip 22, tip 23, tip 30 a tip 50.

Činnost pulzního výstupu (PTO)

PTO generuje obdélníkový signál (střída 50%) s daným počtem pulzů a danou periodou. (Viz obrázek 6-28.) PTO může generovat jednu nebo více sérií pulzů (s použitím pulzu profilu). Specifikuje se počet pulzů a perioda (v mikrosekundových nebo milisekundových přírůstcích):

- ☐ Počet pulzů: 1 až 4 294 967 295
- ☐ Čas cyklu: 50 μ s až 65.535 μ s nebo 2 ms až 65.535 ms.

Pokud má perioda formu lichého čísla (např. 75 ms) způsobuje to určitou deformaci signálu (duty cycle).



Obr. 6-28 Pulzní výstup (PTO)

Omezení počtu pulzů a periody najdete v tabulce 6-31.

Tabulka 6-31 Počet pulzů a periody ve funkci PTO

Počet pulzů, perioda	Odezva
Perioda < 2 časové jednotky	Perioda se implicitně nastaví na 2 časové jednotky.
Počet pulzů = 0	Počet pulzů je implicitně nastaven na 1 pulz.

Funkce PTO umožňuje "zřetězení" nebo "zřetěžené zpracování" sekvence pulzů. Když se ukončí aktivní sekvence pulzů, bezprostředně začne výstup nové sekvence pulzů. Tak se udržuje spojitost mezi za sebou jdoucími výstupními sekvencemi pulzů.

Řetězení samostatných segmentů pulzů PTO

Při řetězení samostatných segmentů odpovídá programátor za aktualizaci SM oblasti pro následující sekvenci. Když byl spuštěn první PTO segment, musíte bezprostředně změnit SM oblasti tak, aby odpovídaly průběhu následujícího signálu, a opětovně provést instrukci PLS. Atributy druhé sekvence se drží v "řetězu", dokud se neukončí první sekvence pulzů. V "řetězu" může být v dané době pouze jedna položka. Když se ukončí první sekvence, začne výstup druhé vlny a "řetěz" je připraven pro zadání atributů nové sekvence. Tento proces můžete zopakovat pro nastavení atributů další sekvence pulzů.

Přechod mezi sekvencemi pulzů je hladký, pokud nedojde ke změně časové základny nebo pokud se neukončí aktivní sekvence pulzů ještě před zadáním atributů nové sekvence pomocí provedení instrukce PLS.

Řetězení více segmentů pulzů PTO

Při řetězení více segmentů S7-200 automaticky čte atributy každé sekvence pulzů z takzvané tabulky profilu, která se nachází v paměti V. Z paměti SM se v tomto režimu pro tabulku profilu používá řídicí byte, stavový byte a počáteční offset paměti V (SMW168 nebo SMW178). Časová základna může být dána v mikrosekundách nebo milisekundách, ale je stejná pro všechny časy cyklů v tabulce profilu a v době běhu profilu se nedá měnit. Řetězení s více segmenty se spouští provedením instrukce PLS.

Každá položka pro segment má délku 8 bytů a skládá se z 16bitové hodnoty času cyklu, 16bitové hodnoty změny periody a 32bitové hodnoty počtu pulzů. Tabulka 6-32 ukazuje formát tabulky profilu. Periodu můžete automaticky zvýšit nebo snížit naprogramováním příslušné hodnoty pro každý pulz. Kladná hodnota pro změnu periody prodlužuje periodu, záporná hodnota ji zkracuje; zadáním 0 se perioda nemění.

V době běhu profilu PTO se počet aktuálně aktivních segmentů dá zjistit v SMB166 (nebo SMB176).

Tabulka 6-32 Formát tabulky profilu - pro režim PTO s více segmenty

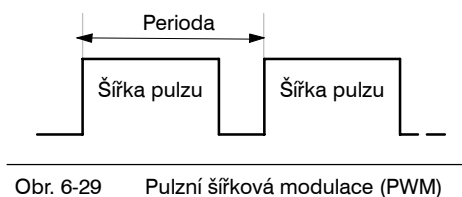
Posunutí bytu	Segment	Popis položek tabulky
0		Počet segmentů: 1 až 255 ¹
1	#1	Počáteční perioda (2 až 65.535 jednotek časové základny)
3		Změna periody na jeden pulz (hodnota se znaménkem) (–32.768 až 32.767 jednotek časové základny)
5		Počet pulzů (1 až 4.294.967.295)
9	#2	Počáteční perioda (2 až 65.535 jednotek časové základny)
11		Změna periody na jeden pulz (hodnota se znaménkem) (–32.768 až 32.767 jednotek časové základny)
13		Počet pulzů (1 až 4.294.967.295)
(Pokračování)	#3	(Pokračování)

¹ Vložení hodnoty 0 jako počtu segmentů generuje nekritickou chybu. Negeneruje se výstup PTO.

Pulzně šířková modulace (PWM)

Funkce PWM dává výstup s pevnou periodou a s proměnnou délkou pulzu (viz obrázek 6-29). Periodu a šířku pulzu můžete zadat v mikrosekundových nebo milisekundových přírůstcích:

- ☐ Perioda: 50 μ s až 65.535 μ s nebo 2 ms až 65.535 ms
- ☐ Šířka pulzu: 0 μ s až 65.535 μ s nebo 0 ms až 65.535 ms



Obr. 6-29 Pulzní šířková modulace (PWM)

Jak je patrné z tabulky 6-33, při nastavení šířky pulzu tak, že se rovná periodě (střída 100 %), je výstup trvale ve stavu log. 1. Nastavení šířky pulzu na 0 (střída 0 %) způsobí, že výstup je v log. 0.

Tabulka 6-33 Šířka pulzu, perioda a odezva funkce PWM

Šířka pulzu, perioda	Odezva
Šířka pulzu $\geq$ Hodnota periody	Střída je 100 %: výstup je trvale zapnutý.
Šířka pulzu = 0	Střída je 0 %: výstup je vypnutý.
Perioda < 2 časové jednotky	Perioda je implicitně nastavena na dvě časové jednotky.

Existují dva různé způsoby, jak změnit charakteristiku PWM signálu:

- ☐ Synchronní aktualizace: Pokud není třeba měnit časovou základnu, je možné použít synchronní aktualizaci. U synchronní aktualizace se změna tvaru signálu děje na konci periody; proto dochází jen k hladkým přechodům.
- ☐ Asynchronní aktualizace: Pro operaci PWM je typické, že se mění šířka pulzu, zatímco perioda zůstává konstantní; proto se nepožaduje změna časové základny. Když je však změna časové základny generátoru PTO/PWM požadována, používá se asynchronní aktualizace. Ta zastaví PTO/PWM generátor asynchronně s PWM signálem. To může být příčinou nežádoucího nestabilního chování řízeného zařízení. Z tohoto důvodu se doporučuje používat synchronní způsob. Zvolte si takovou časovou základnu, která vyhovuje všem předpokládaným délkám periody.

**Tip**

Příznakový bit metody aktualizace PWM (SM67.4 nebo SM77.4) v řídicím bytu specifikuje typ aktualizace použitý při provádění instrukce PLS.

Při změně časové základny dojde k asynchronní aktualizaci nezávisle na stavu příznakového bitu příznaku způsobu aktualizace PWM.

Použití míst v paměti SM pro konfiguraci a řízení operací PTO/PWM

Instrukce PLS čte data uložená na specifikovaných místech paměti SM a programuje podle toho generátor PTO/PWM. SMB67 řídí PTO 0 nebo PWM 0 a SMB77 řídí PTO 1 nebo PWM 1. Tabulka 6-34 popisuje registry použité pro řízení funkcí PTO/PWM. Tabulku 6-35 můžete použít pro rychlé určení hodnoty, kterou je třeba uložit do řídicího registru PTO/PWM pro spuštění požadované operace.

Charakteristiku signálu PTO nebo PWM můžete změnit úpravou bytů v oblasti paměti SM (včetně řídicího bytu) a následujícím provedením instrukce PLS. Generování signálu PTO nebo PWM můžete kdykoliv pozastavit zapsáním 0 do povolovacího bitu PTO/PWM v řídicím bytu (SM67.7 nebo SM77.7) a následujícím provedením instrukce PLS.

Bit "PTO nečinný" ve stavovém bytu (SM66.7 nebo SM76.7) slouží pro indikaci ukončení naprogramované sekvence pulzů. Dále je možné po ukončení sekvence vyvolat přerušení (viz popis instrukcí přerušení a komunikace). Pokud používáte zřetězení více segmentů, je přerušovací podprogram vyvolán po dokončení tabulky profilu.

Následující podmínky nastavují SM66.4 (nebo SM76.4) a SM66.5 (nebo SM76.5):

- ☐ Při zadání hodnoty změny periody tak, že je výsledná perioda po daném počtu pulzů mimo povolený rozsah, se objeví matematické přetečení, které ukončí funkci PTO a nastaví bit chyby výpočtu změny periody (SM66.4 nebo SM76.4) na 1. Výstup se přepne na řízení pomocí registru obrazu procesu.
- ☐ Ruční přerušení (provádění) profilu PTO během činnosti nastaví bit přerušení uživatelem (SM66.5 nebo SM76.5) na 1.
- ☐ Při pokusu o nahrání atributů do plného "řetězu" se nastaví bit přetečení PTO (SM66.6 nebo SM76.6) na 1. Pokud chcete detekovat další přetečení, je nutné po přetečení tento bit ručně vymazat. Přechod do režimu RUN nastaví tento bit na 0.

**Tip**

Jestliže chcete nastavit novou hodnotu počtu pulzů (SMD72 nebo SMD82), šířky pulzu (SMW70 nebo SMW80) nebo periody (SMW68 nebo SMW78), nastavte také příslušné aktualizací bity v řídicím registru dříve, než provedete instrukci PLS. Při použití režimu s více segmenty je též nutné před provedením instrukce PLS zapsat počáteční offset (SMW168 nebo SMW178) tabulky profilu.

Tabulka 6-34 Umístění řídicích registrů PTO / PWM v paměti SM

Q0.0	Q0.1	Stavové bity		
SM66.4	SM76.4	Profil PTO přerušen (chyba výpočtu změny per.):	0 = bez chyby	1 = přerušeno
SM66.5	SM76.5	Profil PTO přerušen uživatelským příkazem:	0 = bez přerušení	1 = přerušení
SM66.6	SM76.6	Přetečení/nenaplnění "řetězu" PTO:	0 = bez přeplnění	1 = přetečení / nenaplnění
SM66.7	SM76.7	PTO nečinný:	0 = PTO pracuje	1 = PTO nečinný
Q0.0	Q0.1	Řídicí bity		
SM67.0	SM77.0	Příznak aktualizace periody PTO/PWM:	0 = bez aktualizace	1 = aktualizuje periodu
SM67.1	SM77.1	Příznak aktualizace času šířky pulzu PWM:	0 = bez aktualizace	1 = aktualizuje šířku pulzu
SM67.2	SM77.2	Příznak aktualizace počtu pulzů PTO:	0 = bez aktualizace	1 = aktualizuje počet pulzů
SM67.3	SM77.3	Volba časové základny PTO/PWM:	0 = 1 μs/pulz	1 = 1 ms/pulz
SM67.4	SM77.4	Příznak metody aktualizace PWM:	0 = asynchronní	1 = synchronní
SM67.5	SM77.5	Režim PTO s jedním/více segmenty:	0 = s 1 segmentem	1 = s více segmenty
SM67.6	SM77.6	Volba režimu PTO/PWM:	0 = PTO	1 = PWM
SM67.7	SM77.7	Povolení PTO/PWM:	0 = PTO/PWM pozastaven 1 = PTO/PWM povolen	
Q0.0	Q0.1	Další registry PTO/PWM		
SMW68	SMW78	Hodnota periody PTO/PWM	: rozsah 2 až 65 535	
SMW70	SMW80	Hodnota šířky pulzu PWM	: rozsah 0 až 65 535	
SMD72	SMD82	Hodnota počtu pulzů PTO	: rozsah 1 až 4 294 967 295	
SMB166	SMB176	Počet zpracovávaných segmentů	Pouze při režimu PTO s více segmenty	
SMW168	SMW178	Počáteční adresa tabulky profilu (posunutí v bytech od V0)	Pouze při režimu PTO s více segmenty	

Tabulka 6-35 Řídicí byte PTO/PWM

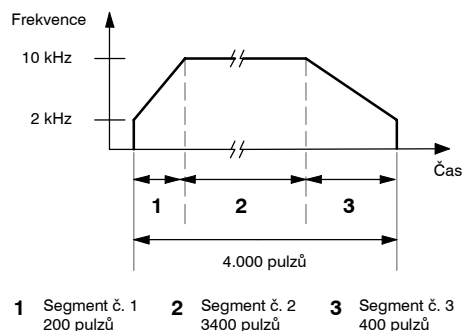
Řídicí registr (Hexa hodnota)	Výsledek provedení instrukce PLS							
	Povoleno	Režim výběru	PTO režim segmentů	PWM Metoda aktualizace	Časová základna	Počet pulzů	Šířka pulzů	Perioda
16#81	Ano	PTO	Jediný		1 μs/perioda			Zadat
16#84	Ano	PTO	Jediný		1 μs/perioda	Zadat		
16#85	Ano	PTO	Jediný		1 μs/perioda	Zadat		Zadat
16#89	Ano	PTO	Jediný		1 ms/perioda			Zadat
16#8C	Ano	PTO	Jediný		1 ms/perioda	Zadat		
16#8D	Ano	PTO	Jediný		1 ms/perioda	Zadat		Zadat
16#A0	Ano	PTO	Více		1 μs/perioda			
16#A8	Ano	PTO	Více		1 ms/perioda			
16#D1	Ano	PWM		Synchronní	1 μs/perioda			Zadat
16#D2	Ano	PWM		Synchronní	1 μs/perioda		Zadat	
16#D3	Ano	PWM		Synchronní	1 μs/perioda		Zadat	Zadat
16#D9	Ano	PWM		Synchronní	1 ms/perioda			Zadat
16#DA	Ano	PWM		Synchronní	1 ms/perioda		Zadat	
16#DB	Ano	PWM		Synchronní	1 ms/perioda		Zadat	Zadat

Výpočet hodnot tabulky profilu

Režim generátorů PTO/PWM s řetězením více segmentů je užitečný v mnoha aplikacích, obzvláště při řízení krokového motoru.

Funkci PTO s profilem pulzu můžete například použít na řízení krokového motoru v průběhu jednoduché sekvence rozběhu, provozu a doběhu nebo pro komplikovanější sekvence tak, že definujete profil pulzu, který se skládá z až 255 segmentů, kde každý segment odpovídá operaci rozběhu, chodu nebo doběhu.

Obrázek 6-30 ukazuje příklad hodnot tabulky profilu, potřebných pro generování výstupního signálu, který rozběhne krokový motor (segment 1), udrží motor na konstantní rychlosti (segment 2) a zpomalí ho (segment 3).



1 Segment č. 1 200 pulzů 2 Segment č. 2 3400 pulzů 3 Segment č. 3 400 pulzů

Obr. 6-30 Časově frekvenční diagram

V tomto příkladu: Počáteční a konečná frekvence pulzů je 2 kHz, maximální frekvence pulzů je 10 kHz. Pro dosažení požadovaných otáček motoru je zapotřebí 4000 pulzů. Protože jsou hodnoty v tabulce profilu vyjádřeny v počtech period místo frekvence, je třeba převést dané hodnoty frekvence na periodu. Proto je počáteční (výchozí) a konečná (závěrečná) perioda 500 μ s a perioda odpovídající maximální frekvenci je 100 μ s. Během rozběhové fáze profilu má být dosaženo maximální frekvence pulzů po přibližně 200 pulzech. Zpomalovací část profilu má být ukončena přibližně za 400 pulzů.

Pro stanovení hodnoty změny periody daného segmentu, který generátor PTO/PWM používá pro přizpůsobení periody každého pulzu, můžete použít následující vzorec:

$$\text{Změna periody pro daný segment} = | \text{End\_CT}_{\text{seg}} - \text{Init\_CT}_{\text{seg}} | / \text{Quantity}_{\text{seg}}$$

kde: $\text{End\_CT}_{\text{seg}}$ = Konečná perioda pro daný segment
 $\text{Init\_CT}_{\text{seg}}$ = Počáteční perioda pro daný segment
 $\text{Quantity}_{\text{seg}}$ = Počet pulzů v tomto segmentu

Použití tohoto vzorce pro výpočet hodnot změny periody pro daný příklad: Tabulka 6-36 Hodnoty tabulky profilu

Segment 1 (rozběh):
 Změna periody = -2

Segment 2 (konstantní rychlost):
 Změna periody = 0

Segment 3 (zpomalení):
 Změna periody = 1

Tabulka 6-36 uvádí hodnoty pro generování signálu z příkladu (za předpokladu, že tabulka profilu se nachází v paměti V se začátkem na V500). Do svého programu můžete zahrnout instrukce pro zápis těchto hodnot do paměti V nebo můžete hodnoty profilu definovat v datovém bloku programu.

Adresa	Hodnota	Popis
VB500	3	Celkový počet segmentů
VW501	500	Počáteční perioda
VW503	-2	Počáteční změna periody
VD505	200	Počet pulzů
VW509	100	Počáteční perioda
VW511	0	Změna periody
VD513	3400	Počet pulzů
VW517	100	Počáteční perioda
VW519	1	Změna periody
VD521	400	Počet pulzů

Abyste stanovili, zda jsou přechody mezi segmenty signálu přijatelné, potřebujete určit periodu posledního pulzu v segmentu. Pokud není změna periody 0, musíte vypočítat periodu posledního pulzu v segmentu, protože tato hodnota není přímo dána v tabulce profilu. Pro výpočet periody posledního pulzu použijte následující vzorec:

$$\text{Perioda posledního pulzu v segmentu} = \text{Init\_CT}_{\text{seg}} + (\text{Delta}_{\text{seg}} * (\text{Quantity}_{\text{seg}} - 1))$$

kde: $\text{Init\_CT}_{\text{seg}}$ = Počáteční perioda pro daný segment

$\text{Delta}_{\text{seg}}$ = Změna periody pro daný segment

$\text{Quantity}_{\text{seg}}$ = Počet pulzu v tomto segmentu

Výše uvedený zjednodušený příklad je užitečný jako úvod, ale skutečná použití mohou používat podstatně složitější profily. Mějte na paměti, že změna periody může být zadána jen jako celé číslo v mikrosekundách nebo milisekundách a výpočet periody se provádí při každém pulzu.

Výsledkem je, že výpočet změny periody pro daný segment může vyžadovat iterativní přístup. Pravděpodobně bude nutné použít proměnné hodnoty času poslední periody, případně počtu pulzů pro daný segment.

Doba trvání daného segmentu profilu může být užitečná při určování správných hodnot tabulky profilu. Na výpočet času potřebného k provedení daného segmentu se používá následující vzorec:

$$\text{Doba trvání segmentu} = \text{Quantity}_{\text{seg}} * (\text{Init\_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{Quantity}_{\text{seg}} - 1)))$$

kde: $\text{Quantity}_{\text{seg}}$ = Počet pulzů v tomto segmentu

$\text{Init\_CT}_{\text{seg}}$ = Počáteční perioda pro daný segment

$\text{Delta}_{\text{seg}}$ = Změna periody pro daný segment

Činnost výstupu PWM



Tip

Následující popis inicializace a činnosti výstupu PWM doporučuje použít pro inicializaci pulzního výstupu bit prvního programového cyklu (SM0.1). Použití tohoto bitu pro vyvolání inicializačního podprogramu zkracuje dobu programového cyklu, protože následující cykly už nemusí tento podprogram volat. (Bit prvního programového cyklu se nastavuje pouze u prvního programového cyklu následujícího po přechodu automatu do režimu RUN.) Aplikace však může mít jiná omezení, která vyžadují inicializaci (nebo opětovou inicializaci) pulzního výstupu. V takovém případě můžete pro vyvolání inicializačního podprogramu použít jinou podmínku.

Inicializace výstupu PWM

Pro inicializaci PWM se běžně používá podprogram. Inicializační podprogram se vyvolává z hlavního programu. Pro inicializaci výstupu PWM použijte bit prvního programového cyklu (SM0.1) a vyvolejte podprogram, který provede inicializační funkce. Když používáte volání podprogramu, následující programové cykly už podprogram nevolají, čímž se zkracuje čas provádění programového cyklu a zlepšuje se strukturovanost programu.

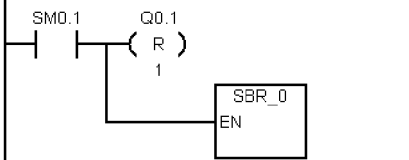
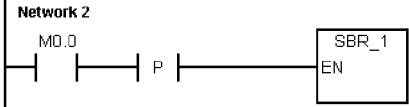
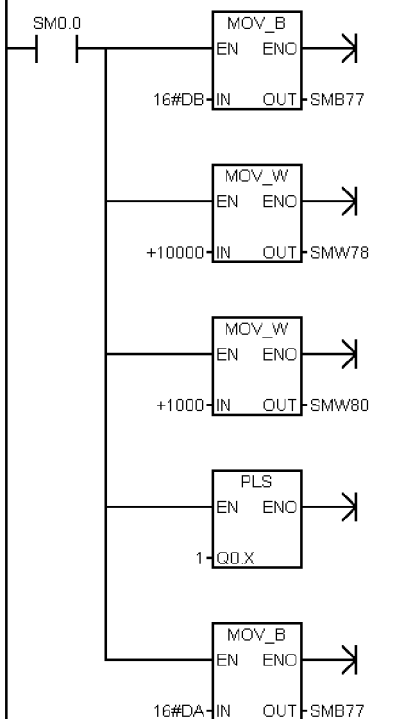
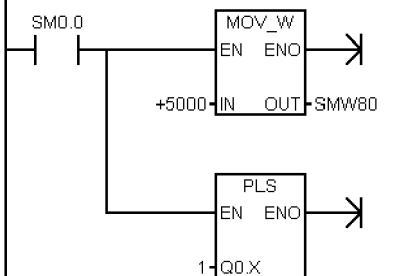
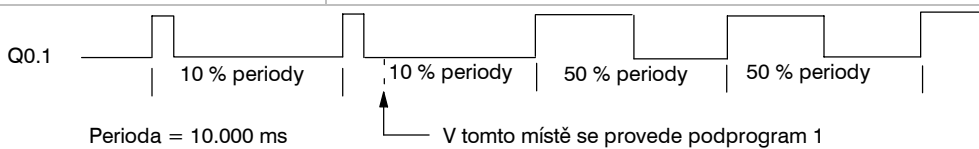
Po vyvolání inicializačního podprogramu z hlavního programu použijte následující kroky, abyste vytvořili řídicí logiku pro konfiguraci pulzního výstupu Q0.0 z inicializačního podprogramu:

1. Konfigurujte řídicí byte tím, že do SMB67 uložíte jednu z následujících hodnot: 16#D3 (zvolíte přírůstky v mikrosekundách) nebo 16#DB (zvolíte přírůstky v milisekundách).
Obě tyto hodnoty povolí funkci PTO/PWM, vyberou operaci PWM, nastaví hodnoty šířky pulzu a doby trvání a vyberou časovou základnu (mikrosekundy nebo milisekundy).
2. Do SMW68 uložíte hodnotu o velikosti word, která určuje periodu.
3. Do SMW70 nahrajte hodnotu o velikosti word pro určení šířky pulzu.
4. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM).
5. Chcete-li předem nahrát novou hodnotu řídicího bytu pro provedení změny šířky pulzu (volitelné), nahrajte do SMB67 jednu z následujících hodnot: 16#D2 (mikrosekundy) nebo 16#DA (milisekundy).
6. Ukončete podprogram.

Změna šířky pulzu pro výstup PWM

Pokud jste do SMB67 předem nahráli 16#D2 nebo 16#DA (viz krok 5. výše), můžete použít podprogram, který změní šířku pulzu pro pulzní výstup (Q0.0). Po vyvolání tohoto podprogramu použijte následující kroky, abyste vytvořili řídicí logiku pro změnu šířky pulzu:

1. Do SMW70 uložíte hodnotu o velikosti word pro novou šířku pulzu.
2. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM).
3. Ukončete podprogram.

Příklad: Pulzní šířková modulace (PWM)		
M A I N	Network 1  Network 2 	Network 1 //V prvním programovém cyklu nastaví bit //registru obrazu na nízkou úroveň a volá SBR_0. <pre>LD SM0.1 R Q0,1, 1 CALL SBR_0</pre> Network 2 //Nastaví M0.0 na jiném místě programu, //aby změnil šířku pulzu na 50 % periody. <pre>LD M0.0 EU CALL SBR_1</pre>
	Network 1 	Network 1 //Začátek podprogramu 0: //1. Nastavení řídicího bytu. // – Zvolí režim PWM. // – Zvolí přírůstky v ms a // synchronní aktualizaci. // – Povolí nahrání hodnot šířky pulzu // a času cyklu. // – Povolí funkci PWM. //2. Nastaví čas cyklu na 10.000 ms. //3. Nastaví šířku pulzu na 1.000 ms. //4. Vyvolá funkci PWM: PLS1=>Q0.1. //5. Předem nahraje řídicí byte pro následující // změny šířky pulzu <pre>LD SM0.0 MOVB 16#DB, SMB77 MOVW +10000, SMW78 MOVW +1000, SMW80 PLS 1 MOVB 16#DA, SMB77</pre>
S B R 0	Network 1 	Network 1 //Začátek podprogramu 1: //Nastaví šířku pulzu na 5000 ms. //Aplikuje změnu šířky pulzu. <pre>LD SM0.0 MOVW +5000, SMW80 PLS 1</pre>
S B R 1		
Časový diagram 		

Příklad práce výstupu PTO



Tip

Následující popis inicializace a činnosti výstupu PTO doporučuje použít pro inicializaci pulzního výstupu paměťový bit prvního programového cyklu (SM0.1). Použití tohoto bitu pro vyvolání inicializačního podprogramu zkracuje dobu programového cyklu, protože následující cykly už nemusí tento podprogram volat. (Bit prvního programového cyklu se nastavuje pouze u prvního programového cyklu následujícího po přechodu do režimu RUN.) Vaše aplikace však může mít jiná omezení, která vyžadují inicializaci (nebo opětovou inicializaci) pulzního výstupu. V takovém případě můžete pro vyvolání inicializačního podprogramu použít jinou podmínku.

Inicializace výstupu PTO v režimu s jedním segmentem

Pro konfiguraci a inicializaci PTO pro pulzní výstup se běžně používá podprogram. Inicializační podprogram se vyvolává z hlavního programu. Pro inicializaci výstupu užívaného PTO na 0 použijte paměťový bit prvního programového cyklu (SM0.1) a vyvolejte podprogram, který provede inicializační funkce. Když používáte volání podprogramu, následující programové cykly už podprogram nevolají, čímž se zkracuje čas provádění programového cyklu a zlepšuje se strukturovanost programu.

Po vyvolání inicializačního podprogramu z hlavního programu použijte následující kroky, abyste vytvořili řídicí logiku pro konfigurování pulzního výstupu Q0.0 z inicializačního podprogramu:

1. Konfigurujte řídicí byte tím, že do SMB67 nahrajete jednu z následujících hodnot: 16#85 (zvolíte přírůstky v mikrosekundách) nebo 16#8D (zvolíte přírůstky v milisekundách).
Obě tyto hodnoty povolí funkci PTO/PWM, vyberou operaci PTO, nastaví aktuální hodnotu počtu pulzů a periodu a vybírají časovou základnu (mikrosekundy nebo milisekundy).
2. Do SMW68 uložte hodnotu o velikosti word určující periodu.
3. Do SMD72 uložte hodnotu o velikosti double word pro počet pulzů.
4. (Volitelné) Chcete-li, aby se následující funkce provedla bezprostředně po ukončení sekvence cyklů, můžete do podprogramu přerušeni naprogramovat přerušeni od události ukončení sekvence pulzů (událost přerušeni 19). Použijte instrukci ATCH a proveďte instrukci globálního povolení přerušeni ENI.
5. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM).
6. Ukončete podprogram.

Změna periody PTO (režim s jedním segmentem)

V režimu cyklu PTO s jedním segmentem můžete pro změnu periody použít přerušeni nebo podprogram. Chcete-li takto změnit periodu PTO, proveďte následující kroky:

1. Nastavte řídicí byte (pro povolení funkce PTO/PWM, vybrání operace PTO, vybrání časové základny a nastavení aktualizace periody) nahráním jedné z následujících hodnot do SMB67: 16#81 (mikrosekundy) nebo 16#89 (milisekundy).
2. Do SMW68 nahrajte hodnotu o velikosti word pro novou periodu.
3. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM). Dříve než začne S7-200 generovat signál s aktualizovanou periodou, dokončí aktuální sekvenci PTO, která se právě zpracovává.
4. Ukončete přerušeni nebo podprogram.

Změna počtu pulzů PTO (režim s jedním segmentem)

PTO v režimu s jedním segmentem můžete pro změnu počtu pulzů použít přerušení nebo podprogram. Chcete-li takto změnit počet pulzů, proveďte následující kroky:

1. Nastavte řídicí byte (pro povolení funkce PTO/PWM, výběru operace PTO, výběru časové základny a nastavení hodnoty aktualizace počtu pulzů) tak, že do SMB67 nahrajete jednu z následujících hodnot: 16#84 (mikrosekundy) nebo 16#8C (milisekundy).
2. Do SMD72 uložte hodnotu o velikosti double word pro nový počet pulzů.
3. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM). Dříve než začne PTO generovat signál s aktualizovaným počtem pulzů, dokončí programovatelný automat S7-200 sekvenci PTO, která se právě zpracovává.
4. Ukončete přerušení nebo podprogram.

Změna periody a počtu pulzů PTO (režim s jedním segmentem)

PTO v režimu cyklu s jedním segmentem můžete pro změnu periody a počtu pulzů použít přerušení nebo podprogram. Chcete-li takto změnit periodu a počet pulzů, proveďte následující kroky:

1. Nastavte řídicí byte (pro povolení PTO/PWM, výběr operace PTO, výběr časové základny a nastavení hodnot aktualizace periody a počtu pulzů) nahráním jedné z následujících hodnot do SMB67: 16#85 (mikrosekundy) nebo 16#8D (milisekundy).
2. Do SMW68 uložte hodnotu o velikosti word pro novou periodu.
3. Do SMC72 uložte hodnotu o velikosti double word pro nový počet pulzů.
4. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM). Dříve než začne PTO generovat signál s aktualizovaným počtem pulzů a periodou, dokončí aktuální sekvenci PTO, která se právě zpracovává.
5. Ukončete přerušení nebo podprogram.

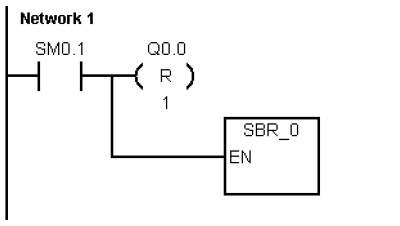
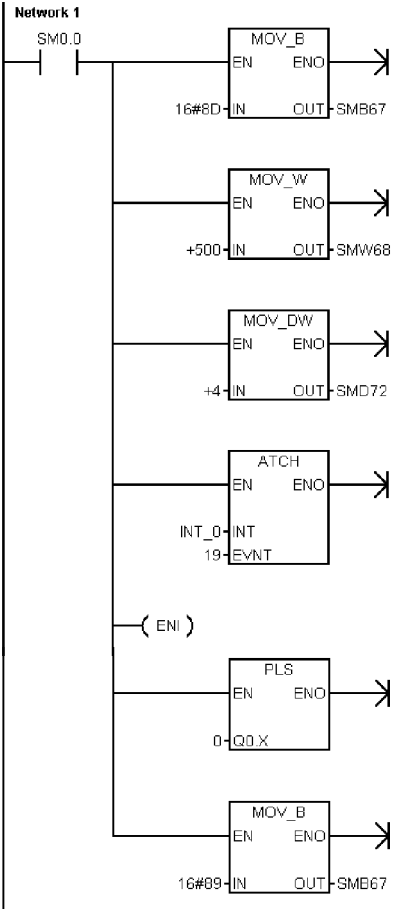
Inicializace výstupu PTO v režimu s více segmenty

Pro konfiguraci a inicializaci PTO pro pulzní výstup v režimu s více segmenty se běžně používá podprogram. Inicializační podprogram se vyvolává z hlavního programu. Pro inicializaci výstupu užívaného PTO na 0 použijte paměťový bit prvního programového cyklu (SM0.1) a vyvolejte podprogram, který provede inicializační funkci. Jestliže pro vyvolání inicializačního podprogramu použijete bit prvního programového cyklu, následující programové cykly už podprogram nevolají, což zkracuje čas provádění cyklu.

Po vyvolání inicializačního podprogramu z hlavního programu použijte následující kroky, abyste vytvořili inicializační podprogram s řídicí logikou pro konfigurování pulzního výstupu Q0.0:

1. Konfigurujte řídicí byte tím, že do SMB67 nahrajete jednu z následujících hodnot: 16#A0 (zvolíte přírůstky v mikrosekundách) nebo 16#A8 (zvolíte přírůstky v milisekundách).
Obě tyto hodnoty povolí funkci PTO/PWM, nastaví operaci PTO, nastaví režim s více segmenty a nastaví časovou základnu (mikrosekundy nebo milisekundy).
2. Do SMW168 uložte hodnotu o velikosti word pro počáteční offset tabulky profilu v paměti V pro.
3. Pro nastavení segmentů v tabulce profilu použijte paměť V. Zkontrolujte, zda je správně vyplněná položka "počet segmentů" (první byte tabulky).
4. (Volitelné) Chcete-li, aby se následující funkce provedla bezprostředně po ukončení profilu PTO, můžete do podprogramu přerušení naprogramovat přerušení od události ukončení sekvence pulzů (událost přerušení 19). Použijte instrukci ATCH a proveďte instrukci globálního povolení přerušení ENI.
5. Proveďte instrukci PLS (aby S7-200 naprogramoval generátor PTO/PWM).
6. Ukončete podprogram.

Příklad: Pulzní výstup (PTO) v režimu s jedním segmentem

MAIN	Network 1 	Network 1 //V prvním programovém cyklu //nastaví bit registru obrazu na 0 a volá podprogram 0. <pre> LD SM0.1 R Q0,0, 1 CALL SBR_0 </pre>
SBR 0	Network 1 	Network 1 //Začátek podprogramu 0: Konfigurace PTO //1. Nastavení řídicího bytu: // - Zvolí režim PTO. // - Zvolí režim s jedním segmentem. // - Zvolí přírůstky v ms. // - Povolí uložení hodnot počtu pulzů a periody. // - Povolí funkci PTO. //2. Nastaví periodu na 500 ms. //3. Nastaví počet pulzů na 4. //4. Definuje, že přerušení 0 je přerušení pro zpracování dokončených PTO. //5. Globální povolení přerušení. //6. Vyvolá funkci PTO, PLS0 => Q0.0. //7. Předem nahraje řídicí byte pro následující změnu periody. <pre> LD SM0.0 MOVB 16#8D, SMB67 MOVW +500, SMW68 MOVD +4, SMD72 ATCH INT_0, 19 ENI PLS 0 MOVB 16#89, SMB67 </pre>

Příklad: Pulzní výstup (PTO) v režimu s jedním segmentem, pokračování

Network 1

SMW68
==I
+500

MOV_W
EN ENO
+1000-IN OUT-SMW68

PLS
EN ENO
0-Q0.X

(RET)

Network 2

SMW68
==I
+1000

MOV_W
EN ENO
+500-IN OUT-SMW68

PLS
EN ENO
0-Q0.X

Network 1 //Pokud je aktuální perioda 500 ms:
//Nastaví periodu na 1000 ms a generuje 4 pulzy.

LDW= SMW68, +500
MOVW +1000, SMW68
PLS 0
CRETI

Network 2 //Pokud je aktuální perioda 1000 ms:
//Nastaví periodu na 500 ms a generuje 4 pulzy.

LDW= SMW68, +1000
MOVW +500, SMW68
PLS 0

Časový diagram

1 cyklus 500 ms

1 cyklus 1000 ms

Q0,0

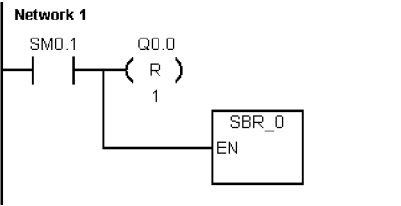
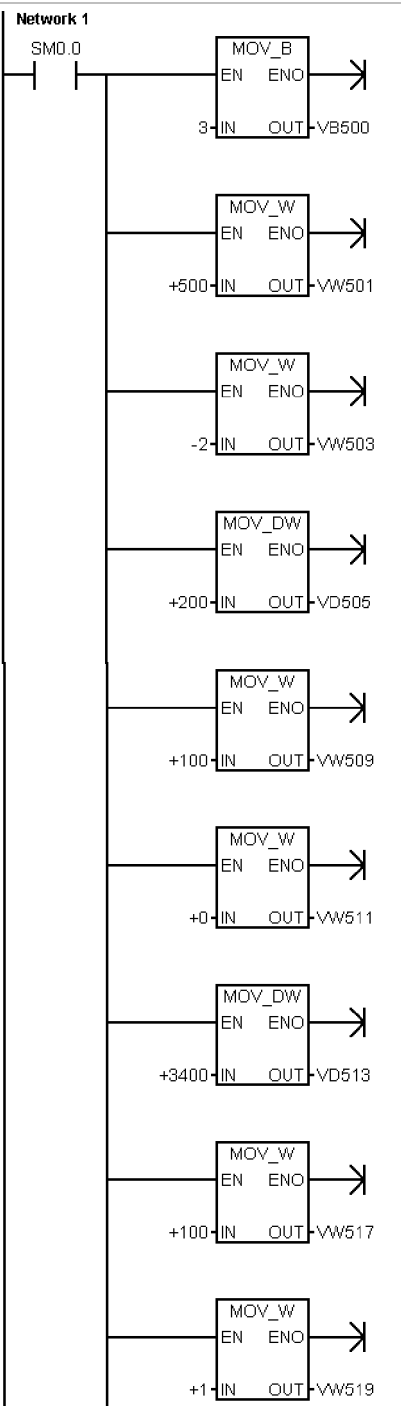
4 cykly nebo 4 pulzy

4 cykly nebo 4 pulzy

Provede se přerušení 0

Provede se přerušení 0

Příklad: Pulzní výstup (PTO) v režimu s více segmenty

M A I N	Network 1 	Network 1 //V prvním programovém cyklu //nastaví bit registru obrazu na log. 0 a volá podprogram 0 <pre> LD SM0.1 R Q0,0, 1 CALL SBR_0 </pre>
S B R 0	Network 1 	Network 1 //Předem nahraje tabulku profilu PTO: //Nastaví počet segmentů profilu na 3. //Konfiguruje každý ze 3 segmentů. // //1. Konfigurace segmentu 1: // - Nastaví počáteční periodu = 500 ms. // - Nastaví změnu periody na -2 ms. // - Nastaví počet pulzů na 200. //2. Konfigurace segmentu 2: // - Nastaví počáteční periodu na 100 ms. // - Nastaví změnu periody na 0 ms. // - Nastaví počet pulzů na 3400. //3. Konfigurace segmentu 3: // - Nastaví počáteční periodu na 100 ms. // - Nastaví změnu periody na 1 ms. // - Nastaví počet pulzů na 400. <pre> LD SM0.0 MOVB 3, VB500 MOVW +500, VW501 //Segment 1 MOVW -2, VW503 MOVD +200, VD505 MOVW +100, VW509 //Segment 2 MOVW +0, VW511 MOVD +3400, VD513 MOVW +100, VW517 //Segment 3 MOVW +1, VW519 MOVD +400, VD521 </pre>

Příklad: Pulzní výstup (PTO) v režimu s více segmenty, pokračování		
S B R O c o n t i n u e d		Netrowk 2 //1. Nastavení řídicího bytu: // – Zvolí režim PTO // – Zvolí režim s více segmenty // – Zvolí přírůstky v ms // – Povolí funkci PTO //2. Zadá počáteční adresu tabulky profilu V500. //3. Definuje, že přerušení 0 je přerušení // pro zpracování dokončených PTO. //4. Globální povolení přerušení. //5. Vymáá funkci PTO, PLS0 => Q0.0. <pre>LD SM0.0 MOVB 16#A8, SMB67 MOVW +500, SMW168 ATCH INT_0, 19 ENI PLS 0</pre>
		Network 1 //Když je profil výstupu PTO ukončen, //nastaví výstup Q0.5 na log. 1. <pre>LD SM0.0 = Q0.5</pre>

Instrukce matematiky

Sčítání, odčítání, násobení a dělení

Sčítání

IN1 + IN2 = OUT
IN1 + OUT = OUT

Odčítání

IN1 – IN2 = OUT
OUT – IN1 = OUT

LAD a FBD
STL

Instrukce "Sečíst integer" (+I) a "Odečíst integer" (–I) sečte nebo odečte dvě 16bitová celá čísla (integer) – výsledkem je 16bitová hodnota. Instrukce "Sečíst double integer" (+D) nebo "Odečíst double integer" (–D) přičte a odečte dvě 32bitová celá čísla (integer), výsledkem je 32bitová hodnota. Instrukce "Sečíst real" (+R) nebo "Odečíst real" (–R) přičte a odečte dvě 32bitová reálná čísla – výsledkem je 32bitová hodnota (reálné číslo).

Násobení

IN1 * IN2 = OUT
IN1 * OUT = OUT

Dělení

IN1 / IN2 = OUT
OUT / IN1 = OUT

LAD a FBD
STL

Instrukce "Násobit integer" (*I) a "Dělit integer" (/I) vynásobí a vydělí dvě 16bitová celá čísla (integer) – výsledkem je 16bitová hodnota. (Při dělení se neuchovává zbytek.) Instrukce "Násobit double integer" (*D) a "Dělit double integer" (/D) vynásobí a vydělí dvě 32bitová celá čísla (integer) – výsledkem je 32bitová hodnota. (Při dělení se neuchovává zbytek.) Instrukce "Násobit real" (*R) a "Dělit real" (/R) vynásobí a vydělí dvě 32bitová reálná čísla – výsledkem je 32bitová hodnota (reálné číslo).

Bity SM a ENO

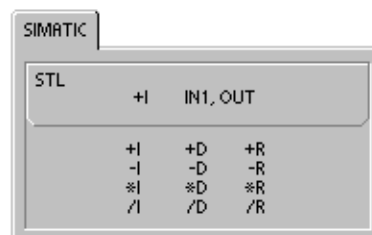
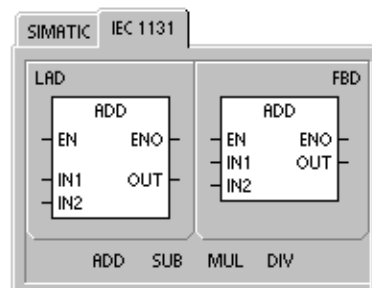
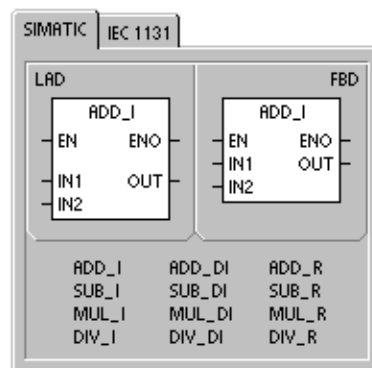
SM1.1 indikuje chyby přetečení a neplatné hodnoty. Pokud je nastaven SM1.1, je stav SM1.0 a SM1.2 neplatný a původní vstupní operandy se nemění. Pokud SM1.1 a SM1.3 nejsou nastaveny, skončila matematická operace s platným výsledkem a SM1.0 a SM1.2 obsahují platný stav. Pokud je SM1.3 nastaven během operace dělení, zůstávají ostatní stavové bity matematických operací nezměněny.

Chyby, které nastavují ENO = 0

- SM1.1 (přetečení)
- SM1.3 (dělení nulou)
- 0006 (nepřímá adresa)

Ovlivněné bity speciální paměti

- SM1.0 (nula)
- SM1.1 (přetečení, během operace generována neplatná hodnota nebo zjištěn neplatný vstupní parametr)
- SM1.2 (záporný výsledek)
- SM1.3 (dělení nulou)



Tabulka 6-37 Operandy platné pro instrukce sčítání, odčítání, násobení a dělení

Vstupy/Výstupy	Typy dat	Operandy
IN1, IN2	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, konstanta
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
OUT	INT	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, *VD, *AC, *LD
	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Reálná čísla (čísla s pohyblivou řádovou čárkou) jsou reprezentována ve formátu popsaném v normě ANSI/IEEE 754–1985 (jednoduchá přesnost). Další informace najdete v této normě.

Příklad: Matematika celých čísel

Network 1

```
graph LR
    I0_0[I0.0] --> EN_ADD[EN]
    I0_0 --> EN_MUL[EN]
    I0_0 --> EN_DIV[EN]
    EN_ADD --> OUT_ADD[OUT]
    EN_MUL --> OUT_MUL[OUT]
    EN_DIV --> OUT_DIV[OUT]
```

Network 1

```
LD I0.0
+I AC1, AC0
*I AC1, VW100
/I VW10, VW200
```

Sčítání

40	+	60	=	100
AC1		AC0		AC0

Násobení

40	*	20	=	800
AC1		VW100		VW100

Dělení

4000	/	40	=	100
VW200		VW10		VW200

Příklad: Matematika reálných čísel

Network 1

```
graph LR
    I0_0[I0.0] --> EN_ADD[EN]
    I0_0 --> EN_MUL[EN]
    I0_0 --> EN_DIV[EN]
    EN_ADD --> OUT_ADD[OUT]
    EN_MUL --> OUT_MUL[OUT]
    EN_DIV --> OUT_DIV[OUT]
```

Network 1

```
LD I0.0
+R AC1, AC0
*R AC1, VD100
/R VD10, VD200
```

Sčítání

4000.0	+	6000.0	=	10000.0
AC1		AC0		AC0

Násobení

400.0	*	200.0	=	80000.0
AC1		VD100		VD100

Dělení

4000.0	/	41.0	=	97.5609
VD200		VD10		VD200

141

Násobení celých čísel s 32bitovým výsledkem a dělení celých čísel se zbytkem

Násobení celých čísel s 32bitovým výsledkem

IN1 * IN2 = OUT LAD a FBD
IN1 * OUT = OUT STL

Instrukce "Násobit integer do double integer" (MUL) vynásobí dvě 16bitová celá čísla (integer) – výsledkem je 32bitová hodnota (double integer). V STL instrukci MUL se nejnižší platný word (16 bitů) z 32bitového OUT používá jako jeden z činitelů.

Dělení celých čísel se zbytkem

IN1 / IN2 = OUT LAD a FBD
OUT / IN1 = OUT STL

Instrukce "Dělit integer se zbytkem" (DIV) vydělí dvě 16bitová celá čísla (integer) s 32bitovým výsledkem (double integer), který se skládá z 16bitového zbytku (nejvyšší word) a 16bitového podílu (nejnižší word).

V STL se nejnižší platná hodnota word (16 bitů) z 32bitového OUT používá jako dělenec.

Bity SM a ENO

U obou instrukcí indikují na této stránce bity speciální paměti (SM) chyby a neplatné hodnoty. Pokud je během dělení nastavena SM1.3 (dělení nulou), zůstávají ostatní stavové bity matematiky nezměněny. Jinak všechny stavové bity matematiky obsahují po dokončení matematické operace platný stav.

Chyby, které nastavují ENO = 0

- SM1.1 (přetečení)
- SM1.3 (dělení nulou)
- 0006 (nepřímá adresa)

Ovlivněné bity speciální paměti

- SM1.0 (nula)
- SM1.1 (přetečení)
- SM1.2 (záporný výsledek)
- SM1.3 (dělení nulou)

Tabulka 6-38 Operandy platné pro násobení celých čísel s 32bitovým výsledkem a dělení celých čísel se zbytkem

Vstupy/Výstupy	Typy dat	Operandy
IN1, IN2	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
OUT	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Příklad: Instrukce násobení celých čísel s 32bitovým výsledkem a instrukce dělení celých čísel se zbytkem

Network 1	Network 1 <pre>LD I0.0 MUL AC1, VD100 DIV VW10, VD200</pre> Násobení integer s 32bitovým výsledkem 400 * 200 = 80000 AC1 VW102 VD100 Dělení integer se zbytkem 4000 / 41 = 23 97 VW202 VW10 zbyt. podíl VW200 VW202 VD200

Poznámka: VD100 obsahuje VW100 a VW102; VD200 obsahuje VW200 a VW202.

Numerické funkce

Sinus, kosinus a tangens

Instrukce "Sinus" (SIN), "Cosinus" (COS) a "Tangens" (TAN) vypočítají trigonometrickou funkci úhlu daného v IN a výsledek zapíší do OUT. Hodnota vstupního úhlu je v radiánech.

$$\text{SIN (IN)} = \text{OUT} \quad \text{COS (IN)} = \text{OUT} \quad \text{TAN (IN)} = \text{OUT}$$

Chcete –li převést úhel ze stupňů na radiány: Použijte instrukci MUL_R (*R) pro vynásobení úhlu ve stupních koeficientem 1.745329E-2 (přibližně $\pi/180$).

Přirozený logaritmus a přirozený exponent

Instrukce "Přirozený logaritmus" (LN) provede přirozený logaritmus z hodnoty IN a výsledek uloží do OUT.

Instrukce "Přirozený exponent" (EXP) umocní e na exponent uvedený v parametru IN a výsledek zapíše do OUT.

$$\text{LN (IN)} = \text{OUT} \quad \text{EXP (IN)} = \text{OUT}$$

Chcete –li získat z přirozených logaritmů dekadické logaritmy: Vydělte přirozený logaritmus 2,302585 (přibližně přirozený logaritmus 10).

Chcete –li vypočítat libovolnou reálnou mocninu reálného čísla včetně zlomkových exponentů: Zkombinujte instrukci "Přirozený exponent" s instrukcí "Přirozený logaritmus". Chcete –li například umocnit X na Y, vložte následující instrukci: EXP (Y * LN (X)).

Druhá odmocnina

Instrukce "Druhá odmocnina" (SQRT) vypočítá druhou odmocninu z reálného čísla (IN) a uloží ji do reálného výsledku v OUT.

$$\text{SQRT (IN)} = \text{OUT}$$

Chcete –li vypočítat jiné odmocniny: 5 na třetí = $5^3 = \text{EXP}(3 * \text{LN}(5)) = 125$
 Třetí odmocnina 125 = $125^{(1/3)} = \text{EXP}((1/3) * \text{LN}(125)) = 5$
 Druhá odmocnina z 5 na třetí = $5^{(3/2)} = \text{EXP}(3/2 * \text{LN}(5)) = 11,18034$

Bity SM a ENO pro instrukce numerických funkcí

U všech instrukcí popsanych na této stránce je SM1.1 použit pro indikaci chyb přetečení a neplatných hodnot. Pokud je nastaven SM1.1, je stav SM1.0 a SM1.2 neplatný a původní vstupní operandy se nemění. Pokud SM1.1 není nastaven, skončila matematická operace s platným výsledkem a SM1.0 a SM1.2 obsahují platný stav.

Chyby, které nastavují ENO = 0

- SM1.1 (přetečení)
- 0006 (nepřímá adresa)

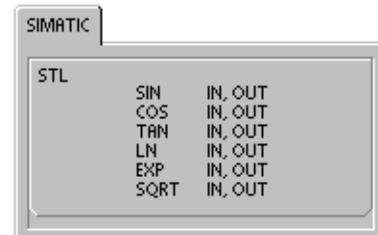
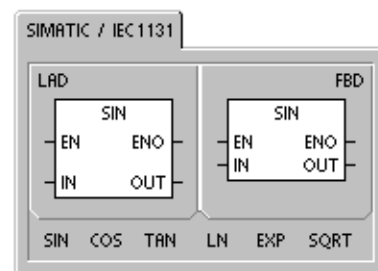
Ovlivněné bity speciální paměti

- SM1.0 (nula)
- SM1.1 (přetečení)
- SM1.2 (záporný výsledek)

Tabulka 6-39 Operandy platné pro numerické funkce

Vstupy/Výstupy	Typy dat	Operandy
IN	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
OUT	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Reálná čísla (čísla s pohyblivou řádovou čárkou) jsou reprezentována ve formátu popsáném v normě ANSI/IEEE 754–1985 (jednoduchá přesnost). Další informace najdete v této normě.



Tabulka 6-40 Operandy platné pro instrukce "Inkrementace" a "Dekrementace"

PID algoritmus

Instrukce "PID algoritmus" (PID) provede výpočet PID smyčky na základě vstupu a konfigurace dané v tabulce (TBL).

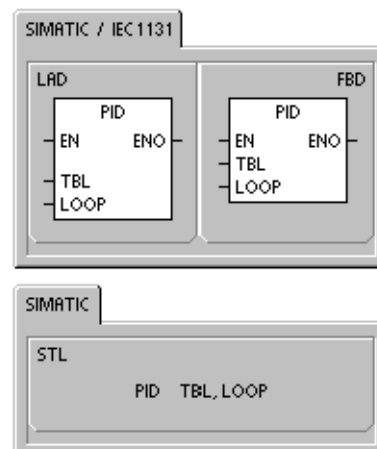
Chyby, které nastavují ENO = 0:

- SM1.1 (přetečení)
- 0006 (nepřímá adresa)

Ovlivněné bity speciální paměti:

- SM1.1 (přetečení)

Instrukce "PID algoritmus" (PID = proporcionální, integrační a derivační regulátor) se používá pro provedení výpočtu PID. Aby mohl být výpočet PID proveden, musí být vrchol zásobníku (TOS) ve stavu log. 1 (signálový tok). Instrukce má dva operandy: TABLE – adresa, kde začíná tabulka algoritmu, LOOP – konstanta, číslo od 0 do 7.



V programu může být použito až osmi instrukcí PID. Pokud mají dvě nebo více instrukcí PID stejné číslo (i kdyby měly různé adresy tabulek), budou se výpočty PID vzájemně rušit a výstup bude nepředvídatelný.

V tabulce algoritmu je uloženo devět parametrů používaných pro řízení a monitorování činnosti algoritmu. Tabulka obsahuje aktuální a předcházející hodnoty regulované veličiny, požadovanou hodnotu výstupu, zesílení, vzorkovací periodu, integrační časovou konstantu (reset), derivační časovou konstantu (rate) a integrační sumu (bias).

Aby byl PID výpočet proveden s požadovanou vzorkovací periodou, musí být instrukce PID vykonána buď v přerušení, nebo v hlavním programu, kde musí být rychlost řízení časovačem. Vzorkovací perioda musí být zadána jako vstup smyčky PID pomocí tabulky algoritmu.

Tabulka 6-41 Operandy platné pro Instrukci "PID algoritmus"

Vstupy/Výstupy	Typy dat	Operandy
TBL	BYTE	VB
LOOP	BYTE	Konstanta (0 až 7)



Průvodce

STEP 7-Micro/WIN nabízí průvodce, který vás provede definováním algoritmu PID pro řídicí proces s uzavřenou řídicí smyčkou. Vyberte příkaz menu **Nástroje > Průvodce instrukcemi** a pak vyberte **PID** z okna průvodce.

Vysvětlení algoritmu PID

V ustáleném stavu provozu řídí regulátor PID hodnotu výstupu tak, aby odchylka (e) byla od požadované hodnoty nulová. Míra odchylky je dána rozdílem mezi požadovanou hodnotou (SP) a regulovanou veličinou (PV). Princip řízení PID je založen na následující rovnici, která vyjadřuje výstup M(t) jako funkci proporcionálního, integračního a derivačního členu:

Výstup	=	Proporcionální člen	+	Integrační člen	+	Derivační člen
M(t)	=	K_C * e	+	K_C ∫₀^t e dt + M_{initial}	+	K_C * de/dt
kde:	M(t)	je výstup algoritmu jako funkce času				
	K _C	je zesílení				
	e	je regulační odchylka (rozdíl mezi požadovanou hodnotou a regulovanou veličinou)				
	M _{initial}	je počáteční hodnota výstupu algoritmu				

Aby bylo možné implementovat tuto řídicí funkci v digitálním počítači, musí být původní spojitá funkce transformována na periodické vzorky regulační odchylky s následným přepočtem výstupu. Příslušná rovnice, která je základem pro implementaci regulačního algoritmu na digitálním počítači, je:

M_n	=	K_C * e_n	+	K_I * ∑₁ⁿ + M_{initial}	+	K_D * (e_n-e_{n-1})
výstup	=	proporcionální člen	+	integrační člen	+	derivační člen
kde:	M _n	je vypočítaná hodnota výstupu algoritmu ve vzorkovací periodě n				
	K _C	je zesílení				
	e _n	je hodnota regulační odchylky ve vzorkovací periodě n				
	e _{n-1}	je předcházející hodnota regulační odchylky (ve vzorkovací periodě n - 1)				
	K _I	je proporcionální konstanta integračního členu				
	M _{initial}	je počáteční hodnota výstupu				
	K _D	je proporcionální konstanta derivačního členu				

Z rovnice vidíme, že integrační člen je funkcí všech odchylek od prvního po aktuální vzorek. Diferenční člen je funkcí aktuálního a předcházejícího vzorku, zatímco proporcionální člen je funkcí pouze aktuálního vzorku. Není praktické ani potřebné uchovávat v digitálním počítači všechny vzorky regulační odchylky.

Protože počítač musí počítat výstupní hodnotu při každém vzorkování odchylky počínaje prvním vzorkem, je nutné uložit pouze předcházející hodnotu odchylky a předcházející hodnotu integračního členu. V důsledku repetitivní povahy řešení je možné provést zjednodušení rovnice, která je řešena v každém vzorkovacím cyklu. Zjednodušená rovnice vypadá takto:

M_n	=	K_C * e_n	+	K_I * e_n + MX	+	K_D * (e_n-e_{n-1})
výstup	=	proporcionální člen	+	integrační člen	+	derivační člen
kde:	M _n	je vypočítaná hodnota výstupu algoritmu ve vzorkovací periodě n				
	K _C	je zesílení				
	e _n	je hodnota regulační odchylky ve vzorkovací periodě n				
	e _{n-1}	je předcházející hodnota regulační odchylky (ve vzorkovací periodě n - 1)				
	K _I	je proporcionální konstanta integračního členu				
	MX	je předcházející hodnota integračního členu (ve vzorkovací periodě n - 1)				
	K _D	je proporcionální konstanta derivačního členu				

Programovatelný automat S7-200 používá pro výpočet výstupní hodnoty smyčky modifikovanou formu výše uvedenou zjednodušenou rovnice. Modifikovaná rovnice je:

M_n	=	MP_n	+	MI_n	+	MD_n
výstup	=	proporcionální člen	+	integrační člen	+	derivační člen
kde:	M_n	je vypočítaná hodnota výstupu algoritmu ve vzorkovací periodě n				
	MP_n	je hodnota proporcionálního členu výstupu algoritmu ve vzorkovací periodě n				
	MI_n	je hodnota integračního členu výstupu algoritmu ve vzorkovací periodě n				
	MD_n	je hodnota derivačního členu výstupu algoritmu ve vzorkovací periodě n				

Vysvětlení proporcionálního členu rovnice PID

Proporcionální člen MP je součinem zesílení (K_C), které řídí citlivost výstupního výpočtu, a odchylky (e), která představuje rozdíl mezi požadovanou hodnotou (SP) a regulovanou veličinou (PV) v dané vzorkovací periodě. Rovnice pro proporcionální člen, jak ji řeší S7-200, zní:

MP_n	=	K_C	*	$(SP_n - PV_n)$
kde:	MP_n	je hodnota proporcionálního členu výstupu algoritmu ve vzorkovací periodě n		
	K_C	je zesílení		
	SP_n	je požadovaná hodnota ve vzorkovací periodě n		
	PV_n	je hodnota regulované veličiny ve vzorkovací periodě n		

Vysvětlení integračního členu rovnice PID

Integrační člen MI je přímo úměrný součtu regulačních odchylek v průběhu času. Rovnice pro integrační člen, jak ji řeší S7-200, zní:

MI_n	=	K_C	*	T_S	/	T_I	*	$(SP_n - PV_n)$	+	MX
kde:	MI_n	je hodnota integračního členu výstupu algoritmu ve vzorkovací periodě n								
	K_C	je zesílení								
	T_S	je vzorkovací perioda								
	T_I	je integrační konstanta (také nazývaná integrační časová konstanta nebo reset)								
	SP_n	je požadovaná hodnota ve vzorkovací periodě n								
	PV_n	je hodnota regulované veličiny ve vzorkovací periodě n								
	MX	je hodnota integračního členu ve vzorkovací periodě n – 1 (nazývaný také integrační suma)								

Integrační suma (MX) je průběžný součet všech předcházejících hodnot integračního členu. Po každém výpočtu MI_n je integrační suma aktualizována hodnotou MI_n , která může být přizpůsobena nebo omezena (podrobnosti viz odstavec "Proměnné a rozsahy"). Počáteční hodnota integračního součtu je nastavena na $M_{initial}$ těsně před výpočtem prvního výstupu algoritmu. Součástí integračního členu je také několik konstant, zesílení (K_C), vzorkovací perioda (T_S), což je perioda, po které PID algoritmus opět počítá výstupní hodnotu, a integrační časová konstanta neboli reset (T_I), což je čas použitý pro ovlivnění výpočtu výstupu integračním členem.

Vysvětlení derivačního členu rovnice PID

Derivační člen MD je přímo úměrný změně odchylky. Programovatelný automat S7-200 používá pro derivační člen následující rovnici:

$$MD_n = K_C * T_D / T_S * ((SP_n - PV_n) - (SP_{n-1} - PV_{n-1}))$$

Aby se zabránilo skokovým změnám a nárazům na výstupu způsobeným derivačním charakterem změny požadované hodnoty, je tato rovnice upravena na základě předpokladu, že požadovaná hodnota je konstanta ($SP_n = SP_{n-1}$). Výsledkem je výpočet změny regulované veličiny namísto změny odchylky, jak je patrné z rovnice:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_n + PV_{n-1})$$

nebo jen:

$$MD_n = K_C * T_D / T_S * (PV_{n-1} - PV_n)$$

kde:

MD_n	je hodnota derivačního členu výstupu algoritmu ve vzorkovací periodě n
K_C	je zesílení
T_S	je vzorkovací perioda
T_D	je derivační konstanta (také nazývaná derivační časová konstanta nebo rate)
SP_n	je požadovaná hodnota ve vzorkovací periodě n
SP_{n-1}	je požadovaná hodnota ve vzorkovací periodě n-1
PV_n	je hodnota regulované veličiny ve vzorkovací periodě n
PV_{n-1}	je hodnota regulované veličiny ve vzorkovací periodě n-1

Pro další výpočet derivačního členu je třeba uchovat spíše regulovanou veličinu než odchylku. V době prvního vzorku je hodnota PV_{n-1} inicializována, aby se rovnala PV_n .

Výběr typu řídicího algoritmu

V mnoha aplikacích může být potřeba použít pouze jednu nebo dvě složky řídicího algoritmu. Může být například požadováno pouze proporcionální řízení nebo proporcionální a integrační řízení. Výběr požadovaného typu řízení se provádí nastavením hodnoty konstantních parametrů.

Pokud chcete vynechat integrační složku (PD regulátor), musí být pro integrační časovou konstantu (reset) zadáno jako hodnota nekonečno. I bez integrace nemusí být hodnota integračního členu nulová díky původní hodnotě integračního součtu MX.

Pokud chcete vynechat derivační složku (PI regulátor), musí být pro derivační časovou konstantu (rate) zadána hodnota 0,0.

Pokud chcete vynechat proporcionální složku a chcete řízení I nebo ID, musí být pro zesílení zadána hodnota 0,0. Protože je v rovnicích zesílení smyčky faktorem pro výpočet integračního a derivačního členu, bude mít ve výpočtech integračních a derivačních členů nastavení hodnoty 0,0 pro zesílení smyčky za důsledek použití hodnoty 1,0.

Konverze a normalizace vstupů algoritmu

Algoritmus má dvě vstupní proměnné, požadovanou hodnotu a regulovanou veličinu. Požadovaná hodnota je zpravidla konstantní hodnota, jako je například nastavení rychlosti pohybu. Regulovaná veličina je hodnota, která je svázána s výstupem algoritmu, a měří proto vliv výstupu algoritmu na řízený systém. Ve výše uvedeném příkladu nastavení rychlosti by regulovanou veličinou byl vstup do tachodynamu, které měří rychlost kol zařízení.

Požadovaná hodnota i regulovaná veličina jsou hodnoty reálného světa, jejichž velikost, rozsah a jednotky mohou být různé. Před zpracováním těchto hodnot reálného světa pomocí instrukce PID je třeba převést je na normalizovaná čísla s pohyblivou řádovou čárkou.

Prvním krokem je převést hodnotu reálného světa z 16bitového celého čísla (integer) na číslo s pohyblivou řádovou čárkou neboli reálné číslo. Konverze integer na reálné číslo se provádí podle následující sekvence instrukcí.

ITD	AIW0, AC0	//Konverze vstupní hodnoty na hodnotu double word
DTR	AC0, AC0	//Konverze 32bitového celého čísla (double integer) na číslo reálné

Dalším krokem je konverze hodnoty reálného světa reprezentované reálným číslem na normalizovanou hodnotu z intervalu 0,0 až 1,0. Pro normalizaci požadované hodnoty nebo regulované veličiny se používá následující rovnice:

R_{Norm}	=	((R_{Raw} / Span) + Offset)
kde:	R _{Norm}	je normalizovaná hodnota reálného čísla reprezentujícího hodnotu reálného světa
	R _{Raw}	je nenormalizovaná neboli nezpracovaná hodnota reálného čísla reprezentujícího hodnotu reálného světa
	Offset	je 0,0 pro unipolární hodnoty je 0,5 pro bipolární hodnoty
	Span	je rozdíl maximální a minimální možné hodnoty: = 32.000 pro unipolární hodnoty (obvykle) = 64.000 pro bipolární hodnoty (obvykle)

Následující sekvence instrukcí ukazuje, jak normalizovat bipolární hodnotu v AC0 (s rozsahem 64.000) jako pokračování předcházející sekvence instrukcí:

/R	64000.0, AC0	//Normalizace hodnoty v akumulátoru
+R	0.5, AC0	//Offset hodnoty do rozsahu 0,0 až 1,0
MOVR	AC0, VD100	//Uložení normalizované hodnoty do tabulky algoritmu

Konverze výstupu na integer s upraveným měřítkem

Výstup algoritmu je řídicí proměnná, například v našem příkladu o řízení rychlosti automobilu je to nastavení průtoku paliva. Výstup je normalizované reálné číslo mezi 0,0 a 1,0. Před použitím výstupu algoritmu na řízení analogového výstupu je nutné ho převést na 16bitový integer s upraveným měřítkem. Tento proces je opakem konverze PV a SP na normalizovanou hodnotu. Prvním krokem je převedení výstupu na reálné číslo s upraveným měřítkem pomocí následujícího vzorce:

R_{Scal}	=	(M_n – Offset) * Span
wzde: R _{Scal}		je reálná hodnota s upraveným měřítkem výstupu smyčky
M _n		je normalizovaná reálná hodnota výstupu smyčky
Offset		je 0,0 pro unipolární hodnoty je 0,5 pro bipolární hodnoty
Span		je rozdíl maximální a minimální možné hodnoty = 32.000 pro unipolární hodnoty (obvykle) = 64.000 pro bipolární hodnoty (obvykle)

Následující sekvence instrukcí ukazuje, jak upravit měřítko výstupu algoritmu:

MOVR	VD108, AC0	//Uloží výstup do akumulátoru
–R	0.5, AC0	//Tuto instrukci provést pouze v případě bipolární hodnoty
*R	64000.0, AC0	//Upraví měřítko hodnoty v akumulátoru

Ted' je třeba převést reálné číslo s upraveným měřítkem, které představuje výstup smyčky, na 16bitový integer. Tato konverze se uskuteční pomocí následující sekvence instrukcí:

ROUND	AC0, AC0	//Konverze reálného čísla na 32bitové celé číslo
DTI	AC0, LW0	//Konverze hodnoty na 16bitový integer.
MOVW	LW0, AQW0	//Zápis hodnoty na analogový výstup

Přímá nebo reverzní smyčka

Smyčka je přímá, je –li zesílení kladné, a reverzní, je –li záporné. (U řízení I a ID, kde je hodnota zesílení 0,0, způsobí zadání kladné hodnoty pro integrační a derivační časovou konstantu přímou činnost smyčky, zadání záporné hodnoty způsobí reverzní činnost smyčky.)

Proměnné a rozsahy

Regulovaná veličina a požadovaná hodnota jsou vstupy do PID výpočtu. Proto může instrukce PID odpovídající políčka tabulky číst, ale nikoliv měnit.

Výstupní hodnota je generována výpočtem PID, proto je políčko výstupu v tabulce smyčky aktualizováno po dokončení každého výpočtu PID. Výstupní hodnota leží mezi 0,0 a 1,0. Políčko výstupní hodnoty může být použito jako vstup pro zadání počáteční hodnoty výstupu uživatelem při přechodu od ručního řízení výstupu na řízení výstupu pomocí instrukce PID (auto). (Další informace najdete v odstavci "Režimy".)

Pokud se používá integrační řízení, je hodnota integračního součtu aktualizována výpočtem PID; aktualizovaná hodnota je použita jako vstup pro další výpočet PID. Jestliže je vypočítaná výstupní hodnota mimo rozsah (výstup by byl menší než 0,0 nebo větší než 1,0), upraví se integrační součet podle následujících vzorců:

MX	=	1.0	– (MP_n + MD_n)	<i>když je vypočítaný výstup M_n > 1.0</i>
nebo				
MX	=	– (MP_n + MD_n)		<i>když je vypočítaný výstup M_n < 0,0</i>
wzde: MX			je hodnota upraveného integračního součtu	
MP _n			je hodnota proporcionálního členu výstupu algoritmu ve vzorkovací periodě n	
MD _n			je hodnota derivačního členu výstupu algoritmu ve vzorkovací periodě n	
M _n			je hodnota výstupu algoritmu ve vzorkovací periodě n	

Upravením integračního součtu výše uvedeným způsobem se dosáhne zlepšení citlivosti systému, jakmile se vypočítaný výstup vrátí do správného rozsahu. Vypočítaný integrační součet se rovněž nachází v rozsahu 0,0 až 1,0; po dokončení každého výpočtu PID se запиše do políčka pro integrační součet v tabulce algoritmu. Hodnota uložená v tabulce algoritmu je použita v příštím výpočtu PID.

Před provedením instrukce PID může uživatel modifikovat hodnotu integrační sumy v tabulce algoritmu, aby v určitých situacích vyřešil problémy s hodnotou integračního součtu. Při ruční úpravě integrační sumy se musí postupovat opatrně; jakákoliv hodnota integrační sumy zapsaná do tabulky algoritmu musí být reálné číslo v rozmezí 0,0 a 1,0.

Pro derivační část výpočtu PID je v tabulce uložena porovnávací hodnota regulované veličiny. Tato hodnota by neměla být modifikována uživatelem.

Režimy

PID algoritmy programovatelného automatu S7-200 nemají žádné zabudované řízení režimu. Výpočet PID se provádí pouze při aktivním signálovém toku do bloku PID. Automatický režim neboli "auto" znamená cyklické provádění výpočtu PID. "Manuální" režim znamená, když se výpočet PID neprovádí.

Instrukce PID má bit historie signálového toku, obdobně jako instrukce čítače. Tento bit používá instrukce pro detekci přechodu signálového toku z 0 na 1. Při detekci takového přechodu instrukce provede řadu činností, aby zabezpečila beznárazovou změnu z manuálního řízení na automatické. Aby byla změna na automatický režim řízení beznárazová, musí být hodnota výstupu (nastavená při manuálním řízení) nastavena i jako vstup pro instrukci PID (zapsána do položky tabulky smyčky pro M_n) dříve, než se přejde na automatické řízení. Instrukce PID provede s hodnotami v tabulce algoritmu následující činnosti, aby zajistila bezproblémovou změnu z manuálního na automatické řízení při přechodu signálového toku z 0 na 1:

- ☐ Požadovaná hodnota (SP_n) = regulovaná veličina (PV_n)
- ☐ Předchozí regulovaná veličina (PV_{n-1}) = regulovaná veličina (PV_n)
- ☐ Integrační suma (MX) = výstupní hodnota (M_n)

Implicitní nastavení bitů historie PID je hodnota log. 1. Tento stav je při spuštění i při každém přechodu programovatelného automatu z režimu STOP do režimu RUN. Při prvním provádění smyčky PID po přepnutí do režimu RUN se při aktivním signálovém toku nespustí činnost beznárazového přechodu.

Kontrola poruchových signálů a speciální funkce

Instrukce PID je jednoduchý, ale účinný nástroj pro provádění výpočtů PID. Pokud se požadují další funkce, jako je například kontrola poruchových signálů nebo proměnných smyčky, je nutné je implementovat pomocí základních instrukcí podporovaných automatem S7-200.

Chyby

Pokud jsou počáteční adresa tabulky algoritmu nebo operandy PID zadány v instrukci mimo rozsah, CPU ohlásí při kompilaci kompilační chybu (chybu rozsahu) a kompilace se zastaví.

Některé vstupní hodnoty v tabulce nejsou instrukcí PID kontrolovány, zda se nacházejí v povoleném rozsahu. Je třeba věnovat velkou pozornost tomu, aby regulovaná veličina a požadovaná hodnota (stejně jako integrační součet a předcházející regulovaná veličina, pokud jsou použity jako vstup) byla reálná čísla v rozsahu mezi 0,0 a 1,0.

Pokud se v době provádění matematických operací výpočtu PID vyskytne jakákoliv chyba, nastaví se SM1.1 (přetečení nebo neplatná hodnota) a provádění instrukce PID se ukončí. (Aktualizace výstupních hodnot v tabulce algoritmu může být neúplná, proto tyto hodnoty neberte v úvahu a před příštím provedením instrukce PID opravte vstupní hodnotu, která matematickou chybu způsobila.)

Tabulka algoritmu

Tabulka má velikost 36 bytů a formát podle tabulky 6-42.

Tabulka 6-42 Tabulka smyčky

Posun utí	Pole	Formát	Typ	Popis
0	Regulovaná veličina (PV_n)	Double word – REAL	vstup	Obsahuje regulovanou veličinu, která musí být v rozsahu mezi 0,0 a 1,0.
4	Požadovaná hodnota (SP_n)	Double word – REAL	vstup	Obsahuje požadovanou hodnotu, která musí být v rozsahu mezi 0,0 a 1,0.
8	Výstup (M_n)	Double word – REAL	vstup/výstup	Obsahuje vypočítanou výstupní hodnotu, která musí být v rozsahu mezi 0,0 a 1,0.
12	Zesílení (K_C)	Double word – REAL	vstup	Obsahuje zesílení, které je proporcionální konstanta. Může to být kladné nebo záporné číslo.
16	Vzorkovací perioda (T_S)	Double word – REAL	vstup	Obsahuje vzorkovací periodu v sekundách. Musí to být kladné číslo.
20	Integrační časová konstanta (T_I)	Double word – REAL	vstup	Obsahuje integrační časovou konstantu v minutách. Musí to být kladné číslo.
24	Derivační časová konstanta (T_D)	Double word – REAL	vstup	Obsahuje derivační časovou konstantu v minutách. Musí to být kladné číslo.
28	Integrační suma (MX)	Double word – REAL	vstup/výstup	Obsahuje hodnotu integrační sumy v rozsahu mezi 0,0 a 1,0.
32	Předcházející regulovaná veličina (PV_{n-1})	Double word – REAL	vstup/výstup	Obsahuje předcházející hodnotu regulované veličiny uloženou od posledního provedení instrukce PID.

Příklad programu smyčky PID

V tomto příkladu je vodní nádrž použita na udržení konstantního tlaku vody. Voda se z nádrže odebírá nepřetržitě s proměnlivou rychlostí. Na doplňování vody do nádrže se používá čerpadlo s proměnnou rychlostí, které má zajistit udržení příslušného tlaku v nádrži a zabránit jejímu vyprázdnění.

Požadovaná hodnota tohoto systému je výška hladiny, která odpovídá 75% naplnění nádrže. Regulovaná veličina je snímána plovákovým snímačem, který poskytuje informace, jak je nádrž plná v rozmezí od 0 % (prázdná) do 100 % (plná). Výstup je hodnota rychlosti čerpadla, která se pohybuje v rozmezí od 0 % do 100 % maximální rychlosti.

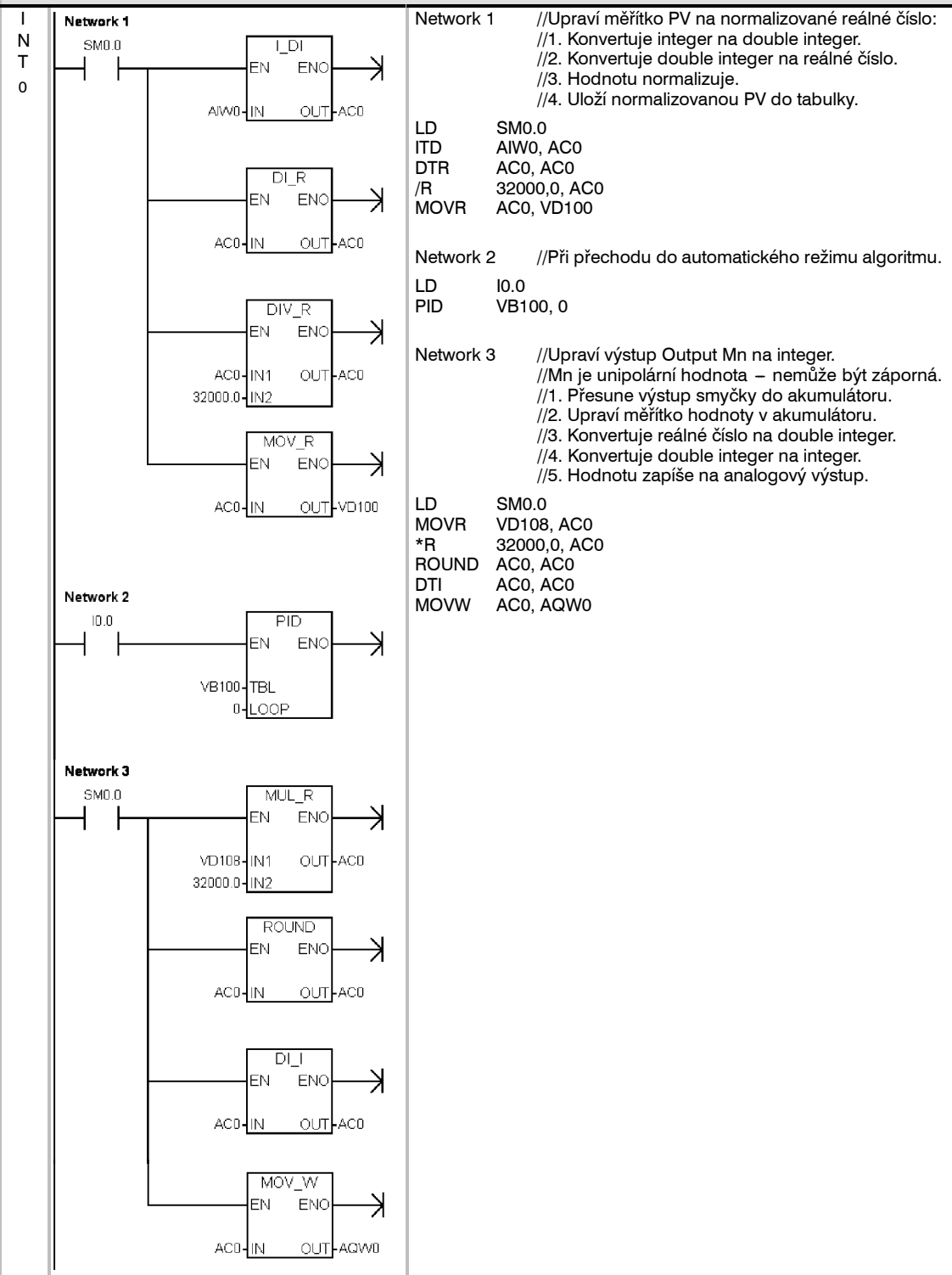
Požadovaná hodnota je předdefinovaná a zadává se přímo do tabulky algoritmu. Regulovaná veličina je unipolární analogová hodnota plovákového snímače. Výstup algoritmu se zapisuje na unipolární analogový výstup, který je použit pro řízení rychlosti čerpadla. Rozsah analogového vstupu i analogového výstupu je 32.000.

V příkladu se používá jen řízení s proporcionální a integrační složkou. Zesílení a časové konstanty byly stanoveny na základě technických výpočtů a pro dosažení optimálního řízení mohou být upraveny. Vypočtené hodnoty časových konstant jsou: $K_C = 0,25$, $T_S = 0,1$ sekundy a $T_I = 30$ minut.

Rychlost čerpadla se řídí manuálně až do naplnění vodní nádrže na 75 %, potom se otevře ventil na vypouštění vody z nádrže. Zároveň se čerpadlo přepíná z manuálního na automatický režim. Pro přepnutí řízení z manuálního na automatické se používá digitální vstup. Vstup (I0.0) slouží pro manuální/automatické řízení: 0 = manuální a 1 = automatické. V manuálním režimu řízení zapisuje rychlost čerpadla operátor na VD108 jako reálné číslo v rozsahu 0,0 až 1,0.

Příklad: Algoritmus PID		
M A I N	Network 1	Network 1 //V prvním programovém cyklu //volá inicializační podprogram. LD SM0.1 CALL SBR_0
S B R 0	Network 1	Network 1 //Nahraje parametry PID a //přiřadí podprogram přerušení PID: //1. Nastaví požadovanou hodnotu = 75 % plné // nádrže. //2. Nastaví zesílení = 0,25. //3. Nastaví vzorkovací periodu smyčky = 0,1 // sekundy. //4. Nastaví integrační konstantu = 30 min. //5. Vyřadí derivační složku. //6. Nastaví časový interval (100 ms) // pro časové přerušení INT_0. //7. Nastaví časové přerušení pro spouštění // provádění PID. //8. Povolí přerušení LD SM0.0 MOVR 0.75, VD104 MOVR 0.25, VD112 MOVR 0.1, VD116 MOVR 30.0, VD120 MOVR 0.0, VD124 MOVB 100, SMB34 ATCH INT_0, 10 ENI

Příklad: Algoritmus PID, pokračování



Instrukce přerušení

Povolení a zakázání přerušení

Instrukce "Povolit přerušení" (ENI) globálně povoluje zpracování všech přiřazených událostí vyvolávajících přerušení. Instrukce "Zakázat přerušení" (DISI) globálně pozastaví zpracování všech událostí vyvolávajících přerušení.

Při přechodu do režimu RUN jsou zpočátku přerušení zakázána. Když už je CPU v režimu RUN, je možné povolit zpracování přerušení provedením instrukce "Povolit přerušení". Po provedení instrukce "Zakázat přerušení" se aktivní události pro přerušení zařadí do fronty, ale podprogramy přerušení se nevyvolají.

Chyby, které nastavují ENO = 0:

- 0004 (pokus o vykonání instrukcí ENI, DISI nebo HDEF v přerušení)

Podmíněný návrat z přerušení

Instrukci "Podmíněný návrat z přerušení" (CRETI) je možné použít pro návrat z přerušení na základě předchozího stavu programu.

Připojení přerušení

Instrukce "Připojit přerušení" (ATCH) spojuje událost vyvolávající přerušení EVNT s položkou podprogramu přerušení INT a povoluje událost přerušení.

Chyby, které nastavují ENO = 0:

- 0002 (konfliktní přiřazení vstupů k HSC)

Odpojení přerušení

Instrukce "Odpojit přerušení" (DTCH) zruší přiřazení události EVNT ode všech podprogramů přerušení a zakáže tuto událost.

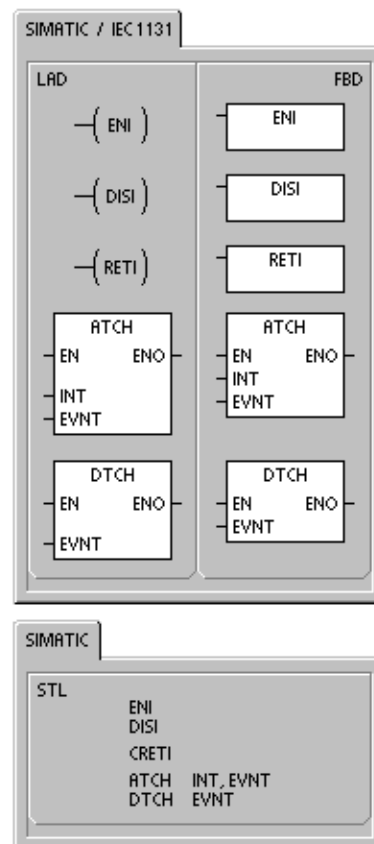
Tabulka 6-43 Operandy platné pro instrukce "Připojit přerušení" a "Odpojit přerušení"

Vstupy/Výstupy	Typy dat	Operandy
INT	BYTE	Konstanta (0 až 127)
EVNT	BYTE	Konstanta CPU 221 a CPU 222: 0 až 12, 19 až 23 a 27 až 33 CPU 224: 0 až 23 a 27 až 33 CPU 226 a CPU 226XM: 0 až 33

Vysvětlení instrukcí "Připojit přerušení" a "Odpojit přerušení"

Dříve, než může být vyvolán podprogram přerušení, se musí vytvořit přiřazení mezi událostí a segmentem programu, který je třeba vykonat po výskytu události. Pro přiřazení události (specifikované jejím číslem) k programovému segmentu (specifikovanému číslem podprogramu přerušení) použijte instrukci "Připojit přerušení". Je možné přiřadit více událostí k jednomu podprogramu přerušení, ale jedna událost nemůže být současně přiřazena k více podprogramům přerušení.

Při přiřazení události k podprogramu přerušení je toto přerušení automaticky povoleno. Pokud zakážete všechna přerušení pomocí globální instrukce zakázání přerušení, zařadí se každá událost vyvolávající přerušení, která se objeví, do fronty do doby, než jsou přerušení opět povolena pomocí globální instrukce povolení přerušení nebo dokud není fronta přerušení přeplněna.



Jednotlivé události vyvolávající přerušení můžete zakázat zrušením spojení mezi přerušovací událostí a podprogramem přerušení pomocí instrukce "Odpojit přerušení". Tato instrukce vrací přerušení do neaktivního stavu. Tabulka 6-44 uvádí různé typy událostí.

Tabulka 6-44 Události vyvolávající přerušení

Událost	Popis	CPU 221 CPU 222	CPU 224	CPU 226 CPU 226XM
0	I0.0 Náběžná hrana	Y	Y	Y
1	I0.0 Sestupná hrana	Y	Y	Y
2	I0.1 Náběžná hrana	Y	Y	Y
3	I0.1 Sestupná hrana	Y	Y	Y
4	I0.2 Náběžná hrana	Y	Y	Y
5	I0.2 Sestupná hrana	Y	Y	Y
6	I0.3 Náběžná hrana	Y	Y	Y
7	I0.3 Sestupná hrana	Y	Y	Y
8	Port 0 Příjem znaku	Y	Y	Y
9	Port 0 Ukončení přenosu	Y	Y	Y
10	Časované přer. 0 SMB34	Y	Y	Y
11	Časované přer. 1 SMB35	Y	Y	Y
12	HSC0 CV=PV (aktuální hodnota = nastavená hodnota)	Y	Y	Y
13	HSC1 CV=PV (aktuální hodnota = nastavená hodnota)		Y	Y
14	HSC1 Změna směru		Y	Y
15	HSC1 Externí reset		Y	Y
16	HSC2 CV=PV (aktuální hodnota = nastavená hodnota)		Y	Y
17	HSC2 Změna směru		Y	Y
18	HSC2 Externí reset		Y	Y
19	PLS0 Přerušení od ukončení počítání impulzů PTO	Y	Y	Y
20	PLS1 Přerušení od ukončení počítání impulzů PTO	Y	Y	Y
21	Časovač T32 přerušení CT=PT	Y	Y	Y
22	Časovač T96 přerušení CT=PT	Y	Y	Y
23	Port 0 Ukončení příjmu zprávy	Y	Y	Y
24	Port 1 Ukončení příjmu zprávy			Y
25	Port 1 Příjem znaku			Y
26	Port 1 Ukončení přenosu			Y
27	HSC0 Změna směru	Y	Y	Y
28	HSC0 Externí reset	Y	Y	Y
29	HSC4 CV=PV (aktuální hodnota = nastavená hodnota)	Y	Y	Y
30	HSC4 Změna směru	Y	Y	Y
31	HSC4 Externí reset	Y	Y	Y
32	HSC3 CV=PV (aktuální hodnota = nastavená hodnota)	Y	Y	Y
33	HSC5 CV=PV (aktuální hodnota = nastavená hodnota)	Y	Y	Y

Vysvětlení, jak automat S7-200 zpracovává podprogramy přerušení

Podprogram přerušení se provádí jako odpověď na přiřazenou vnitřní (interní) nebo vnější (externí) událost. Po provedení poslední instrukce podprogramu přerušení se řízení vrátí do hlavního programu. Podprogram je možné opustit provedením instrukce podmíněného návratu z přerušení (CRETI). Tabulka 6-45 zdůrazňuje některé pokyny a omezení pro použití přerušovacích podprogramů v uživatelském programu.

Tabulka 6-45 Pokyny a omezení pro použití podprogramů přerušení

Pokyny
Zpracování přerušení poskytuje možnost rychlé reakce na speciální vnitřní nebo vnější události. Podprogramy přerušení byste měli optimalizovat pro provádění konkrétní úlohy a pak vrátit řízení do hlavního programu. Jestliže jsou podprogramy krátké a účelné, je jejich provedení rychlé a ostatní procesy nejsou dlouho zdržovány. Pokud tomu tak není, mohou nepředpokládané stavy způsobit mimořádné chování zařízení řízeného hlavním programem. Pro přerušení zcela jistě platí axiom "čím kratší, tím lepší".
Omezení
V podprogramu přerušení není možné použít instrukce "Odpojit přerušení" (DISI), "Připojit přerušení" (ENI), "Vysokorychlostní spínač" (HDEF) a "Konec" (END).

Systémová podpora pro přerušení

Protože mohou přerušení ovlivnit logiku kontaktů, cívek a akumulátorů, systém zálohuje a opět obnoví logický zásobník, registry akumulátoru a bity speciální paměti (SM), které indikují stav akumulátoru a instrukcí. Tím se předchází chybám v hlavním uživatelském programu způsobeným větvením do podprogramu přerušení.

Sdílení dat mezi hlavním programem a přerušením

Je možné sdílet data mezi hlavním programem a jedním nebo více podprogramy. Protože není možné předvídat, kdy může programovatelný automat S7-200 generovat přerušení, je žádoucí omezit počet proměnných, které jsou užívány v přerušení i v jiných místech programu. V případech, kdy je provádění instrukcí hlavního programu přerušeno událostmi vyvolávajícími přerušení, mohou nastat problémy s konzistencí sdílených dat způsobené činnostmi podprogramů přerušení. Pro zajištění toho, že podprogram přerušení používá pouze dočasnou paměť a nepřepisuje data používaná jinde v programu, používejte tabulku lokálních proměnných z přerušení.

Existuje mnoho programovacích technik, které můžete použít pro zajištění správného sdílení dat mezi hlavním programem a podprogramy přerušení. Tyto techniky buďto omezují způsob přístupu do sdílených míst paměti, nebo nepovolují přerušení sekvencí instrukcí, které používají sdílená místa paměti.

- ☐ Pro program STL, který sdílí jednu proměnnou: Pokud jsou sdílená data o velikosti jeden byte, word nebo double word a program je napsán v STL, je možné zajistit správný sdílený přístup uložením přechodných hodnot z operací se sdílenými daty pouze do nesdílených míst paměti nebo akumulátorů.
- ☐ Pro program LAD, který sdílí jednu proměnnou: Pokud jsou sdílená data o velikosti jeden byte, word nebo double word a program je napsán v LAD, pak je možné zajistit správný sdílený přístup stanovením konvence, že přístup ke sdíleným místům paměti bude prováděn pouze pomocí instrukcí "Přesun" (MOVB, MOVW, MOVD, MOVR). Zatímco mnoho instrukcí LAD se skládá z přerušitelných sekvencí instrukcí STL, instrukce "Přesun" se skládají pouze z jediné instrukce STL, jejíž provádění nemůže být ovlivněno událostmi vyvolávajícími přerušení.
- ☐ Pro program STL nebo LAD, který sdílí více proměnných: Pokud se sdílená data skládají z více souvisejících hodnot byte, word nebo double word, je možné pro řízení provádění podprogramu přerušení použít instrukce povolení/zakázání přerušení (DISI a ENI). V bodu hlavního programu, kde mají začít operace se sdílenými místy paměti, zakažte přerušení. Jakmile jsou dokončeny všechny činnosti, které se týkají sdílených míst, přerušení povolte. Během doby, kdy jsou přerušení zakázána, není možné provádět podprogramy přerušení, a nemohou tedy přistupovat ke sdíleným místům paměti; to však může mít za následek opožděnou odezvu na události vyvolávající přerušení.

Volání podprogramů z přerušení

Z přerušení je možné volat jednu vnořenou úroveň podprogramu. Akumulátory a logický zásobník jsou sdíleny přerušením a volaným podprogramem.

Typy přerušení podporované programovatelným automatem S7-200

S7-200 podporuje následující typy přerušení:

- ☐ Přerušení od komunikačního portu: S7-200 generuje události, které umožní uživatelskému programu řídit komunikační port.
- ☐ Přerušení od vstupů/výstupů: S7-200 generuje události pro různé změny stavu různých I/O. Tyto události umožňují uživatelskému programu reagovat na vysokorychlostní čítače, pulzní výstupy či náběžné nebo sestupné hrany na vstupech.
- ☐ Cyklická přerušení: S7-200 generuje události, které umožní uživatelskému programu reagovat v přesně daných intervalech.

Přerušení od komunikačního portu

Sériový komunikační port automatu S7-200 může být řízen uživatelským programem. Tento režim práce komunikačního portu se nazývá režim Freeport. V režimu Freeport definuje uživatelský program přenosovou rychlost v bodech, počet bitů na znak, paritu a protokol. Přerušení přijetí a vyslání zprávy usnadňují komunikaci řízenou programem. Více informací najdete v odstavcích, týkajících se instrukcí vysílání a příjmu.

Přerušení od vstupů/výstupů

Přerušení I/O zahrnují přerušení od náběžné/sestupné hrany, přerušení od vysokorychlostních čítačů a přerušení od sekvence pulzů na výstupu. Automat S7-200 může generovat přerušení při náběžné nebo sestupné hraně na vstupu (I0.0, I0.1, I0.2 nebo I0.3). Událost náběžné i sestupné hrany se dá zachytit pro každý z těchto vstupních bodů. Tyto náběžné/sestupné hrany se dají použít jako označení podmínky, které se musí věnovat okamžitá pozornost při výskytu události.

Přerušení od vysokorychlostního čítače umožňují reagovat na podmínky jako "aktuální hodnota dosáhla nastavené hodnoty", "změna směru čítání", která může odpovídat změně směru otáčení hřídele, nebo "externí vynulování čítače". Každá z těchto událostí vysokorychlostního čítače umožňuje reagovat v reálném čase na vysokorychlostní události, které není možné řídit při rychlostech programového cyklu programovatelného automatu.

Přerušení od sekvence pulzů na výstupu poskytují okamžitou informaci o dokončení výstupu předepsaného počtu pulzů. Typickým použitím výstupů se sekvencí pulzů je řízení krokového motoru.

Každé z výše uvedených přerušení můžete povolit přiřazením přerušovacího podprogramu k příslušné v/v události.

Časovaná přerušení

Časovaná přerušení zahrnují cyklické přerušení a přerušení od časovače T32/T96. Na základě cyklického přerušení můžete specifikovat cyklicky prováděné činnosti. Čas cyklu je nastaven v přírůstcích po 1 ms od 1 ms do 255 ms. Pro cyklické přerušení 0 musíte zapsat čas cyklu do SMB34 a pro cyklické přerušení 1 do SMB35.

Událost cyklického přerušení předá řízení příslušnému přerušovacímu podprogramu při každém ukončení intervalu časovače. Typickým použitím časovaných přerušení je řízení vzorkování analogových vstupů nebo provádění PID algoritmu v pravidelných intervalech.

Když přiřadíte podprogram přerušení události časovaného přerušení, je povoleno časované přerušení a začíná odpočítávání času. Během přiřazování systém uchovává hodnotu času cyklu, takže následující změny SMB34 a SMB35 periodu neovlivní. Chcete-li změnit periodu, musíte změnit hodnotu času a pak opět přiřadit podprogram přerušení k události časovaného přerušení. Když dojde k opětovnému přiřazení, funkce časovaného přerušení vymaže veškerý akumulovaný čas z předcházejícího přiřazení a odpočítávání času začne s novou hodnotou.

Pokud je to povoleno, běží časované přerušení trvale a provádí přiřazený podprogram přerušení při každém ukončení specifikovaného časového intervalu. Pokud opustíte režim RUN nebo odpojíte časované přerušení, je časované přerušení zablokováno. Pokud je provedena instrukce globálního zakázání přerušení, dochází k časovaným přerušením i nadále. Každý výskyt časovaného přerušení je zařazen do fronty (do doby, než jsou přerušení povolena nebo je fronta přeplněna).

Přerušení časovače T32/T96 umožňují včasnou odezvu na dokončení specifikovaného časového intervalu. Tato přerušení jsou podporována pouze pro zpožděné zapnutí (TON) a zpožděné vypnutí (TOF) časovače T32 a T96 s rozlišením 1 ms. Jinak se časovače T32 a T96 chovají standardně. Jakmile je přerušení povoleno, provede se přiřazený podprogram přerušení, pokud se aktuální hodnota aktivního časovače rovná nastavené hodnotě času během normální 1 ms aktualizace časovače prováděné automatem S7-200.

Priorita přerušení a jejich řazení do fronty

S7-200 obsluhuje přerušení na základě pravidla: kdo první přijde, je jako první obslužen, ale s přihlédnutím k prioritě. V kterémkoliv časovém bodě je vždy prováděn pouze jeden podprogram přerušení. Jakmile je provádění podprogramu jednou zahájeno, je vykonáván až do svého ukončení. Nemůže být přerušen jiným podprogramem přerušení, ani podprogramem s vyšší prioritou. Přerušení, která se objeví během zpracovávání jiného přerušení, jsou zařazena do fronty pro pozdější zpracování.

Tabulka 6-46 ukazuje tři fronty pro přerušení a maximální počty přerušení, která do nich mohou být založena.

Tabulka 6-46 Maximální počet položek ve frontě přerušení

Fronta	CPU 221, CPU 222, CPU 224	CPU 226 a CPU 226XM
Komunikační fronta	4	8
Fronta přerušení od vstupů/výstupů	16	16
Fronta časovaných přerušení	8	8

Potenciálně může dojít k více přerušením, než je fronta schopna pojmout. Proto má systém paměťové bity pro přeplnění fronty (s identifikací událostí vyvolávající přerušení, které byly ztraceny). Tabulka 6-47 ukazuje bity pro přeplnění fronty přerušení. Tyto bity byste měli používat pouze v podprogramu přerušení, protože jsou při vyprázdnění fronty vynulovány a řízení se vrací do hlavního programu.

Tabulka 6-48 uvádí všechny události vyvolávající přerušení s jejich prioritou a přiřazeným číslem události.

Tabulka 6-47 Bity pro přeplnění fronty přerušení

Popis (0 = bez přeplnění, 1 = přeplnění)	SM Bit
Komunikační fronta	SM4.0
Fronta přerušení od vstupů/výstupů	SM4.1
Fronta časovaných přerušení	SM4.2

Tabulka 6-48 Pořadí priority událostí vyvolávající přerušení

Událost	Popis	Prioritní skupina	Priorita ve skupině
8	Port 0 Příjem znaku	Komunikace <i>Nejvyšší priorita</i>	0
9	Port 0 Ukončení přenosu		0
23	Port 0 Ukončení příjmu zprávy		0
24	Port 1 Ukončení příjmu zprávy		1
25	Port 1 Příjem znaku		1
26	Port 1 Ukončení přenosu		1
19	PLS0 Přerušení od ukončení počítání pulzů PTO	Diskrétní <i>Střední priorita</i>	0
20	PLS1 Přerušení od ukončení počítání pulzů PTO		1
0	I0.0 Náběžná hrana		2
2	I0.1 Náběžná hrana		3
4	I0.2 Náběžná hrana		4
6	I0.3 Náběžná hrana		5
1	I0.0 Sestupná hrana		6
3	I0.1 Sestupná hrana		7
5	I0.2 Sestupná hrana		8
7	I0.3 Sestupná hrana		9
12	HSC0 CV=PV (aktuální hodnota = nastavená hodnota)		10
27	HSC0 Změna směru		11
28	HSC0 Externí reset		12
13	HSC1 CV=PV (aktuální hodnota = nastavená hodnota)		13
14	HSC1 Změna směru		14
15	HSC1 Externí reset		15
16	HSC2 CV=PV (aktuální hodnota = nastavená hodnota)		16
17	HSC2 Změna směru		17
18	HSC2 Externí reset		18
32	HSC3 CV=PV (aktuální hodnota = nastavená hodnota)		19
29	HSC4 CV=PV (aktuální hodnota = nastavená hodnota)		20
30	HSC4 Změna směru		21
31	HSC4 Externí reset		22
33	HSC5 CV=PV (aktuální hodnota = nastavená hodnota)		23
10	Časované přer. 0 SMB34	Časované <i>Nejnižší priorita</i>	0
11	Časované přer. 1 SMB35		1
21	Časovač T32 přerušení CT=PT		2
22	Časovač T96 přerušení CT=PT		3

Příklad: Instrukce přerušení

M A I N	Network 1 Network 2 Network 3	Network 1 //V prvním programovém cyklu: //1. Definuje podprogram přerušení INT_0 jako přerušení od //sestupné hrany pro I0.0 //2. Globálně povolí přerušení. LD SM0.1 ATCH INT_0, 1 ENI Network 2 //Pokud se detekuje chyba I/O, //zakáže se přerušení od sestupné hrany pro I0.0. //Tento network je volitelný. LD SM5.0 DTCH 1 Network 3 //Když je M5.0 v hodnotě logické 1, //zakáže všechna přerušení. LD M5.0 DISI
	Network 1	Network 1 //Podprogram přerušení: //Podmíněný návrat na základě chyby I/O. LD SM5.0 CRETI

Příklad: Časované přerušení pro čtení hodnoty analogového vstupu

M A I N	Network 1	Network 1 //V prvním programovém cyklu volá podprogram 0. LD SM0.1 CALL SBR_0
S B R 0	Network 1	Network 1 //1. Nastaví interval pro časované přerušení 0 na hodnotu 100 ms. //2. Přiřadí časované přerušení 0 (událost 10) k INT_0. //3. Globální povolení přerušení. LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI
I N T 0	Network 1	Network 1 //Snímá hodnotu AIW4 každých 100 ms LD SM0.0 MOVW AIW4, VW100

Instrukce logických operací

Inverze

Inverze byte, word a double word

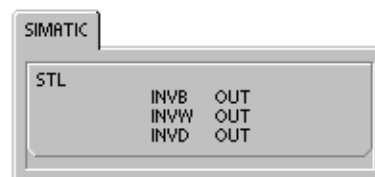
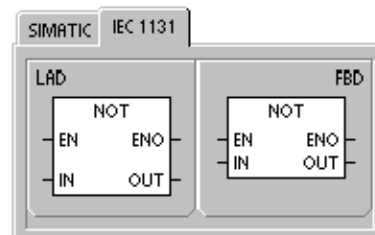
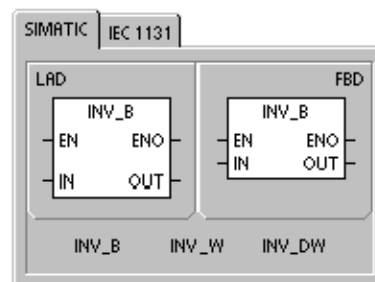
Instrukce "Invertovat byte" (INVB), "Invertovat word" (INW) a "Invertovat double word" (INVD) vytvoří jedničkový doplněk vstupu IN a výsledek uloží do místa paměti OUT.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Ovlivněné bity SM:

- SM1.0 (nula)



Tabulka 6-49 Operandy platné pro instrukce inverze

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta
OUT	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Příklad: Instrukce inverze

Network 1 	Network 1 <pre>LD I4.0 INW AC0</pre> Inverze word AC0 1101 0111 1001 0101 doplňek AC0 0010 1000 0110 1010

AND, OR a XOR

AND byte, AND word a AND double word

Instrukce "AND byte" (ANDB), "AND word" (ANDW) a "AND double word" (ANDD) provedou logický součin odpovídajících bitů dvou vstupních hodnot IN1 a IN2 a výsledek uloží do paměťového místa OUT.

OR byte, OR word a OR double word

Instrukce "OR byte" (ORB), "OR word" (ORW) a "OR double word" (ORD) provedou logický součet odpovídajících bitů dvou vstupních hodnot IN1 a IN2 a výsledek uloží do paměťového místa OUT.

XOR byte, XOR word a XOR double word

Instrukce "XOR byte" (XORB), "XOR word" (XORW) a "X double word" (XORD) provedou nonekvivalenci odpovídajících bitů dvou vstupních hodnot IN1 a IN2 a výsledek uloží do paměťového místa OUT.

Bity SM a ENO

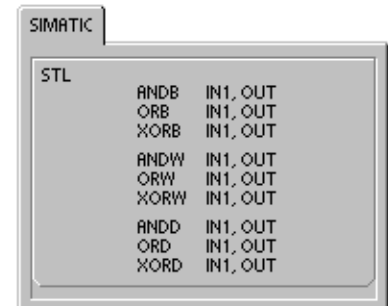
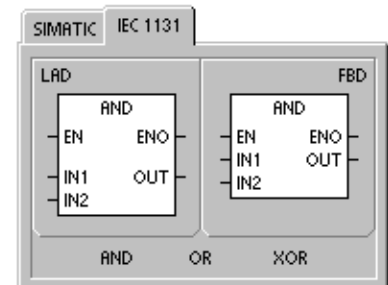
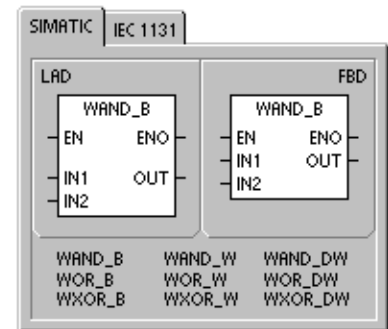
Bity SM a ENO ovlivňují u všech instrukcí popsaných na této stránce následující podmínky.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Ovlivněné bity SM:

- SM1.0 (nula)



Tabulka 6-50 Operandy platné pro instrukce AND, OR a XOR

Vstupy/Výstupy	Typy dat	Operandy
IN1, IN2	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *AC, *LD
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

6

Network 1

I4.0

WAND_W

EN

ENO

AC1-IN1

AC0-IN2

OUT-AC0

WOR_W

EN

ENO

AC1-IN1

VW100-IN2

OUT-VW100

WXOR_W

EN

ENO

AC1-IN1

AC0-IN2

OUT-AC0

Network 1

LD I4.0

ANDW AC1, AC0

ORW AC1, VW100

XORW AC1, AC0

AND word

AC1 0001 1111 0110 1101

AND

AC0 1101 0011 1110 0110

rovná se

AC0 0001 0011 0110 0100

OR word

AC1 0001 1111 0110 1101

OR

VW100 1101 0011 1010 0000

rovná se

VW100 1101 1111 1110 1101

XOR word

AC1 0001 1111 0110 1101

XOR

AC0 0001 0011 0110 0100

rovná se

AC0 0000 1100 0000 1001

164

Instrukce přesunu

Přesun byte, word, double word nebo real

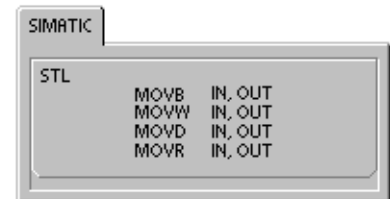
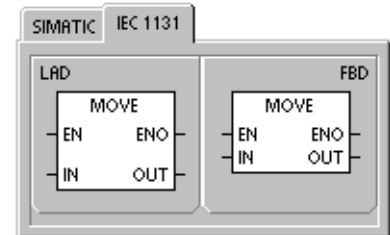
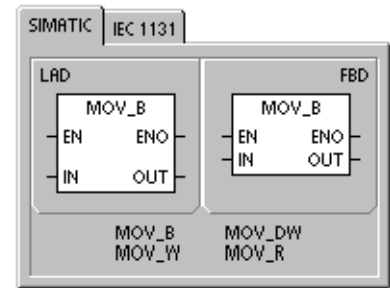
Instrukce "Přesunout byte" (MOVB), "Přesunout word" (MOVW), "Přesunout double word" (MOVD) a "Přesunout real" (MOVR) přesunou hodnotu z paměťového místa IN na nové paměťové místo OUT, aniž by změnily původní hodnotu.

Pro vytvoření ukazatele použijte instrukci "Přesunout double word". Více informací naleznete v odstavcích o ukazatelích a nepřímé adresaci v kapitole 4.

V instrukci IEC pro přesun se mohou typy vstupních a výstupních dat lišit, ale musí mít stejnou velikost.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)



Tabulka 6-51 Operandy platné pro instrukce přesunu

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, konstanta
	DWORD, DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, &IB, &QB, &VB, &MB, &SB, &T, &C, *VD, *LD, *AC, konstanta
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, konstanta
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC
	DWORD, DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Okamžitý přesun bytu (čtení a zápis)

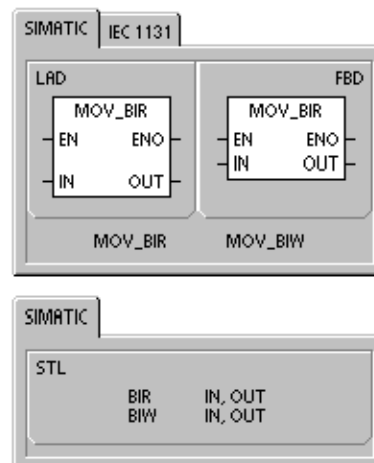
Instrukce "Přesunout byte okamžitě" umožňuje okamžitý přesun bytu mezi fyzickým vstupem/výstupem a paměťovým místem.

Instrukce "Přesunout byte na čtení okamžitě" (BIR) snímá fyzický vstup (IN) a zapisuje výsledek do adresy paměti (OUT), ale registr obrazu procesu není aktualizován.

Instrukce "Přesunout byte na zápis okamžitě" (BIW) snímá data z adresy paměti (IN) a zapisuje je na fyzický výstup (OUT) a odpovídající místo obrazu procesu.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- Nemožnost přístupu do rozšiřovacího modulu



Tabulka 6-52 Operandy platné pro instrukci "Přesunout byte na čtení okamžitě"

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE	IB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Tabulka 6-53 Operandy platné pro instrukci "Přesunout byte na zápis okamžitě"

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
OUT	BYTE	QB, *VD, *LD, *AC

Přesun bloku

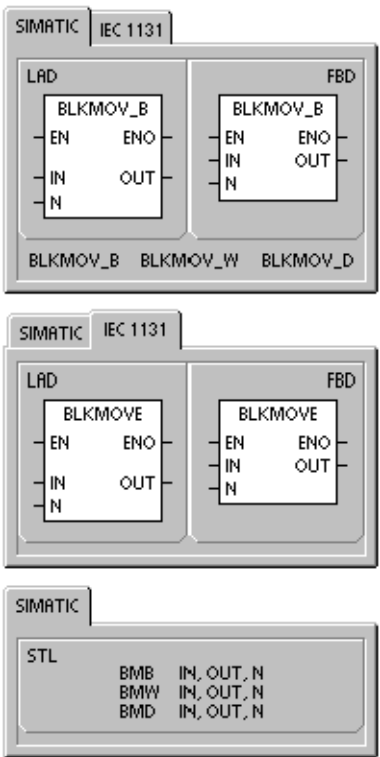
Přesun bloku byte, word nebo double word

Instrukce "Přesunout blok byte" (BMB), "Přesunout blok word" (BMW) a "Přesunout blok double word" (BMD) přesunou daný objem dat na nové paměťové místo přesunutí zadaného počtu bytů, word nebo double word v parametru N, který začíná na vstupní adrese IN, do nového bloku začínajícího na výstupní adrese OUT.

N má rozsah od 1 do 255.

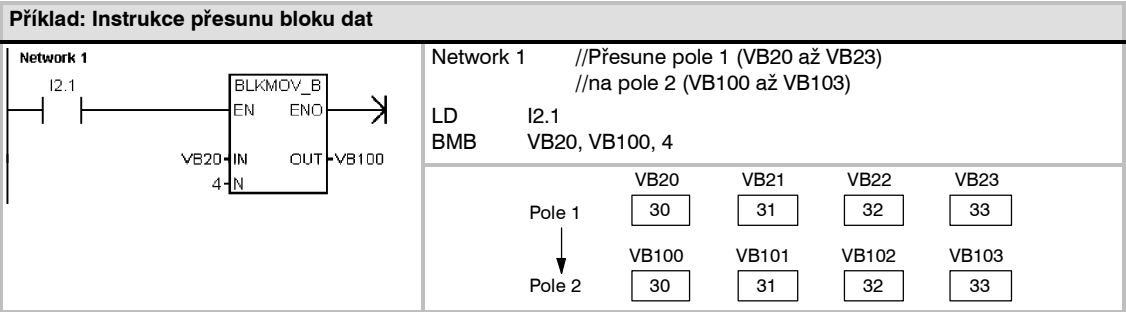
Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)



Tabulka 6-54 Operandy platné pro instrukce přesunu bloku dat

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE WORD, INT DWORD, DINT	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, *VD, *LD, *AC
OUT	BYTE WORD, INT DWORD, DINT	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AQW, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, *VD, *LD, *AC
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, konstanta, *VD, *LD, *AC



Řízení programu

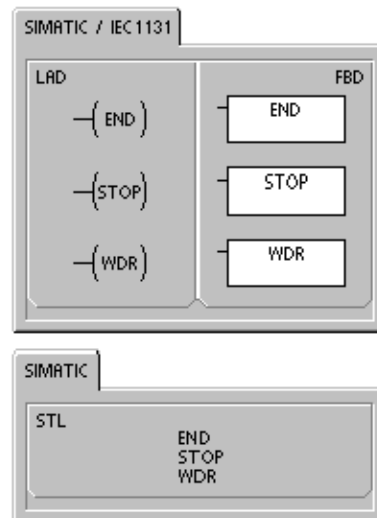
Podmíněný konec programu

Instrukce "Podmíněný konec programu" (END) ukončí současný programový cyklus na základě předcházející logiky. Instrukci "Podmíněný konec programu" je možné použít v hlavním programu; není možné ji však použít v podprogramech nebo v přerušení.

Stop

Instrukce "Stop" (STOP) ukončí provádění uživatelského programu tak, že způsobí přechod CPU S7-200 z režimu RUN do režimu STOP.

Pokud se instrukce "Stop" provede v podprogramu přerušení, je tento podprogram okamžitě ukončen a všechna nevyřízená přerušení jsou ignorována. Zbývající činnosti současného programového cyklu jsou dokončeny včetně provedení hlavního uživatelského programu a přechod z režimu RUN do režimu STOP je proveden na konci aktuálního programového cyklu.



Reset Watchdogu

Instrukce "Reset Watchdogu" (WDR) umožňuje opětovné spuštění obvodu sledování funkce (watchdog timer) CPU S7-200, čímž se prodlouží čas, po který může trvat programový cyklus, aniž by došlo k chybě watchdogu.

Instrukci "Reset Watchdogu" je nutné používat opatrně. Pokud použijete instrukce k zabránění dokončení programového cyklu nebo pro výrazné prodloužení jeho trvání, jsou následující procesy zablokovány až do dokončení cyklu:

- ☐ Komunikace (s výjimkou režimu Freeport)
- ☐ Aktualizace vstupů/výstupů (s výjimkou bezprostředních I/O)
- ☐ Aktualizace force (vnucení hodnot)
- ☐ Aktualizace SM bitů (SM0, SM5 až SM29 nejsou aktualizovány)
- ☐ Diagnostika v době vykonávání programu (run-time)
- ☐ 10 ms a 100 ms časovače nebudou správně akumulovat čas u programových cyklů delších než 25 sekund
- ☐ Instrukce STOP použita v podprogramu přerušení
- ☐ Rozšiřovací moduly s diskretními výstupy mají také obvod sledování funkce (watchdog timer), který vypne výstupy, pokud automat S7-200 nezapíše do modulu. V případě diskretních výstupů použijte bezprostřední zapisování do každého rozšiřovacího modulu, aby se zachovaly zapnuty správné výstupy během prodlouženého trvání programového cyklu. Viz příklad, který následuje po tomto popisu.

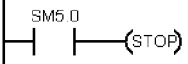
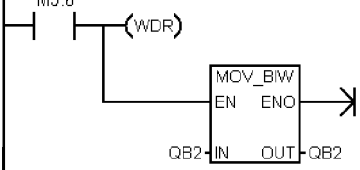
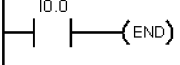


Tip

Pokud předpokládáte, že programový cyklus bude delší než 500 ms nebo pokud očekáváte shluk přerušovacích událostí, které by mohly zabránit návratu do hlavního cyklu po dobu delší než 500 ms, měli byste použít instrukci "Reset Watchdogu" pro opětovné spuštění obvodu sledování funkce.

Když použijete instrukci "Reset Watchdogu", měli byste v každém diskretním rozšiřovacím modulu pro resetování watchdogu každého rozšiřovacího modulu pokaždé provést i bezprostřední zápis do jednoho výstupního bytu (QB).

Pokud použijete instrukci "Reset Watchdogu", abyste umožnili provádění programu, který vyžaduje dlouhý čas programového cyklu, přepnutí přepínače režimů do polohy STOP způsobí, že S7-200 přejde do režimu STOP do 1,4 sekundy.

Příklad: Instrukce "Stop", "Konec" a "Reset Watchdogu"	
Network 1  Network 2  Network 3 	Network 1 //Když se detekuje chyba I/O, //vnutí se přechod do režimu STOP. LD SM5.0 STOP Network 2 //Když je M5.6 v hodnotě logická 1, umožní prodloužení //programového cyklu: //1. Znovu spustí instrukci "Reset Watchdogu" pro S7-200. //2. Znovu spustí watchdog pro první výstupní modul. LD M5.6 WDR BIW QB2, QB2 Network 3 //Když je I0.0 v hodnotě logická 1, ukončí aktuální //programový cyklus. LD I0.0 END

Smyčka For–Next

Smyčky "For" (FOR) a "Next" (NEXT) se používají pro definování smyčky, která má specifikovaný počet opakování. Každá instrukce "For" vyžaduje instrukci "Next". Smyčky For–Next je možné vnořit (umístit smyčku For–Next do jiné smyčky For–Next) až do hloubky osm.

Instrukce "For" provede instrukce mezi instrukcemi "For" a "Next". Je třeba specifikovat hodnotu indexu neboli počet opakování pro aktuální smyčku INDX, počáteční hodnotu INIT a konečnou hodnotu FINAL.

Instrukce "Next" označuje konec smyčky FOR.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

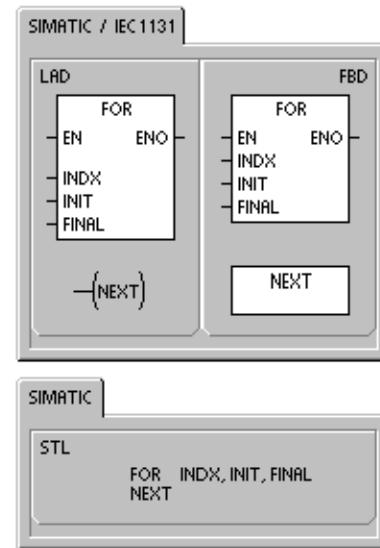
Pokud provedete smyčku For–Next, její práce pokračuje až do ukončení opakování, pokud nezměníte konečnou hodnotu zvnitřku smyčky samotné. Hodnoty je možné měnit i během práce smyčky For–Next. Pokud je smyčka spouštěna opakovaně, zkopíruje se počáteční hodnota do hodnoty indexu (číslo aktuální smyčky).

Instrukce "For–Next" se sama vynuluje při příštím spuštění.

Jestliže má například INIT hodnotu 1 a FINAL hodnotu 10, budou instrukce mezi instrukcí "For" a instrukcí "Next" provedeny desetkrát a hodnota INDX naroste: 1, 2, 3, ...10.

Pokud je počáteční hodnota větší než konečná, smyčka se neprovede. Po každém provedení instrukcí mezi instrukcemi "For" a "Next" je hodnota INDX inkrementována a výsledek je porovnán s konečnou hodnotou. Pokud je INDX větší než konečná hodnota, smyčka je ukončena.

Pokud je na vrcholu zásobníku hodnota log. 1, když program vstoupí do smyčky For–Next, bude hodnota na vrcholu zásobníku 1, když program ze smyčky For–Next vystoupí.



Tabulka 6-55 Operandy platné pro instrukce "For" a "Next"

Vstupy/Výstupy	Typy dat	Operandy
INDX	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
INIT, FINAL	INT	VW, IW, QW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta

Příklad: Instrukce smyčky For – Next

Network 1

I2.0

FOR

EN ENO

VW100 INDX

+1 INIT

+100 FINAL

1

Network 2

I2.1

FOR

EN ENO

VW225 INDX

+1 INIT

+2 FINAL

2

Network 3

(NEXT)

Network 4

(NEXT)

Network 1 //Když se zapne I2.0, je vnější smyčka
 //(šipka 1) provedena stokrát

LD I2.0

FOR VW100, +1, +100

Network 2 //Vnitřní smyčka (šipka 2) je provedena dvakrát
 //na každé provedení vnější smyčky, pokud je
 //ovšem I2.1 zapnutý.

LD I2.1

FOR VW225, +1, +2

Network 3 //Konec smyčky 2.

NEXT

Network 4 //Konec smyčky 1.

NEXT

Skok

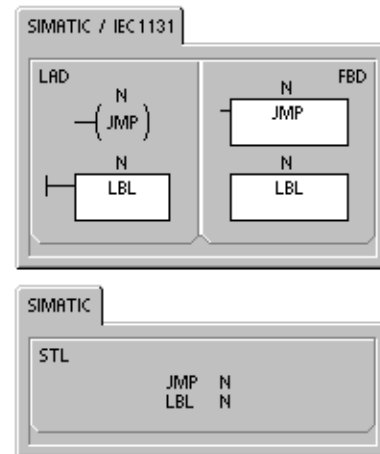
Instrukce "Skok" uskutečňuje skok na specifikované návěští N v programu.

Instrukce "Návěští" (LBL) označuje cílové místo pro skok N.

Skokové instrukce je možné používat v hlavním programu, v podprogramech nebo v přerušení. Instrukce "Skok" a k ní příslušející instrukce "Návěští" musí být vždy umístěny ve stejné části programu (v hlavním programu, podprogramu nebo přerušení).

Není možné skočit z hlavního programu na návěští v podprogramu nebo přerušení. Obdobně nemůžete skočit z podprogramu nebo z přerušení na návěští mimo tento podprogram nebo přerušení.

Instrukci "Skok" můžete použít v segmentu SCR, ale příslušná instrukce "Návěští" musí být umístěna ve stejném segmentu SCR.



Tabulka 6-56 Operandy platné pro skokové instrukce

Vstupy/Výstupy	Typy dat	Operandy
N	WORD	Konstanta (0 až 255)

Příklad: Instrukce Skok	
Network 1 Network 2	Network 1 //Pokud nebyla ztracena remanentní data, skočí na LBL4 <pre>LDN SM0.2 JMP 4</pre> Network 2 <pre>LBL 4</pre>

Sekvenční řídicí relé (SCR)

Instrukce SCR poskytují jednoduchou, ale účinnou programovací techniku stavového řízení, která se přirozeně hodí pro LAD, FBD nebo STL.

Kdykoliv se aplikace skládá ze sledu operací, které musí být prováděny opakovaně, je možné pomocí SCR strukturalizovat program tak, aby odpovídal přímo dané aplikaci. V důsledku toho je možné aplikaci rychleji a snáze programovat a odladit.

Instrukce "Načíst SCR" (LSCR) uloží do SCR a logických zásobníků hodnotu bitu S daného instrukcí N.

Segment SCR je aktivován nebo deaktivován výslednou hodnotou zásobníku SCR. Hodnota zásobníku SCR je zkopírována na vrchol logického zásobníku, takže bloky nebo výstupní kontakty mohou být připojeny přímo k levé napájecí liště bez vloženého kontaktu.

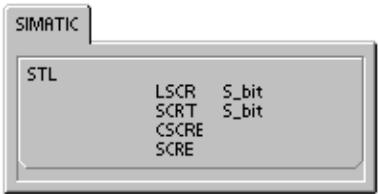
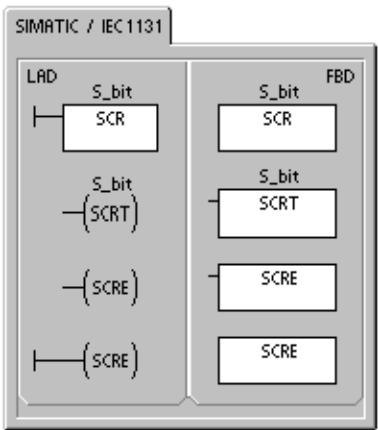
Omezení

Při používání SCR mějte na paměti následující omezení:

- ☐ Stejný bit S není možné použít ve více než jedné části programu. Jestliže použijete například S0.1 v hlavním programu, nepoužívejte ho v podprogramu.
- ☐ Není možné provést skok do segmentu SCR nebo z něho ven; můžete ale použít instrukce "Skok" a "Návěští" pro skok kolem segmentů SCR nebo skok uvnitř segmentu SCR.
- ☐ V segmentu SCR nemůžete použít instrukci END.

Tabulka 6-57 Operandy platné pro instrukce sekvenčních řídicích relé

Vstupy/Výstupy	Typy dat	Operandy
S_bit	BOOL	S

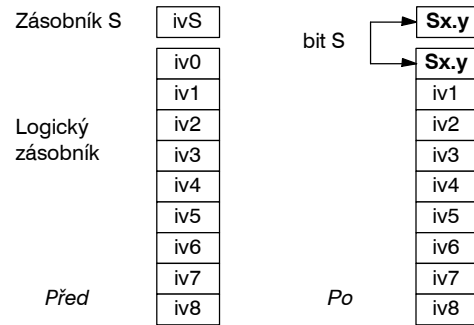


Obrázek 6-31 ukazuje zásobník S, logický zásobník a funkci instrukce "Načíst SCR". Pro instrukce sekvenčního řídicího relé platí:

- ❑ Instrukce "Načíst SCR" (LSCR) označuje začátek segmentu SCR a instrukce "Konec SCR" (SCRE) označuje konec segmentu SCR. Provedení veškeré logiky mezi instrukcemi "Načíst SCR" a "Konec SCR" závisí na hodnotě zásobníku S. Logika mezi "Konec SCR" a příští instrukcí "Načíst SCR" na hodnotě zásobníku S nezávisí.
- ❑ Instrukce "Přechod SCR" (SCRT) dává možnost přenést řízení z aktivního segmentu SCR na jiný segment SCR.

Provedení instrukce "Přechod SCR" při aktivním signálovém toku vynuluje bit S právě aktivního segmentu a nastaví bit S daného segmentu. Vynulování bitu S aktivního segmentu neovlivní zásobník S v době provedení instrukce "Přechod SCR". V důsledku toho zůstane segment SCR aktivní, dokud není opuštěn.

Nahraje hodnotu Sx.y do SCR a do logických zásobníků.



Obr. 6-31 Účinek LSCR na logický zásobník

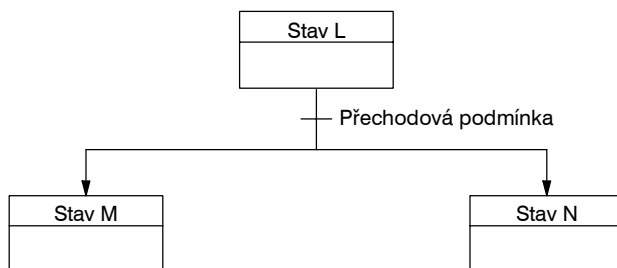
- ❑ Instrukce "Podmíněný konec SCR" (CSCRE) poskytuje možnost vystoupit z aktivního segmentu SCR bez provedení instrukcí mezi instrukcemi "Podmíněný konec SCR" a "Konec SCR". Instrukce "Podmíněný konec SCR" neovlivní žádný bit S ani zásobník S.

V následujícím příkladu bit prvního programového cyklu SM0.1 nastaví S0.1, což je aktivní Stav 1 při prvním programovém cyklu. Po zpoždění 2. sekundy způsobí T37 přechod do Stavu 2. Tento přechod deaktivuje segment SCR (S0.1) Stavu 1 a aktivuje segment SCR (S0.2) Stavu 2.

Příklad: Instrukce sekvenčního řídicího relé	
Network 1 SM0.1 S0.1 (s) 1 Network 2 S0.1 SCR Network 3 SM0.0 Q0.4 1 (s) Q0.5 (R) 2 T37 IN TON +20 PT Network 4 T37 S0.2 (SCRT) Network 5 (SCRE) Network 6 S0.2 SCR Network 7 SM0.0 Q0.2 1 (s) T38 IN TON +250 PT Network 8 T38 S0.3 (SCRT) Network 9 (SCRE)	Network 1 //V prvním programovém cyklu povolí Stav 1. LD SM0.1 S S0.1, 1 Network 2 //Začátek oblasti řízení Stavů 1. LSCR S0.1 Network 3 //Řídí signály pro ulici 1: //1. Povolení: Rozsvítí červené světlo. //2. Nulování: Zhasne žluté a zelené světlo. //3. Spustí 2sekundový časovač. LD SM0.0 S Q0.4, 1 R Q0.5, 2 TON T37, +20 Network 4 //Po zpoždění 2 sekundy, přechod do Stavů 2. LD T37 SCRT S0.2 Network 5 //Konec oblasti SCR pro Stav 1. SCRE Network 6 //Začátek oblasti řízení Stavů 2. LSCR S0.2 Network 7 //Řídí signály pro ulici 2: //1. Nastavení: Rozsvítí zelené světlo. //2. Spustí 25sekundový časovač. LD SM0.0 S Q0.2, 1 TON T38, +250 Network 8 //Po zpoždění 25 sekund přechod do Stavů 3. LD T38 SCRT S0.3 Network 9 //Konec oblasti SCR pro Stav 2. SCRE

Větvení programu

V mnoha aplikacích musí být jeden tok sekvenčních stavů rozdělen do dvou nebo více různých toků. Jestliže se tok řízení rozvětví, musí být všechny výstupní větve aktivovány současně. To ukazuje obrázek 6-32.



Obr. 6-32 Větvení toku řízení

Větvení toků řízení může být v programu SCR implementováno použitím více instrukcí SCRT povolených stejnou přechodovou podmínkou, jak ukazuje následující příklad.

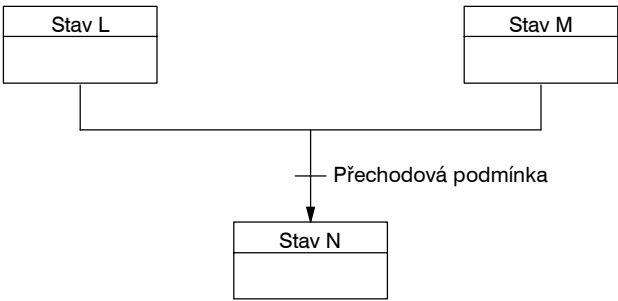
Příklad: Větvení toků řízení

Network 1 S3.4 SCR	Network 1 //Začátek oblasti řízení Stavů L. LSCR S3.4
Network 2 M2.3 I2.1 S3.5 (SCRT) S6.5 (SCRT)	Network 2 LD M2.3 A I2.1 SCRT S3.5 //Přechod do Stavů M SCRT S6.5 //Přechod do Stavů N
Network 3 (SCRE)	Network 3 //Konec oblasti stavu pro Stav L. SCRE

Sloučení větví

Obdobná situace nastane, když se dva nebo více toků sekvenčních stavů musí sloučit do toku jediného. Sloučení více toků do jediného se nazývá konvergence. Když toky konvergují, musí být před provedením následujícího stavu všechny vstupující toky ukončené. Obrázek 6-33 znázorňuje konvergenci dvou toků řízení.

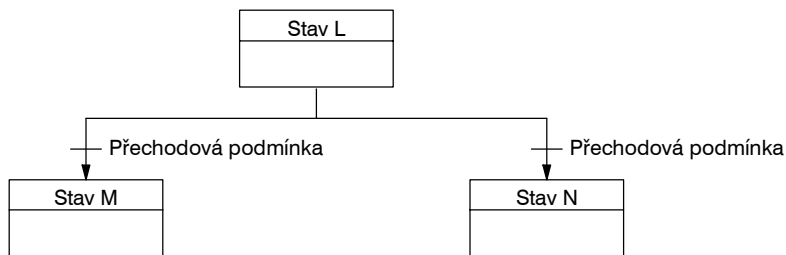
Konvergenci toků řízení je možné v programu SCR implementovat provedením přechodu ze stavu L do stavu L' a přechodem ze stavu M do stavu M'. Když jsou oba bity SCR představující L' a M' pravdivé, je možné povolit stav N, jak ukazuje následující příklad.



Obr. 6-33 Konvergence toku řízení

Příklad: Konvergence toků řízení	
Network 1 S3.4 SCR Network 2 V100.5 S3.5 (SCRT) Network 3 (SCRE) Network 4 S6.4 SCR Network 5 C50 S6.5 (SCRT) Network 6 (SCRE) Network 7 S3.5 S6.5 S5.0 (S)1 S3.5 (R)1 S6.5 (R)1	Network 1 //Začátek oblasti řízení Stavů L LSCR S3.4 Network 2 //Přechod do stavu L' LD V100.5 SCRT S3.5 Network 3 //Konec oblasti SCR pro Stav L. SCRE Network 4 //Začátek oblasti řízení Stavů M LSCR S6.4 Network 5 //Přechod do stavu M' LD C50 SCRT S6.5 Network 6 //Konec oblasti SCR pro Stav M SCRE Network 7 //Když je aktivován Stav L' a Stav M': //1. Povolí Stav N (S5.0) //2. Resetuje Stav L' (S3.5) //3. Resetuje Stav M' (S6.5) LD S3.5 A S6.5 S S5.0, 1 R S3.5, 1 R S6.5, 1

V ostatních situacích je možné tok řízení nasměrovat do jednoho z několika možných toků řízení podle toho, která přechodová podmínka se uskuteční jako první. Taková situace je ukázána na obrázku 6-34, který ukazuje tento program pro SCR.



Obr. 6-34 Větvení toku řízení v závislosti na přechodové podmínce

6

Příklad: Podmíněné přechody

Network 1 S3.4 SCR	Network 1 //Začátek oblasti řízení Stavů L LSCR S3.4
Network 2 M2.3 S3.5 — —(SCRT)	Network 2 //Přechod do Stavů M LD M2.3 SCRT S3.5
Network 3 I3.3 S6.5 — —(SCRT)	Network 3 //Přechod do Stavů N LD I3.3 SCRT S6.5
Network 4 —(SCRE)	Network 4 //Konec oblasti SCR pro Stav L. SCRE

Instrukce posunu a rotace

Posun vpravo a vlevo

Instrukce posunou vstupní hodnotu IN doprava nebo doleva o počet bitů N a výsledek uloží na výstup OUT.

Při posunu bitu ven doplňují instrukce posunu hodnotu nulami. Pokud je počet posunu (N) větší nebo pokud se rovná povolenému maximu (8 pro operace s byty, 16 pro operace s word a 32 pro operace s double word), je hodnota posunuta tolikrát, kolik je maximální počet pro tuto operaci. Pokud je počet posunu větší než 0, přebírá bit přetečení paměti (SM1.1) hodnotu posledního vysunutého bitu. Nulový bit paměti (SM1.0) je nastaven, jestliže je výsledkem operace posunu nula.

Operace s byty jsou bez znaménka. U operací s jednotkami word a double word je při použití datových typů se znaménky posunut i znaménkový bit.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Ovlivněné bity SM:

- SM1.0 (nula)
- SM1.1 (přetečení)

Rotace vpravo a vlevo

Instrukce rotují vstupní hodnotu (IN) doprava nebo doleva o počet bitů (N) a výsledek uloží na paměťové místo (OUT). Rotace je kruhová.

Pokud je počet rotací větší nebo pokud se rovná maximu pro operaci (8 pro operaci s byty, 16 pro operaci s word nebo 32 pro operaci s double word), provede S7-200 operaci modulo s počtem posunu, aby získal před provedením rotace platný počet. Výsledkem je počet posunu 0 až 7 pro operaci s byty, 0 až 15 pro operaci s jednotkami word a 0 až 31 pro operaci s jednotkami double word.

Pokud je počet posunů 0, operace rotace není provedena. Po provedení operace rotace je hodnota posledního rotovaného bitu zkopírována do bitu přeplnění (SM1.1).

Pokud není počet posunů celočíselným násobkem 8 (pro operace s byty), 16 (pro operace s word) nebo 32 (pro operace s double word), zkopíruje se poslední rotovaný bit do paměťového bitu přeplnění (SM1.1). Nulový bit paměti (SM1.0) je nastaven, jestliže je hodnota rotována na nulu.

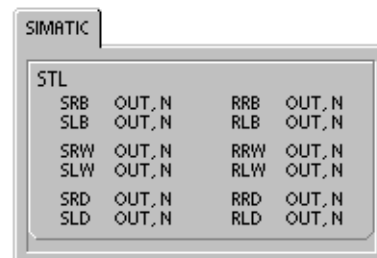
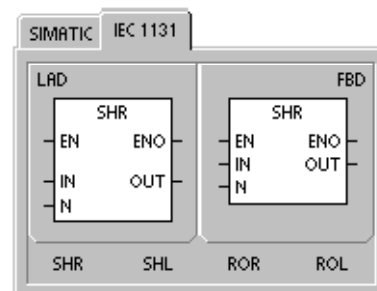
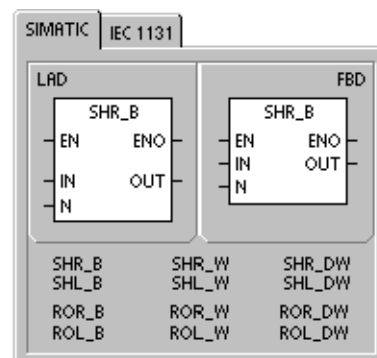
Operace s byty jsou bez znaménka. U operací s jednotkami word a double word je při použití datových typů se znaménky rotován i znaménkový bit.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)

Ovlivněné bity SM:

- SM1.0 (nula)
- SM1.1 (přetečení)



Tabulka 6-58 Operandy platné pro instrukce posunu a rotace

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, konstanta
OUT	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta

Příklad: Instrukce posunu a rotace

Network 1

Network 1

LD I4.0

RRW AC0, 2

SLW VW200, 3

Rotace

	Před rotací	Přetečení
AC0	0100 0000 0000 0001	x
	Po první rotaci	Přetečení
AC0	1010 0000 0000 0000	1
	Po druhé rotaci	Přetečení
AC0	0101 0000 0000 0000	0
Nulový bit paměti (SM1.0) = 0		
Bit přetečení paměti (SM1.1) = 0		

Posun

	Před posunem	Přetečení
VW200	1110 0010 1010 1101	x
	Po prvním posunu	Přetečení
VW200	1100 0101 0101 1010	1
	Po druhém posunu	Přetečení
VW200	1000 1010 1011 0100	1
	Po třetím posunu	Přetečení
VW200	0001 0101 0110 1000	1
Nulový bit paměti (SM1.0) = 0		
Bit přetečení paměti (SM1.1) = 1		

180

Posunutí bitů registru

Instrukce "Posunout bit registru" posune hodnotu do posuvného registru. Tato instrukce nabízí snadný způsob pro kontrolu následnosti a řízení toku výrobků nebo dat. Instrukci použijte pro posunutí celého registru o jeden bit jednou za programový cyklus.

Instrukce "Posunout bit registru" posune hodnotu DATA do posuvného registru. S_BIT specifikuje nejnižší platný bit posuvného registru. N specifikuje délku posuvného registru a směr posunu (kladné posunutí = N, záporné posunutí = -N).

Každý bit vysunutý instrukcí SHRB je umístěn do bitu přetečení paměti (SM1.1).

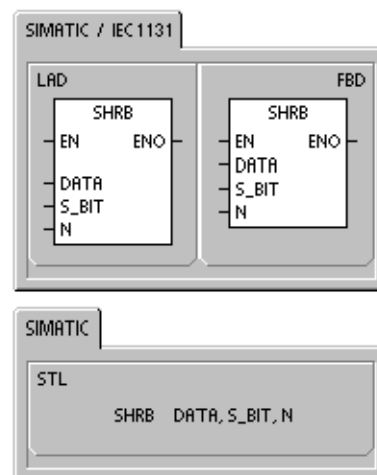
Tato instrukce je definována jak nejnižším platným bitem (S_BIT), tak počtem bitů specifikovaných délkou (N).

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)
- 0092 (chyba v poli počtu)

Ovlivněné bity SM:

- SM1.1 (přetečení)



Tabulka 6-59 Operandy platné pro instrukci "Posunout bit registru"

Vstupy/Výstupy	Typy dat	Operandy
DATA, S_Bit	BOOL	I, Q, V, M, SM, S, T, C, L,
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta

Pro výpočet adresy nejvyššího platného bitu posuvného registru (MSB.b) použijte následující rovnici:

$$\text{MSB.b} = [(\text{Byte } S\_BIT) + ([N] - 1 + (\text{bit } S\_BIT))] / 8 \cdot [\text{zbytek dělení } 8]$$

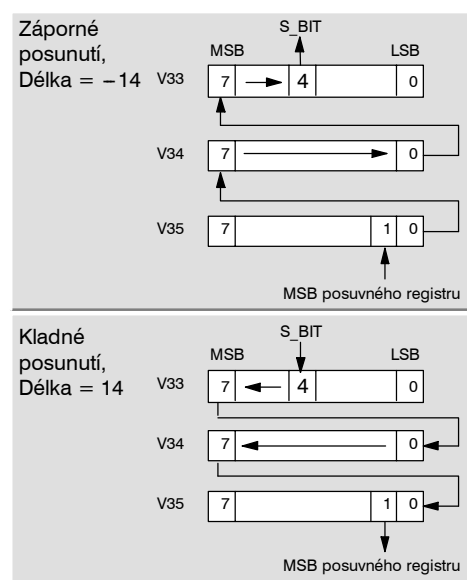
Například: jestliže S_BIT je V33.4 a N je 14, ukazuje následující výpočet, že MSB.b je V35.1.

$$\begin{aligned} \text{MSB.b} &= V33 + ([14] - 1 + 4) / 8 \\ &= V33 + 17 / 8 \\ &= V33 + 2 \text{ se zbytkem } 1 \\ &= V35.1 \end{aligned}$$

Při záporném posunutí indikovaném zápornou hodnotou délky (N) se vstupní data posunou do nejvyššího platného bitu posuvného registru a vysunou nejnižší platný bit (S_BIT). Vysunutá data jsou pak umístěna v bitu přetečení paměti (SM1.1).

Při kladném posunutí indikovaném kladnou hodnotou délky (N) se vstupní data (DATA) vysunou ven z nejvyššího platného bitu posuvného registru a posunou se do nejnižšího platného bitu posuvného registru specifikovaného v S_BIT. Vysunutá data jsou pak umístěna v bitu přetečení paměti (SM1.1).

Maximální délka posuvného registru je 64 bitů kladných nebo záporných. Obrázek 6-35 ukazuje posunutí bitů pro záporné a kladné hodnoty N.



Obr. 6-35 Vstup a výstup posuvného registru

Network 1

Network 1

```
LD      I0.2
EU
SHRB    I0.3, V100.0, +4
```

Časový diagram

Před prvním posunem

Po prvním posunu

Po druhém posunu

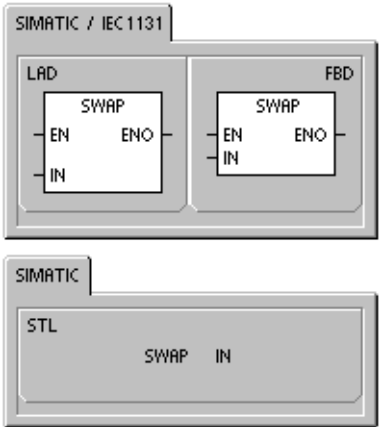
	7 (MSB)	6	5	4	3	2	1	0 (LSB)	S_BIT	
Před prvním posunem							0	1	0	1
	↓									
Přetečení (SM1.1)	x									
Po prvním posunu							1	0	1	1
	↓									
Přetečení (SM1.1)	0									
Po druhém posunu							0	1	1	0
	↓									
Přetečení (SM1.1)	1									

Výměna bytů

Instrukce "Vyměnit byty" vymění nejvyšší platný byte slova (word) IN s jeho nejnižším platným byte.

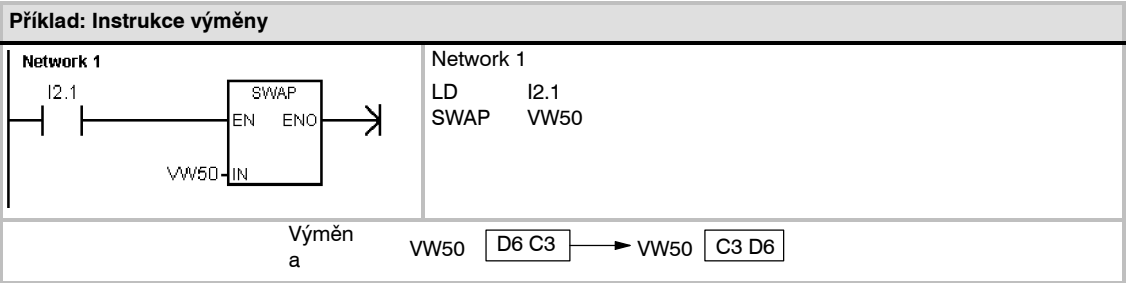
Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)



Tabulka 6-60 Operandy platné pro instrukci "Vyměnit byty"

Vstupy/Výstupy	Typy dat	Operandy
IN	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC,*VD, *LD, *AC



Instrukce pro řetězce

Délka řetězce

Instrukce "Délka řetězce" (SLEN) vrací délku řetězce specifikovanou v IN.

Kopírování řetězce

Instrukce "Kopírovat řetězec" (SCPY) kopíruje řetězec specifikovaný v IN do řetězce specifikovaného v OUT.

Spojování řetězců

Instrukce "Sloučit řetězce" (SCAT) připojí řetězec specifikovaný v IN na konec řetězce specifikovaného v OUT.

Bity SM a ENO

U instrukcí "Délka řetězce", "Kopírovat řetězec" a "Sloučit řetězce" ovlivní ENO následující podmínky.

Chyby, které nastavují ENO = 0

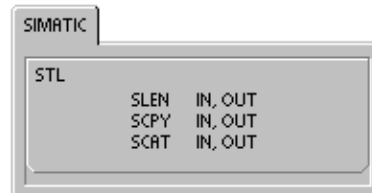
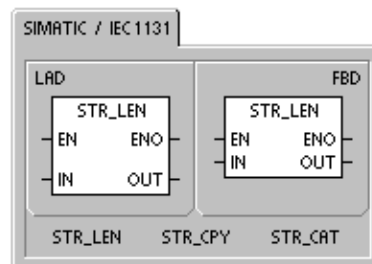
- 0006 (nepřímá adresa)
- 0091 (chyba rozsahu)

Tabulka 6-61 Operandy platné pro instrukci "Délka řetězce"

Vstupy/Výstupy	Typy dat	Operandy
IN	BYTE (řetězec)	VB, LB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Tabulka 6-62 Operandy platné pro instrukce "Kopírovat řetězec" a "Sloučit řetězce"

Vstupy/Výstupy	Typy dat	Operandy
IN, OUT	BYTE (řetězec)	VB, LB, *VD, *LD, *AC



Příklad: Instrukce "Sloučit řetězec", "Kopírovat řetězec" a "Délka řetězce"

Network 1

STR_CAT
EN ENO
VB20 IN OUT VB0

STR_CPY
EN ENO
VB0 IN OUT VB100

STR_LEN
EN ENO
VB100 IN OUT AC0

Network 1

//1. Připojí řetězec na VB20
// k řetězci na VB0
//2. Zkopíruje řetězec na VB0
// na nový řetězec na VB100
//3. Získá délku řetězce,
// který začíná na VB100

LD I0.0
SCAT VB20, VB0
STRCPY VB0, VB100
STRLEN VB100, AC0

Před provedením programu

VB0	VB6	VB20	VB25
6	'H' 'E' 'L' 'L' 'O' '	5	'W' 'O' 'R' 'L' 'D'

Po provedení programu

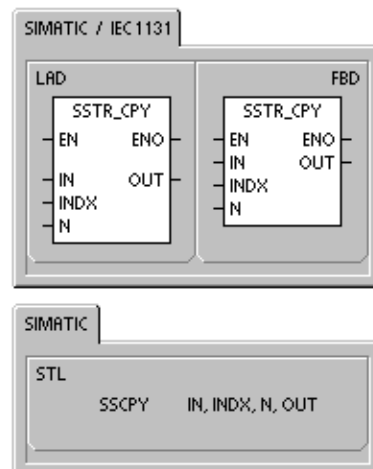
VB0	VB11
11	'H' 'E' 'L' 'L' 'O' ' ' 'W' 'O' 'R' 'L' 'D'
VB100	VB111
11	'H' 'E' 'L' 'L' 'O' ' ' 'W' 'O' 'R' 'L' 'D'
AC0	
11	

Kopírování části řetězce

Instrukce "Kopírovat část řetězce" (SSCPY) zkopíruje specifikovaný počet znaků N z řetězce specifikovaného v IN, počínaje indexem INDX, do nového řetězce specifikovaného v OUT.

Chyby, které nastavují ENO = 0

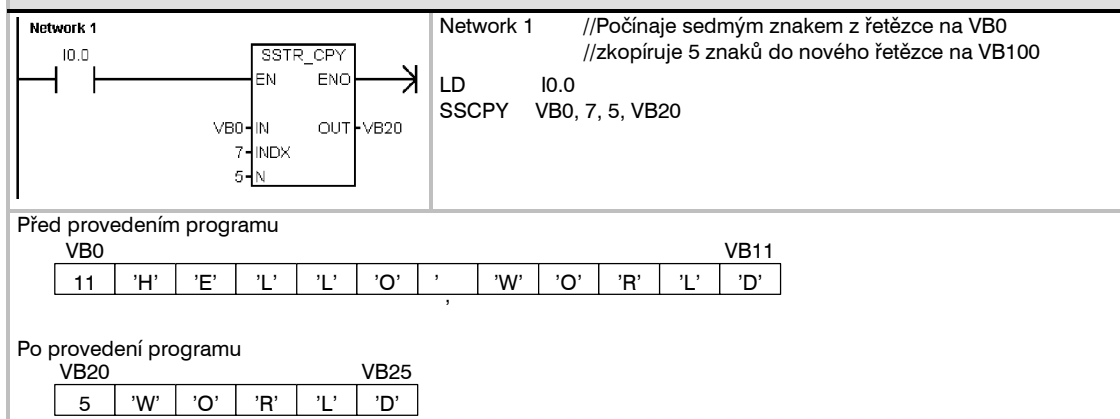
- 0006 (nepřímá adresa)
- 0091 (chyba rozsahu)
- 009B (index=0)



Tabulka 6-63 Operandy platné pro instrukci "Kopírovat část řetězce"

Vstup/Výstupy	Typy dat	Operandy
IN, OUT	BYTE (řetězec)	VB, LB, *VD, *LD, *AC
INDX, N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta

Příklad: Instrukce kopírování části řetězce



Nalezení části řetězce

Instrukce "Najít část řetězce" (SFND) hledá první výskyt řetězce IN2 uvnitř řetězce IN1. Vyhledávání začíná na počátečním místě specifikovaném v OUT. Jestliže je nalezena sekvence znaků, která přesně odpovídá řetězci IN2, je umístění prvního znaku sekvence řetězce zapsáno do OUT. Pokud řetězec IN2 nebyl v řetězci IN1 nalezen, je v OUT 0.

Chyby, které nastavují ENO = 0

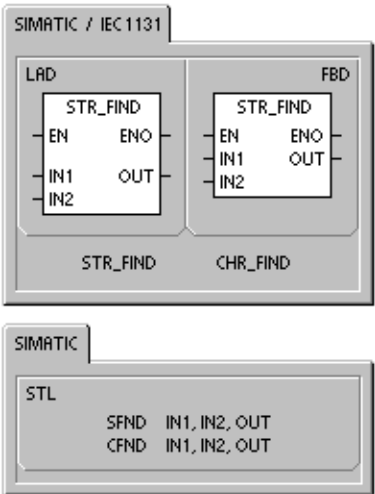
- 0006 (nepřímá adresa)
- 0091 (chyba rozsahu)
- 009B (index=0)

Nalezení znaku uvnitř řetězce

Instrukce "Najít znak v řetězci" (CFND) prohledává řetězec IN1 na první výskyt kteréhokoliv znaku ze souboru znaků popsaného v řetězci IN2. Vyhledávání začíná na výchozím místě v OUT. Jestliže je nalezen odpovídající znak, je jeho umístění zapsáno do OUT. Není-li nalezen žádný odpovídající znak, je v OUT 0.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (chyba rozsahu)
- 009B (index=0)

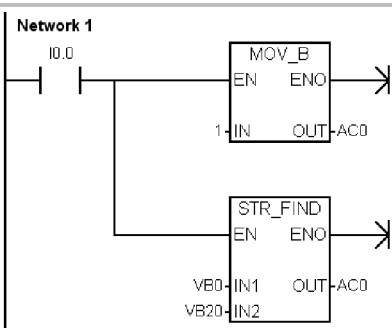


Tabulka 6-64 Operandy platné pro instrukce "Najít část řetězce" a "Najít znak v řetězci"

Vstupy/Výstupy	Typy dat	Operandy
IN1, IN2	BYTE (řetězec)	VB, LB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Příklad: Instrukce nalezení části řetězce

V následujícím příkladu je řetězec uložený na VB0 použit jako povel pro zapnutí nebo vypnutí čerpadla. Řetězec 'On' (zapnutí) je uložen na VB20 a řetězec 'Off' (vypnutí) je uložen na VB30. Výsledek instrukce "Najít část řetězce" je uložen v AC0 (parametr OUT). Pokud výsledek není 0, byl řetězec 'On' nalezen v povelovém řetězci (VB12).



Network 1 //1. Nastaví AC0 na 1.
// (AC0 je použit jako parametr OUT.)
//2. Prohledá řetězec na VB0 na vložený řetězec
// na VB20 ('On'), počínaje od prvního
// místa (AC0=1).

LD I0.0
MOVB 1, AC0
SFND VB0, VB20, AC0

VB0										VB12									
12	'T'	'u'	'r'	'n'	'P'	'u'	'm'	'p'	'O'	'n'									
VB20										VB22									
2	'O'	'n'																	
VB30										VB33									
3	'O'	'f'	'f'																

Je-li řetězec ve
VB20 nalezen:

AC0

11

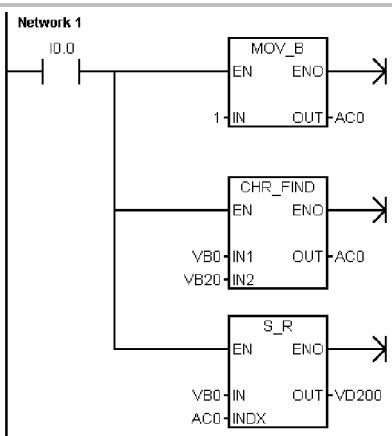
Není-li řetězec ve
VB20 nalezen:

AC0

0

Příklad: Instrukce nalezení znaku uvnitř řetězce

V následujícím příkladu obsahuje řetězec uložený ve VB0 teplotu. Řetězec ve VB20 uchovává všechny číslice (a + a -), které mohou identifikovat v řetězci teplotu. Vzorový program nalezne počáteční umístění čísla v daném řetězci a pak převede číslice na reálné číslo. VD200 uchovává hodnotu teploty danou reálným číslem.



Network 1 //1. Nastaví AC0 na 1.
// (AC0 je použit jako parametr OUT)
// a ukazuje na první místo řetězce.)
//2. Najde číslici
// v řetězci na VB0.
//3. Konvertuje řetězec na reálné číslo.

LD I0.0
MOVB 1, AC0
CFND VB0, VB20, AC0
STR VB0, AC0, VD200

VB0										VB11									
11	'T'	'e'	'm'	'p'	'9'	'8'	'6'	'F'											
VB20										VB32									
12	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'0'	'+'	'-'							

Počáteční místo teploty uložené ve VB0:

AC0

7

Hodnota teploty udaná reálným číslem:

VD200

98.6

Instrukce pro práci s tabulkou

Přidávání do tabulky

Instrukce "Přidat do tabulky" přidává hodnoty velikosti word (DATA) do tabulky (TBL). První hodnota v tabulce je její maximální délka (TL). Druhá hodnota je počet položek (EC), které specifikuje počet položek v tabulce. Nové údaje se do tabulky přidávají za poslední položku. Pokaždé, když jsou do tabulky přidána nová data, se zvětší počet položek.

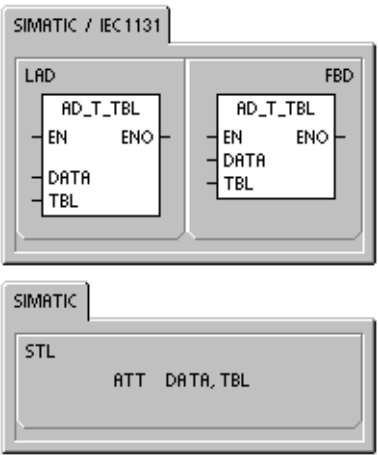
Tabulka může obsahovat až 100 datových položek.

Chyby, které nastavují ENO = 0

- SM1.4 (přetečení)
- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)

Ovlivněné bity SM:

- SM1.4 je nastaven na 1, jestliže se snažíte tabulku přepsat



Tabulka 6-65 Operandy platné pro instrukce tabulky

Vstupy/Výstupy	Typy dat	Operandy
DATA	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
TBL	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, *VD, *LD, *AC

Příklad: Instrukce "Přidat do tabulky"

Network 1

SM0.1

MOV_W

EN ENO

+6 IN OUT VW200

Network 2

I0.0

AD_T_TBL

EN ENO

VW100 DATA

VW200 TBL

Network 1 //Nahraje maximální délku tabulky

LD SM0.1

MOVW +6, VW200

Network 2

LD I0.0

ATT VW100, VW200

Před provedením ATT

VW100 1234

VW200 0006

VW202 0002

VW204 5431

VW206 8942

VW208 xxxx

VW210 xxxx

VW212 xxxx

VW214 xxxx

TL (max. počet položek)

EC (počet položek)

d0 (data 0)

d1 (data 1)

Po provedení ATT

VW200 0006

VW202 0003

VW204 5431

VW206 8942

VW208 1234

VW210 xxxx

VW212 xxxx

VW214 xxxx

TL (max. počet položek)

EC (počet položek)

d0 (data 0)

d1 (data 1)

d2 (data 2)

Instrukce FIFO a LIFO

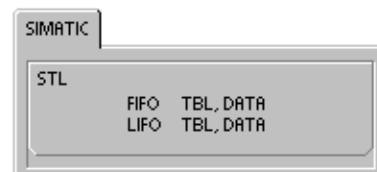
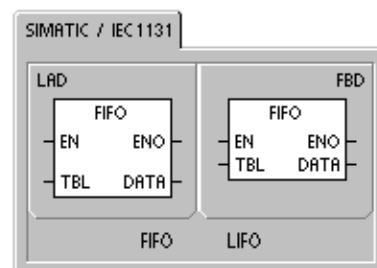
Tabulka může obsahovat až 100 datových položek.

První dovnitř, první ven (FIFO)

Instrukce "První dovnitř, první ven" (FIFO) posune nejstarší (neboli první) položku tabulky na adresu výstupní paměti tak, že odstraní první položku v tabulce (TBL) a posune ji na místo specifikované v DATA. Všechny ostatní položky tabulky se posunou o jedno místo. Počet položek v tabulce se po každém provedení instrukce snižuje.

Poslední dovnitř, první ven (LIFO)

Instrukce "Poslední dovnitř, první ven" (LIFO) posune nejnovější (neboli poslední) položku tabulky na adresu výstupní paměti tak, že odstraní poslední položku v tabulce (TBL) a posune ji na místo specifikované v DATA. Počet položek v tabulce se po každém provedení instrukce snižuje.



Chyby, které nastavují ENO = 0

- SM1.5 (prázdná tabulka)
- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)

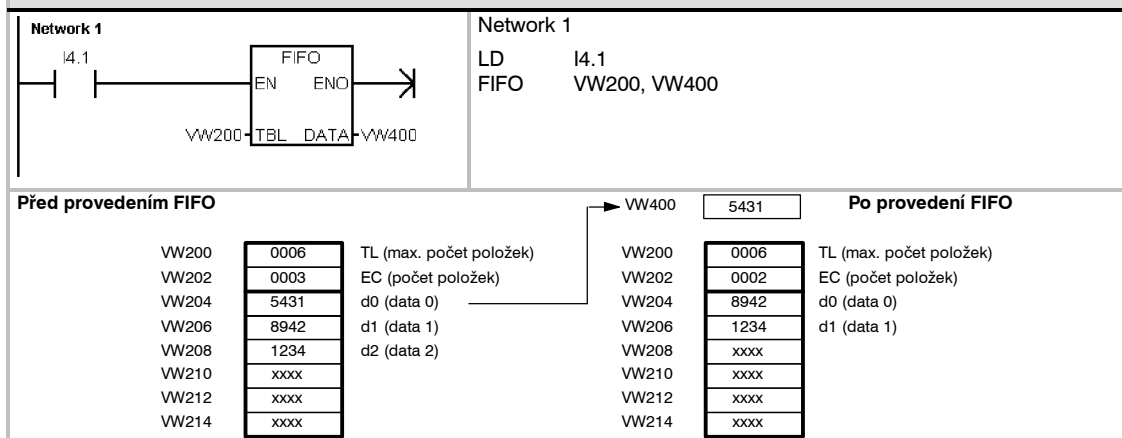
Ovlivněné bity SM:

- SM1.5 se nastaví na 1, jestliže se pokusíte odstranit položku z prázdné tabulky

Tabulka 6-66 Operandy platné pro instrukce "První dovnitř, první ven" a "Poslední dovnitř, první ven"

Vstupy/Výstupy	Typy dat	Operandy
TBL	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, *VD, *LD, *AC
DATA	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC

Příklad: Instrukce "První dovnitř, první ven"



Příklad: Instrukce "Poslední dovnitř, první ven"

Network 1

Network 1
LD I0.1
LIFO VW200, VW300

Před provedením LIFO

VW200	0006	TL (max. počet položek)
VW202	0003	EC (počet položek)
VW204	5431	d0 (data 0)
VW206	8942	d1 (data 1)
VW208	1234	d2 (data 2)
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

Po provedení LIFO

VW200	0006	TL (max. počet položek)
VW202	0002	EC (počet položek)
VW204	5431	d0 (data 0)
VW206	8942	d1 (data 1)
VW208	xxxx	
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

→ VW300

1234

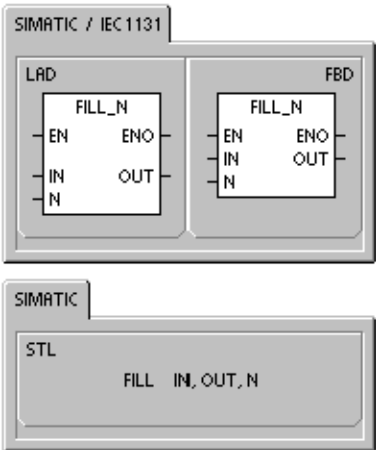
Vyplnění paměti

Instrukce "Vyplnit paměť" (FILL) zapisuje N za sebou jdoucích wordů, které začínají na adrese OUT, s hodnotou obsaženou na adrese IN.

N má rozsah od 1 do 255.

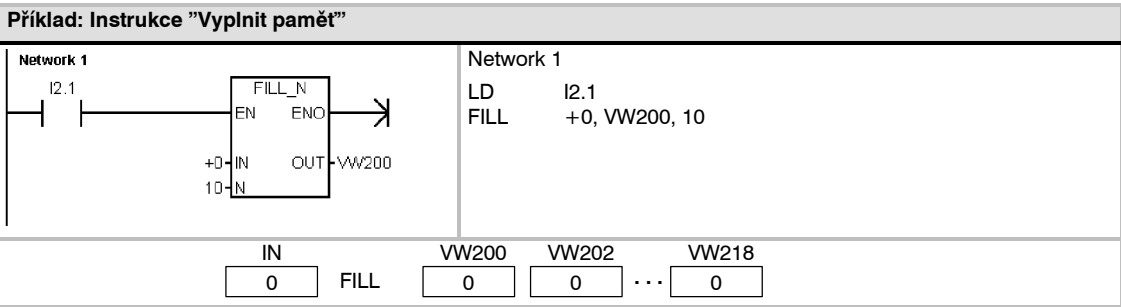
Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)



Tabulka 6-67 Operandy platné pro instrukci "Vyplnit paměť"

Vstupy/Výstupy	Typy dat	Operandy
IN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, konstanta
OUT	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AQW, *VD, *LD, *AC



Prohledávání tabulky

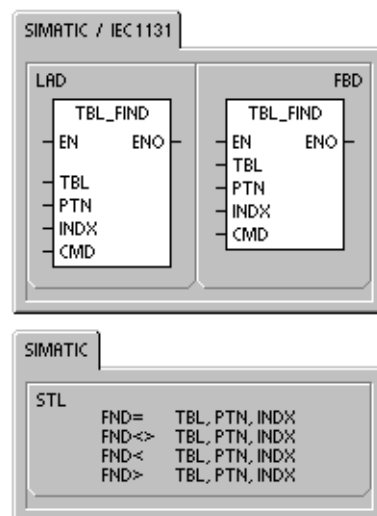
Instrukce "Najít v tabulce" (FND) prohledává tabulku a hledá data, která odpovídají určitým kritériím. Instrukce "Najít v tabulce" prohledává tabulku TBL počínaje položkou tabulky INDX a hledá datovou hodnotu nebo obrazec, které odpovídají vyhledávacím kritériím definovaným v CMD. Parametr povelu CMD je daný číselnou hodnotou 1 až 4, kde 1 znamená =, 2 znamená <>, 3 znamená < a 4 znamená >.

Když se najde vhodná hodnota, ukáže INDX na odpovídající položku tabulky. Chcete-li hledat další odpovídající položku, je třeba nejdříve inkrementovat INDX a teprve pak opět spustit instrukci "Najít v tabulce". Pokud se odpovídající hodnota nenajde, rovná se hodnota INDX počtu položek.

Tabulka může obsahovat až 100 datových položek. Datové položky (prohledávaný prostor) jsou číslovány od 0 do maximálně 99.

Chyby, které nastavují ENO = 0

- 0006 (nepřímá adresa)
- 0091 (operand mimo rozsah)



Tabulka 6-68 Operandy platné pro instrukci "Najít v tabulce"

Vstupy/Výstupy	Typy dat	Operandy
TBL	WORD	IW, QW, VW, MW, SMW, T, C, LW, *VD, *LD, *AC
PTN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta
INDX	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
CMD	BYTE	(Konstanta) 1: Rovná se (=), 2: Nerovná se (<>), 3: Menší než (<), 4: Větší než (>)



Tip

Pokud instrukci "Najít v tabulce" použijete pro tabulky generované pomocí instrukcí "Přidat do tabulky", "Poslední dovnitř, první ven" a "První dovnitř, první ven", počet položek přímo odpovídá datovým položkám. Parametr maximálního počtu položek, který je vyžadován pro instrukce "Přidat do tabulky", "Poslední dovnitř, první ven" nebo "První dovnitř, první ven", instrukce "Najít v tabulce" nevyžaduje. Viz obr. 6-36.

Proto musíte nastavit operand TBL instrukce "Najít v tabulce" na adresu o jeden word (dva byty) vyšší než operand TBL odpovídající instrukce "Přidat do tabulky", "Poslední dovnitř, první ven" nebo "První dovnitř, první ven".

Formát tabulky pro ATT, LIFO a FIFO

VW200	0006	TL (max. počet položek)
VW202	0006	EC (počet položek)
VW204	xxxx	d0 (data 0)
VW206	xxxx	d1 (data 1)
VW208	xxxx	d2 (data 2)
VW210	xxxx	d3 (data 3)
VW212	xxxx	d4 (data 4)
VW214	xxxx	d5 (data 5)

Formát tabulky pro TBL_FIND

VW202	0006	EC (počet položek)
VW204	xxxx	d0 (data 0)
VW206	xxxx	d1 (data 1)
VW208	xxxx	d2 (data 2)
VW210	xxxx	d3 (data 3)
VW212	xxxx	d4 (data 4)
VW214	xxxx	d5 (data 5)

Obr. 6-36 Rozdíl ve formátu tabulky instrukce "Najít v tabulce" a tabulky instrukcí ATT, LIFO a FIFO

Network 1

Network 1

LD I2.1

FND= VW202, 16#3130, AC1

Pokud má I2.1 hodnotu logické 1, prohledá tabulku a hledá hodnotu, která se rovná 3130 HEX.

	0006	EC (počet položek)
VW202	3133	d0 (data 0)
VW204	4142	d1 (data 1)
VW206	3130	d2 (data 2)
VW208	3030	d3 (data 3)
VW210	3130	d4 (data 4)
VW212	4541	d5 (data 5)

Pokud byla tabulka vytvořena pomocí instrukcí ATT, LIFO a FIFO, obsahuje VW200 maximální počet položek a instrukce prohledávání ho nevyžadují.

AC1

AC1 musí být nastaven na 0, aby vyhledávání probíhalo v tabulce shora dolů.

Provede prohledávání tabulky

AC1

AC1 obsahuje číslo datové položky, která odpovídá první nalezené shodě v tabulce (d2).

AC1

Než začne prohledávat zbývající položky tabulky, inkrementuje INDX o jednu.

Provede prohledávání tabulky

AC1

AC1 obsahuje číslo datové položky, která odpovídá druhé nalezené shodě v tabulce (d4).

AC1

Než začne prohledávat zbývající položky tabulky, inkrementuje INDX o jednu.

Provede prohledávání tabulky

AC1

AC1 obsahuje hodnotu, která je rovna počtu položek. Celá tabulka byla prohledána a další shoda nebyla nalezena.

AC1

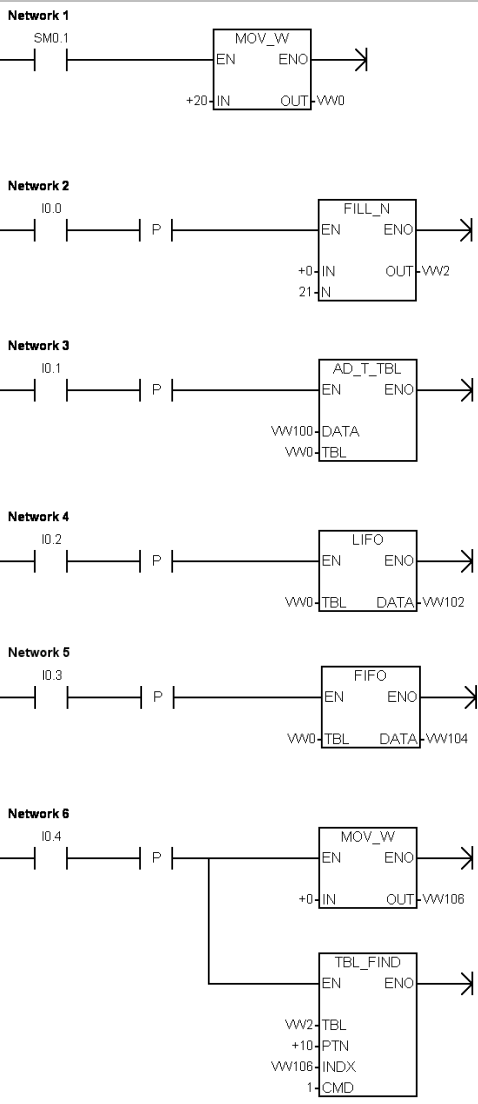
Před opětovným prohledáním tabulky musí být hodnota INDX resetována na 0.

Příklad: Vytvoření tabulky

Následující program vytvoří tabulku s 20 položkami. První paměťové místo tabulky obsahuje délku tabulky (v tomto případě 20 položek). Druhé paměťové místo obsahuje aktuální počet položek tabulky. Ostatní místa obsahují jednotlivé položky. Tabulka může obsahovat až 100 položek. Příklad nezahrnuje parametry definující maximální délku tabulky nebo skutečný počet položek (v tomto případě VW0 a VW2). Při každém příkazu CPU automaticky zvyšuje nebo snižuje aktuální počet položek v tabulce (zde VW2).

Dříve než začnete pracovat s tabulkou, stanovte maximální počet položek tabulky. Jinak nemůžete položky do tabulky vkládat. Také zkontrolujte, že všechny příkazy pro čtení a zápis jsou aktivovány hranou.

Chcete-li tabulku prohledávat, musí být před spuštěním prohledávání nastaven index (VW106) na 0. Jestliže je nalezena shoda, bude mít index číslo položky tabulky; jestliže se však žádná shoda nenajde, bude se index rovnat aktuálnímu počtu položek v tabulce (VW2).



Network 1 //Vytvoří tabulku s 20 položkami, která začíná na
//paměťovém místě 4.
//1. Při prvním programovém cyklu
// definuje maximální délku tabulky.

LD SM0.1
MOVW +20, VW0

Network 2 //Resetuje tabulku vstupem I0.0
//Při náběžné hraně na I0.0
//vyplní paměťová místa od VW2 " +0" .

LD I0.0
EU
FILL +0, VW2, 21

Network 3 //Zapíše hodnotu do tabulky vstupem I0.1
//Při náběžné hraně na I0.1 zkopíruje hodnotu
//paměťového místa VW100 do tabulky.

LD I0.1
EU
ATT VW100, VW0

Network 4 //Načte první hodnotu tabulky vstupem I0.2
//Přesune poslední hodnotu v tabulce do
//VW102. Tím se sníží počet položek.
//Při náběžné hraně na I0.2
//Přesune poslední hodnotu v tabulce
//na místo VW102.

LD I0.2
EU
LIFO VW0, VW102

Network 5 //Načte poslední hodnotu tabulky vstupem I0.3
//Přesune první hodnotu v tabulce do VW102.
//Tím se sníží počet položek.
//Při náběžné hraně na I0.3
//Přesune první hodnotu v tabulce na VW104

LD I0.3
EU
FIFO VW0, VW104

Network 6 //Najde v tabulce první místo, které má
//hodnotu 10.
//1. Při náběžné hraně na I0.4
// resetuje ukazatel indexu.
//2. Najde položku tabulky, jejíž hodnota je
// rovna 10.

LD I0.4
EU
MOVW +0, VW106
FND= VW2, +10, VW106

Instrukce časovače

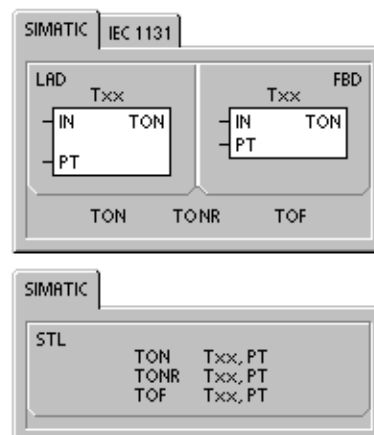
Časovač SIMATIC

Časovač zapnutí Remanentní časovač zapnutí

Instrukce "Časovač zapnutí" (TON) a "Remanentní časovač zapnutí" (TONR) odpočítávají čas od sepnutí vstupu. Číslo (Txx) rozlišuje jednotlivé časovače.

Časovač vypnutí

Instrukce "Časovač vypnutí" (TOF) se používá na zpoždění rozepnutí výstupu o pevný čas po rozepnutí vstupu. Číslo (Txx) rozlišuje jednotlivé časovače.



Tabulka 6-69 Operandy platné pro SIMATIC instrukce časovače

Vstupy/Výstupy	Typy dat	Operandy
Txx	WORD	Konstanta (T0 až T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
PT	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, konstanta



Tip

Není možné mít stejné číslo časovače (Txx) pro časovač vypnutí (TOF) a časovač zapnutí (TON). Nemůžete mít například TON T32 i TOF T32.

Jak ukazuje tabulka 6-70, uvedené tři typy časovačů provádějí různé typy časovacích úloh:

- ☐ TON je možné použít pro intervalové časování.
- ☐ TONR je možné použít pro akumulaci více časovaných intervalů.
- ☐ TOF je možné použít pro prodloužení času po vypnutí, například pro vychlazení motoru po jeho vypnutí.

Tabulka 6-70 Operace instrukcí časovače

Typ	Aktuální = Nastavená	Stav povolovacího vstupu (IN)	Výpadek napájení nebo první cyklus
TON	Časovací bit zapnut Pokračuje počítání aktuální hodnoty do 32.767	ZAPNUTO: Aktuální hodnota odpočítává čas VYPNUTO: Bit časovače vypnutý, aktuální hodnota = 0	Bit časovače vypnutý Aktuální hodnota = 0
TONR	Časovací bit zapnut Pokračuje počítání aktuální hodnoty do 32.767	ZAPNUTO: Aktuální hodnota odpočítává čas VYPNUTO: Bit časovače a aktuální hodnota udržují poslední stav	Bit časovače vypnutý Je možné udržet aktuální hodnotu ¹
TOF	Bit časovače vypnutý Aktuální = nastavená, konec počítání	ZAPNUTO: Bit časovače zapnutý, aktuální hodnota = 0 VYPNUTO: Časovač počítá po přechodu ze zapnutého do vypnutého stavu	Bit časovače vypnutý Aktuální hodnota = 0

¹ Pro uchování hodnoty časovače i po výpadku napájení je možné nastavit remanenci aktuální hodnoty. Více informací o remanentní paměti CPU S7-200 najdete v kapitole 4.



Tipy a triky

Vzorový program s použitím časovače zpožděného zapnutí (TON) najdete v tipech a tricích na CD s dokumentací. Viz tip 31

Instrukce TON a TONR odpočítávají čas, když je jejich vstup zapnutý. Pokud je aktuální hodnota rovna nebo větší než nastavený čas, zapne se bit časovače.

- ☐ Když se rozezne jeho vstup IN, aktuální hodnota časovače TON je smazána, zatímco aktuální hodnota časovače TONR se při rozepnutí vstupu uchová.
- ☐ Časovač TONR je možné použít pro akumulaci času při sepnutí a rozepnutí vstupu IN. Aktuální hodnotu TONR lze vymazat pomocí instrukce "Reset" (R).
- ☐ Jak časovač TON, tak časovač TONR pokračují v počítání i po dosažení nastavené hodnoty a přestanou počítat na maximální hodnotě 32.767.

Instrukce TOF se používá na zpoždění rozepnutí výstupu o pevný čas po rozepnutí vstupu. Když se sepne povolovací vstup, zapne se okamžitě bit časovače a aktuální hodnota je nastavena na 0. Když se vstup rozezne, časovač počítá, dokud uplynulý čas nedosáhne nastavené hodnoty.

- ☐ Když je dosažena nastavená hodnota, vypne se bit časovače a přestane se zvětšovat aktuální hodnota; pokud se však vstup opět sepne dříve, než TOF dosáhne nastavené hodnoty, zůstává bit časovače zapnutý.
- ☐ Aby začal TOF počítat časové intervaly, musí povolovací vstup provést přechod ze sepnutého do rozepnutého stavu.
- ☐ Pokud je časovač TOF uvnitř oblasti SCR a oblast SCR je neaktivní, je aktuální hodnota nastavena na 0, bit časovače se vypne a aktuální hodnota se nezvyšuje.



Tip

TONR je možné resetovat pouze pomocí instrukce Reset (R). Instrukci "Reset" můžete také použít pro resetování kteréhokoli časovače TON nebo TOF. Instrukce "Reset" provádí následující operace:

- Bit časovače => vypnutý
- Aktuální hodnota časovače => 0

Mají-li časovače TOF opět začít počítat, potřebují po resetování přechod vstupu IN ze sepnutého do rozepnutého stavu.

Stanovení rozlišení časovače

Časovače počítají časové intervaly. Rozlišení (neboli časová základna) časovače určuje délku nejmenšího časového intervalu. Například časovač TON s rozlišením 10 ms počítá, kolik 10ms intervalů uplyne po tom, co byl časovač TON spuštěn: počet 50 představuje na 10ms časovači 500 ms. Časovače SIMATIC nabízí tři rozlišení: 1 ms, 10 ms a 100 ms. Jak je patrné z tabulky 6-71, číslo časovače ukazuje jeho rozlišení.



Tip

Pro zaručení minimálního časového intervalu zvýšte požadovanou hodnotu (PV) o 1; například pro zajištění minimálního časovaného intervalu nejméně 2100 ms u 100 ms časovače nastavte PV na 22.

Tabulka 6-71 Čísla a rozlišení časovačů

Typ časovače	Rozlišení	Maximální hodnota	Číslo časovače
TONR (remanentní)	1 ms	32,767 s (0,546 min)	T0, T64
	10 ms	327,67 s (5,46 min)	T1 až T4, T65 až T68
	100 ms	3276,7 s (54,6 min)	T5 až T31, T69 až T95
TON, TOF (bez remanence)	1 ms	32,767 s (0,546 min)	T32, T96
	10 ms	327,67 s (5,46 min)	T33 až T36, T97 až T100
	100 ms	3276,7 s (54,6 min)	T37 až T63, T101 až T255

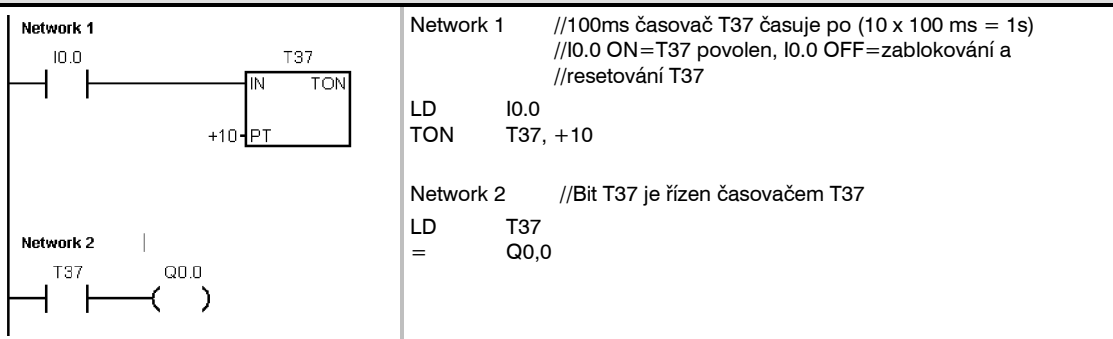
Vysvětlení, jak rozlišení ovlivňuje činnost časovače

U časovače s rozlišením 1 ms jsou bit časovače a aktuální hodnota aktualizovány asynchronně s programovým cyklem. U programových cyklů delších než 1 ms jsou bit časovače a aktuální hodnota aktualizovány během cyklu několikrát.

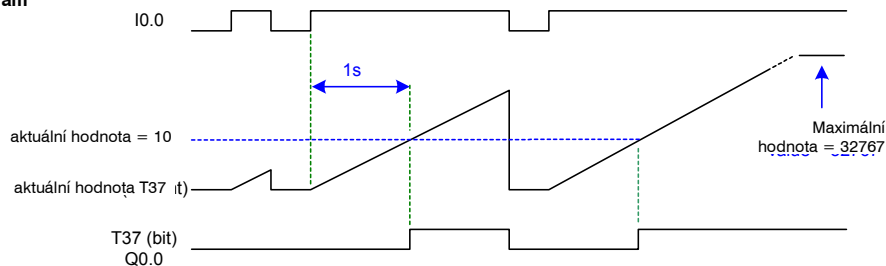
U časovače s rozlišením 10 ms jsou bit časovače a aktuální hodnota aktualizovány na začátku každého programového cyklu. Bit časovače a aktuální hodnota zůstávají během programového cyklu konstantní; časové intervaly, které se během cyklu nakumulují, se na začátku každého programového cyklu přičtou k aktuální hodnotě.

U časovače s rozlišením 100 ms jsou bit časovače a aktuální hodnota aktualizovány při provedení instrukce; proto zajistěte, aby program vykonával instrukci pro 100ms časovač pouze jednou za programový cyklus, aby si časovač uchoval správné časování.

Příklad: Časovač zpoždění náběžné hrany (On-Delay) SIMATIC -

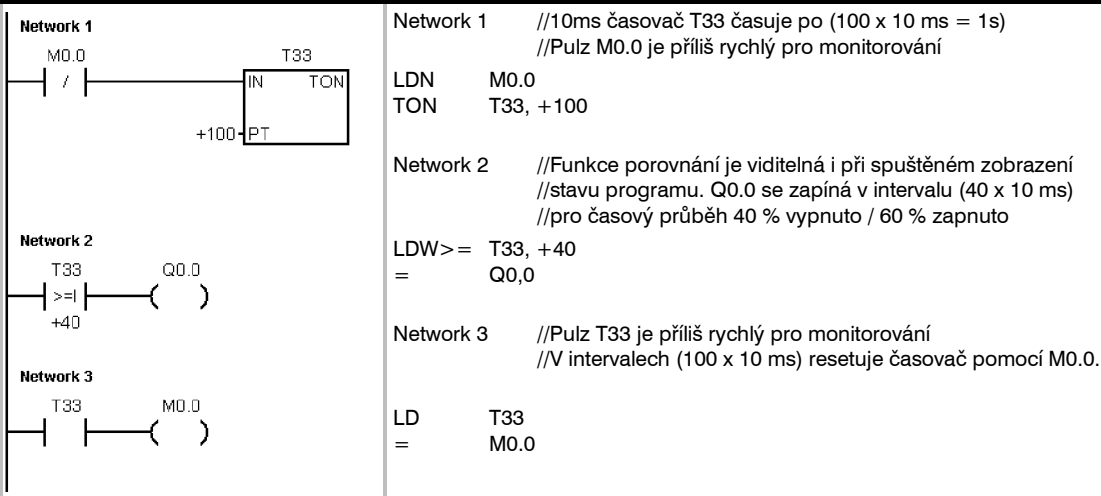
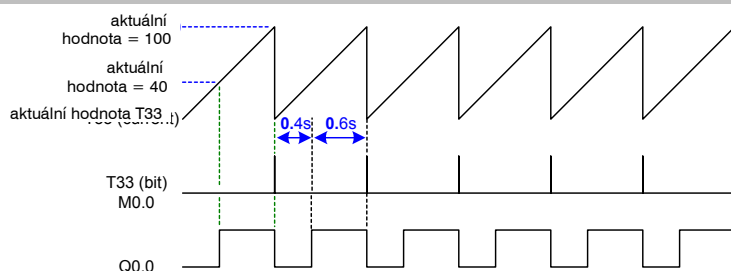
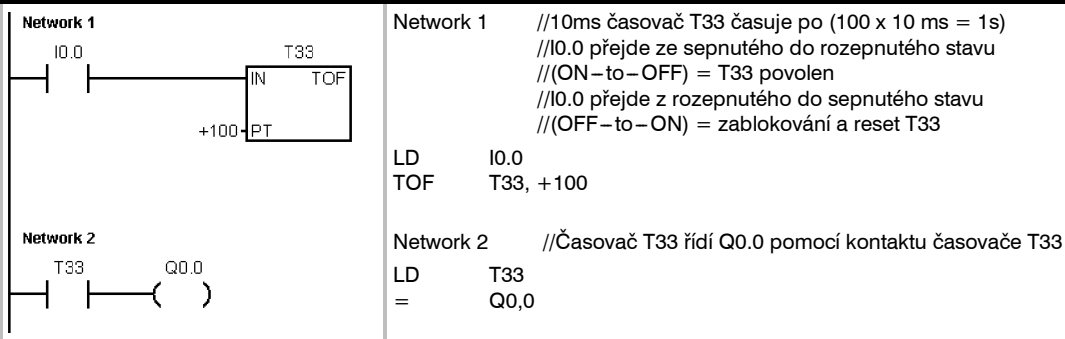
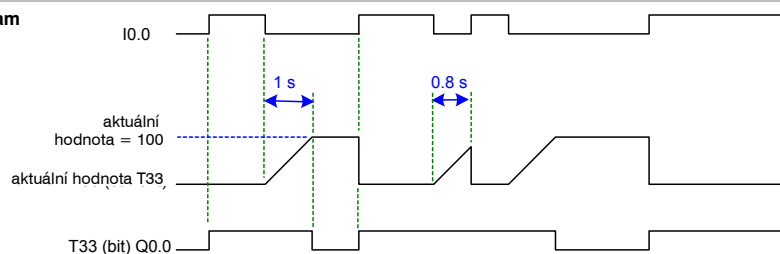


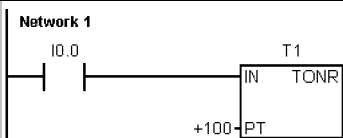
Časový diagram



**Tip**

Aby bylo zaručeno, že se výstup časovače s automatickým opakováním sepne v jednom programovém cyklu pokaždé, když časovač dosáhne požadované hodnoty, použijte jako vstup časovače kontakt, který je v klidu zavřen (namísto bitu časovače).

Příklad: Časovač zpožděného zapnutí SIMATIC s automatickým opakováním**Časový diagram****Příklad: Časovač zpožděného vypnutí SIMATIC****Časový diagram**

Příklad: Remanentní časovač zpožděného zapnutí SIMATIC


Network 1 //10ms časovač TONR T1
//rozepne při PT=(100 x 10 ms=1s)

LD I0.0
TONR T1, +100

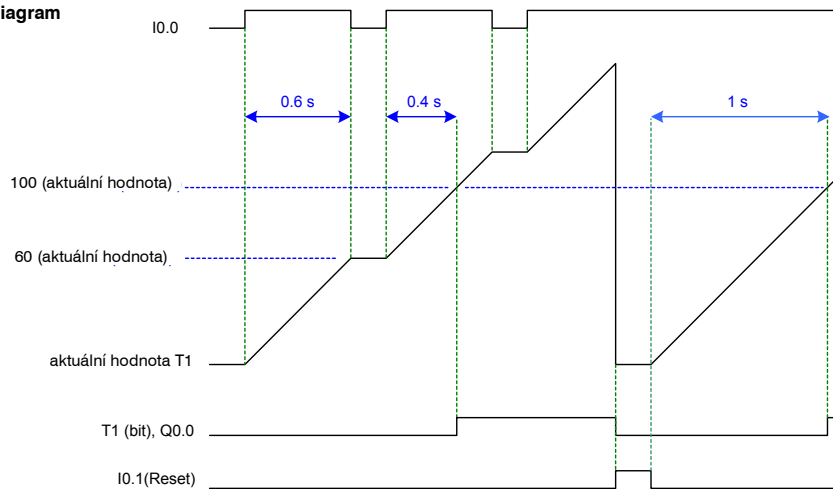
Network 2 //Bit T1 je řízen časovačem T1.
//Sepne Q0.0 po tom, co časovač
//souhrnně napočítá 1 sekundu.

LD T1
= Q0,0

Network 3 //Časovače TONR musí být resetovány
//pomocí instrukce "Reset".

//Když je sepnutý I0.1, resetuje časovač T1
//(aktuální hodnotu a bit).

LD I0.1
R T1, 1

Časový diagram


Časovač IEC

Časovač zpožděného zapnutí

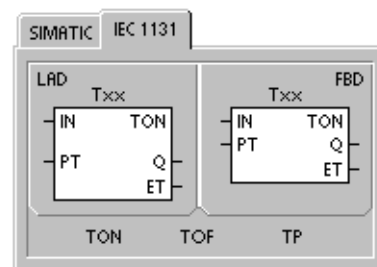
Instrukce "Časovač zapnutí" (TON) odpočítává čas, když je sepnut povolovací vstup.

Časovač zpožděného vypnutí

Instrukce "Časovač vypnutí" (TOF) zpožďuje rozeznutí výstupu o pevný čas po rozeznutí vstupu.

Pulzní časovač

Instrukce "Pulzní časovač" (TP) generuje pulzy specifické délky.



Tabulka 6-72 Operandy platné pro IEC instrukce časovače

Vstupy/Výstupy	Typy dat	Operandy
Txx	TON, TOF, TP	Konstanta (T32 až T63, T96 až T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
PT	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, konstanta
Q	BOOL	I, Q, V, M, SM, S, L
ET	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC



Tip

Časovače TOF, TON a TP nemohou být označeny stejnými čísly. Nemůžete mít například TON T32 i TOF T32.

- ☐ Instrukce TON počítá časové intervaly až do požadované hodnoty, když je vstup (IN) v log.1. Když se uplynulý čas (ET) rovná nastavenému času (PT), nastaví se výstupní bit (Q) časovače na log. 1. Výstupní bit je resetován po deaktivaci vstupu IN. Když je dosaženo požadované hodnoty, časování se zastaví a časovač je zablokován.
- ☐ Instrukce TOF zpožďuje rozeznutí výstupu o pevný čas po nastavení vstupu na log. 0. Když se rozezne vstup IN, začne TOF počítat čas až do požadované hodnoty. Když se uplynulý čas (ET) rovná nastavenému času (PT), nastaví se výstupní bit (Q) časovače na log. 0. Když je dosaženo požadované hodnoty, nastaví se výstupní bit časovače na log. 0 a uplynulý čas je uchováván, dokud vstup IN nepřejde do stavu log. 1. Jestliže vstup IN nastaví přechod do log. 0 na dobu kratší, než je nastavený čas, zůstane výstupní bit v log. 1.
- ☐ Instrukce TP generuje pulzy specifické délky. Když se sepne vstup IN, nastaví se výstupní bit (Q). Výstupní bit zůstane v log. 1 po dobu trvání pulzu v rámci nastaveného času (PT). Dosáhne-li uplynulý čas (ET) požadované hodnoty (PT), výstupní bit se nastaví na log. 0. Uplynulý čas je uchováván až do doby, než se rozezne vstup IN. Když se nastaví výstupní bit, zůstane zapnutý, dokud neuplyne čas pulzu.

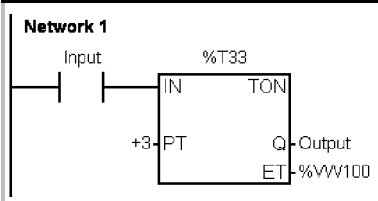
Každá aktuální hodnota je násobkem časové základny. Například počet 50 představuje na 10ms časovači 500 ms. Časovače IEC (TON, TOF a TP) jsou k dispozici se třemi rozlišeními: rozlišení je dáno číslem časovače, jak ukazuje tabulka 6-73.

Tabulka 6-73 Rozlišení časovačů IEC

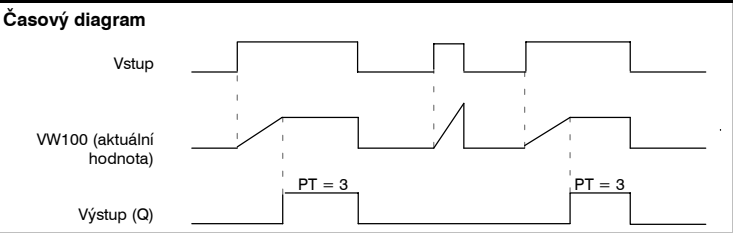
Rozlišení	Maximální hodnota	Číslo časovače
1 ms	32,767 s (0,546 minut)	T32, T96
10 ms	327,67 s (5,46 minut)	T33 až T36, T97 až T100
100 ms	3276,7 s (54,6 minut)	T37 až T63, T101 až T255

Příklad: Časovače zpožděného zapnutí IEC

Network 1

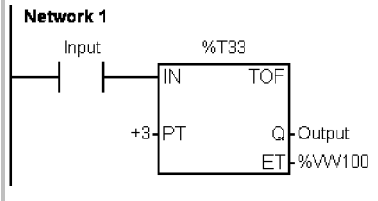


Časový diagram

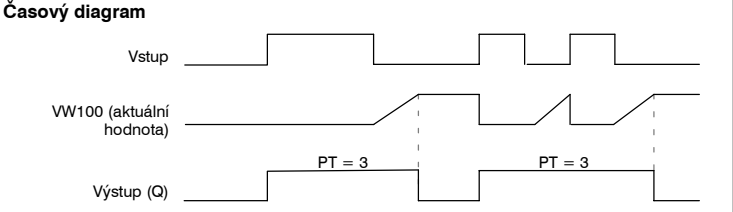


Příklad: Časovače zpožděného vypnutí IEC

Network 1

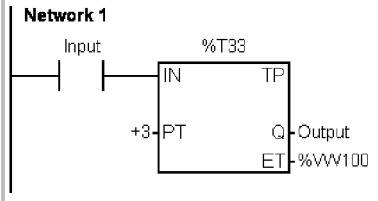


Časový diagram

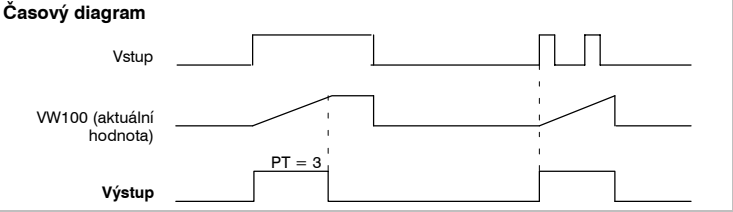


Příklad: Pulzní časovač IEC

Network 1



Časový diagram



202

Instrukce podprogramů

Instrukce volání podprogramu (CALL) přenáší řízení do podprogramu SBR_N. Instrukci "Volání podprogramu" můžete použít s parametry, nebo bez nich. Po skončení provádění podprogramu se řízení vrací na instrukci, která následuje po instrukci "Volání podprogramu".

Instrukce "Podmíněný návrat z podprogramů" (CRET) ukončuje podprogram na základě předcházející logiky.

Chcete-li přidat podprogram, vyberte příkaz menu **Úpravy > Vložit > Podprogram**.

Chyby, které nastavují ENO = 0

- 0008 (překročen maximální povolený počet vnoření podprogramů)
- 0006 (nepřímá adresa)

Z hlavního programu můžete vnořovat podprogramy (umístit volání podprogramu do jiného podprogramu) až do hloubky osm. Z přerušení podprogramy vnořovat nemůžete.

Volání podprogramu nemůže být umístěno v podprogramu, který je volán z přerušení. Rekurse (podprogram, který volá sám sebe) není zakázána; s jejím použitím byste však měli u podprogramů zacházet opatrně.

Tabulka 6-74 Operandy platné pro instrukce podprogramu

Vstupy/Výstupy	Typy dat	Operandy
SBR_N	WORD	Konstanta pro CPU 221, CPU 222, CPU 224 a CPU 226: 0 až 63 pro CPU 226XM: 0 až 127



Tip

STEP 7–Micro/WIN automaticky doplní nepodmíněný návrat z každého podprogramu.

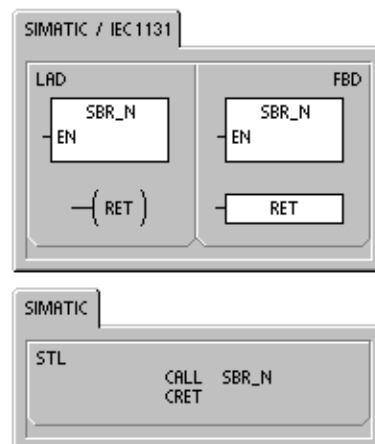
Když je volán podprogram, uloží se celý logický zásobník, vrchol zásobníku se nastaví na jedna, všechna ostatní místa zásobníku se nastaví na nulu a řízení se přenesou do volaného podprogramu. Pokud je tento podprogram ukončen, zásobník se obnoví hodnotami uloženými v místě volání a řízení se vrátí do volajícího programu.

Akumulátory jsou společné pro podprogramy a volající program. Při použití podprogramu se s nimi neprovádí operace uložení nebo obnovení.

Volání podprogramu s parametry

Podprogramy mohou obsahovat odevzdávané parametry. Ty jsou definovány v tabulce lokálních proměnných podprogramu. Parametry musí mít symbolický název (maximálně 23 znaků), typ proměnné a typ dat. Do podprogramu nebo z podprogramu je možné odevzdávat až šestnáct parametrů.

Pole typu proměnné v tabulce lokálních proměnných definuje, zda je proměnná odevzdávána do podprogramu (IN), odevzdávána do podprogramu i z něho ven (IN_OUT) nebo pouze odevzdávána z podprogramu (OUT). Tabulka 6-75 popisuje typy parametrů podprogramu. Chcete-li přidat další parametr, umístěte kurzor na pole typu proměnné (IN, IN_OUT nebo OUT), kterou chcete přidat. Klikněte pravým tlačítkem myši a zobrazí se menu voleb. Vyberte volbu "Vložit" a pak "Řádek pod". Pod aktuální položkou se objeví další položka požadovaného typu.



Tabulka 6-75 Typy parametrů podprogramu

Parametr	Popis
IN	Parametry se odevzdávají do podprogramu. Pokud je parametrem přímá adresa (např. VB10), odevzdá se hodnota na daném místě do podprogramu. Pokud je parametrem nepřímá adresa (např. *AC1), odevzdá se hodnota do podprogramu na místo, na které ukazuje. Pokud je parametrem datová konstanta (16#1234) nebo adresa (&VB100), je do podprogramu odevzdána hodnota konstanty nebo adresy.
IN_OUT	Hodnota na dané adrese parametru se odevzdá do podprogramu, zatímco výsledná hodnota se z podprogramu vrátí na stejnou adresu. Konstanty (např. 16#1234) a adresy (např. &VB100) nejsou jako vstupní/výstupní parametry povoleny.
OUT	Výsledná se hodnota z podprogramu zapíše na danou adresu parametru. Konstanty (jako 16#1234) a adresy (jako &VB100) nejsou jako výstupní parametry povoleny.
TEMP	Kterákoliv lokální paměť, která se nepoužívá pro předávání parametrů, může být použita pro dočasné uložení v rámci podprogramu.

Jak ukazuje obrázek 6-37, pole typu dat v tabulce lokálních proměnných definuje velikost a formát parametru. Typy parametrů jsou následující:

- ☐ **BOOL:** Tento typ dat se používá pro vstupy a výstupy o velikosti jednoho bitu. IN3 v následujícím příkladu je booleovský (bitový) vstup.
- ☐ **BYTE, WORD, DWORD:** Tyto typy dat označují vstupní nebo výstupní parametr bez znaménka o velikosti 1, 2 a 4 byty.
- ☐ **INT, DINT:** Tyto typy dat označují vstupní nebo výstupní parametr se znaménkem o velikosti 2 a 4 byty.
- ☐ **REAL:** Tento typ dat označuje hodnotu s plovoucí desetinou čárkou a jednoduchou přesností (4 byty) podle IEEE.
- ☐ **Signálový tok:** Booleovský signálový tok je povolen pouze pro bitové (booleovské) vstupy. Z této deklarace pro STEP 7-Micro/WIN vyplývá, že tento vstupní parametr je výsledkem signálového toku založeného na kombinaci bitových logických instrukcí. Booleovské vstupy signálového toku se musí v tabulce lokálních proměnných objevit jako první před jakýmkoliv jiným typem vstupu. Tímto způsobem je možné použít pouze vstupní parametry. Povolovací vstup (EN) a vstup IN1 v následujícím příkladu používají booleovské typy parametrů.

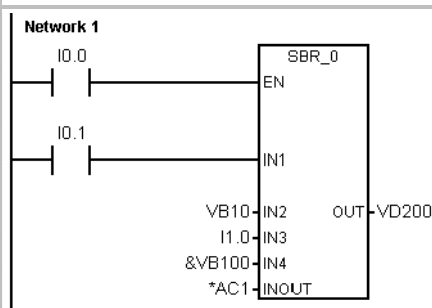
SIMATIC LAD				
	Název	Typ proměnné	Datový typ	Komentář
L0.0	Addr	IN	BOOL	Existence adresy
LB1	Value01	IN_OUT	BYTE	Hodnota parametru 01
L2.0	Ventil	OUT	BOOL	Stav ventilu
		OUT		

Obr. 6-37 Tabulka lokálních proměnných

Příklad: Volání podprogramu

Následují dva příklady v STL. První příklad může být zobrazen pouze editorem STL, protože parametry BOOL použité jako vstupy signálového toku nejsou uloženy do paměti L.

Druhý soubor instrukcí STL může být zobrazen také editory LAD a FBD, protože je použita paměť L pro uchování stavu vstupních BOOL parametrů, které jsou zobrazeny jako vstupy signálového toku v LAD a FBD.



Pouze STL:

Network 1

```
LD    IO.0
CALL  SBR_0, IO.1, VB10, I1.0, &VB100, *AC1, VD200
```

Pro správné zobrazení v LAD a FBD:

Network 1

```
LD    IO.0
=     L60,0
LD    IO.1
=     L63,7
LD    L60.0
CALL  SBR_0, L63.7, VB10, I1.0, &VB100, *AC1, VD200
```


Adresy jako parametry, například IN4 (&VB100), se do podprogramu odevzdávají jako hodnota DWORD (bez znaménka). Typ konstantního parametru musí být ještě před jeho použitím specifikován jako parametr ve volajícím podprogramu deskriptorem konstanty. Aby byla například konstanta double word bez znaménka o hodnotě 12.345 předána jako parametr, musí být konstantní parametr specifikován jako DW#12345. Pokud by se z parametru deskriptor konstanty vynechal, mohlo by se předpokládat, že konstanta je jiný typ.

Se vstupními a výstupními parametry se neprovádějí žádné automatické konverze typu dat. Jestliže například tabulka lokálních proměnných specifikuje, že parametr má typ dat REAL a ve volajícím podprogramu je pro tento parametr specifikováno double word (DWORD), bude hodnota v podprogramu double word.

Když se hodnoty odevzdávají do podprogramu, ukládají se do jeho lokální paměti. První sloupec tabulky lokálních proměnných uvádí adresu v lokální paměti pro každý předaný parametr. Když je podprogram volán, jsou hodnoty vstupních parametrů zkopírovány do jeho lokální paměti. Hodnoty výstupních parametrů jsou po ukončení provádění podprogramu zkopírovány z lokální paměti podprogramu na specifikované adresy výstupních parametrů.

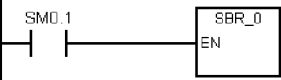
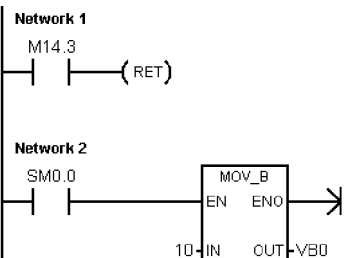
Velikost a typ datového prvku jsou dány parametrem. Přiřazení hodnot parametrů k lokální paměti v podprogramu je následující:

- ☐ Hodnoty parametrů jsou přiřazeny do lokální paměti v pořadí daném instrukcí volání podprogramu s parametry začínajícími na L.0.
- ☐ Jedna až osm za sebou jdoucích hodnot bitových parametrů jsou přiřazeny jednomu bytu na adrese Lx.0 až Lx.7.
- ☐ Hodnoty byte, word a double word jsou zapsány do lokální paměti podle bytů (LBx, LWx nebo LDx).

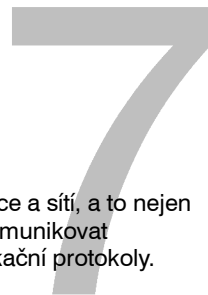
V případě instrukce "Volání podprogramu" s parametry musí být parametry uspořádány tak, aby byly na prvním místě vstupní parametry, za nimi vstupní/výstupní parametry a po nich parametry výstupní.

Pokud programujete v STL, je formát instrukce CALL následující:

CALL číslo podprogramu, parametr 1, parametr 2, ... , poslední parametr

Příklad: Volání podprogramu a návrat z podprogramu		
M A I N		Network 1 //V prvním programovém cyklu volá podprogram 0 pro //inicializaci. LD SM0.1 CALL SBR_0
		Network 1 //Před posledním networkem můžete použít podmíněný //návrat pro opuštění podprogramu. LD M14.3 CRET Network 2 //Pokud je M14.3 zapnutý, bude tento network //přeskočen LD SM0.0 MOVB 10, VB0

Síťová komunikace



Programovatelný automat S7-200 je určen k řešení potřeb uživatele v oblasti komunikace a sítí, a to nejen těch nejjednodušších, ale i složitějších. S7-200 je vybaven funkcemi, které umožňují komunikovat s jinými zařízeními, jako jsou například tiskárny a váhy, které užívají své vlastní komunikační protokoly.

Díky STEP 7 – Micro/WIN je nastavení a konfigurace sítě jednoduché a snadné.

Přehled kapitoly

Vysvětlení základů síťové komunikace S7-200	208
Výběr síťového komunikačního protokolu	211
Instalace a odstranění komunikačních rozhraní	216
Návrh sítě	218
Vytvoření protokolů definovaných uživatelem v režimu Freeport	222
Použití modemu a STEP 7 – Micro/WIN	224
Pro pokročilé	228

Vysvětlení základů síťové komunikace S7-200

Použití zařízení typu master a slave

Síť S7-200 pracuje v režimu master-slave, což znamená, že může fungovat jako master i jako slave, přičemž STEP 7 – Micro/WIN je vždy master.



Tip

Pokud používáte Windows NT a kabel PC/PPI, nesmí být v síti žádné jiné zařízení typu master.

Zařízení typu master

Zařízení, které v síti pracuje jako master, může generovat požadavek na jiné síťové zařízení. Master může také reagovat na požadavky jiných masterů, zapojených v síti. Typickými představiteli zařízení typu master jsou STEP 7 – Micro/WIN, zařízení pro rozhraní člověk – stroj, jako je např. TD 200 a programovatelné automaty S7-300 nebo S7-400. Programovatelný automat S7-200 funguje jako master tehdy, požaduje – li informace od jiného S7-200 (komunikace peer-to-peer).



Tip

TP070 nebude v síti spolupracovat s jiným zařízením typu master.

Zařízení typu slave

Zařízení, které je konfigurováno jako slave, může odpovídat pouze na požadavky masteru. Slave žádost nikdy negeneruje. Ve většině sítí funguje automat S7-200 jako slave. Ve funkci slave odpovídá S7-200 na požadavky masteru, jako je například operátorský panel nebo STEP 7 – Micro/WIN.

Nastavení přenosové rychlosti a síťové adresy

Rychlost, kterou jsou data přenášena po síti, se nazývá přenosová rychlost; měří se v kilobaudech (kbaud) nebo megabaudech (Mbaud). Přenosová rychlost udává, jaké množství dat je možné přenést za daný časový interval. Například přenosová rychlost 19,2 kbaud znamená přenos 19.200 bitů za sekundu.

Všechna zařízení, která komunikují prostřednictvím dané sítě, musí být nakonfigurována tak, aby přenášela data stejnou přenosovou rychlostí. Proto je nejvyšší přenosová rychlost dána nejpomalejším zařízením, připojeným do sítě.

Tabulka 7-1 uvádí přenosové rychlosti podporované automatem S7-200.

Síťová adresa je jedinečné číslo, které přidělíte každému zařízení v síti. Adresa zajišťuje, aby byla data přenášena nebo vyhledávána ve správném zařízení. Programovatelný automat S7-200 podporuje síťové adresy od 0 do 126. V případě S7-200 se dvěma porty může mít každý z nich vlastní síťovou adresu. Tabulka 7-2 uvádí implicitní nastavení adres pro automaty S7-200.

Tabulka 7-1 Přenosové rychlosti podporované S7-200

Síť	Přenosová rychlost
Standardní síť	9.6 kbaud až 187,5 kbaud
S použitím modulu EM 277	9.6 kbaud až 12 Mbaud
Režim Freeport	1200 baud až 115,2 kbaud

Tabulka 7-2 Implicitně nastavené adresy automatů S7-200

S7-200	Implicitní adresa
STEP 7 – Micro/WIN	0
HMI (TD 200, TP nebo OP)	1
S7-200 CPU	2

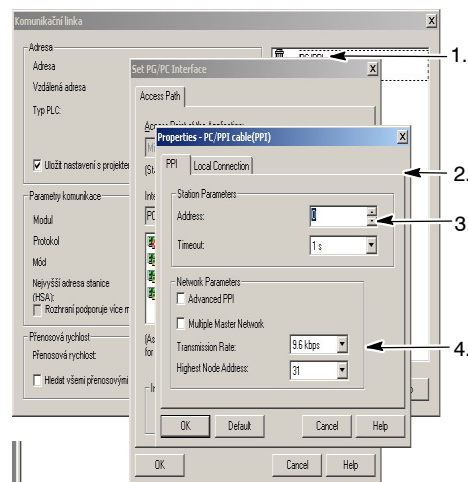
Nastavení přenosové rychlosti a síťové adresy pro STEP 7–Micro/WIN

Pro STEP 7–Micro/WIN je třeba nastavit adresu a přenosovou rychlost. Přenosová rychlost musí být stejná, jako mají ostatní zařízení v síti, síťová adresa musí být jedinečná.

Běžně není nutné síťovou adresu STEP 7–Micro/WIN (0) měnit. Pokud ale síť obsahuje jiný programovací SW, například STEP 7, může být nutné změnit síťovou adresu STEP 7–Micro/WIN.

Jak je vidět z obrázku 7-1, nastavení přenosové rychlosti a síťové adresy pro STEP 7–Micro/WIN je jednoduché. Nejprve klikněte na ikonu "Komunikace" v menu "Navigace" a proveďte následující kroky:

1. Dvakrát klikněte na ikonu v okně "Nastavení komunikace".
2. V dialogovém okně "Nastavení rozhraní PG/PC" klikněte na tlačítko "Vlastnosti".
3. Vyberte síťovou adresu pro STEP 7–Micro/WIN.
4. Vyberte přenosovou rychlost pro STEP 7–Micro/WIN.



Obr. 7-1 Nastavení STEP 7–Micro/WIN

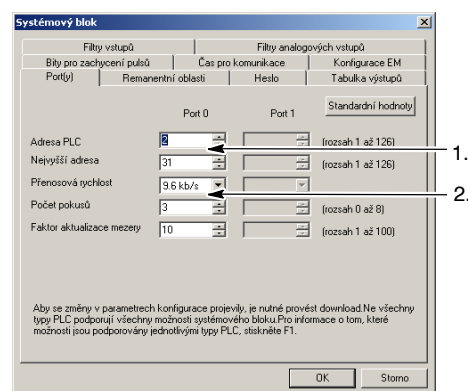
Nastavení přenosové rychlosti a síťové adresy pro S7-200

Přenosovou rychlost a síťovou adresu je nutné nastavit také pro S7-200. Rychlost i adresa jsou uloženy v systémovém bloku S7-200. Po nastavení parametrů musíte systémový blok downloadovat do S7-200.

Implicitně nastavená rychlost portů S7-200 je 9,6 kbaud, implicitní síťová adresa je 2.

Jak je vidět na obrázku 7-2, pro nastavení přenosové rychlosti a síťové adresy S7-200 se používá STEP 7–Micro/WIN. Nejprve vyberte ikonu "Systémový blok" nebo vyberte příkaz menu **Zobrazit > Komponenty > Systémový blok** a proveďte následující kroky:

1. Vyberte síťovou adresu.
2. Vyberte přenosovou rychlost.
3. Proveďte download systémového bloku.



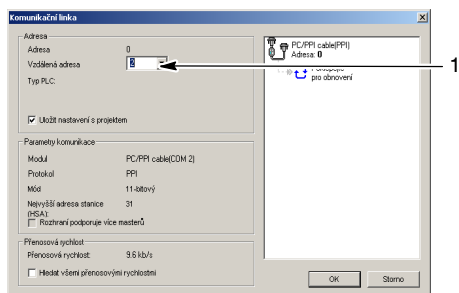
Obr. 7-2 Nastavení CPU S7-200

Nastavení vzdálené adresy

Před downloadem nového nastavení do S7-200 musíte nastavit jak komunikační (COM) port STEP 7–Micro/WIN, tak vzdálenou adresu S7-200 tak, aby odpovídaly aktuálnímu nastavení vzdáleného S7-200. Viz obrázek 7-3.

Po provedení downloadu nového nastavení musíte překonfigurovat COM port (pokud se liší od nastavení použitého pro download vzdáleného S7-200). Abyste otevřeli dialogové okno "Komunikace", klikněte na ikonu "Komunikace" v menu "Navigace" nebo vyberte příkaz menu **Zobrazit > Komponenty > Komunikace**.

1. Vyberte vzdálenou adresu.
2. Zkontrolujte, zda odpovídají parametry (přenosová rychlost) COM portu, portu vzdáleného S7-200 a kabelu PC/PPI. Jinak komunikace neproběhne.



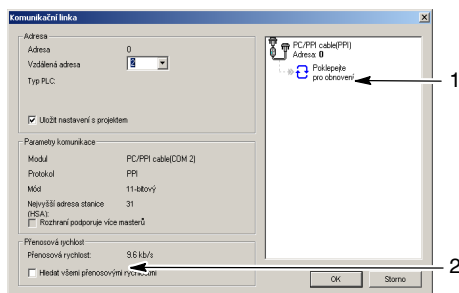
Obr. 7-3 Nastavení CPU S7-200

Vyhledávání CPU S7-200 v síti

CPU S7-200, které jsou k síti připojeny, je možné vyhledávat a identifikovat. Při vyhledávání je možné prohledávat síť určenou buď přenosovou rychlostí, nebo všemi přenosovými rychlostmi.

Pokud použijete kabel PC/PPI, může STEP 7–Micro/WIN vyhledávat pouze rychlostmi 9,6 kbaud a 19,2 kbaud. V případě CP karty hledá STEP 7–Micro/WIN rychlostmi 9,6 kbaud, 19,2 kbaud a 187,5 kbaud. Vyhledávání začne takovou přenosovou rychlostí, která je právě nastavena.

1. Chcete-li zahájit vyhledávání, otevřete dialogové okno "Komunikace" a dvakrát klikněte na ikonu "Obnovit".
2. Pro vyhledávání ve všech přenosových rychlostech vyberte zaškrtnuté políčko "Hledat všemi přenosovými rychlostmi".



Obr. 7-4 Vyhledávání CPU v síti

Výběr síťového komunikačního protokolu

CPU S7-200 podporuje jednu nebo více následujících typů komunikace, které umožňují nakonfigurovat síť na takový výkon a funkčnost, jakou vaše aplikace vyžaduje:

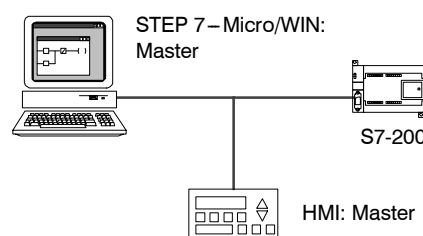
- ☐ Dvoubodové spojení (PPI)
- ☐ Mnohobodové spojení (MPI)
- ☐ PROFIBUS

Tyto protokoly jsou založeny na sedmivrstvé architektuře (OSI). Jsou implementovány v síti s kruhovou topologií, která odpovídá standardu PROFIBUS tak, jak je definován evropskou normou EN 50170. Tyto protokoly jsou asynchronní, znakově orientované s jedním start bitem, osmi datovými bity, sudou paritou a jedním stop bitem. Přenosové rámce obsahují speciální znaky start a stop, adresy zdrojové a cílové stanice, délky u rámce a kontrolní součet. Pokud mají všechny protokoly stejnou přenosovou rychlost, mohou v síti běžet, aniž by se vzájemně rušily.

Protokol PPI

PPI je protokol typu master-slave: zařízení master posílají požadavky zařízením slave, které jim odpovídají. Viz obr. 7-5. Zařízení slave negenerují zprávy, ale čekají, dokud master nepošle požadavek nebo je nevyzve k odpovědi.

Zařízení master komunikují se zařízeními slave prostřednictvím sdíleného spojení, které je řízeno protokolem PPI. Tento protokol neomezuje počet masterů, které mohou komunikovat s kterýmkoliv zařízením slave; není však možné instalovat více než 32 masterů.



Obr. 7-5 Síť PPI

Volba PPI Advanced umožňuje navázat vzájemné logické spojení mezi zařízeními. V případě PPI Advanced má každé zařízení omezený počet připojení. Tabulka 7-3 uvádí počet připojení podporovaných S7-200.

CPU S7-200 se mohou v režimu RUN chovat jako master, pokud v uživatelském programu povolíte režim PPI master. (Viz popis SMB30 v příloze D.) Po povolení režimu PPI master můžete použít instrukce "Čtení ze sítě" pro čtení nebo "Zápis do sítě" pro zápis do jiných jednotek S7-200. I když pracuje automat jako PPI master, stále reaguje jako zařízení typu slave na požadavky jiných masterů.

Protokol PPI můžete použít pro komunikaci se všemi CPU S7-200. Pro komunikaci s EM 277 musíte aktivovat PPI Advanced.

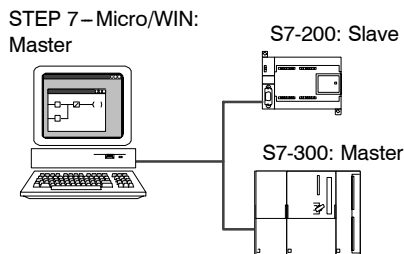
Tabulka 7-3 Počet spojení CPU S7-200 a modulů EM 277

Modul	Přenosová rychlost	Počet spojení
CPU S7-200,	Port 0	9,6 kbaud, 19,2 kbaud nebo 187,5 kbaud
	Port 1	9,6 kbaud, 19,2 kbaud nebo 187,5 kbaud
Modul EM 277	9,6 kbaud až 12 Mbaud	6 na modul

Protokol MPI

Protokol MPI umožňuje komunikaci master-master i master-slave. Viz obr. 7-6. Pro komunikaci s CPU S7-200 naváže STEP 7 – Micro/WIN spojení master – slave. Protokol MPI nepodporuje CPU S7-200 ve funkci master.

Síťová zařízení spolu komunikují prostřednictvím oddělených spojení (řízených protokolem MPI) mezi libovolnými dvěma zařízeními. Komunikace mezi zařízeními je omezena na počet spojení podporovaných CPU S7-200 nebo moduly EM 277. Počet spojení, podporovaných S7-200, uvádí tabulka 7-3.



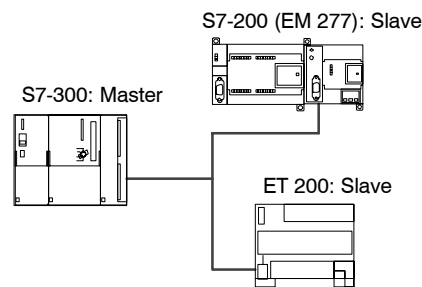
Obr. 7-6 Síť MPI

Pro čtení a zápis dat používají programovatelné automaty S7-300 a S7-400 v protokolu MPI instrukce XGET a XPUT. Informace o těchto instrukcích najdete v příručce pro programování S7-300 nebo S7-400.

Protokol PROFIBUS

Protokol PROFIBUS je určen pro vysokorychlostní komunikaci s distribuovanými vstupními/výstupními zařízeními (vzdálené vstupy/výstupy). Mnoho různých výrobců nabízí řadu zařízení PROFIBUS. Tato zařízení se pohybují od jednoduchých vstupních nebo výstupních modulů až po regulátory motorů a programovatelné automaty.

Sítě PROFIBUS běžně obsahují jedno zařízení typu master a několik vstupních/výstupních zařízení typu slave. Viz obr. 7-7. Zařízení typu master je nakonfigurováno tak, že zná, jaké typy vstupních/výstupních zařízení typu slave jsou k němu připojeny a jaké mají adresy. Master inicializuje síť a ověřuje, zda zařízení typu slave zapojená do sítě odpovídají konfiguraci. Master nepřetržitě zapisuje výstupní data do zařízení typu slave a čte z nich data vstupní.



Obr. 7-7 Síť PROFIBUS

Po úspěšné konfiguraci zařízení typu slave DP masterem je slave přiřazen určitému masteru. Pokud je v síti druhý master, má jen velmi omezený přístup k zařízením typu slave přiřazeným prvnímu masteru.

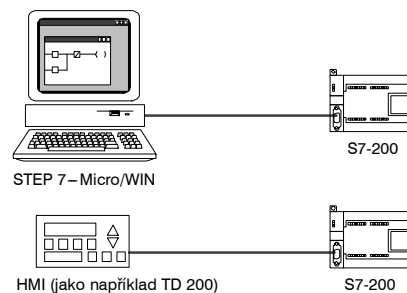
Vzorové konfigurace sítě s použitím pouze automatů S7-200

Síť PPI s jedním masterem

V síti s jedním masterem jsou počítač a CPU S7-200 spojeny buď kabelem PC/PPI, nebo kartou komunikačního procesoru (CP karta), instalovanou v počítači.

Na obrázku 7-8 nahoře je počítač (STEP 7 – Micro/WIN) master této sítě. Na obrázku 7-8 dole funguje jako master zařízení pro rozhraní člověk – stroj (HMI) (jako například TD 200, TP nebo OP).

V obou vzorových příkladech pracuje CPU S7-200 jako slave, který reaguje na požadavky masteru.



Obr. 7-8 Síť PPI s jedním masterem

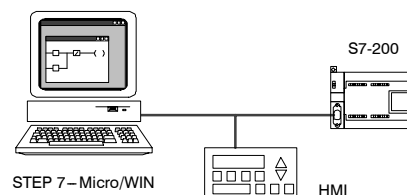
V případě použití PPI s jedním masterem konfiguruje STEP 7 – Micro/WIN pro používání protokolu PPI: vyberte jeden nebo více masterů nebo PPI Advanced.

Síť PPI s více mastery

Na obrázku 7-9 je příklad sítě obsahující více zařízení typu master a jednoho slave. Síť sdílí počítač, který pro STEP 7 – Micro/WIN používá CP kartu, nebo kabel PC/PPI, a zařízení HMI.

Jak STEP 7 – Micro/WIN, tak zařízení HMI jsou zařízení typu master a musí mít svoje vlastní síťové adresy. CPU S7-200 je zařízení typu slave.

V případě sítě, v níž přistupuje více masterů k jedinému zařízení typu slave, konfiguruje STEP 7 – Micro/WIN tak, aby používal protokol PPI s aktivovaným ovladačem "multi-master". Optimální je PPI Advanced.

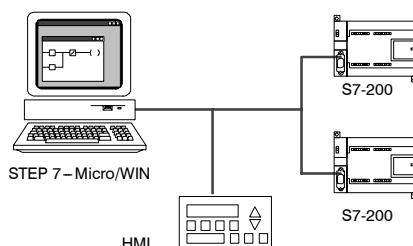


Obr. 7-9 Více masterů s jedním slave

Na obrázku 7-10 je síť PPI, v níž více masterů komunikuje s více zařízeními typu slave. V tomto příkladu může jak STEP 7 – Micro/WIN, tak HMI požadovat data od jakékoliv CPU S7-200 ve funkci slave. STEP 7 – Micro/WIN a zařízení HMI síť sdílejí.

Všechna zařízení (typu master i slave) mají různé adresy.

V případě sítě PPI s více zařízeními typu master i slave konfiguruje STEP 7 – Micro/WIN tak, aby používal protokol PPI s aktivovaným ovladačem "multi-master". Optimální je PPI Advanced.



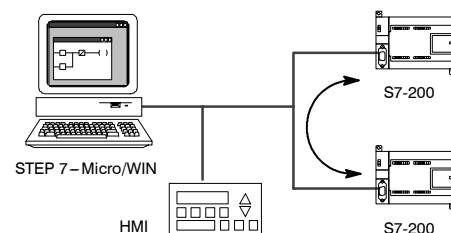
Obr. 7-10 Více zařízení typu master i slave

Složité síť PPI

Na obrázku 7-11 je vidět příklad sítě, která používá více zařízení typu master pro dvoubodovou komunikaci (peer-to-peer).

STEP 7 – Micro/WIN a zařízení HMI čtou a zapisují data do CPU S7-200; CPU S7-200 používají instrukce "Čtení ze sítě" a "Zápis do sítě" pro vzájemné čtení a zápis (dvoubodová komunikace – peer-to-peer).

Pro tento typ složité sítě PPI konfiguruje STEP 7 – Micro/WIN tak, aby používal protokol PPI s aktivovaným ovladačem "multi-master". Optimální je PPI Advanced.

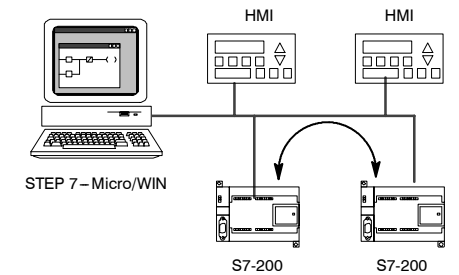


Obr. 7-11 Peer-to-peer komunikace

Na obrázku 7-12 vidíme další příklad složité sítě PPI, která používá více masterů pro dvoubodovou komunikaci (peer-to-peer). V tomto příkladu každé HMI monitoruje jedno CPU S7-200.

CPU S7-200 používá pro vzájemné čtení a zápis instrukce NETR a NETW (komunikace peer-to-peer).

Pro tento typ sítě nakonfigurujte STEP 7 – Micro/WIN tak, aby používal protokol PPI s aktivovaným ovladačem "multi-master". Optimální je PPI Advanced.



Obr. 7-12 Zařízení HMI a komunikace peer-to-peer

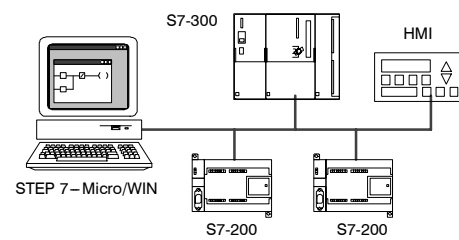
Příklady konfigurací sítě s použitím S7-200, S7-300 a S7-400

Sítě s přenosovou rychlostí až do 187,5 kbaud

U příkladu sítě na obrázku 7-13 používá S7-300 pro komunikaci s CPU S7-200 instrukce XPUT a XGET. S7-300 nemůže komunikovat s CPU S7-200 v režimu master.

Pro přenosové rychlosti vyšší než 19,2 kbaud musí být STEP 7 – Micro/WIN připojen kartou komunikačního procesoru (CP).

Pro komunikaci s CPU S7-200 nakonfigurujte STEP 7 – Micro/WIN tak, aby používal protokol PPI s aktivovaným ovladačem "multi-master". Optimální je PPI Advanced.



Obr. 7-13 Rychlost přenosu do 187,5 kbaud

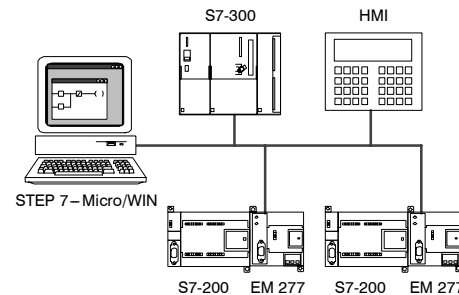
Sítě s přenosovými rychlostmi nad 187,5 kbaud (až do 12 Mbaud)

Pro přenosové rychlosti vyšší než 187,5 kbaud musí CPU S7-200 použít pro připojení k síti modul EM 277. Viz obr. 7-14. STEP 7 – Micro/WIN musí být připojen kartou komunikačního procesoru (CP).

V této konfiguraci může S7-300 komunikovat s automaty S7-200 pomocí instrukcí XPUT a XGET a HMI může monitorovat buď automaty S7-200, nebo S7-300.

EM 277 je vždy zařízení typu slave.

STEP 7 – Micro/WIN může prostřednictvím EM 277 programovat nebo monitorovat kterýkoliv CPU S7-200. Pro komunikaci s EM 277 nakonfigurujte STEP 7 – Micro/WIN tak, aby používal protokol PPI s aktivovaným ovladačem PPI Advanced.



Obr. 7-14 Rychlost přenosu nad 187,5 kbaud

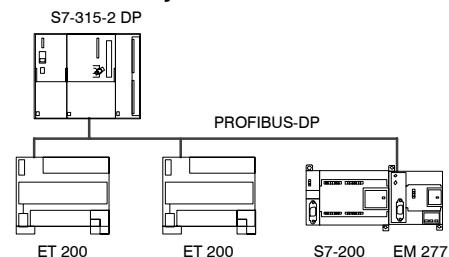
Příklady konfigurací sítě PROFIBUS-DP

Sítě s S7-315–2 DP jako zařízením PROFIBUS master a EM 277 jako PROFIBUS slave

Na obrázku 7-15 je příklad sítě PROFIBUS, která používá S7-315–2 DP jako PROFIBUS master. Modul EM 277 je PROFIBUS slave.

S7-315–2 DP může z EM 277 data číst nebo zapisovat v rozsahu od 1 bytu až do 128 bytů. S7-315–2 DP používá pro čtení a zápis paměti V automatu S7-200.

Tato síť podporuje přenosové rychlosti od 9600 baud do 12 Mbaud.



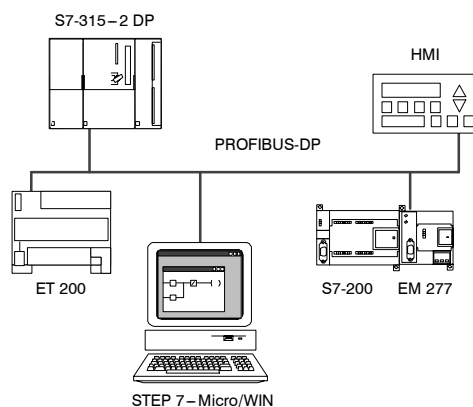
Obr. 7-15 Síť s S7-315–2 DP

Sítě se STEP 7–Micro/WIN a HMI

Na obrázku 7-16 je příklad sítě s S7-315–2 DP jako PROFIBUS master a EM 277 jako PROFIBUS slave. V této konfiguraci monitoruje HMI automat S7-200 přes EM 277. STEP 7–Micro/WIN programuje S7-200 přes EM 277.

Tato síť podporuje přenosové rychlosti od 9600 baud do 12 Mbaud. STEP 7–Micro/WIN vyžaduje pro přenosové rychlosti kartu CP vyšší než 19,2 kbaud.

STEP 7–Micro/WIN nakonfiguruje tak, aby pro kartu CP používal protokol PROFIBUS. Pokud jsou v síti pouze zařízení DP, vyberte profil DP nebo Standard. Pokud jsou v síti jiná zařízení než DP, vyberte pro všechna zařízení PROFIBUS master profil Universal (DP/FMS).



Obr. 7-16 Síť PROFIBUS

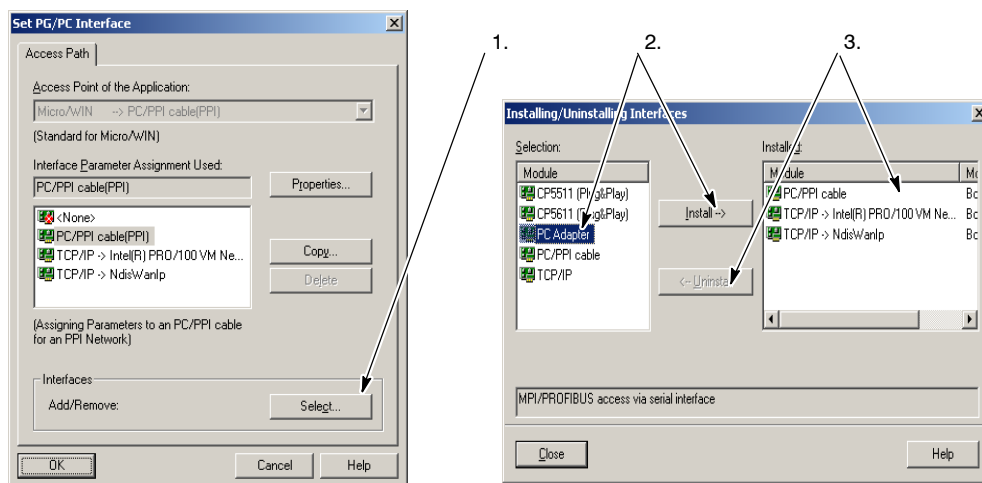
Instalace a odstranění komunikačních rozhraní

Z dialogového okna "Set PG/PC Interface" (Nastavení rozhraní PG/PC) použijte pro instalaci nebo odstranění komunikačních rozhraní dialogové okno "Installing/Uninstalling Interfaces" (Instalace/odstranění rozhraní)

1. V dialogovém okně "Set PG/PC Interface" klikněte na "Select" (Vybrat); objeví se dialogové okno "Installing/Uninstalling Interfaces".

V poli "Selection" (Výběr) je seznam dostupných rozhraní, v poli "Installed" (Instalováno) jsou uvedena rozhraní, která už jsou v počítači nainstalována.

2. *Chcete-li přidat komunikační rozhraní:* Vyberte komunikační hardware a klikněte na "Install" (Instalovat). Po zavření dialogového okna "Installing/Uninstalling Interfaces" se zobrazí dialogové okno "Set PG/PC Interface" daného rozhraní v poli "Interface Parameter Assignment Used" (Použité přiřazení parametrů rozhraní).
3. *Chcete-li odstranit komunikační rozhraní:* Vyberte rozhraní, které má být odstraněno, a klikněte na "Uninstall" (Odstranit). Po zavření dialogového okna "Installing/Uninstalling Interfaces" se dané rozhraní odstraní z pole "Interface Parameter Assignment Used" v dialogovém okně "Set PG/PC Interface".



Obr. 7-17 Nastavení rozhraní PG/PC a instalace/odstranění dialogových oken pro rozhraní



Tip

Speciální informace pro instalaci hardwaru pro uživatele Windows NT

Instalace hardwarových modulů pod operačním systémem Windows NT je mírně odlišná od instalace ve Windows 95. Přestože jsou hardwarové moduly pro oba systémy stejné, vyžaduje instalace ve Windows NT více znalostí o hardwaru, který chcete instalovat. Windows 95 se za vás automaticky snaží nastavit systémové prostředky, což Windows NT nečiní. Windows NT pouze nabídne implicitní hodnoty. Tyto hodnoty mohou, ale nemusí odpovídat konfiguraci hardwaru. Parametry je však možné snadno upravit tak, aby odpovídaly potřebnému nastavení systému.

Po instalaci hardwarového zařízení jej vyberte v poli "Installed" a klikněte na tlačítko "Resources" (Prostředky). Objeví se dialogové okno "Resources". V tomto dialogovém okně můžete měnit systémová nastavení hardwaru, který jste právě nainstalovali. Pokud je toto tlačítko nepřístupné (šedé), nemusíte už dělat nic víc.

V tomto bodě budete možná muset nahlédnout do příručky instalovaného hardwaru, abyste zjistili nastavení všech parametrů, uvedených v dialogovém okně. Možná budete muset vyzkoušet několik různých přerušení, abyste komunikaci správně dokončili.

Úprava nastavení portů počítače pro PPI Multi-Master

Pokud použijete kabel PC/PPI s operačním systémem, který podporuje konfiguraci PPI Multi-Master (Windows NT PPI Multi-Master nepodporují), budete pravděpodobně muset upravit nastavení portů na svém počítači:

1. Klikněte pravým tlačítkem myši na ikonu "Tento počítač" na ploše a vyberte **Vlastnosti**.
2. Vyberte záložku "Správce zařízení". U Windows 2000 nejprve vyberte záložku "Hardware" a pak tlačítko "Správce zařízení".
3. Dvakrát klikněte na "Porty" (COM & LPT).
4. Vyberte komunikační port, který aktuálně používáte (například COM1).
5. V záložce "Nastavení portu" klikněte na tlačítko "Upřesnit".
6. Nastavte hodnotu pro "Přijímací vyrovnávací paměť" a "Přenosová vyrovnávací paměť" na nejnižší hodnotu (1).
7. Klikněte na "OK", zavřete všechna okna a počítač restartujte, aby se nové nastavení projevilo.

Návrh sítě

Obecné pokyny

Vždy instalujte přepětovou ochranu pro všechny okruhy, které by mohly být zasaženy bleskem.

Neumísťujte nízkonapětové signálové vodiče a komunikační kabely do stejné lišty s napájecími kabely střídavého napětí a vysokonapětovými vodiči stejnosměrného napětí, které jsou rychle spínány. Vždy pokládejte vodiče v párech s nulovým nebo společným vodičem, spárovaným s živým nebo signálovým vodičem.

Komunikační port CPU S7-200 není napětově oddělen. Abyste zajistili oddělení sítě, zvažte použití repeateru RS-485 nebo modulu EM 277.

Upozornění

Jestliže propojíte zařízení s různými referenčními potenciály, mohou spojovacím kabelem protékat nežádoucí proudy.

Tyto nežádoucí proudy mohou způsobit chyby při komunikaci nebo poškodit zařízení.

Zkontrolujte, zda všechny části zařízení, které chcete propojit komunikačním kabelem, mají buď společný referenční potenciál, nebo jsou odděleny, aby se zabránilo nežádoucím proudům. Více informací o uzemnění a referenčních bodech oddělených obvodů najdete v kapitole 3.

Stanovení vzdáleností, přenosové rychlosti a kabelů

Jak je vidět z tabulky 7-4, maximální délka segmentu sítě je dána dvěma faktory: napětovým oddělením (použití zesilovače RS-485) a přenosovou rychlostí.

Jestliže spojujete zařízení s různými potenciály země, je nutné je galvanicky oddělit. Tyto různé potenciály se mohou vyskytnout v případě, kdy jsou země fyzicky odděleny dlouhou vzdáleností. Na krátkých vzdálenostech mohou rozdíly potenciálů způsobit také napájecí proudy velkých strojů.

Tabulka 7-4 Maximální délka kabelu v síti

Přenosová rychlost	Neoddělený - port CPU ¹	Port CPU s repeaterem nebo modulem EM 277
9,6 kbaud až 187,5 kbaud	50 m	1 000 m
500 kbaud	Nepodporuje	400 m
1 Mbaud až 1,5 Mbaud	Nepodporuje	200 m
3 Mbaud až 12 Mbaud	Nepodporuje	100 m

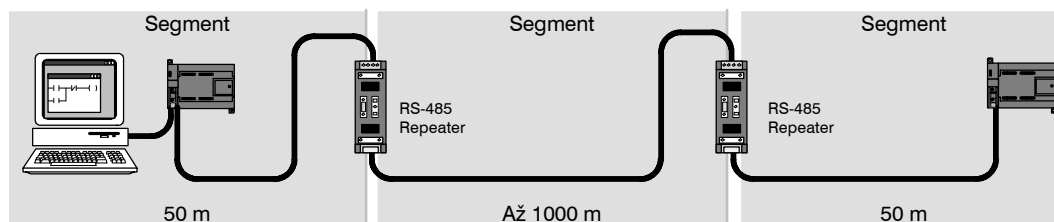
¹ Maximální povolená vzdálenost bez použití oddělení nebo repeateru je 50 m. Tato vzdálenost se měří od prvního k poslednímu uzlu v segmentu.

Použití repeaterů

Repeater RS-485 zajišťuje symetrizaci a zakončení segmentů sítě. Je možné ho použít k následujícím účelům:

- ☐ *Pro prodloužení délky sítě:* Přidání repeateru do sítě umožní její prodloužení o dalších 50 m. Pokud propojíte dva repeater, mezi nimiž neleží žádné jiné uzly (jak je to znázorněno na obrázku 7-18), můžete síť prodloužit na maximální délku kabelu pro danou přenosovou rychlost. Do sítě můžete sériově zapojit až 9 repeaterů, ale celková délka sítě nesmí přesáhnout 9600 m.
- ☐ *Pro přidání zařízení do sítě:* Každý segment může mít maximálně 32 zařízení, propojených do vzdálenosti 50 m při rychlosti 9600 baud. Použití repeateru umožní přidat do sítě další segment (32 zařízení).
- ☐ *Pro oddělení různých segmentů sítě:* Oddělení sítě zlepšuje kvalitu přenosu tím, že oddělí síťové segmenty, které mohou mít různé zemní potenciály.

Repeater se v síti počítá za jeden z uzlů v segmentu, i když nemá přiřazenou síťovou adresu.



Obr. 7-18 Příklad sítě s repeaterem

Výběr síťového kabelu

Sítě S7-200 používají pro standard RS-485 kroucenou dvojlinku. Tabulka 7-5 uvádí požadované vlastnosti kabelu. V síťovém segmentu může být zapojeno až 32 zařízení.

Tabulka 7-5 Obecné vlastnosti kabelů

Vlastnosti	Popis
Typ kabelu	Stíněná kroucená dvojlinka
Odpor smyčky	$\leq 115 \Omega/\text{km}$
Kapacita	30 pF/m
Jmenovitá impedance	Přibližně 135Ω až 160Ω (<i>frekvence</i> = 3 MHz až 20 MHz)
Útlum	0.9 dB/100 m (<i>frekvence</i> = 200 kHz)
Průřez kabelu	0,3 mm ² až 0,5 mm ²
Průměr kabelu	8 mm $\pm$ 0,5 mm

Rozmístění vývodů konektoru

Komunikační porty na CPU S7-200 jsou kompatibilní s RS-485 a jsou realizovány jako miniaturní konektory typu D s devíti vývody podle standardu PROFIBUS, jak je definován v evropské normě EN 50170.

Tabulka 7-6 ukazuje konektor, který zajišťuje fyzické spojení komunikačního portu, a popisuje rozmístění vývodů komunikačního portu.

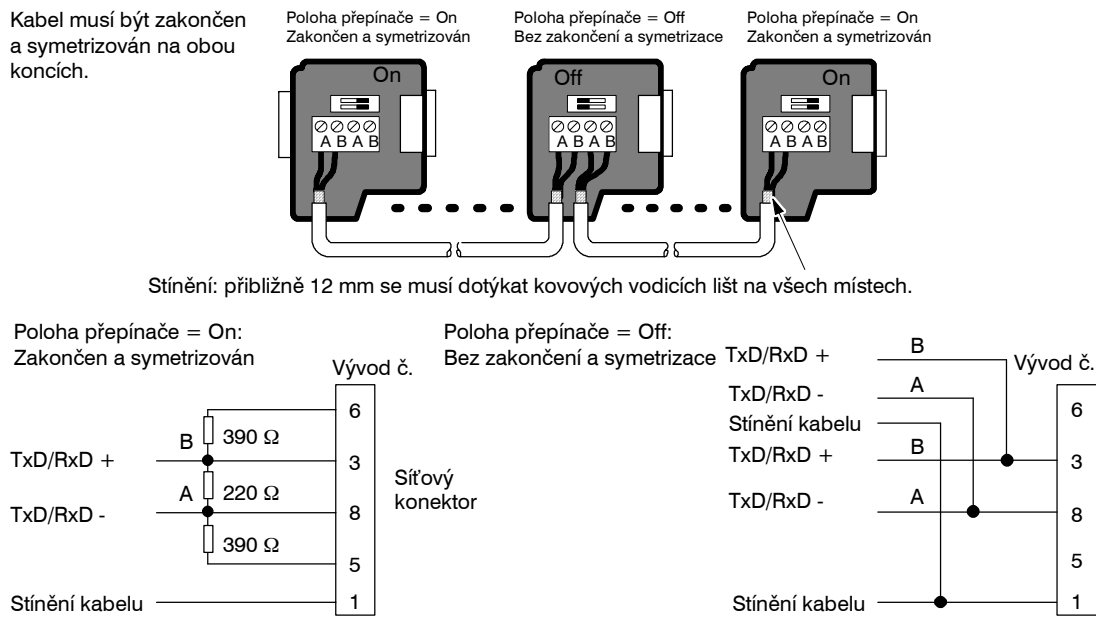
Tabulka 7-6 Rozmístění vývodů v komunikačním portu automatu

Konektor	Číslo vývodu	PROFIBUS Signal	Port 0/Port 1
	1	Stínění	Uzemnění skříně
	2	24 V zpětný	Logický společný
	3	RS-485 signál B	RS-485 signál B
	4	Požadavek na vysílání	RTS (TTL)
	5	5 V zpětný	Logický společný
	6	+5 V	+5 V, 100 Ω sériový odpor
	7	+24 V	+24 V
	8	RS-485 signál A	RS-485 signál A
	9	Nepoužitý	10bitová volba protokolu (vstup)
Pouzdro konektoru		Stínění	Uzemnění skříně

Symetrizace a zakončení síťového kabelu

Siemens nabízí dva typy síťových konektorů, které lze použít pro snadné připojení více zařízení k síti: standardní síťový konektor (rozmístění vývodů najdete v tabulce 7-6) a konektor, který má programovací port, umožňující připojení programovací stanice nebo zařízení HMI do sítě, aniž by se musela rozebírat stávající kabeláž sítě. Konektor s programovacím portem předává všechny signály (včetně výkonových vývodů) z S7-200 přes programovací port, což je obzvláště užitečné pro připojení zařízení, která jsou napájena z S7-200 (jako například TD 200).

Oba konektory mají dvě sady šroubovacích svorkovnic, které umožňují připojení vstupních a výstupních síťových kabelů. Oba konektory také mají přepínače pro nastavení symetrizace a zakončení sítě. Obrázek 7-19 ukazuje typickou symetrizaci a zakončení kabelových konektorů.



Obr. 7-19 Symetrizace a zakončení síťového kabelu

Volba karty CP nebo kabelu PC/PPI

Jak ukazuje tabulka 7-7, STEP 7–Micro/WIN podporuje několik karet CP, které umožňují, aby se počítač (nebo programovací zařízení SIMATIC) choval v síti jako master.

Karty CP obsahují specializovaný hardware, který pomáhá počítači při řízení sítě s více mastery a který podporuje různé protokoly s různými přenosovými rychlostmi. Kabel PC/PPI také umožňuje použití více masterů.

Každá karta CP má jeden port RS-485 pro připojení k síti. Karta CP 5511 PCMCIA má adaptér pro D konektor s devíti vývody. Jeden konec kabelu se připojí do portu RS-485 komunikační karty, druhý konec k programovacímu portu síťového konektoru.

Pokud použijete kartu CP s komunikací PPI, nepodporuje STEP 7–Micro/WIN současný běh dvou různých aplikací na stejné kartě CP. Druhou aplikaci musíte zavřít dříve, než připojíte STEP 7–Micro/WIN prostřednictvím karty CP k síti.

Upozornění

Použití neodděleného převodníku RS-485 na RS-232 může poškodit port RS-232 počítače.

PC/PPI kabel Siemens (objednávací číslo 6ES7 901–3BF21–0XA0) zajišťuje elektrické oddělení portu RS-485 na CPU S7-200 a portu RS-232 počítače. Pokud nepoužijete PC/PPI kabel Siemens, musíte zajistit oddělení portu RS-232 počítače.

Tabulka 7-7 Karty CP a protokoly podporované STEP 7–Micro/WIN

Konfigurace	Přenosová rychlost	Protokol
PC/PPI kabel ¹ Připojen do portu COM na počítači	9,6 kbaud nebo 19,2 kbaud	PPI
CP 5511 Typ II, karta PCMCIA (pro notebook)	9,6 kbaud až 12 Mbaud	PPI, MPI a PROFIBUS
CP 5611 (verze 3 nebo vyšší) Karta PCI	9,6 kbaud až 12 Mbaud	PPI, MPI a PROFIBUS
MPI Buď integrovaný port na programovacím zařízení SIMATIC, nebo karta CP pro počítač (ISA karta)	9,6 kbaud až 12 Mbaud	PPI, MPI a PROFIBUS

¹ PC/PPI kabel zajišťuje elektrické oddělení portu RS-485 (na CPU S7-200) a portu RS-232 na vašem počítači. Použití neodděleného převodníku RS-232/485 může poškodit port RS-232 počítače.

Použití zařízení HMI v síti

CPU S7-200 podporuje mnoho typů zařízení HMI jak od firmy Siemens, tak i od jiných výrobců. Některá zařízení HMI (jako například TD 200 nebo TP070) neumožňují výběr komunikačního protokolu, který používají, ale jiná zařízení (jako například OP7 a TP170) ano.

Pokud zařízení HMI výběr komunikačního protokolu umožňuje, vezměte v úvahu následující pokyny:

- ☐ Pro zařízení HMI připojené ke komunikačnímu portu CPU S7-200 v síti, která neobsahuje žádná jiná zařízení, zvolte protokol PPI nebo MPI.
- ☐ Pro zařízení HMI připojené k modulu EM 277 PROFIBUS vyberte buď protokol MPI, nebo PROFIBUS.
 - Pokud jsou v síti spolu se zařízením HMI programovatelné automaty S7-300 nebo S7-400, zvolte pro zařízení HMI protokol MPI.
 - Pokud je zařízení HMI součástí sítě PROFIBUS, zvolte pro toto zařízení protokol PROFIBUS a zvolte stejný profil, jaký mají ostatní mastery v dané síti PROFIBUS.
- ☐ Pro zařízení HMI připojené na komunikační port CPU S7-200, které je nakonfigurováno jako zařízení typu master, zvolte protokol PPI. Optimální je PPI Advanced. Protokoly MPI a PROFIBUS nepodporují CPU S7-200 jako master.

Vytvoření protokolů definovaných uživatelem v režimu Freeport

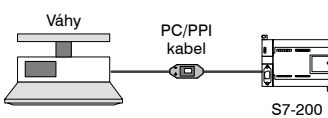
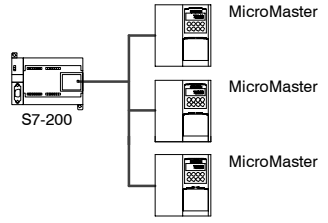
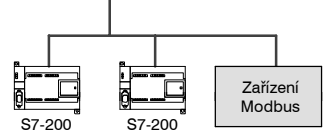
Režim Freeport umožňuje programově řídit komunikační port CPU S7-200. Je možné ho použít pro implementaci komunikačních protokolů, definovaných uživatelem při komunikaci s mnoha typy inteligentních zařízení. Režim Freeport podporuje ASCII protokol i binární protokoly.

Pro aktivaci režimu Freeport se používají byty speciální paměti SMB30 (pro port 0) a SMB130 (pro port 1). Uživatelský program používá pro řízení komunikačního portu následující:

- ☐ Instrukce a přerušení ("Vysílání", XMT) pro vysílání: instrukce "Vysílání" umožňuje přenést až 255 znaků. Přerušení informuje program v S7-200 o tom, že byl přenos ukončen.
- ☐ Přerušení od příjmu znaku: přerušení od příjmu znaku informuje uživatelský program o tom, že byl na portu COM přijat znak. Program pak může tento znak zpracovat podle implementovaného protokolu.
- ☐ Instrukce příjmu ("Příjem", RCV): instrukce "Příjem" přijme celou zprávu z portu COM; po jejím přijetí generuje přerušení uživatelského programu. Pro konfiguraci začátku a konce přijímání zpráv na základě definovaných podmínek se používá paměť SM automatu S7-200. Instrukce "Příjem" umožňuje uživatelskému programu zahájit nebo zastavit zprávu na základě konkrétních znaků nebo časových intervalů. Pomocí instrukce "Příjem" může být implementována většina protokolů.

Režim Freeport je aktivní pouze tehdy, je-li S7-200 v režimu RUN. Převedení automatu S7-200 do režimu STOP zastaví veškerou komunikaci Freeport a komunikační port se pak vrací do protokolu PPI s nastaveními, která byla konfigurována v systémovém bloku automatu S7-200.

Tabulka 7-8 Použití režimu Freeport

Konfigurace sítě		Popis
Použití při připojení přes RS-232		Příklad: Použití S7-200 s elektronickými vahami, které mají port RS-232. ■ Kabel PC/PPI spojuje port RS-232 na vahách s portem RS-485 CPU S7-200. ■ CPU S7-200 používá pro komunikaci s vahami režim Freeport. ■ Přenosová rychlost se může pohybovat od 1200 baud do 115,2 kbaud. ■ Protokol je definován uživatelským programem.
Použití USS protokolu		Příklad: Použití S7-200 s pohony SIMODRIVE MicroMaster. ■ STEP 7 – Micro/WIN s USS knihovnou. ■ CPU S7-200 funguje jako master, pohony jako slave.  <i>Příklad programu najdete v Típech a tricích na CD s dokumentací. Viz tip 28.</i> Tipy a triky
Vytvoření uživatelského programu, který emuluje zařízení typu slave v jiné síti		Příklad: Připojení CPU S7-200 k síti Modbus. ■ Uživatelský program v S7-200 emuluje zařízení Modbus slave. ■ STEP 7 – Micro/WIN s knihovnou Modbus.  <i>Příklad programu Modbus najdete v Típech a tricích na CD s dokumentací. Viz tip 41.</i> Tipy a triky

Použití kabelu PC/PPI a režimu Freeport s rozhraním RS-232

Kabel PC/PPI a komunikační funkce Freeport můžete použít pro připojení CPU S7-200 k zařízením, která splňují standard RS-232.

PC/PPI kabel je ve vysílacím režimu, pokud dochází k přenosu dat z portu RS-232 na port RS-485. Kabel je v přijímacím režimu, pokud je nečinný nebo pokud se přenáší data z portu RS-485 na port RS-232. Kabel okamžitě přejde z přijímacího do vysílacího režimu, když na lince RS-232 detekuje znaky.

PC/PPI kabel podporuje přenosové rychlosti od 1200 baud do 115,2 kbaud. Pro nastavení PC/PPI kabelu na správnou přenosovou rychlost použijte DIP přepínače na kabelu. Tabulka 7-9 ukazuje přenosové rychlosti a polohy přepínače.

Kabel se přepíná zpět do přijímacího režimu, je-li vysílací linka RS-232 v nečinném stavu po dobu definovanou jako doba přepnutí. Volba přenosové rychlosti kabelu určuje dobu přepnutí, jak je patrné z tabulky 7-9.

Pokud jste použili PC/PPI kabel v systému, kde se používá komunikace Freeport, musí program v S7-200 počítat s dobou přepnutí v následujících situacích:

Tabulka 7-9 Doba obrátky a nastavení

Přenosová rychlost	Doba přepnutí	Nastavení (1 = nahoru)
38400 až 115200	0,5 ms	000
19200	1,0 ms	001
9600	2,0 ms	010
4800	4,0 ms	011
2400	7,0 ms	100
1200	14,0 ms	101

- ☐ Automat S7-200 odpovídá na zprávy vysílané ze zařízení s rozhraním RS-232.

Po přijetí požadavku z RS-232 musí S7-200 pozastavit vyslání odpovědi po dobu delší nebo rovnou době přepnutí kabelu.

- ☐ Zařízení s rozhraním RS-232 odpovídá na zprávy vysílané z automatu S7-200.

Po přijetí odpovědi z RS-232 musí S7-200 pozastavit vyslání dalšího požadavku po dobu delší nebo rovnou době přepnutí kabelu.

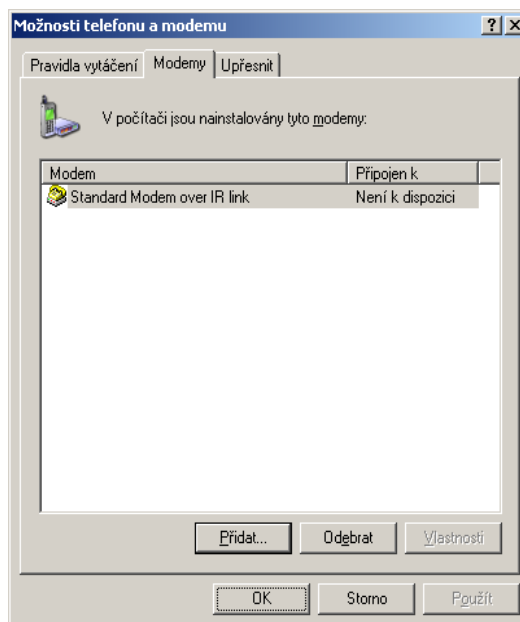
V obou situacích poskytne zpoždění PC/PPI kabelu dost času na přepnutí z vysílacího do přijímacího režimu, takže je možné přenášet data z portu RS-485 na port RS-232.

Použití modemu a STEP 7–Micro/WIN

STEP 7–Micro/WIN verze 3.2 používá pro konfiguraci telefonních modemů standardní telefonní připojení a modemy Windows. Konfigurace telefonního připojení a modemu se provádí pomocí ovládacího panelu Windows. Nastavení modemů pod Windows umožňuje:

- ☐ Použít většinu interních a externích modemů, podporovaných Windows.
- ☐ Použít standardní konfigurace pro většinu modemů, podporovaných Windows.
- ☐ Použít standardní pravidla vytáčení Windows pro výběr míst, čísla země a směrovacího čísla, pulzní nebo tónové volby a volací karty.
- ☐ Použít vyšší přenosové rychlosti při komunikaci s modemovým modulem EM 241.

Pro zobrazení dialogového okna "Možnosti telefonu a modemu" použijte ovládací panely Windows. Toto dialogové okno umožňuje konfigurovat modem; ten vyberete ze seznamu modemů podporovaných Windows. Pokud není váš typ uveden v seznamu modemů v dialogovém okně, vyberte typ, který vašemu nejbližší odpovídá, nebo kontaktujte výrobce modemu a získejte od něj ovladače modemu pro Windows.



Obr. 7-20 Konfigurace modemu

STEP 7–Micro/WIN umožňuje také použití rádiových a GSM modemů. Tyto typy modemů se neobjevují v dialogovém okně "Vlastnosti" ve Windows, ale jsou dostupné při konfiguraci připojení ve STEP 7–Micro/WIN.

Konfigurace připojení modemu

Připojení sdružuje identifikační jméno a fyzické vlastnosti připojení. U telefonního modemu k těmto vlastnostem patří typ modemu, výběr 10 nebo 11bitového protokolu a časové prodlevy. U celulárních modemů připojení umožňuje nastavení PIN a dalších parametrů. K vlastnostem rádiového modemu patří výběr přenosové rychlosti, parity, řízení toku a další parametry.

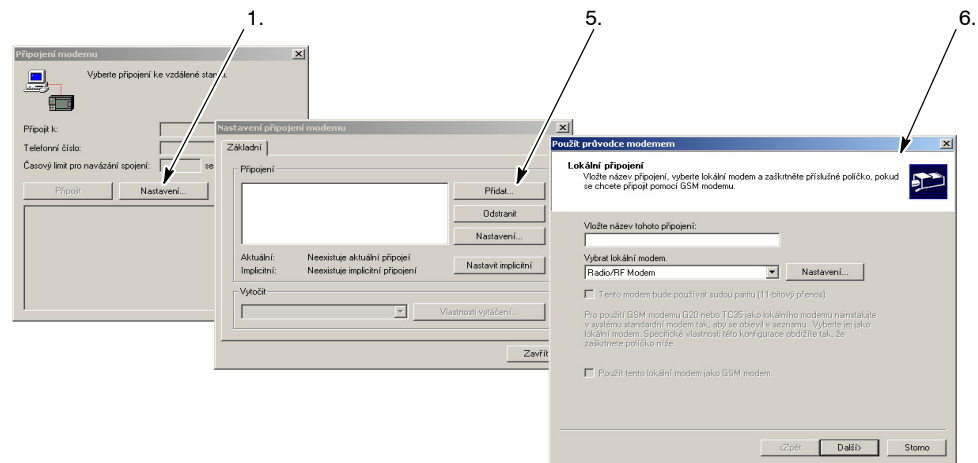


"Průvodce"

Vytvoření připojení

Pro vytvoření nového připojení, jeho odstranění nebo editaci použijte "Průvodce", jak ukazuje obrázek 7-21.

1. Dvakrát klikněte na ikonu v okně "Nastavení komunikace".
2. Abyste otevřeli nastavení rozhraní PG/PC, dvakrát klikněte na PC/PPI kabel. Vyberte kabel PPI a klikněte na tlačítko "Vlastnosti". V záložce "Místní připojení" zaškrtněte políčko "Připojení modemu".
3. Otevřete znovu dialogové okno "Komunikace" a dvakrát klikněte na ikonu "Připojení".
4. Abyste otevřeli dialogové okno "Nastavení připojení modemu", klikněte na tlačítko "Nastavení".
5. Pro spuštění "Použit průvodce modemem" klikněte na tlačítko "Přidat".
6. Nakonfigurujte připojení podle toho, jak průvodce napovídá.

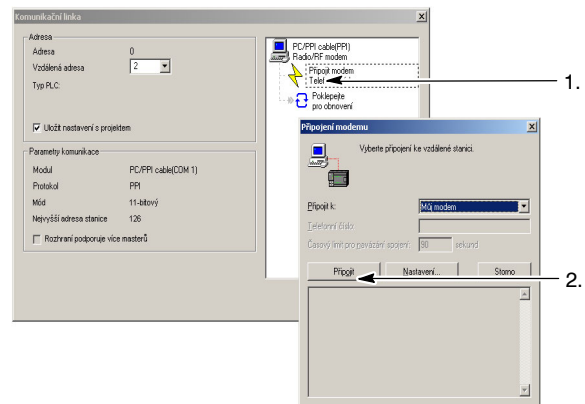


Obr. 7-21 Vytvoření připojení modemu

Připojení modemu k S7-200

Po vytvoření připojení se můžete připojit k CPU S7-200.

1. Abyste otevřeli dialogové okno "Komunikační linka", klikněte dvakrát na ikonu "Připojit".
2. V dialogovém okně "Připojení modemu" klikněte na "Připojit", čímž modem vytvoříte.



Obr. 7-22 Připojení k S7-200

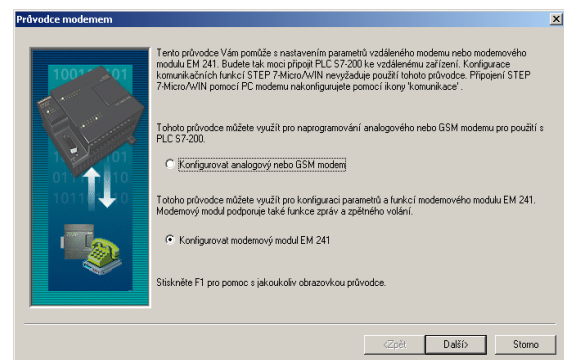
Konfigurace vzdáleného modemu

Vzdálený modem je takový modem, který je připojený k S7-200. Pokud je vzdáleným modemem modul EM 241, není nutná žádná konfigurace. Pokud připojujete externí modem nebo GSM modem, musíte připojení nakonfigurovat.



"Průvodce modemem"

"Průvodce modemem" konfiguruje vzdálený modem, který je připojený k CPU S7-200. Pro správnou komunikaci s polovičním duplexem RS-485 CPU S7-200 je potřeba speciální konfigurace modemu. Jednoduše vyberte typ modemu a vložte informace podle průvodce. Více informací najdete v online nápovědě.



Obr. 7-23 "Průvodce modemem"

Použití modemu s PC/PPI kabelem

Pro spojení komunikačního portu RS-232 modemu a CPU S7-200 je možné použít PC/PPI kabel. Viz obr. 7-24. Přepínače 1, 2 a 3 na PC/PPI kabelu nastavují přenosovou rychlost. Přepínač 4 volí buď 10bitový, nebo 11bitový PPI protokol. Přepínač 5 volí buď režim "Komunikační zařízení" (DCE), nebo "Koncové datové zařízení" (DTE). Přepínač 6 (pokud ho zařízení má) volí činnost signálu RTS na portu RS-232 PC/PPI kabelu.

SIEMENS		Napěťově oddělený PC/PPI kabel	
	Přepínač DIP č.	123	4
	115.2–38.4K	000	1 = 10 BIT
	19.2K	001	0 = 11 BIT
	9.6K	010	5
	2.4K	100	1 = DTE
	1.2K	101	0 = DCE
			6
			1 = RTS pro XMT
			0 = RTS

Obr. 7-24 Nastavení PC/PPI kabelu

Aby byl počítač schopen modem řídit, používají se řídicí signály RS-232 (jako například RTS, CTS a DTR). Jestliže modem použijete s PC/PPI kabelem, musíte modem nakonfigurovat tak, aby pracoval bez těchto signálů. Abyste zjistili příkazy potřebné pro jeho konfiguraci, nahlédněte do dokumentace modemu.

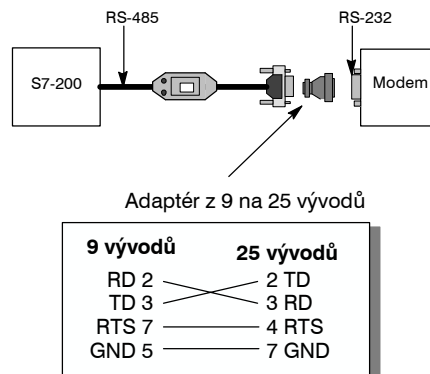
Pro protokol PPI volí přepínač 4 kabelu PC/PPI 10bitový, nebo 11bitový režim. Přepínač 4 přepněte pouze tehdy, je-li automat S7-200 připojen ke STEP 7–Micro/WIN s modemem. Jinak nastavte přepínač 4 na 11bitový režim, abyste zajistili správnou součinnost s jinými přístroji.

Přepínač 5 kabelu PC/PPI umožňuje nastavit port RS-232 kabelu na režim DCE nebo DTE. Pokud PC/PPI kabel používáte se STEP 7–Micro/WIN nebo pokud je PC/PPI kabel připojen k počítači, nastavte PC/PPI kabel na režim DCE. Pokud PC/PPI kabel používáte s modemem (který pracuje v režimu DCE), nastavte PC/PPI kabel na režim DTE. Díky tomu není nutné instalovat adaptér "nulový modem" mezi PC/PPI kabel a modem.

Podle typu konektoru modemu může ještě potřebovat adaptér z 9 na 25 vývodů.

Přepínač 6 PC/PPI kabelu volí činnost signálu RTS na konektoru RS-232. Nastavení "RTS for XMT" způsobí, že signál RTS bude aktivní, bude-li S7-200 vysílat z portu RS-485, a nečinný, bude-li S7-200 přijímat data. Nastavení "RTS Always" způsobí, že signál RTS bude vždy aktivní na portu RS-232 kabelu PC/PPI bez ohledu na to, zda automat S7-200 vysílá nebo přijímá data. Přepínač 6 ovlivňuje signál RTS pouze tehdy, je-li PC/PPI kabel nastaven do režimu DTE.

Obrázek 7-25 znázorňuje rozmístění vývodů běžného modemu.



Obr. 7-25 Rozmístění vývodů pro adaptér

Tabulka 7-10 ukazuje čísla a funkce vývodů portů RS-485 a RS-232 kabelu PC/PPI v režimu DTE. Tabulka 7-11 ukazuje čísla a funkce vývodů portů RS-485 a RS-232 kabelu PC/PPI v režimu DCE. PC/PPI kabel předává RTS pouze v režimu DTE.

Tabulka 7-10 Vývody rozhraní RS-485 a RS-232 v režimu DTE

Vývody rozhraní RS-485		Vývody rozhraní RS-232v režimu DTE ¹	
Vývod	Popis signálu	Vývod	Popis signálu
1	Uzemnění (logická zem RS-485)	1	<i>Nepoužito</i> : Detekce nosiče dat (DCD)
2	24V zpětný vodič (logická zem RS-485)	2	Přijímaná data (RD) (vstup PC/PPI kabelu)
3	Signál B (Rx/D/TxD+)	3	Vysílaná data (TD) (výstup PC/PPI kabelu)
4	RTS (úroveň TTL)	4	<i>Nepoužito</i> : Datový terminál připraven (DTR)
5	Uzemnění (logická zem RS-485)	5	Uzemnění (logická zem RS-232)
6	NC (Nepřipojeno)	6	<i>Nepoužito</i> : Pohotovost (DSR)
7	24V napájení	7	Požadavek na vysílání (RTS) (volitelné přepínačem)
8	Signál A (Rx/D/TxD-)	8	<i>Nepoužito</i> : Připraven k vysílání (CTS)
9	Volba protokolu	9	<i>Nepoužito</i> : Indikátor vyzvánění (RI)

¹ Pro modem můžete potřebovat adaptér ze zásuvky na vidlici a z 9 na 25 vývodů

Tabulka 7-11 Vývody rozhraní RS-485 a RS-232 v režimu DCE

Vývody rozhraní RS-485		Vývody rozhraní RS-232 v režimu DCE	
Vývod	Popis signálu	Vývod	Popis signálu
1	Uzemnění (logická zem RS-485)	1	<i>Nepoužito</i> : Detekce nosiče dat (DCD)
2	24V zpětný vodič (logická zem RS-485)	2	Přijímaná data (RD) (výstup PC/PPI kabelu)
3	Signál B (Rx/D/TxD+)	3	Vysílaná data (TD) (vstup PC/PPI kabelu)
4	RTS (úroveň TTL)	4	<i>Nepoužito</i> : Datový terminál připraven (DTR)
5	Uzemnění (logická zem RS-485)	5	Uzemnění (logická zem RS-232)
6	NC (Nepřipojeno)	6	<i>Nepoužito</i> : Pohotovost (DSR)
7	24V napájení	7	<i>Nepoužito</i> : Požadavek na vysílání (RTS)
8	Signál A (Rx/D/TxD-)	8	<i>Nepoužito</i> : Připraven k vysílání (CTS)
9	Volba protokolu	9	<i>Nepoužito</i> : Indikátor vyzvánění (RI)

Pro pokročilé

Optimalizace výkonu sítě

Výkon sítě ovlivňují následující faktory (největší vliv má přenosová rychlost a počet masterů):

- ☐ Přenosová rychlost: Provoz sítě při nejvyšší přenosové rychlosti podporované všemi částmi zařízení má na výkon sítě největší vliv.
- ☐ Počet masterů v síti: Minimalizace počtu masterů také zvyšuje výkon sítě. Každý master zvyšuje požadavky režie sítě; jestliže síť obsahuje méně masterů, snižuje to režii.
- ☐ Volba adres zařízení typu master a slave: adresy masterů by měly být nastaveny tak, aby následovaly za sebou bez mezer. Jakmile je mezi adresami mezera, kontroluje master nepřetržitě adresy v mezeře, zda se tam nenachází další master, žádající o připojení do sítě. Tato kontrola vyžaduje čas a zvyšuje režii sítě. Pokud mezi adresami masterů mezera není, kontrola se neprovádí, a režie je tak minimalizována. Adresy zařízení typu slave můžete nastavit libovolně; výkon sítě tím neovlivníte, pokud ovšem nejsou tato zařízení mezi zařízeními typu master. Slave umístěný mezi mastery zvyšuje režii sítě stejným způsobem jako mezery mezi adresami jednotlivých masterů.
- ☐ Faktor obnovení mezery (GUF): GUF se používá pouze tehdy, pracuje-li CPU S7-200 jako PPI master a říká S7-200, jak často má hledat jiné mastery v mezeře mezi adresami. Pro nastavení GUF se při konfiguraci CPU používá STEP 7–Micro/WIN. Automat S7-200 se tak konfiguruje na periodickou kontrolu mezery mezi adresami. Při GUF = 1 kontroluje S7-200 mezeru mezi adresami pokaždé, když dostane token; při GUF = 2 kontroluje S7-200 tuto mezeru jednou na každé dvě obdržení tokenu. Pokud jsou mezi adresami masterů mezery, snižuje vyšší GUF režii sítě. Pokud mezi nimi mezery nejsou, nemá GUF na výkon žádný vliv. Nastavení velkého GUF způsobuje velká zpoždění v připojování masterů do sítě, protože jsou adresy méně často kontrolovány. Implicitní nastavení GUF je 10.
- ☐ Adresa nejvyšší stanice (HSA): HSA se používá pouze tehdy, pracuje-li CPU S7-200 jako PPI master a definuje nejvyšší adresu, na které master hledá další mastery. Pro nastavení HSA se při konfiguraci portu CPU používá STEP 7–Micro/WIN. Nastavením HSA můžete eliminovat mezeru mezi adresami, kterou musí kontrolovat poslední master (nejvyšší adresa) v síti. Omezení velikosti mezery mezi adresami minimalizuje dobu potřebnou pro nalezení jiného mastera a jeho připojení do sítě. Adresa nejvyšší stanice nemá žádný vliv na adresy zařízení typu slave: master může komunikovat se zařízeními typu slave, která mají adresu vyšší než HSA. Obecně platí, že se má adresa nejvyšší stanice nastavit u všech zařízení typu master na stejnou hodnotu. Tato adresa má být vyšší nebo se rovnat nejvyšší adrese zařízení typu master. Implicitně nastavená hodnota HSA je 31.

Výpočet doby oběhu tokenu

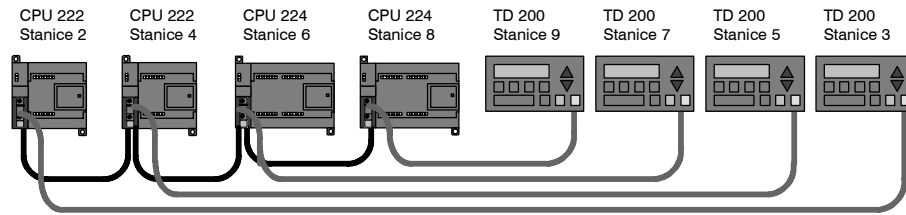
V síti, kde se předává token, může iniciovat komunikaci pouze ta stanice, která obdržela token. Doba oběhu tokenu (doba, za kterou token oběhne všechny mastery v logickém okruhu) je úměrná výkonu dané sítě.

Na obrázku 7-26 je vzorová síť jako příklad pro výpočet doby oběhu tokenu v síti s více mastery. V tomto příkladu komunikuje TD 200 (stanice 3) s CPU 222 (stanice 2), TD 200 (stanice 5) komunikuje s CPU 222 (stanice 4) a tak dále. Dva moduly CPU 224 používají instrukce "Čtení ze sítě" a "Zápis do sítě" pro shromažďování dat z ostatních automatů S7-200: CPU 224 (stanice 6) posílá zprávy stanicím 2, 4 a 8, CPU 224 (stanice 8) posílá zprávy stanicím 2, 4 a 6. V této síti je šest stanic master (čtyři displeje TD 200 a dva moduly CPU 224) a dvě stanice slave (dva moduly CPU 222).



Tipy a triky

Podrobnosti o oběhu tokenu najdete v *Tipech a tricích* na CD s dokumentací. Viz tip 42.



Obr. 7-26 Příklad sítě s oběhem tokenu

Aby mohl master vyslat zprávu, musí mít token. Když má například token stanice 3, generuje požadavek pro stanici 2 a pak předá token stanici 5. Stanice 5 pak generuje požadavek pro stanici 4 a následně předá token stanici 6. Stanice 6 pak generuje zprávu pro stanici 2, 4 nebo 8 a předá token stanici 7. Proces generování zpráv a předávání tokenu pokračuje kolem celého logického kruhu od stanice 3 do stanic 5, 6, 7, 8, 9 a nakonec zpátky do stanice 3. Token musí oběhnout celý logický kruh, aby mohl master vyslat žádost

o informace. Pro logický kruh šesti stanic, z nichž každá vysílá na jedno obdržení tokenu jeden požadavek na zápis nebo čtení hodnoty jednoho double word (čtyři byty dat), je doba oběhu tokenu přibližně 900 ms při 9600 baud. Zvýšení počtu bytů dat na požadavek nebo zvýšení počtu stanic prodlužuje dobu oběhu tokenu.

Doba oběhu tokenu je určena tím, jak dlouho ho každá stanice drží. Doba oběhu tokenu v síti s více mastery můžete stanovit sečtením časů, po které každý master token drží. Pokud byl aktivován režim PPI master (pod protokolem PPI), můžete posílat zprávy ostatním automatům pomocí instrukcí "Čtení ze sítě" a "Zápis do sítě". Pokud posíláte zprávy pomocí těchto instrukcí, můžete pro výpočet přibližné doby oběhu tokenu použít následující vzorec při dodržení těchto předpokladů: každá stanice vysílá jeden požadavek na jedno držení tokenu, požadavek představuje čtení nebo zápis za sebou jdoucích oblastí dat, použití jednoho komunikačního zásobníku v S7-200 nezpůsobuje konflikty a žádný automat S7-200 nemá delší programový cyklus než cca 10 ms.

Doba držení tokenu (T_{hold}) = (128 režie + n datových znaků) x 11 bitů/znak x 1/přenosová rychlost v baudech

Doba oběhu tokenu (T_{rot}) = $T_{\text{hold}} \text{ master 1} + T_{\text{hold}} \text{ master 2} + \dots + T_{\text{hold}} \text{ master } m$

kde n je počet datových znaků (bytů)
 m je počet zařízení typu master

Následující rovnice počítají doby oběhu (jeden "bit time" se rovná trvání jedné signální periody) pro příklad ukázaný na obrázku 7-26:

$$\begin{aligned} T (\text{doba držení tokenu}) &= (128 + 4 \text{ znaky}) \times 11 \text{ bitů/znak} \times 1/9600 \text{ bit time/s} \\ &= 151,25 \text{ ms/master} \\ T (\text{doba oběhu tokenu}) &= 151,25 \text{ ms/master} \leq 6 \text{ masterů} \\ &= 907,5 \text{ ms} \end{aligned}$$

**Tip**

Software SIMATIC NET COM PROFIBUS zahrnuje analyzátor pro vyhodnocení výkonu sítě.

Porovnání doby oběhu tokenu

Tabulka 7-12 přináší srovnání doby oběhu tokenu při různých počtech stanic, velikosti přenášených dat a přenosových rychlostech. Časy jsou vypočítané pro případ použití instrukcí "Čtení ze sítě" a "Zápis do sítě" CPU S7-200 nebo jiným masterem.

Tabulka 7-12 Doba oběhu tokenu (v sekundách)

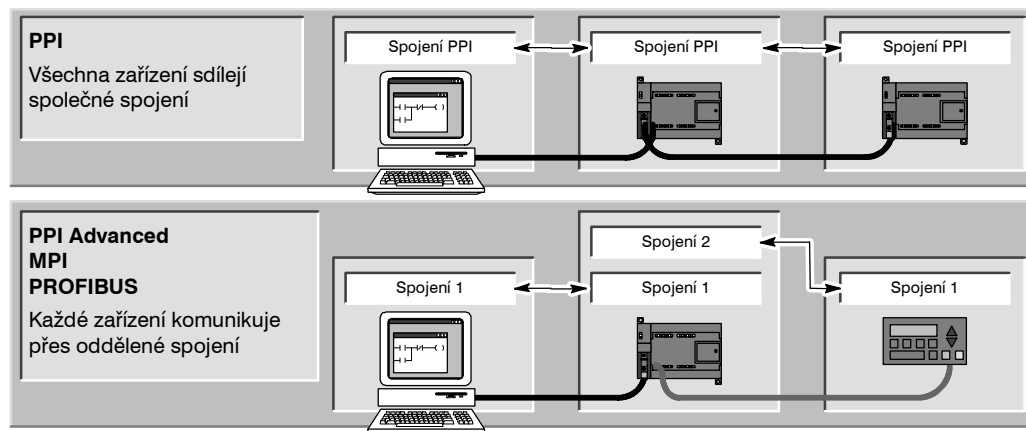
Přenosová rychlost	Počet přenášených bytů	Počet masterů								
		2	3	4	5	6	7	8	9	10
9.6 kbaud	1	0.30	0.44	0.59	0.74	0.89	1.03	1.18	1.33	1.48
	16	0.33	0.50	0.66	0.83	0.99	1.16	1.32	1.49	1.65
19.2 kbaud	1	0.15	0.22	0.30	0.37	0.44	0.52	0.59	0.67	0.74
	16	0.17	0.25	0.33	0.41	0.50	0.58	0.66	0.74	0.83
187.5 kbaud	1	0.009	0.013	0.017	0.022	0.026	0.030	0.035	0.039	0.043
	16	0.011	0.016	0.021	0.026	0.031	0.037	0.042	0.047	0.052

Vysvětlení datových spojení mezi zařízeními v síti

Zařízení v síti komunikují přes jednotlivá spojení, která tvoří spoje mezi zařízeními typu master a slave. Jak je vidět na obrázku 7-27, komunikační protokoly se liší v tom, jak se spojeními zacházejí:

- ☐ Protokol PPI využívá jedno spojení, sdílené všemi zařízeními v síti.
- ☐ Protokoly PPI Advanced, MPI a PROFIBUS využívají oddělená spojení mezi libovolnými dvěma zařízeními, která spolu komunikují.

Při použití protokolu PPI Advanced, MPI nebo PROFIBUS nemůže druhý master zasahovat do spojení, které bylo navázáno mezi zařízeními typu master a slave. CPU S7-200 a EM 277 mají vždy vyhrazeno jedno spojení pro STEP 7 – Micro/WIN a jedno spojení pro zařízení HMI. Jiná zařízení typu master tato vyhrazená spojení nemohou používat. Tím je zajištěno, že je vždy možné připojit nejméně jednu programovací stanici a nejméně jedno zařízení HMI s CPU S7-200 nebo EM 277, jestliže master používá protokol, který spojení podporuje (jako je například PPI Advanced).



Obr. 7-27 Řízení komunikačních spojení

Jak je vidět z tabulky 7-13, CPU S7-200 a EM 277 mají určitý počet spojení. Každý port (Port 0 a Port 1) CPU S7-200 podporuje až čtyři oddělená spojení. (To umožňuje maximálně osm spojení na CPU S7-200.) Tato spojení jsou navíc ke sdílenému spojení PPI. EM 277 podporuje šest spojení.

Tabulka 7-13 Možnosti CPU S7-200 a modulů EM 277

Modul		Přenosová rychlost	Počet	Podporované protokoly
CPU S7-200,	Port 0	9.6 kbaud, 19,2 kbaud nebo 187,5 kbaud	4	PPI, PPI Advanced, MPI a PROFIBUS ¹
	Port 1	9.6 kbaud, 19,2 kbaud nebo 187,5 kbaud	4	PPI, PPI Advanced, MPI a PROFIBUS ¹
Modul EM 277		9.6 kbaud až 12 Mbaud	6 na modul	PPI Advanced, MPI a PROFIBUS

¹ V případě Portu 0 a Portu 1 CPU S7-200 můžete MPI a PROFIBUS použít pouze při komunikaci s automatem ve funkci slave.

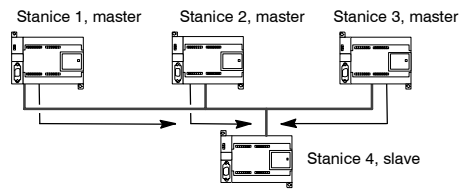
Práce se složitými sítěmi

Pokud jde o S7-200, mají složité sítě více masterů S7-200, které pro komunikaci s jinými zařízeními v síti PPI užívají instrukce "Čtení ze sítě" (NETR) a "Zápis do sítě" (NETW). Tyto sítě mívají speciální problémy, které mohou zablokovat komunikaci mezi zařízeními typu master a slave.

Pokud síť pracuje s nižší přenosovou rychlostí (jako například 9,6 kbaud nebo 19,2 kbaud), dokončí každý master transakci (čtení nebo zápis) dříve, než předá token. Při rychlosti 187,5 kbaud však vyše master požadavek na zařízení typu slave a pak předá token, čímž na slave ponechává nevyřízený požadavek.

Obrázek 7-28 znázorňuje síť s potenciálními komunikačními konflikty. V této síti jsou stanice 1, 2 a 3 zařízeními typu master, která používají instrukce "Čtení ze sítě" nebo "Zápis do sítě" pro komunikaci se stanicí 4. Instrukce "Čtení ze sítě" a "Zápis do sítě" používají protokol PPI, takže všechny automaty S7-200 sdílejí jediné spojení se stanicí 4.

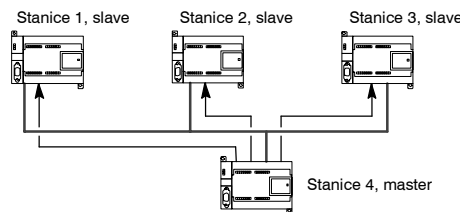
V tomto příkladu stanice 1 generuje požadavek na stanici 4. Při přenosových rychlostech vyšších než 19,2 kbaud pak stanice 1 předá token stanici 2. Pokud se stanice 2 pokusí vyslat požadavek na stanici 4, je požadavek stanice 2 zamítnut, protože je stále přítomen požadavek stanice 1. Všechny požadavky na stanici 4 budou zamítány, dokud stanice 4 nedokončí odpověď pro stanici 1. Až po dokončení této odpovědi může jiná master stanice vyslat požadavek na stanici 4.



Obr. 7-28 Komunikační konflikt

Abyste zabránili konfliktu na komunikačním portu stanice 4, zvažte možnost konfigurace stanice 4 jako jediného masteru v síti, jak ukazuje obrázek 7-29. Potom stanice 4 generuje požadavky na čtení nebo zápis ostatním automatům S7-200.

Nejen že tato konfigurace zabraňuje komunikačnímu konfliktu, ale také snižuje režii, způsobenou více masterly a umožňuje efektivnější práci sítě.



Obr. 7-29 Zabránění konfliktu

U některých aplikací však není možné snížit počet masterů v síti. Jestliže je v síti několik masterů, musíte řídit dobu oběhu tokenu a zajistit, aby nepřekročila celkovou dobu oběhu tokenu. (Doba oběhu tokenu je časový interval, který uplyne od doby, kdy master předal token, do doby, kdy ho opět obdrží.)

Tabulka 7-14 HSA a celková doba oběhu tokenu

HSA	9.6 kbaud	19.2 kbaud	187.5 kbaud
HSA = 15	0,613 s	0,307 s	31 ms
HSA = 31	1,040 s	0,520 s	53 ms
HSA = 63	1,890 s	0,950 s	97 ms
HSA = 126	3,570 s	1,790 s	183 ms

Pokud je doba potřebná pro návrat tokenu na master větší než celková doba oběhu tokenu, nemůže master generovat požadavek. Master může generovat požadavek pouze tehdy, je-li skutečná doba oběhu tokenu kratší než celková doba oběhu tokenu.

Adresa nejvyšší stanice (HSA) a nastavená přenosová rychlost S7-200 určují celkovou dobu oběhu tokenu. Tabulka 7-14 uvádí cílové doby oběhu.

U nižších přenosových rychlostí, jako například 9,6 kbaud a 19,2 kbaud, vyčká master na odpověď na svůj požadavek, než předá token. Protože zpracování požadavku/odpovědi může v porovnání s programovým cyklem trvat relativně dlouho, je vysoce pravděpodobné, že každý master bude v síti připraven vyslat požadavek pokaždé, když obdrží token. Pak poroste skutečná doba oběhu tokenu a některé mastery by nemusely být schopny zpracovat požadavky. V některých situacích by master mohl zpracovávat požadavky opravdu jen zřídka.

Například: Představte si síť 10 masterů, které vysílají 1 byte rychlostí 9,6 kbaud, přičemž je tato síť konfigurována s HSA 15. V tomto příkladu má každý master vždy připravenou zprávu k vyslání. Jak je patrné z tabulky 7-14, celková doba oběhu tokenu je pro tuto síť 0,613 s. Ale na základě údajů o výkonu, uvedených v tabulce 7-12, je skutečná doba oběhu tokenu potřebná pro danou síť 1,48 s. Protože je skutečná doba oběhu tokenu delší než cílová doba oběhu tokenu, budou některé mastery moci vyslat zprávu teprve při nějakém pozdějším oběhu tokenu.

Pro zlepšení situace, kdy je skutečná doba oběhu tokenu delší než cílová doba oběhu tokenu, máte dvě základní volby:

- ☐ Můžete zkrátit skutečnou dobu oběhu tokenu snížením počtu masterů v síti; záleží na vaší aplikaci. Toto řešení ale nemusí být proveditelné.
- ☐ Můžete prodloužit celkovou dobu oběhu tokenu zvýšením HSA pro všechny mastery v síti.

Zvýšení HSA může v síti způsobit jiný problém tím, že ovlivní délku času, který automat S7-200 potřebuje, aby se přepnul do režimu master a vstoupil do sítě. Pokud používáte časovač, abyste zajistili, že bude instrukce "Čtení ze sítě" nebo "Zápis do sítě" dokončena v požadované době, může zpoždění aktivace režimu master a přidání automatu do sítě jako master způsobit, že čas pro instrukci vyprší. Zpoždění můžete při připojování masteru minimalizovat snížením faktoru obnovení mezery (GUF) pro všechny mastery v síti.

Vzhledem ke způsobu, jakým jsou požadavky do slave zapisovány a jakým jsou v něm při rychlosti 187,5 kbaud uchovávány, měli byste při volbě celkové doby oběhu tokenu počítat s nějakým časem navíc. Při rychlosti 187,5 kbaud by se skutečná doba oběhu tokenu měla rovnat přibližně polovině celkové doby oběhu tokenu.

Pro zjištění doby oběhu tokenu použijte údaje o výkonu z tabulky 7-12. Tím určíte dobu potřebnou k dokončení operací "Čtení ze sítě" a "Zápis do sítě". Pro výpočet doby potřebné pro zařízení HMI (jako například TD 200), použijte údaje o výkonu pro přenos 16 bytů. Doba oběhu tokenu vypočítejte součtem časů pro každé zařízení v síti. Nejhorší případ nastane v případě součtu časů tehdy, kdy chtějí všechna zařízení zpracovat požadavek během jednoho oběhu tokenu. Tím je definována maximální doba oběhu tokenu potřebná pro danou síť.

Příklad: Představte si síť s přenosovou rychlostí 9,6 kbaud a se čtyřmi displeji TD 200 a čtyřmi automaty S7-200, kde každý S7-200 zapisuje každou sekundu 10 bytů dat do jiného S7-200. Použijte tabulku 7-12 a vypočítejte příslušné doby přenosu pro síť:

4 displeje TD 200 přenáší 16 bytů dat =	0,66 s
4 automaty S7-200 přenáší 10 bytů dat =	<u>0,63 s</u>
Celková doba oběhu tokenu =	1,29 s

Abyste síti poskytli dostatek času na zpracování všech požadavků během jednoho oběhu tokenu, nastavte HSA na 63. (Viz tabulku 7-14.) Volba cílové doby oběhu tokenu (1,89 s), která je delší než maximální doba oběhu tokenu (1,29 s), zajišťuje, že každé zařízení může přenášet data při každém oběhu tokenu.

Pro zlepšení spolehlivosti sítě s více mastery byste měli zvážit také následující vylepšení:

- ☐ Změna rychlosti aktualizace zařízení HMI, která ponechá delší čas mezi jednotlivými aktualizacemi. Změňte například rychlost aktualizace displeje TD 200 z "Co možná nejrychleji" na "Jedenkrát za sekundu".
- ☐ Snížení počtu požadavků (a režie sítě na zpracování požadavků) zkombinováním operací "Čtení ze sítě" a "Zápis do sítě". Například místo dvou operací použijte "Čtení ze sítě", z nichž každá čte 4 byty, jednu operaci "Čtení ze sítě", která čte 8 bytů. Doba zpracování dvou požadavků po 4 bytech je daleko delší, než doba zpracování jednoho požadavku o 8 bytech.
- ☐ Změňte rychlost aktualizace automatů S7-200 master, aby se nepokoušely o aktualizaci rychleji, než je doba oběhu tokenu.

Návod pro odstraňování poruch hardwaru a nástroje pro ladění softwaru



STEP 7 – Micro/WIN má softwarové nástroje, které pomohou při ladění a testování programu. Tyto prvky zahrnují prohlížení stavu programu při jeho vykonávání, zadání běhu S7-200 po určený počet programových cyklů a vnucování hodnot.

Jako návod pro stanovení příčiny a možného řešení při odstraňování hardwarových problémů S7-200 použijte tabulku 8-1.

Přehled kapitoly

Funkce pro ladění uživatelského programu	236
Zobrazení stavu programu	238
Použití stavové tabulky pro monitorování a modifikaci dat	239
Force (vnucení) specifických hodnot	240
Zadání běhu programu po určený počet programových cyklů	240
Návod pro odstraňování poruch hardwaru	241

Funkce pro ladění uživatelského programu

STEP 7 – Micro/WIN má několik funkcí, které pomáhají při ladění programu: záložky, tabulku křížových odkazů a editaci v době běhu programu.

Použití záložek pro snadný přístup k programu

V programu je možné nastavit záložky, které usnadňují přesuny dopředu a zpátky mezi označenými (záložkou opatřenými) řádky dlouhého programu. Je možné přejít na následující nebo předcházející řádek programu opatřený záložkou.

Použití tabulky křížových odkazů pro kontrolu odkazů v programu



Křížové odkazy

Tabulka křížových odkazů zobrazuje informace o křížových odkazech a použití prvků v programu.

Tabulka křížových odkazů uvádí všechny operandy použité v programu a identifikuje umístění v programovém bloku, networku nebo řádku a kontext instrukce pro operand při každém jeho použití.

Chcete-li změnit zobrazení všech operandů, můžete přepínat mezi symbolickým a absolutním náhledem.

Křížové odkazy				
	3	4	5	6
	7	8	9	10
	11	12	13	14
	15	16	17	18
	Element	Blok	Umístění	Kontext
1	Disp_Current	MAIN (OB1)	Network 3, Řádek 2	MOVW
2	Disp_Preset	MAIN (OB1)	Network 2, Řádek 2	MOVW
3	Disp_Preset	MAIN (OB1)	Network 3, Řádek 3	MOVW
4	Saved_Preset	MAIN (OB1)	Network 2, Řádek 2	MOVW
5	Saved_Preset	MAIN (OB1)	Network 3, Řádek 3	MOVW
6	Saved_Preset	MAIN (OB1)	Network 6, Řádek 2	TON

Obr. 8-1 Tabulka křížových odkazů

8



Tip

Pokud kliknete dvakrát na položku v tabulce křížových odkazů, přeskočíte do dané části programu nebo bloku.

Editace uživatelského programu v režimu RUN

Model CPU 224 verze 1.10 (a vyšší) a model CPU 226 verze 1.00 (a vyšší) podporují editaci v režimu RUN. Schopnost editování v režimu RUN umožňuje provádění malých změn programu s minimálním narušením řízeného procesu. Implementace této možnosti ale také umožňuje zásadní změny programu, které mohou program narušit, nebo být dokonce nebezpečné.



Varování

Jestliže nahrajete změny do S7-200 v režimu RUN, ovlivní tyto změny okamžitě průběh procesu. Změna programu v režimu RUN může mít za následek neočekávané chování systému, které může způsobit smrt nebo vážné poranění obsluhy nebo škodu na zařízení.

Editaci v režimu RUN by měli provádět pouze oprávnění pracovníci, kteří znají účinek úprav v režimu RUN na provoz systému.

Chcete-li provést editaci programu v režimu RUN, musí připojená CPU S7-200 podporovat editaci v režimu RUN a musí být v režimu RUN.

1. Vyberte příkaz menu **Odladit > Editace programu v RUN**.
2. Pokud se projekt liší od programu v automatu S7-200, jste vyzváni k jeho uložení. Editaci v režimu RUN je možné provádět pouze v programu uloženém v S7-200.
3. STEP 7 – Micro/WIN vás upozorní, editujete-li program v režimu RUN, a vyzve k pokračování nebo ke zrušení operace. Pokud kliknete na "Pokračovat", STEP 7 – Micro/WIN provede upload programu z S7-200. Nyní můžete program editovat v režimu RUN. Pro editaci neplatí žádná omezení.



Tip

Instrukce nástupné (EU) a sestupné hrany (ED) jsou zobrazeny s operandem. Chcete-li si prohlédnout informace o instrukcích na vyhodnocení hran, vyberte v "Zobrazit" ikonu "Křížové odkazy". Pod záložkou "Využití hran" je seznam čísel pro instrukce na vyhodnocení hran v uživatelském programu. Při editaci programu je třeba dávat pozor, abyste nepřiradili stejné číslo dvěma hranám.

Download programu v režimu RUN

Editace v režimu RUN umožňuje provést download programového bloku, pouze pokud je S7-200 v režimu RUN. Dříve než provedete download programového bloku v režimu RUN, zvažte účinek úpravy v režimu RUN na práci S7-200 v následujících situacích:

- ☐ Pokud jste zrušili řídicí logiku nějakého výstupu, S7-200 uchová poslední stav toho výstupu do příštího vypnutí a obnovení napájení nebo přechodu do režimu STOP.
- ☐ Pokud jste zrušili funkce vysokorychlostního čítače nebo pulzního výstupu, které byly v chodu, pokračují tyto funkce v činnosti až do příštího vypnutí a obnovení napájení nebo přechodu do režimu STOP.
- ☐ Pokud jste zrušili instrukci "Připojit přerušení", ale nezrušili jste podprogram přerušení, bude S7-200 nadále pokračovat v provádění podprogramu přerušení až do příštího vypnutí a obnovení napájení nebo přechodu do režimu STOP. Obdobně, pokud jste zrušili instrukci "Odpojit přerušení", nejsou přerušení zastavena až do příštího vypnutí a obnovení napájení nebo přechodu do režimu STOP.
- ☐ Pokud jste přidali instrukci "Připojit přerušení", která je podmíněna bitem prvního programového cyklu, není tato událost aktivována až do příštího vypnutí a obnovení napájení nebo přechodu režimu STOP do režimu RUN.
- ☐ Pokud jste zrušili instrukci "Povolit přerušení", budou přerušení pokračovat ve své činnosti až do příštího vypnutí a obnovení napájení nebo přechodu režimu RUN do režimu STOP.
- ☐ Pokud jste modifikovali adresu příjmové schránky v tabulce a příjmová schránka je aktivní v době, kdy S7-200 přepne ze starého na modifikovaný program, pokračuje S7-200 v zapisování přijímaných dat na starou adresu v tabulce. Instrukce "Čtení ze sítě" a "Zápis do sítě" fungují stejným způsobem.
- ☐ Jakákoliv logika, která je podmíněna stavem bitu prvního programového cyklu, nebude provedena až do příštího vypnutí a obnovení napájení nebo přechodu z režimu STOP do režimu RUN. Bit prvního programového cyklu je nastaven pouze přechodem do režimu RUN a není ovlivněn editací v režimu RUN.



Tip

Abyste mohli provést download programu v režimu RUN, musí S7-200 podporovat editaci v režimu RUN, musí být program přeložen bez chyb a nesmí docházet k chybám v komunikaci mezi STEP 7–Micro/WIN a S7-200.

Je možné nahrát pouze programový blok.

Chcete-li provést download programu v režimu RUN, klikněte na tlačítko Download nebo vyberte příkaz menu **Soubor > Download**. Pokud se program úspěšně přeloží, nahraje STEP 7–Micro/WIN programový blok do S7-200.

Ukončení editace v režimu RUN

Chcete-li ukončit editaci v režimu RUN, vyberte příkaz menu **Odladit > Editace programu v RUN** a zrušte zaškrtnutí. Pokud jste provedli změny, které nebyly uloženy, STEP 7–Micro/WIN zobrazí výzvu, abyste pokračovali v editaci, nebo provedli download změn a ukončili editaci v režimu RUN, anebo ji ukončili bez provedení downloadu.

Zobrazení stavu programu

STEP 7–Micro/WIN umožňuje monitorovat stav uživatelského programu během jeho provádění. Pokud sledujete stav programu, zobrazuje programový editor stav hodnot operandů instrukcí.

Chcete-li stav zobrazit, klikněte na tlačítko "Stav programu" nebo vyberte příkaz menu **Odladit > Stav programu**.

Zobrazení stavu programu v LAD a FBD

STEP 7–Micro/WIN nabízí pro zobrazení stavu programů LAD a FBD dvě možnosti:

- ☐ Stav na konci programového cyklu: STEP 7–Micro/WIN získává hodnoty pro zobrazení stavu během více programových cyklů; pak aktualizuje zobrazení na obrazovce. Zobrazení stavu neodráží aktuální stav každého prvku v době provádění. Stav na konci programového cyklu nezahrnuje stav paměti L nebo akumulátorů.

Při vyhodnocování stavu na konci programového cyklu jsou stavové hodnoty aktualizovány ve všech pracovních módech CPU.

- ☐ Stav při běhu programu: STEP 7–Micro/WIN zobrazuje hodnoty networků tak, jak jsou jednotlivé prvky prováděny S7-200. Pro zobrazení stavu při běhu programu vyberte příkaz menu **Odladit > Při běhu programu**.

Stavové hodnoty pro stav provádění jsou aktualizovány pouze tehdy, je-li CPU v režimu RUN.

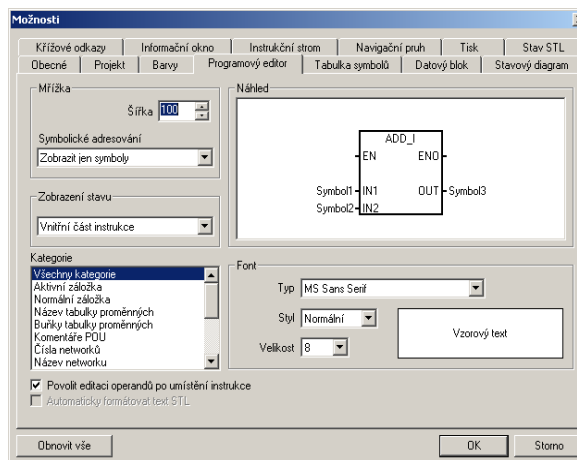
Tip

STEP 7–Micro/WIN obsahuje jednoduchý způsob, jak změnit stav proměnné. Vyberte proměnnou a klikněte pravým tlačítkem myši, aby se zobrazilo menu voleb.

Nastavení způsobu zobrazení stavu v programech LAD a FBD

STEP 7–Micro/WIN má různé možnosti, jak zobrazit stav programu.

Jestliže chcete změnit možnosti zobrazení pro stavovou obrazovku, vyberte příkaz menu **Nástroje > Možnosti** a pak záložku "Programový editor", jak ukazuje obrázek 8-2.



Obr. 8-2 Možnosti zobrazení stavu

Zobrazení stavu programu v STL

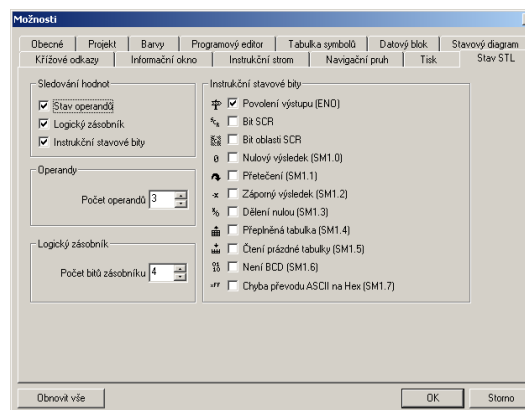
Stav provádění programu STL můžete sledovat při postupném provádění jedné instrukce za druhou. V programu STL zobrazuje STEP 7–Micro/WIN ukazuje stav instrukcí, které jsou zobrazeny na obrazovce.

STEP 7–Micro/WIN shromažďuje stavové informace z automatu S7-200, počínaje prvním příkazem STL nahoře v okně editoru. Při přetáčení okna editoru směrem dolů jsou z automatu S7-200 vyčteny nové informace.

STEP 7–Micro/WIN trvale aktualizuje hodnoty na obrazovce. Chcete-li pozastavit aktualizace obrazovky, vyberte tlačítko "Spuštěná pauza". Současná data zůstanou na obrazovce, dokud nezrušíte volbu tlačítka "Spuštěná pauza".

Konfigurace zobrazovaných parametrů programu STL

STEP 7–Micro/WIN umožňuje zobrazení stavu mnoha parametrů instrukcí STL. Vyberte příkaz menu **Nástroje > Možnosti** a pak záložku "Stav STL". Viz obr. 8-3.



Obr. 8-3 Možnosti zobrazení stavu STL

Použití stavového diagramu pro monitorování a modifikaci dat

Stavový diagram umožňuje čtení, zápis, vncení (force) a monitorování proměnných v době, kdy S7-200 provádí uživatelský program. Chcete-li vytvořit stavovou tabulku, vyberte příkaz menu **Zobrazit > Komponenty > Stavový diagram**. Obrázek 8-4 ukazuje příklad stavového diagramu.

Můžete vytvořit několik stavových diagramů.

STEP 7–Micro/WIN obsahuje v nástrojové liště ikony pro manipulace se stavovým diagramem: "Setřídít vzestupně", "Setřídít sestupně", "Jednorázové čtení", "Zápis všeho", "Force", "Zrušení force", "Zrušení všech force" a "Čtení všech force".

Chcete-li vybrat formát buňky, vyberte ji a klikněte na pravé tlačítko myši, aby se zobrazilo kontextové menu.

Stavový diagram																	
	Adresa	Formát	Aktuální hodnota	Nová hodnota													
1	Always_True	Bit															
2	Timer	Bez znaménka															
3	Disp_Msg_Two	Bit															
4	Disp_Msg_One	Bit															
5	Msg_Ack	Bit															
6	Edit_Bit	Bit															
7	Disp_Current	Se znaménkem															
8	Disp_Preset	Bez znaménka															
9	Saved_Preset	ASCII															
10		Se znaménkem															
11	TD200_F_Keys	Hexadecimálně															
12		Se znaménkem															
13	Add_TD200_Param	Bez znaménka															
14	TD200_Param_Table	Bez znaménka															
15	Lang_and_Update_Time	Bez znaménka															

Obr. 8-4 Stavový diagram

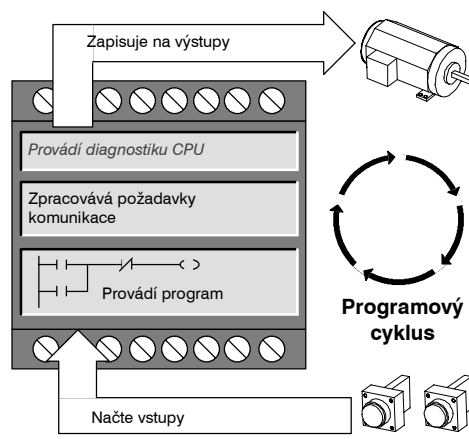
Force (vnucení) specifických hodnot

Programovatelný automat S7-200 umožňuje vnutit hodnoty všech, nebo jen některých vstupů a výstupů (bity I a Q). Dále můžete vnutit hodnoty až do 16 paměťových míst (V nebo M) nebo hodnoty analogových vstupů a výstupů (AI nebo AQ). Pamětím V nebo M můžete vnutit hodnoty typu byte, word nebo double word. Analogové hodnoty mohou mít vnucenu pouze hodnotu typu word na sudých bytech, jako například AIW6 nebo AQW14. Všechny vnucené hodnoty jsou uloženy v permanentní paměti EEPROM automatu S7-200.

Protože mohou být vnucená data během programového cyklu změněna (programem, cyklem aktualizace vstupů a výstupů, nebo komunikací), aplikuje S7-200 opakovaně vnucené hodnoty v různých dobách programového cyklu.

- ☐ **Čtení vstupů:** S7-200 aplikuje vnucené hodnoty na vstupy při jejich načítání.
- ☐ **Provádění řídicí logiky programu:** Automat S7-200 aplikuje vnucené hodnoty vždy, když přistupujete ke vstupům a výstupům. 16 vnucených hodnot je aplikováno po provedení programu.
- ☐ **Zpracování požadavků komunikace:** Automat S7-200 aplikuje vnucené hodnoty na všechny požadavky čtení a zápisu komunikace.
- ☐ **Zápis na výstupy:** S7-200 aplikuje vnucené hodnoty na výstupy při jejich zápisu.

Pro vnucení hodnot můžete použít stavový diagram. Chcete-li vnutit novou hodnotu, vložte ji do sloupce "Nová hodnota" ve stavové tabulce a klikněte na nástrojové liště na tlačítko "Force". Chcete-li vnutit existující hodnotu, vyberte ji ve sloupci "Současná hodnota" a klikněte na tlačítko "Force".



Obr. 8-5 Programový cyklus automatu S7-200



Tip

Funkce "Force" vyřadí instrukce "Okamžité čtení" nebo "Okamžitý zápis". Funkce "Force" také vyřadí tabulku výstupů, která byla nakonfigurována pro přechod do režimu STOP. Pokud přejde automat S7-200 do režimu STOP, odráží výstup vnucenou hodnotu, nikoliv hodnotu, která byla konfigurována v tabulce výstupů.

Zadání běhu programu na zadaný počet programových cyklů

Jako pomoc při ladění programu umožňuje STEP 7—Micro/WIN spustit program na zadaný počet programových cyklů.

Můžete nechat provést pouze první programový cyklus. To umožní monitorovat data v automatu S7-200 po prvním programovém cyklu. Pro spuštění prvního programového cyklu vyberte příkaz menu **Odladit > První krok**.

Můžete také zadat, aby S7-200 prováděl program po několika programových cyklech (od 1 programového cyklu do 65.535 cyklů). To umožňuje sledovat program při změně proměnných. Chcete-li specifikovat počet programových cyklů, které mají být provedeny, vyberte příkaz menu **Odladit > Několik kroků**.

Návod pro odstraňování poruch hardwaru

Tabulka 8-1 Návod pro odstraňování poruch hardwaru automatu S7-200

Příznak	Možné příčiny	Možné řešení
Výstupy přestanou pracovat	- Elektrický impuls od zařízení poškodil výstup - Chyba uživatelského programu - Uvolněná nebo nesprávná kabeláž - Přetížení - Výstup je vnucený (force)	- Při připojení indukční zátěž (jako například motoru nebo relé) použijte vhodný obvod pro potlačení elektrických impulsů. Viz kapitolu 3. - Opravte uživatelský program - Zkontrolujte a opravte kabeláž - Zkontrolujte zátěž podle jmenovitých hodnot výstupů - Zkontrolujte zda není aktivní vnucení hodnot vstupů a výstupů
Rozsvítí se signálka SF (System Fault – systémová chyba) na S7-200	Následující seznam popisuje nejběžnější chybové kódy a příčiny: - Chyba uživatelského programu 0003 Chyba obvodu sledování funkce (watchdog) 0011 Nepřímé adresování 0012 Neplatná hodnota s plovoucí řádovou čárkou 0014 Chyba rozsahu - Elektrické rušení (0001 až 0009) - Poškozený modul (0001 až 0010)	Přečtěte si číslo kritické chyby; Informace o typu chyby najdete v příloze C: - Pokud jde o chybu programování, zkontrolujte použití instrukcí FOR, NEXT, JMP, LBL a "Porovnání". - Pokud jde o elektrické rušení: - Nahlédněte do návodu pro kabeláž v kapitole 3. Je velmi důležité, aby byl řídicí panel dobře uzemněn a aby nebyly vodiče vysokého napětí vedeny paralelně s vodiči nízkého napětí. - Připojte svorku M napájení snímačů 24 V DC na zem.
Nesvítí žádná LED dioda	- Přepálená pojistka - Obrácené napájecí kabely 24 V - Nesprávné napětí	Připojte k systému analyzátor vedení a zkontrolujte velikost a trvání přepětových špiček. Na základě naměřených hodnot přidejte do systému příslušný typ přepětové ochrany. Informace o instalaci kabeláže najdete v návodu v kapitole 3.
Přerušovaný provoz ve spojení s vysokoenergetickým zařízením	- Nesprávné uzemnění - Vedení kabeláže uvnitř rozváděče - Příliš krátký čas vstupních filtrů	Nahlédněte do návodu pro kabeláž v kapitole 3. Je velmi důležité, aby byl řídicí systém dobře uzemněn a aby nebyly vodiče vysokého napětí vedeny paralelně s vodiči nízkého napětí. Připojte svorku M napájení snímačů 24 V DC na zem. Zvětšete zpoždění vstupních filtrů v systémovém datovém bloku.
Komunikační síť se poškodí při připojení na externí zařízení Je poškozen port počítače, port S7-200 nebo kabel PC/PPI	Komunikační kabel může vést pro nežádoucí proudy u zařízení bez galvanického oddělení, jako jsou programovatelné automaty, počítače nebo jiná zařízení, která jsou zapojena do sítě bez společného referenčního bodu. Tyto nežádoucí proudy mohou způsobit chyby v komunikaci nebo poškodit obvody.	- Návod pro kabeláž naleznete v kapitole 3 a návod pro síť v kapitole 7. - Použijte galvanicky oddělený PC/PPI kabel. - Pro propojení zařízení bez společného referenčního potenciálu použijte galvanicky oddělené repeatery RS-485. Informace o objednacích číslech pro vybavení S7-200 najdete v příloze E.
Jiné problémy s komunikací (STEP 7–Micro/WIN)	Informace o síťové komunikaci najdete v kapitole 7.	
Ošetření chyb	Informace o chybových kódech najdete v příloze C.	

Tvorba programu s polohovacím modulem

9

Polohovací modul EM 253 je speciální funkční modul, který generuje sekvence pulzů používané pro řízení rychlosti a polohy krokových motorů nebo servomotorů v otevřené smyčce. S automatem komunikuje přes rozšiřovací sběrnici. V konfiguraci vstupů a výstupů se objevuje jako inteligentní modul s osmi digitálními výstupy.

Na základě konfiguračních informací uložených v paměti V generuje polohovací modul sekvence pulzů potřebných pro řízení pohybu.

Aby se zjednodušilo použití řízení polohy v aplikaci, obsahuje STEP 7–Micro/WIN "Průvodce polohováním", který umožňuje kompletně nakonfigurovat polohovací modul během několika minut. STEP 7–Micro/WIN má také "Control panel pro EM 253", který umožňuje řídit, monitorovat a odzkoušet pohybové operace.

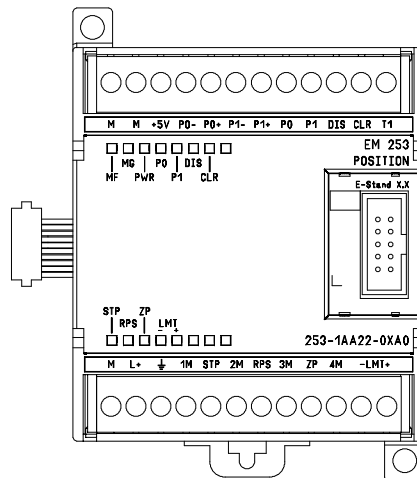
Přehled kapitoly

Funkce polohovacího modulu	244
Konfigurace polohovacího modulu	246
Polohovací instrukce vytvořené pomocí "Průvodce polohováním"	257
Vzorové programy	269
Monitorování polohovacího modulu "Control panelem pro EM 253"	274
Chybové kódy pro polohovací modul a polohovací instrukce	276
Témata pro pokročilé	278

Funkce polohovacího modulu

Polohovací modul přidává do řídicího systému funkce a výkon, které potřebujete pro řízení polohy v otevřené smyčce:

- ☐ Umožňuje vysokorychlostní řízení s rozsahem od 12 pulzů za sekundu až do 200.000 pulzů za sekundu
- ☐ Podporuje hladké (S křivka) nebo lineární zrychlení a zpomalení
- ☐ Obsahuje konfigurovatelný měřicí systém, který umožňuje vkládat data v inženýrských jednotkách (např. milimetry nebo centimetry) nebo jako počet pulzů
- ☐ Má konfigurovatelnou kompenzaci vůle
- ☐ Podporuje absolutní, relativní a manuální způsob řízení polohy
- ☐ Nabízí kontinuální provoz
- ☐ Nabízí až 25 pohybových profilů – každý s až 4 změnami rychlosti
- ☐ Nabízí čtyři různé režimy vyhledávání referenčního bodu s volbou počátečního směru vyhledávání a směru konečného pohybu pro každou sekvenci
- ☐ Má odnímatelné svorkovnice kabeláže pro snadnou instalaci a demontáž



Obr. 9-1 Polohovací modul EM 253

Pro vytvoření celé konfigurace a všech profilových informací použitých polohovacím modulem se používá STEP 7–Micro/WIN. Tyto informace jsou ukládány do automatu spolu s programovými bloky. Protože jsou všechny potřebné informace pro řízení polohy uloženy v S7-200, můžete polohovací modul vyměnit, aniž byste jej museli znovu programovat nebo konfigurovat.

Automat S7-200 rezervuje osm bitů registru obrazu výstupů (paměť Q) pro rozhraní s polohovacím modulem. Uživatelský program v S7-200 tyto bity používá pro řízení práce polohovacího modulu. Těchto osm výstupních bitů není spojeno s žádným fyzickým výstupem polohovacího modulu.

Polohovací modul má pět digitálních vstupů a čtyři digitální výstupy, které představují rozhraní pro řízení pohybu. Viz tabulku 9-1. Tyto vstupy a výstupy jsou fyzické vstupy a výstupy polohovacího modulu. V příloze A je uvedena podrobná specifikace polohovacího modulu; v ní jsou i schémata zapojení pro připojení polohovacího modulu k nejběžnějším zesilovačům motoru.

Tabulka 9-1 Vstupy a výstupy polohovacího modulu

Signál	Popis
STP	Vstup STP (STOP) způsobí, že modul zastaví pohyb, který právě probíhá. Požadovanou práci STP můžete zvolit v "Průvodci polohováním".
RPS	Vstup RPS (Reference Point Switch – snímač referenčního bodu) stanovuje referenční bod neboli výchozí polohu pro operace s absolutním pohybem.
ZP	Vstup ZP (Zero Pulse – nulový pulz) pomáhá stanovit referenční bod nebo výchozí polohu. Běžně zesilovač motoru vysílá ZP jednou za otáčku motoru.
LMT+ LMT–	Vstupy LMT+ a LMT– stanovují maximální meze pohybu při posunu. "Průvodce polohováním" umožňuje práci vstupů LMT+ a LMT– konfigurovat.
P0 P1 P0+, P0– P1+, P1–	P0 a P1 jsou tranzistorové pulzní výstupy s otevřeným kolektorem, které řídí pohyb směr a motoru. P0+, P0– a P1+, P1– jsou diferenční pulzní výstupy, které mají stejné funkce jako P0 a P1, ale dávají špičkovou kvalitu signálu. Výstupy s otevřeným kolektorem i diferenční výstupy jsou všechny aktivní ve stejnou dobu. V závislosti na požadavcích rozhraní zesilovače motoru si vyberete, kterou sadu pulzních výstupů budete používat.

Tabulka 9-1 Vstupy a výstupy polohovacího modulu, pokračování

Signál	Popis
DIS	DIS (Disable – Zablokování) je tranzistorový výstup s otevřeným kolektorem, používaný pro zablokování nebo povolení zesilovače motoru.
CLR	CLR (Clear – Smazání) je tranzistorový výstup s otevřeným kolektorem, používaný pro mazání registru počtu pulzů servopohonu.

Programování polohovacího modulu

STEP 7–Micro/WIN umožňuje snadno konfigurovat a programovat polohovací modul. Jednoduše proveďte tyto kroky:

1. Nakonfigurujte polohovací modul. STEP 7–Micro/WIN obsahuje “Průvodce polohováním”, pomocí kterého vypracujete tabulku konfigurace, profilů a polohovacích instrukcí. Informace o konfigurování polohovacího modulu najdete na straně 246.
2. Přezkoušejte činnost polohovacího modulu. STEP 7–Micro/WIN má “Control panel pro EM 253”, pomocí kterého lze přezkoušet zapojení vstupů a výstupů, konfiguraci polohovacího modulu a činnost profilů pohybu. Informace o “Control panelu pro EM 253” najdete na straně 274.
3. Vytvořte program, který má S7-200 vykonávat. “Průvodce polohováním” automaticky vytvoří instrukce, které do programu vložíte. Informace o instrukcích najdete na straně 257. Do programu vložte následující instrukce:
 - Chcete-li povolit polohovací modul, vložte instrukci POSx_CTRL. Abyste zajistili, že se tato instrukce provede v každém programovém cyklu, použijte SM0.0 (vždy zapnuto).
 - Chcete-li motor posunout do konkrétní polohy, použijte instrukce POSx_GOTO nebo POSx_RUN. Instrukce POSx_GOTO provede posun na místo specifikované vstupy uživatelského programu. Instrukce POSx_RUN provede profily pohybu, které jste nakonfigurovali pomocí “Průvodce polohováním”.
 - Abyste pro daný pohyb mohli použít absolutní souřadnice, musíte stanovit nulovou polohu vaší aplikace. Pro stanovení nulové polohy použijte instrukce POSx_RSEEK nebo POSx_LDPOS.
 - Ostatní instrukce vytvořené “Průvodcem polohováním” poskytují funkce pro typické aplikace a pro konkrétní aplikaci jsou jen volitelné.
4. Zkompilujte uživatelský program a proveďte download systémového, datového a programového bloku do S7-200.



Tip

Informace o připojení polohovacího modulu k několika běžným zesilovačům krokových motorů najdete v příloze A.



Tip

Pro sesouhlasení implicitních nastavení v “Průvodci polohováním” nastavte DIP přepínače na regulátoru krokového motoru na 10.000 pulzů na otáčku.

Konfigurace polohovacího modulu



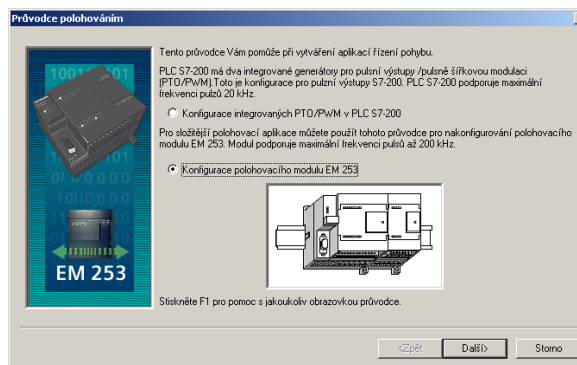
Řízení polohy

Aby polohovací modul řídil aplikaci, musíte pro něj vytvořit tabulku konfigurace a profilů. Díky "Průvodci polohováním" je postup konfigurace rychlý a snadný – průvodce vás tímto procesem provede krok za krokem. Detailní informace o tabulce konfigurace a profilů najdete v Tématech pro pokročilé na straně 278.

"Průvodce polohováním" umožní vytvořit tabulku konfigurace a profilů offline. Konfiguraci můžete vytvořit, aniž byste byli připojeni na CPU S7-200 s instalovaným polohovacím modulem.

Abyste mohli spustit "Průvodce polohováním", musí být váš projekt zkompileován a nastaven do režimu symbolického adresování.

Chcete-li spustit "Průvodce polohováním", klikněte nejprve na ikonu "Nástroje" v navigační liště a pak dvakrát na ikonu "Průvodce polohováním" nebo vyberte příkaz menu **Nástroje > Průvodce polohováním**.



Obr. 9-2 "Průvodce polohováním"

"Průvodce polohováním" umožňuje konfigurovat činnost polohovacího modulu nebo činnost instrukce pulzních výstupů PTO/PWM. Po zvolení polohovacího modulu klikněte na "Další"; průvodce vás provede jednotlivými kroky, potřebnými pro konfigurování polohovacího modulu.

Vložení umístění polohovacího modulu

Musíte definovat parametry modulu a sady profilů pohybu pro aplikaci tím, že vložíte typ a umístění modulu. "Průvodce polohováním" tuto úlohu zjednodušuje automatickým načtením umístění inteligentního modulu. Stačí kliknout na tlačítko "Čtení modulů".

U CPU S7-200 s verzí firmware starší než 1.2 musíte instalovat inteligentní modul přímo vedle CPU, aby "Průvodce polohováním" modul konfiguroval.

Výběr systému odměřování

Je třeba vybrat systém odměřování, který bude v průběhu konfigurace používán. Můžete si zvolit inženýrské jednotky, nebo pulzy. Pokud si vyberete pulzy, nemusíte uvádět žádné další informace. Pokud zvolíte inženýrské jednotky, musíte vložit následující data: počet pulzů na jednu otáčku motoru (údaje najdete ve specifikaci motoru nebo pohonu), základní jednotku měření (jako například palec, stopa, milimetr nebo centimetr) a velikost pohybu danou jednou otáčkou motoru.

STEP 7–Micro/WIN podporuje "Control panel pro EM253", který umožňuje měnit počet jednotek na otáčku po tom, co byl polohovací modul nakonfigurován.

Pokud byste někdy v pozdější době měnili systém měření, musíte v "Průvodci polohováním" smazat celou konfiguraci včetně všech generovaných instrukcí. Následně musíte vložit hodnoty, které jsou v souladu s novým systémem měření.

Editace implicitní konfigurace vstupů a výstupů

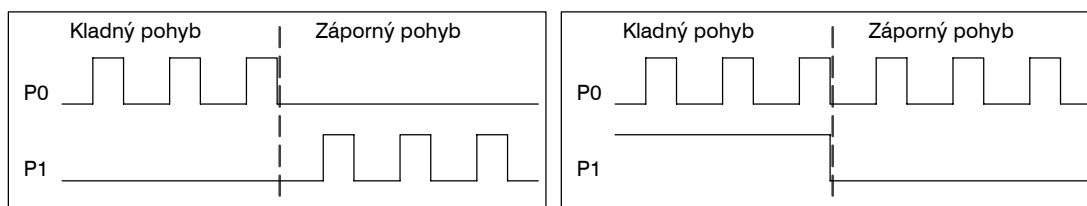
V "Průvodci polohováním" je i volba pro pokročilé "Další možnosti", která umožňuje prohlížet a editovat implicitní konfiguraci vstupů a výstupů polohovacího modulu:

- ☐ Záložka "Aktivní úroveň vstupů" mění nastavení úrovně aktivace. Jestliže je úroveň signálu nastavena na "Vysoký", je logická 1 čtena tehdy, protéká-li vstupem proud. Jestliže je úroveň nastavena na "Nízký", je logická 1 čtena tehdy, když vstupem proud neprotéká. Úroveň logická 1 se vždy interpretuje tak, že znamená aktivní podmínku. LED diody svítí, jestliže vstupem protéká proud bez ohledu na úroveň aktivace. (Implicitní nastavení = aktivní vysoká úroveň)
- ☐ Záložka "Časy vstupních filtrů" umožňuje specifikovat dobu zpoždění (rozsah od 0,20 ms do 12,80 ms) pro filtraci vstupů STP, RPS, LMT+ a LMT-. Toto zpoždění pomáhá filtrovat šum ze vstupní kabeláže, který by mohl způsobit nepředpokládanou změnu stavu vstupů. (Implicitně nastavená hodnota = 6,4 ms)
- ☐ Záložka "Výstupy pro pulzy a směr" umožňuje specifikovat způsob řízení směru. Nejprve musíte specifikovat polaritu výstupů.

Zvolena kladná polarita

V případě aplikace, která používá kladnou polaritu, vyberte jeden z následujících způsobů (ukázaných na obrázku 9-3), který bude vyhovovat pohonu a orientaci aplikace:

- ☐ Polohovací modul vysílá pulzy z výstupu P0 pro kladný pohyb a pulzy z výstupu P1 pro záporný pohyb.
- ☐ Polohovací modul vysílá pulzy z výstupu P0. Modul zapne výstup P1 pro kladný pohyb a vypne výstup P1 pro záporný pohyb. (Toto je implicitní nastavení.)

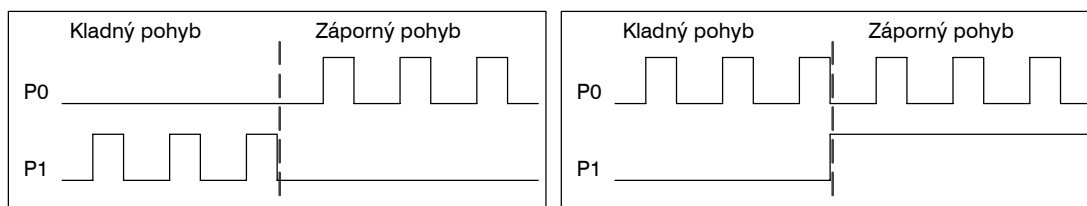


Obr. 9-3 Možnosti volby směru pohybu u kladné polaritě

Zvolena záporná polarita

V případě aplikace, která používá zápornou polaritu, vyberte jeden z následujících způsobů (ukázaných na obrázku 9-4), který bude vyhovovat pohonu a orientaci aplikace:

- ☐ Polohovací modul vysílá pulzy z výstupu P0 pro záporný pohyb a pulzy z výstupu P1 pro kladný pohyb.
- ☐ Polohovací modul vysílá pulzy z výstupu P0. Modul vypne výstup P1 pro kladný pohyb a zapne výstup P1 pro záporný pohyb.



Obr. 9-4 Možnosti volby směru pohybu u záporné polaritě

Konfigurace odezvy modulu na fyzické vstupy

Musíte specifikovat, jak polohovací modul reaguje na vstupy LMT+, LMT– a STP: nereaguje (ignoruje vstupní podmínku), zastaví se zpomalením (implicitní) nebo okamžitě.



Varování

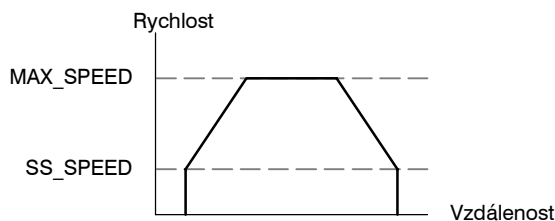
Řídicí zařízení mohou selhat a dostat se do nebezpečného stavu, který může mít za následek nepředvídatelné chování řízeného zařízení. Toto nepředvídatelné chování by mohlo mít za následek smrt nebo vážné zranění osob nebo poruchu zařízení.

Funkce omezení a zastavení polohovacího modulu jsou implementovány logikou a neposkytují úroveň ochrany danou elektromechanickými ovládacími prvky. Použijte funkci nouzového zastavení, elektromechanické spínače nebo jiné doplňující ochranné obvody nezávislé na CPU S7-200.

Vložení maximální rychlosti a spuštění/zastavení

Musíte specifikovat maximální rychlost (MAX_SPEED) a rychlost pro spuštění/zastavení (SS_SPEED):

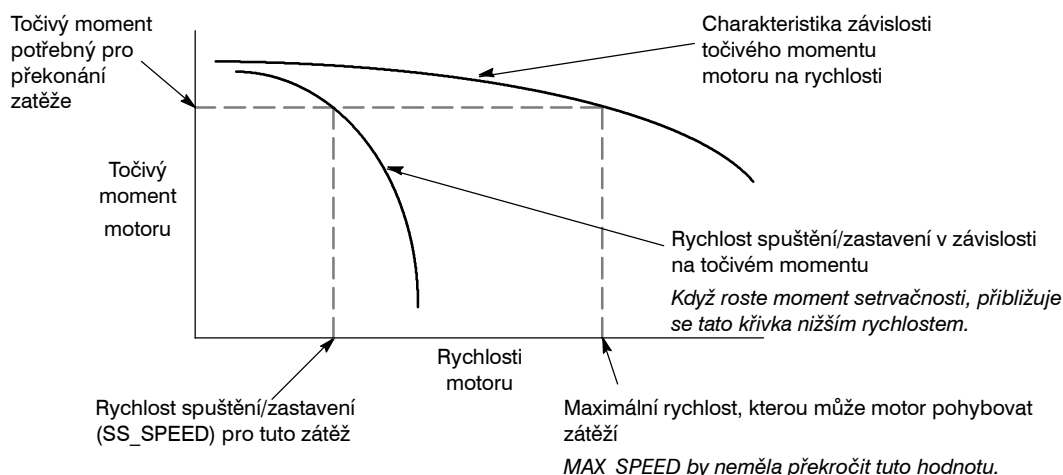
- ☐ **MAX_SPEED:** vložte hodnotu optimální provozní rychlosti pro vaši aplikaci v rámci možností točivého momentu motoru. Točivý moment potřebný pro pohyb zatížení je určen třením, setrvačností a dobou zrychlení/zpomalení. "Průvodce polohováním" vypočítá a zobrazí minimální rychlost, která může být nastavena polohovacím modulem pro specifikované MAX_SPEED.
- ☐ **SS_SPEED:** vložte hodnotu v rámci možností motoru, která dokáže pohybovat s daným zatížením při nízké rychlosti. Pokud je hodnota SS_SPEED příliš nízká, mohou se motor i zátěž chvět nebo se pohybovat krátkými skoky na začátku a na konci posunu. Pokud je hodnota SS_SPEED příliš vysoká, může motor při spuštění ztrácet pulzy a zátěž může při pokusu o zastavení motor přetížit.



Obr. 9-5 Maximální rychlost a rychlost spuštění/zastavení

Technické specifikace motoru uvádějí rychlost spuštění/zastavení pro motor a dané zatížení různě. Běžně se používá hodnota SS_SPEED 5 % až 15 % hodnoty MAX_SPEED. Hodnota SS_SPEED musí být větší než minimální rychlost, zobrazená při specifikaci MAX_SPEED.

Při volbě správné rychlosti pro vaši aplikaci pomůže technická specifikace motoru. Obrázek 9-6 ukazuje typickou momentovou charakteristiku motoru.



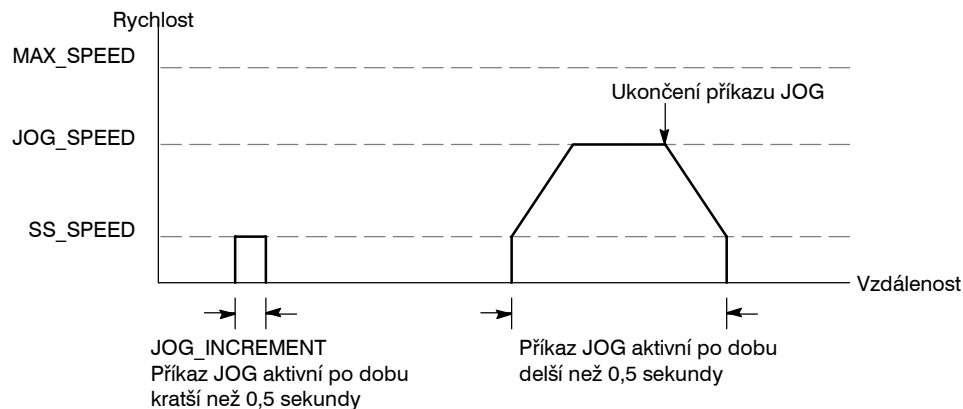
Obr. 9-6 Typická momentová charakteristika motoru

Vložení parametrů manuálního posuvu

Příkaz "Jog" (manuální posuv) se používá pro ruční posuv nástroje na požadované místo. V "Průvodci polohováním" specifikujte následující hodnoty parametrů pomalého posuvu:

- ☐ JOG_SPEED: JOG_SPEED (rychlost motoru pro pomalý posuv) je maximální rychlost, které je možné docílit, dokud zůstává aktivní příkaz JOG.
- ☐ JOG_INCREMENT: Vzdálenost, o kterou je nástroj posunut krátkodobým příkazem JOG.

Obrázek 9-7 ukazuje funkci příkazu "Jog". Když polohovací modul dostane příkaz "Jog", spustí časovač. Jestliže příkaz "Jog" skončí dříve, než uplyne 0,5 sekundy, posune polohovací modul nástroj o vzdálenost specifikovanou v JOG_INCREMENT rychlostí definovanou v SS_SPEED. Jestliže je příkaz "Jog" stále aktivní i po uplynutí 0,5 sekundy, zrychlí polohovací modul až na JOG_SPEED. Pohyb pokračuje, dokud není ukončen příkaz "Jog". Polohovací modul pak zastavení se zpomalením. Příkaz "Jog" můžete aktivovat buď z "Control panelu pro EM 253", nebo pomocí polohovací instrukce.

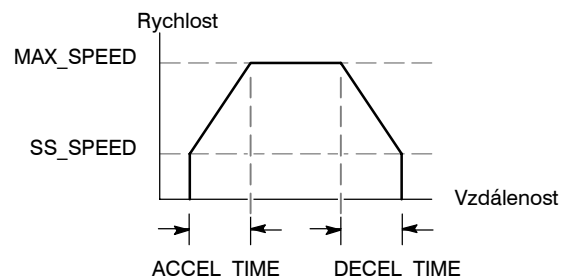


Obr. 9-7 Znáznornění funkce příkazu JOG

Vložení doby zrychlení a zpomalení

Jako součást konfigurace polohovacího modulu nastavte dobu zrychlení a zpomalení. Implicitní nastavení pro dobu zrychlení i pro dobu zpomalení je 1 sekunda. Motory mohou běžně pracovat s dobou kratší, než je 1 sekunda. Specifikujte následující doby v milisekundách:

- ☐ ACCEL_TIME: Doba potřebná k tomu, aby motor zrychlil z SS_SPEED na MAX_SPEED.
Implicitní nastavení = 1000 ms
- ☐ DECEL_TIME: Doba potřebná k tomu, aby motor zpomalil z MAX_SPEED na SS_SPEED.
Implicitní nastavení = 1000 ms



Obr. 9-8 Doba zrychlení a zpomalení



Tip

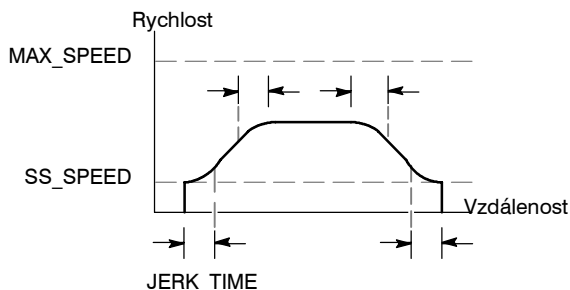
Doby zrychlení a zpomalení motoru jsou stanoveny pomocí pokusu a omylu. Měli byste začít vložím větší hodnoty pomocí "Průvodce polohováním". Při testování aplikace pak můžete používat "Control panel pro EM 253" pro seřízení hodnot podle potřeby. Optimalizujte toto nastavení pro aplikaci postupným zkracováním těchto dob, dokud se motor nezačne zastavovat.

Vložení doby zaoblení

Kompensace rázů dává plynulejší řízení polohy zmenšením změn rychlosti ve zrychlovací a zpomalovací části pohybového profilu. Viz obr. 9-9. Zmenšení rázů zlepšuje polohování. Kompensace rázu je také známá jako profilování "S křivky." Kompensaci rázů je možné aplikovat pouze na jednoduché profily s jedním krokem. Tato kompensace se aplikuje stejně na počáteční i koncovou část křivky zrychlení i křivky zpomalení. Kompensace rázu se neaplikuje na počáteční a konečný krok mezi nulovou rychlostí a SS_SPEED.

Kompensaci rázů specifikujete vložení hodnoty času (JERK_TIME). Je to čas potřebný na to, aby se zrychlení změnilo z nulového na maximální, definované v MAX_SPEED, SS_SPEED a ACCEL_TIME, nebo ekvivalentně pro DECEL_TIME. Delší doba zaoblení způsobí plynulejší chod s menším prodloužením celkové doby cyklu, než k jakému by se dospělo pouhým prodloužením ACCEL_TIME a DECEL_TIME. Nulová hodnota znamená, že nemá být kompensace aplikována.

(Implicitně nastavená hodnota = 0 ms)



Obr. 9-9 Kompensace rázů



Tip

Vhodná implicitní hodnota pro JERK_TIME je 40 % ACCEL_TIME.

Konfigurace referenčního bodu a parametry jeho vyhledávání

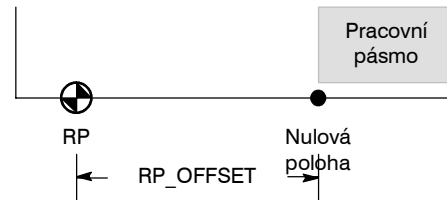
Pokud aplikace vyžaduje pohyby od absolutní polohy, musíte stanovit nulovou polohu, která váže měření polohy na známý bod ve fyzickém systému. Jeden způsob je mít ve fyzickém systému referenční bod (RP). Polohovací modul má vnější vstup pro referenční bod (RPS), který se používá pro hledání RP.

Můžete nakonfigurovat parametry hledání referenčního bodu (RP Seek), které řídí to, jak pohybová aplikace hledá RP. RP může být umístěn ve středu aktivní zóny RPS, může být umístěn na hraně aktivní zóny RPS nebo může ležet o specifikovaný počet přechodů vstupu nulových pulzů (ZP) od hrany aktivní zóny RPS. Chcete-li nakonfigurovat RP, vložte následující informace:

- ☐ Specifikujte rychlost motoru pro hledání RP:
 - RP_FAST je výchozí rychlost, kterou modul použije při provádění příkazu vyhledání RP. Běžně se hodnota RP_FAST rovná přibližně 2/3 hodnoty MAX_SPEED.
 - RP_SLOW je rychlost použitá při konečném přiblížení k RP. Abyste ho neminuli, používá se pro přiblížování k RP pomalejší rychlost. Běžně se hodnota RP_SLOW rovná hodnotě SS_SPEED.
- ☐ Pro vyhledání RP specifikujte počáteční směr hledání (RP_SEEK_DIR) a směr konečného přiblížení (RP_APPR_DIR). Směr se specifikuje jako kladný nebo záporný.
 - RP_SEEK_DIR je počáteční směr operace vyhledávání RP. Běžně je to směr z pracovního pásma do blízkosti RP. Mezní spínače hrají důležitou roli při definování oblasti, která je prohledávána na RP. Při provádění operace vyhledávání RP může mít setkání s mezním spínačem za následek obrácení směru, který umožňuje pokračovat v hledání. (Implicitní nastavení = Záporný)
 - RP_APPR_DIR je směr konečného přiblížení k RP. Pro redukci vůle a větší přesnost se RP_APPR_DIR pohybuje ve stejném směru jako normální pracovní cyklus. (Implicitní nastavení = Kladný)

”Průvodce polohováním” má pro referenční bod volbu pro pokročilé, která umožňuje specifikovat offset RP (RP_OFFSET), což je vzdálenost mezi RP a nulovou polohou. Viz obr. 9-10. RP se určí metodou lokalizace přesné polohy vzhledem k RPS. Chcete-li nakonfigurovat posunutí RP, vložte následující hodnoty:

- ☐ RP_OFFSET: Vzdálenost RP od nulové polohy fyzického odměřovacího systému. Implicitně nastavená hodnota = 0
- ☐ Kompenzace vůle: Vzdálenost, o kterou se musí motor pohnout, aby eliminoval vůli (např. v převodech) v systému při změně směru. Kompenzace vůle je vždy kladná hodnota. Implicitně nastavená hodnota = 0

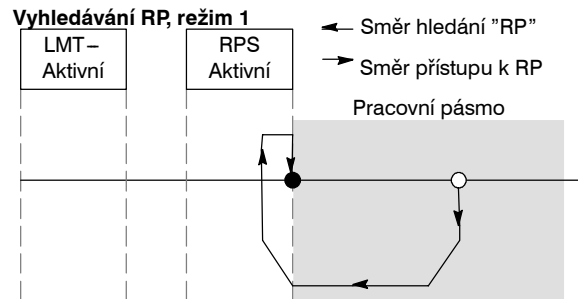


Obr. 9-10 Vztah mezi RP a nulovou polohou

Konfigurace sekvence vyhledávání RP

Můžete nakonfigurovat sekvenci, kterou polohovací modul používá pro hledání referenčního bodu. Obrázek 9-11 ukazuje zjednodušený diagram implicitní sekvence vyhledávání RP. Pro sekvenci hledání RP si můžete vybrat následující volby:

- ☐ Vyhledávání RP, režim 0: Neprovádí sekvenci ”RP Seek”
- ☐ Vyhledávání RP, režim 1: RP je tam, kde vstup RPS přejde do aktivního stavu při přiblížení ze strany pracovního pásma. (Implicitně)
- ☐ Vyhledávání RP, režim 2: RP leží ve středu aktivní oblasti vstupu RPS.
- ☐ Vyhledávání RP, režim 3: RP je umístěn mimo aktivní oblast vstupu RPS. RP_Z_CNT specifikuje, kolik má být obdrženo pulzů na vstupu ZP po tom, co RPS přestane být aktivní.
- ☐ Vyhledávání RP, režim 4: RP leží obvykle uvnitř aktivní oblasti vstupu RPS. RP_Z_CNT specifikuje, kolik má být obdrženo pulzů na vstupu ZP po tom, co RPS přejde do aktivního stavu.



Obr. 9-11 Implicitní sekvence hledání RP (zjednodušená)



Tip

Aktivní oblast RPS (což je vzdálenost, po kterou vstup RPS zůstává aktivní) musí být větší než vzdálenost potřebná ke zpomalení z rychlosti RP_FAST na rychlost RP_SLOW. Pokud je tato vzdálenost příliš krátká, generuje polohovací modul chybu.

Více informací o různých sekvencích hledání RP pro polohovací modul najdete na obrázcích 9-14 až 9-17 na stranách 254 až 255.

Konfigurace profilů pohybu pro polohovací modul

Profil je předdefinovaný popis pohybu, skládající se z jedné nebo více hodnot rychlosti, které způsobují pohyb od výchozího do koncového bodu. Abyste mohli modul používat, nemusíte definovat žádný profil. "Průvodce polohováním" má instrukční podprogram (POSx_GOTO), který můžete používat pro řízení pohybu.

- ☐ Počet profilů: můžete si vybrat maximálně 25 profilů.
- ☐ Adresa příkazového bytu: musíte vložit adresu příkazového bytu ve výstupní paměti (Q) pro polohovací modul. Popis číslování vstupů a výstupů najdete na obrázku 4-10 na straně 31.
- ☐ Adresa tabulky konfigurace/profilů: musíte vložit počáteční paměťovou adresu tabulky konfigurace/profilů, ve které jsou uložena konfigurační data polohovacího modulu a data všech profilů. Pro konfigurační data polohovacího modulu je zapotřebí 92 bytů paměti V, pro každý profil je potřeba 34 bytů paměti V. Například pro tabulku konfigurace profilů polohovacího modulu s jedním profilem je zapotřebí 126 bytů paměti V.

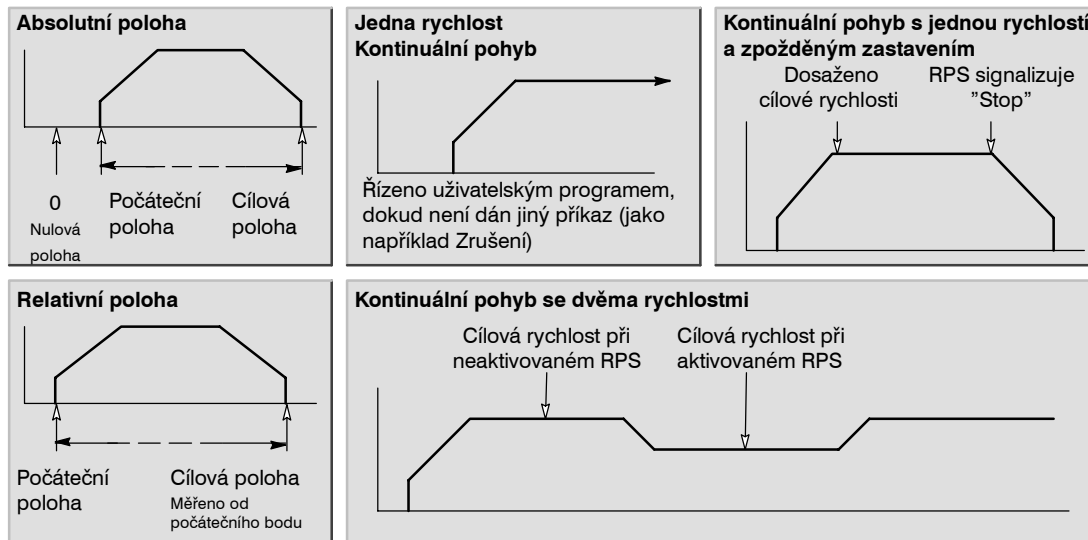
"Průvodce polohováním" umí navrhnout adresu nevyužitého bloku správné velikosti v paměti V.

Definice profilů pohybu

"Průvodce polohováním" obsahuje funkce pro Definici profilů pohybu, pomocí kterých definujete profily aplikace. Pro každý profil vyberte provozní režim a definujte specifické vlastnosti každého jednotlivého kroku profilu. "Průvodce polohováním" také umožňuje definovat pro každý profil symbolický název tím, že ho jednoduše vložíte při definování profilu. Když jste konfiguraci profilu dokončili, můžete ji uložit a vytisknout kopii parametrů.

Výběr pracovního režimu profilu

Profil konfiguruje podle pracovního režimu buď jako absolutní polohu, relativní polohu, kontinuální pohyb s jednou rychlostí nebo kontinuální pohyb se dvěma rychlostmi. Na obrázku 9-12 jsou ukázány různé pracovní režimy.



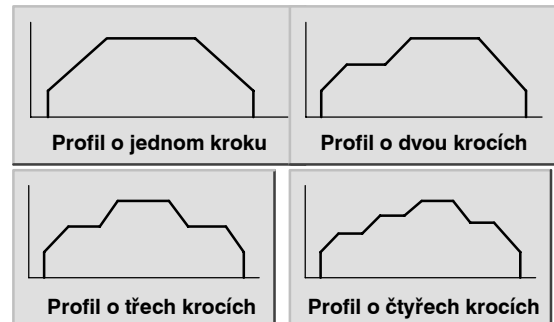
Obr. 9-12 Výběr režimu polohovacího modulu

Tvorba kroků profilu

Krok je vzdálenost, o kterou se posune nástroj včetně vzdálenosti pokryté během doby zrychlení a zpomalení. Každý profil může mít až čtyři jednotlivé kroky.

Specifikujte cílovou rychlost a polohu pro každý krok. Pokud potřebujete více než jeden krok, jednoduše klikněte na tlačítko "Nový krok" a vložte informace pro každý krok profilu. Na obrázku 9-13 vidíte čtyři možné profily; existují ale také jiné možné kombinace.

Pouhým kliknutím na tlačítko "Nakreslit krok" si můžete prohlédnout grafické zobrazení tohoto kroku tak, jak ho vypočítal "Průvodce polohováním". To umožňuje kontrolovat a editovat každý krok snadno a interaktivně.



Obr. 9-13 Příklad profilů pohybu

Dokončení konfigurace polohovacího modulu

Když jste nakonfigurovali operace polohovacího modulu, jednoduše klikněte na "Dokončit" a "Průvodce polohováním" provede následující úlohy:

- ☐ Vloží konfiguraci modulu a tabulku profilů do datového bloku programu S7-200
- ☐ Vytvoří tabulku globálních symbolů pro parametry pohybu
- ☐ Do programového bloku projektu přidá podprogramy pro pohybové instrukce pro použití ve vaší aplikaci

Chcete-li modifikovat jakýkoliv prvek konfigurace nebo hodnotu profilu, můžete znovu spustit "Průvodce polohováním".



Tip

Protože "Průvodce polohováním" provádí změny v programovém, datovém i systémovém bloku, zkontrolujte, je-li proveden download všech tří bloků do CPU S7-200. Jinak by polohovací modul nemusel mít všechny komponenty programu, které potřebuje pro správný provoz.

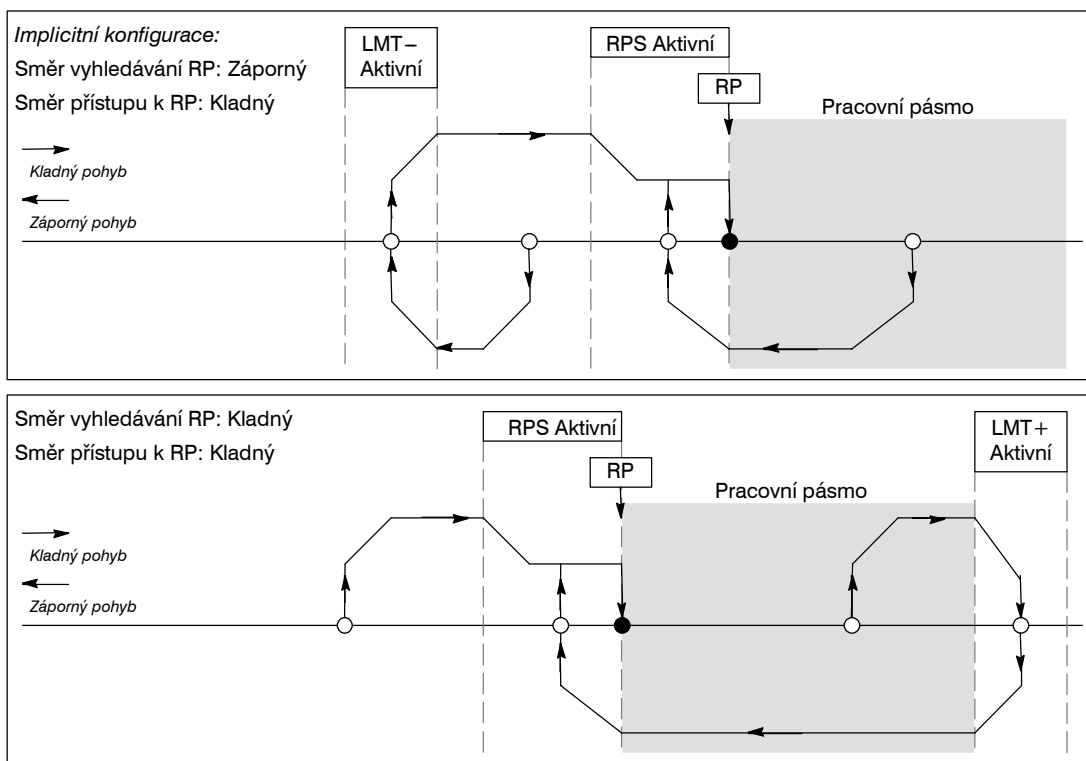
Vysvětlení režimů vyhledávání RP podporovaných polohovacím modulem

Následující obrázky ukazují schémata různých voleb pro každý režim vyhledávání RP.

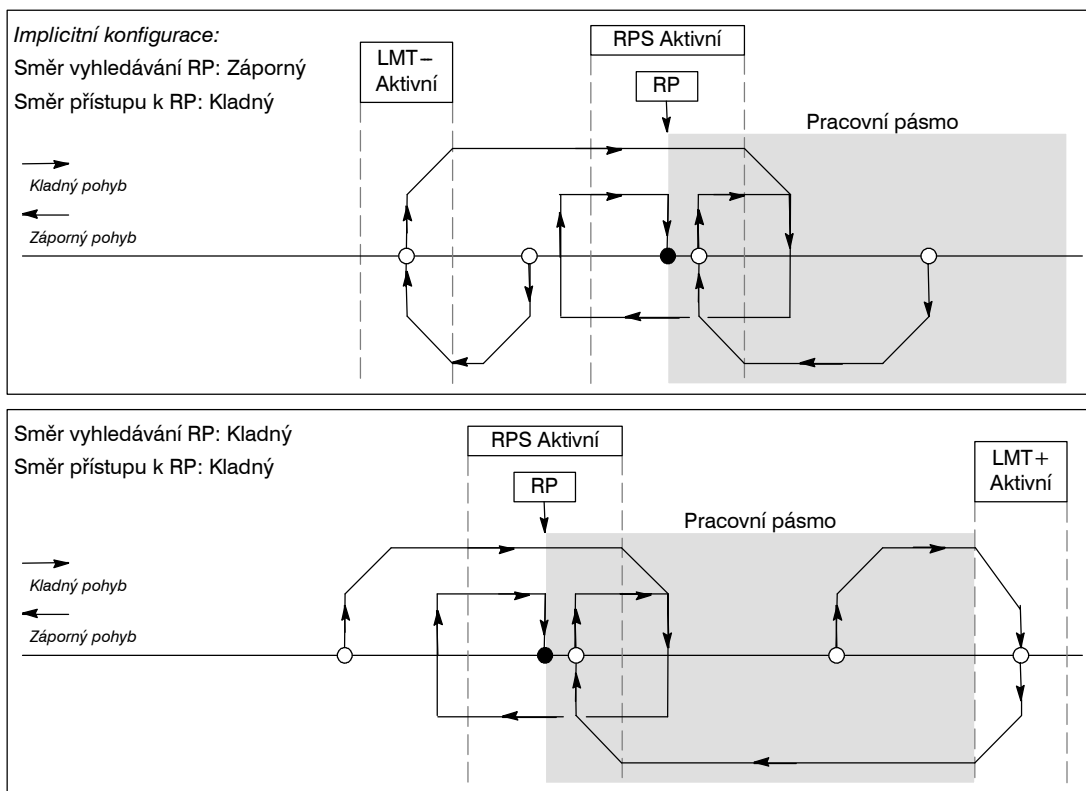
- ☐ Obrázek 9-14 ukazuje dvě z možností volby pro vyhledávání RP, režim 1. V tomto režimu je RP lokalizován tam, kde vstup RPS přejde do aktivního stavu při přiblížení ze strany pracovního pásma.
- ☐ Obrázek 9-15 ukazuje dvě z možností volby pro vyhledávání RP, režim 2. V tomto režimu je RP umístěn ve středu aktivní oblasti vstupu RPS.
- ☐ Obrázek 9-16 ukazuje dvě z možností volby pro vyhledávání RP, režim 3. V tomto režimu je RP umístěn o specifikovaný počet pulzů na vstupu (ZP) vně aktivní oblasti vstupu RPS.
- ☐ Obrázek 9-17 ukazuje dvě z možností volby pro vyhledávání RP, režim 4. V tomto režimu je RP umístěn o specifikovaný počet pulzů na vstupu (ZP) uvnitř aktivní oblasti vstupu RPS.

Pro každý režim existují čtyři kombinace směru vyhledávání RP a směru přístupu k RP. (Na obrázku jsou ukázány pouze dvě kombinace.) Tyto kombinace určují tvar operace vyhledávání RP. Pro každou kombinaci také existují čtyři různé počáteční body:

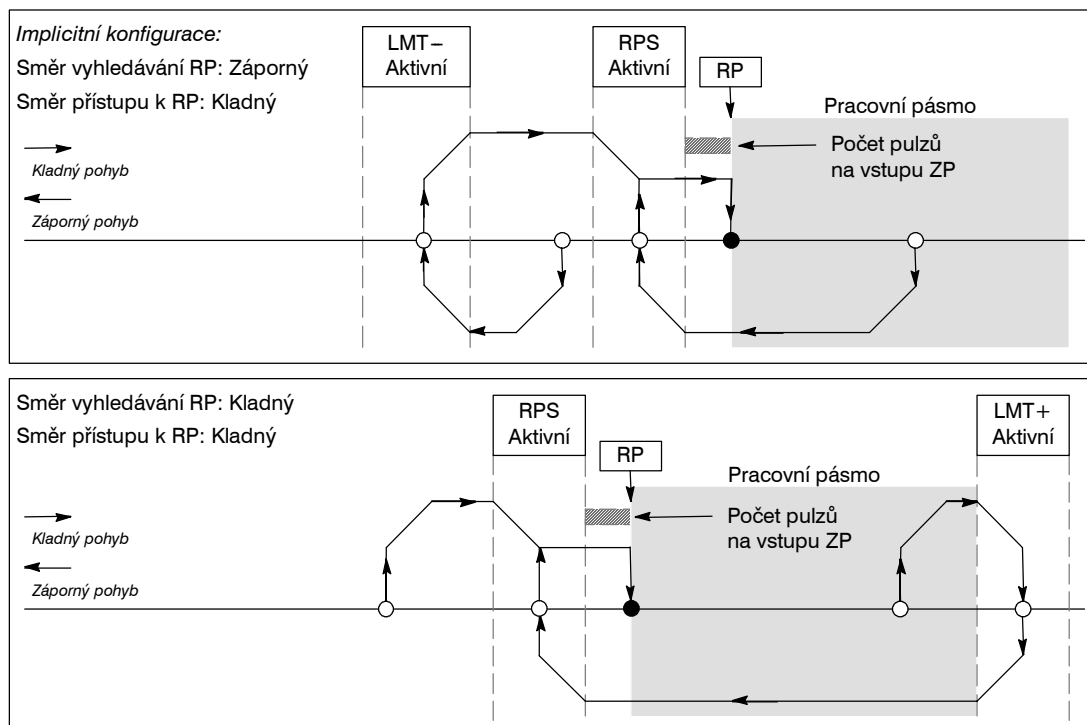
Pracovní pásma každého diagramu byla umístěna tak, aby pohyb od referenčního bodu k pracovnímu pásmu vyžadoval pohyb ve stejném směru, jako je směr přístupu k RP. Díky tomu, že je umístění pracovního pásma vybráno tímto způsobem, jsou eliminovány vůle mechanického převodového systému pro první pohyb směrem k pracovnímu pásmu po vyhledání referenčního bodu.



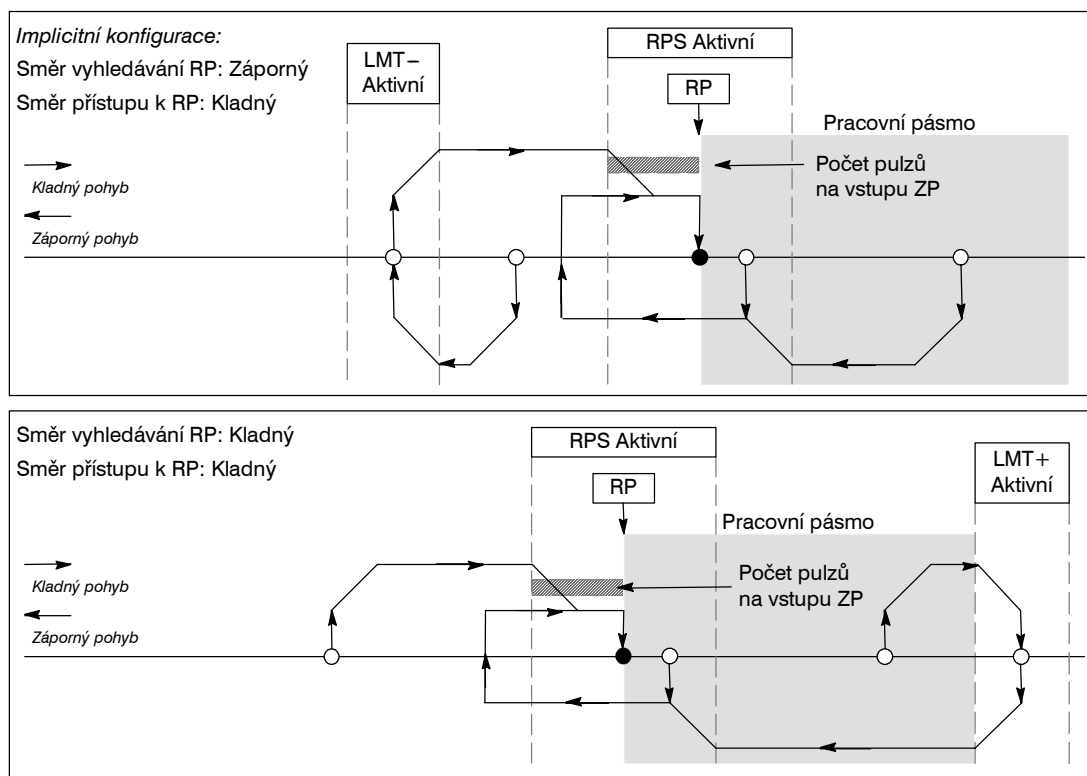
Obr. 9-14 Vyhledávání RP, režim 1



Obr. 9-15 Vyhledávání RP: Režim 2



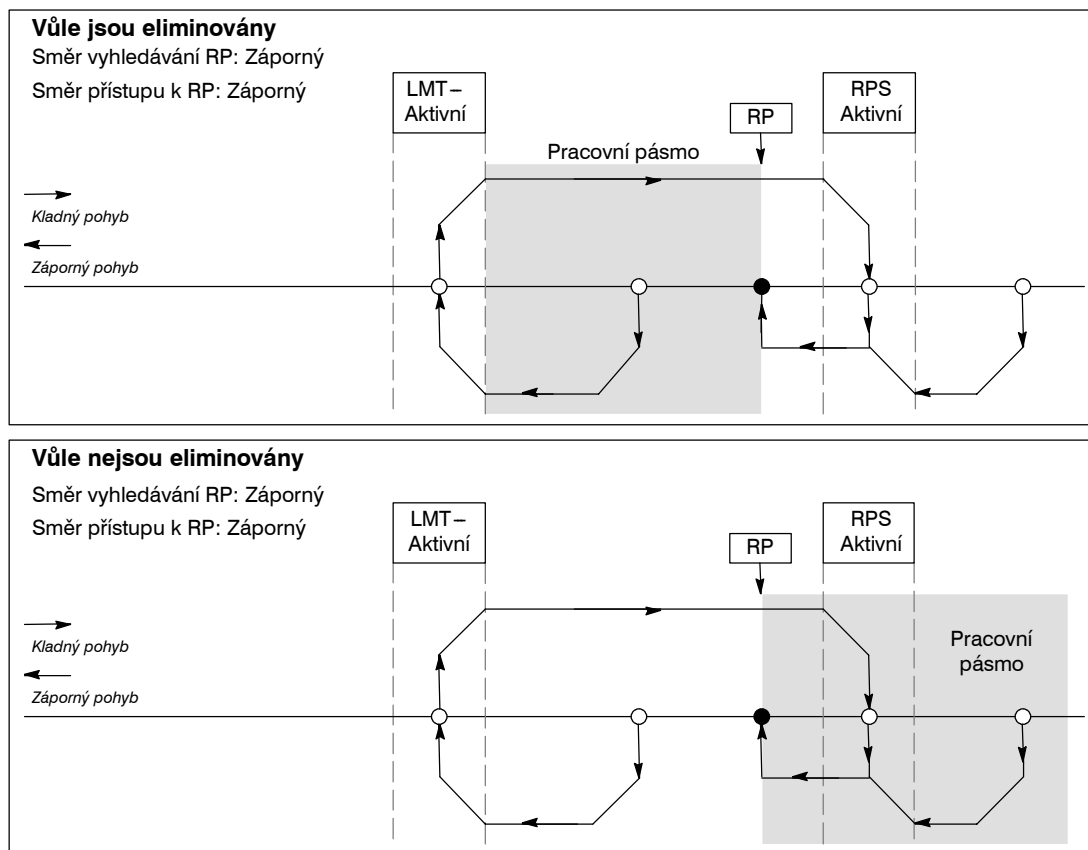
Obr. 9-16 Vyhledávání RP: Režim 3



Obr. 9-17 Vyhledávání RP: Režim 4

Výběr takového umístění pracovního pásma, které eliminuje vůli

Obrázek 9-18 ukazuje pracovní pásmo ve vztahu k referenčnímu bodu (RP), aktivní zóně RPS a koncovým spínačům (LMT+ a LMT–) pro směr přístupu, který vylučuje mechanickou vůli. V druhé části obrázku je pracovní pásmo umístěno tak, že nejsou eliminovány mechanické vůle. Na obrázku 9-18 je vyhledávání RP, režim 3. Obdobné umístění pracovního pásma je možné, i když se nedoporučuje, pro každou vyhledávací sekvenci všech ostatních režimů vyhledávání RP.



Obr. 9-18 Umístění pracovního pásma s eliminací vůle a bez eliminace

Polohovací instrukce vytvořené pomocí "Průvodce polohováním"

"Průvodce polohováním" usnadňuje řízení polohovacího modulu tím, že vytváří jednoznačné podprogramy na základě polohy modulu a voleb konfigurace, které jste si vybrali. Každá polohovací instrukce má prefix "POSx_", kde x znamená umístění modulu. Protože je každá polohovací instrukce podprogramem, používá 11 polohovacích instrukcí 11 podprogramů.



Tip

Díky polohovacím instrukcím se zvětšuje část paměti potřebná pro uživatelský program až o 1700 bytů. Abyste zmenšili spotřebovanou část paměti, můžete vymazat nepoužívané polohovací instrukce. Chcete-li obnovit vymazanou polohovací instrukci, jednoduše znovu spusťte "Průvodce polohováním".

Návod pro užívání polohovacích instrukcí

Musíte zajistit, aby byla v danou dobu aktivní pouze jedna polohovací instrukce.

Instrukce POSx_RUN a POSx_GOTO můžete provést z podprogramu přerušení. Je ale velmi důležité, abyste se nepokoušeli spustit instrukci v podprogramu přerušení, pokud modul právě zpracovává jiný příkaz. Když spustíte některou instrukci v podprogramu přerušení, můžete použít výstupy instrukce POSx_CTRL pro sledování toho, kdy polohovací modul pohyb dokončí.

"Průvodce polohováním" automaticky nakonfiguruje hodnoty parametrů rychlosti (Speed a C_Speed) a parametrů polohy (Pos nebo C_Pos) v souladu se systémem odměřování, který jste vybrali. V případě pulzů mají tyto parametry hodnotu DINT. U inženýrských jednotek jsou parametry hodnoty REAL. Například: výběr centimetrů (cm) uloží parametry polohy jako hodnoty REAL v centimetrech a parametry rychlosti jako hodnoty REAL v centimetrech za sekundu (cm/s).

Pro konkrétní úkoly řízení polohy jsou zapotřebí následující polohovací instrukce:

- ☐ Vložte do svého programu instrukci POSx_CTRL a použijte kontakt SM0.0 pro její provedení v každém programovém cyklu.
- ☐ Chcete-li specifikovat pohyb k absolutní poloze, musíte pro stanovení nulové polohy nejprve použít instrukci POSx_RSEEK, nebo instrukci POSx_LDPOS.
- ☐ Pro pohyb na konkrétní místo na základě vstupů z uživatelského programu použijte instrukci POSx_GOTO.
- ☐ Pro spuštění profilů pohybu, které jste nakonfigurovali pomocí "Průvodce polohováním", použijte instrukci POSx_RUN.

Ostatní polohovací instrukce jsou volitelné.

Instrukce POSx_CTRL

Instrukce POSx_CTRL (Řízení) povoluje a inicializuje polohovací modul tak, že mu automaticky přikazuje, aby nahrál tabulku konfigurace profilů pokaždé, když automat S7-200 přejde do režimu RUN.

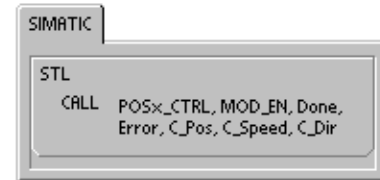
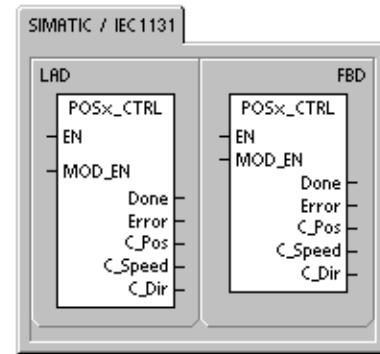
Tuto instrukci použijte ve svém projektu pouze jednou a zajistěte, aby ji váš program volal při každém programovém cyklu. Jako vstup pro parametr EN použijte SM0.0 (Vždy zapnutý).

Parametr EN musí být zapnutý, aby povolil ostatním polohovacím instrukcím posílat příkazy polohovacímu modulu. Pokud se parametr EN vypne, polohovací modul přeruší jakýkoliv probíhající příkaz.

Výstupní parametry instrukce POSx_CTRL ukazují aktuální stav polohovacího modulu.

Parametr "Done" (Dokončeno) přejde do stavu log. 1, když polohovací modul dokončí jakoukoliv instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.



Parametr C_Pos je aktuální poloha modulu. V závislosti na jednotkách měření je tato hodnota buď počet pulzů (DINT), nebo počet inženýrských jednotek (REAL).

Parametr C_Speed uvádí aktuální rychlost modulu. Pokud jste systém odměřování polohovacího modulu nakonfigurovali na pulzy, C_Speed je hodnota DINT, která obsahuje počet pulzů/sekundu. Pokud jste systém odměřování nakonfigurovali na inženýrské jednotky, C_Speed je hodnota REAL, která obsahuje vybrané inženýrské jednotky/sekundu (REAL).

Parametr C_Dir označuje aktuální směr pohybu.

Tabulka 9-2 Parametry instrukce POSx_CTRL

Vstupy/Výstupy	Typ dat	Operandy
MOD_EN	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Done, C_Dir	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD



Tip

Polohovací modul načítá tabulku konfigurace profilů pouze při studeném startu nebo když dostane příkaz nahrát konfiguraci.

- Pokud použijete pro modifikaci konfigurace "Průvodce polohováním", dá instrukce POSx_CTRL automaticky příkaz polohovacímu modulu, aby nahrál tabulku konfigurace profilů pokaždé, když CPU S7-200 přejde do režimu RUN.
- Pokud pro modifikaci konfigurace použijete "Control panel pro EM 253", kliknutí na tlačítko "Aktualizace konfigurace" přikáže polohovacímu modulu, aby nahrál novou tabulku konfigurace profilů.
- Jestliže pro modifikaci konfigurace použijete jiný způsob, musíte dát polohovacímu modulu také příkaz "Znovu nahrát konfiguraci", aby nahrál tabulku konfigurace profilů. V opačném případě polohovací modul nadále používá starou tabulku konfigurace profilů.

Instrukce POSx_MAN

Instrukce POSx_MAN (Manuální režim) uvede polohovací modul do manuálního režimu, což umožňuje použít různou rychlost motoru nebo jej uvést do pomalého chodu kladným nebo záporným směrem. V době, kdy je povolena instrukce POSx_MAN, jsou přípustné pouze instrukce POSx_CTRL a POSx_DIS.

Naráz můžete povolit pouze jeden ze vstupů RUN, JOG_P nebo JOG_N.

Povolení parametru RUN (Chod/Zastavení) dá polohovacímu modulu příkaz zrychlit na specifikovanou rychlost (parametr "Speed") a směr (parametr "Dir"). V době chodu motoru můžete měnit hodnotu parametru "Speed", ale parametr "Dir" musí zůstat konstantní. Blokování parametru RUN dá polohovacímu modulu příkaz zpomalit až do zastavení motoru.

Povolení parametru JOG_P (Manuální chod kladným směrem) nebo parametru JOG_N (Manuální chod záporným směrem) dá polohovacímu modulu příkaz manuálního chodu v kladném nebo záporném směru. Pokud parametr JOG_P nebo JOG_N zůstane povolen na dobu kratší než 0,5 sekundy, generuje polohovací modul pulzy pro posun o vzdálenost specifikovanou v JOG_INCREMENT. Pokud parametr JOG_P nebo JOG_N zůstane povolen po dobu 0,5 sekundy nebo déle, začne polohovací modul zrychlovat na specifikované JOG_SPEED.

Parametr "Speed" (Rychlost) určuje rychlost, když je povolen RUN. Pokud jste nastavili systém odměřování polohovacího modulu na pulzy, má rychlost pro pulzy/sekundu hodnotu DINT. Pokud jste nastavili systém odměřování polohovacího modulu na inženýrské jednotky, má rychlost pro jednotky/sekundu hodnotu REAL. Tento parametr můžete měnit v době chodu motoru.

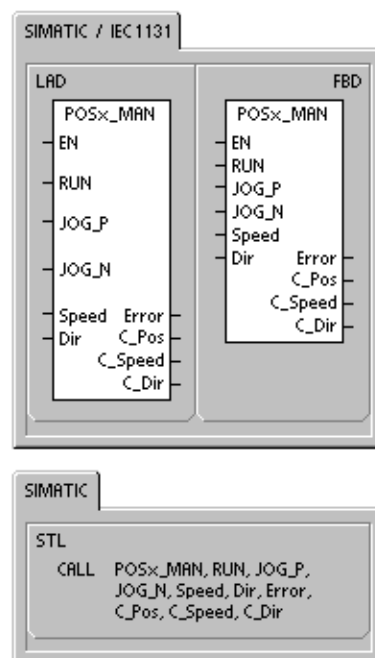
Parametr "Dir" (Směr) určuje směr pohybu, když je povolen RUN. Tuto hodnotu nemůžete měnit, pokud je povolen parametr RUN.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Parametr C_Pos obsahuje aktuální polohu modulu. V závislosti na vybraných jednotkách odměřování je tato hodnota buď počet pulzů (DINT), nebo počet inženýrských jednotek (REAL).

Parametr C_Speed obsahuje aktuální rychlost modulu. V závislosti na vybraných jednotkách odměřování je tato hodnota buď počet pulzů/sekundu (DINT), nebo počet inženýrských jednotek/sekundu (REAL).

Parametr C_Dir označuje aktuální směr motoru.



Tabulka 9-3 Parametry instrukce POSx_MAN

Vstupy/Výstupy	Typ dat	Operandy
RUN, JOG_P, JOG_N	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, konstanta
Dir, C_Dir	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Instrukce POSx_GOTO

Instrukce POSx_GOTO přikazuje polohovacímu modulu přejít do požadované polohy.

Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit DONE signalizovat, že je instrukce ukončena.

Zapnutí parametru START vyšle příkaz GOTO polohovacímu modulu. Při každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul není právě v činnosti, vyšle instrukce polohovacímu modulu příkaz GOTO. Aby bylo jisté, že je vyslán pouze jediný příkaz GOTO, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr "Pos" obsahuje hodnotu, která představuje buď cíl pohybu (pro absolutní pohyb), nebo délku pohybu (pro relativní pohyb). V závislosti na vybraných jednotkách měření je tato hodnota buď počet pulzů (DINT), nebo inženýrských jednotek (REAL).

Parametr "Speed" určuje maximální rychlost tohoto pohybu. V závislosti na jednotkách odměřování je tato hodnota buď počet pulzů/sekundu (DINT) nebo inženýrských jednotek/sekundu (REAL).

Parametr "Mode" volí typ pohybu:

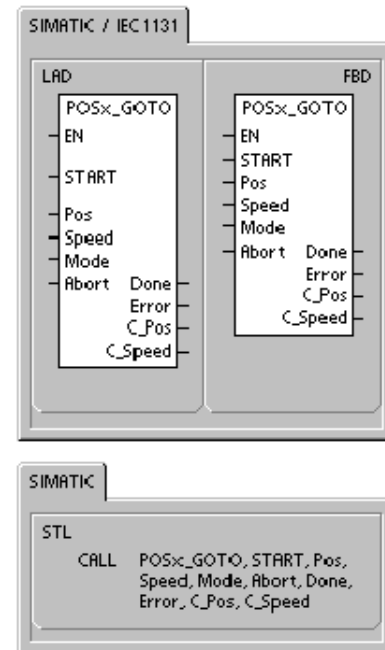
- 0 – Absolutní poloha
- 1 – Relativní poloha
- 2 – Kontinuální kladný pohyb s jednou rychlostí
- 3 – Kontinuální záporný pohyb s jednou rychlostí

Parametr "Done" (Dokončeno) se zapne ve chvíli, kdy polohovací modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Parametr C_Pos obsahuje aktuální polohu modulu. V závislosti na jednotkách měření je tato hodnota buď počet pulzů (DINT), nebo počet inženýrských jednotek (REAL).

Parametr C_Speed obsahuje aktuální rychlost modulu. V závislosti na jednotkách měření je tato hodnota buď počet pulzů/sekundu (DINT), nebo počet inženýrských jednotek/sekundu (REAL).



Tabulka 9-4 Parametry instrukce POSx_GOTO

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Pos, Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, konstanta
Mode	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, konstanta
Abort, Done	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Instrukce POSx_RUN

Instrukce POSx_RUN (Spuštění profilu) přikazuje polohovacímu modulu, aby provedl pohyb podle konkrétního profilu uloženého v tabulce konfigurace profilu.

Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je instrukce ukončena.

Zapnutí parametru START vyše příkaz RUN polohovacímu modulu. Při každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul právě není v činnosti, vyše instrukce polohovacímu modulu příkaz RUN. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr "Profile" (Profil) obsahuje číslo symbolického názvu profilu pohybu. Můžete také vybrat pohybové příkazy pro pokročilé (118 až 127). Informace o pohybových příkazech najdete v tabulce 9-19 na straně 284.

Zapnutí parametru "Abort" (Zrušení) dá polohovacímu modulu příkaz, aby zastavil aktuální profil a zpomaloval motor až do zastavení.

Parametr "Done" (Dokončeno) se zapne ve chvíli, kdy modul dokončí tuto instrukci.

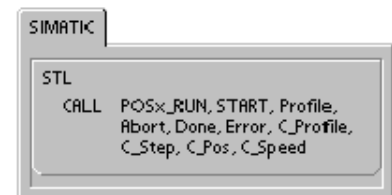
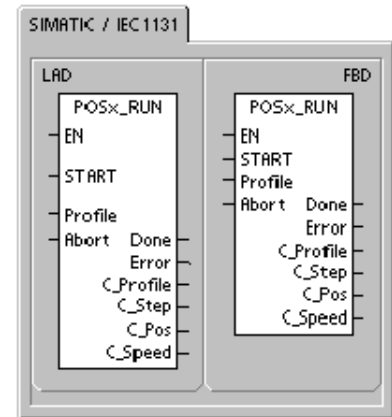
Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Parametr C_Profile obsahuje profil, který v současné době polohovací modul provádí.

Parametr C_Step obsahuje krok profilu, který se právě provádí.

Parametr C_Pos obsahuje aktuální polohu modulu. V závislosti na jednotkách odměřování je tato hodnota buď počet pulzů (DINT), nebo počet inženýrských jednotek (REAL).

Parametr C_Speed obsahuje aktuální rychlost modulu. V závislosti na jednotkách odměřování je tato hodnota buď počet pulzů/sekundu (DINT), nebo počet inženýrských jednotek/sekundu (REAL).



Tabulka 9-5 Parametry instrukce POSx_RUN

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Profile	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, konstanta
Abort, Done	BOOL	I, Q, V, M, SM, S, T, C, L,
Error, C_Profile, C_Step	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Instrukce POSx_RSEEK

Instrukce POSx_RSEEK (Vyhledání referenčního bodu) iniciuje operaci hledání referenčního bodu s použitím způsobu hledání z tabulky konfigurace profilů. Když polohovací modul lokalizuje referenční bod a pohyb se zastaví, nahraje polohovací modul hodnotu parametru RP_OFFSET na aktuální polohu a generuje 50milisekundový pulz na výstupu CLR.

Implicitní hodnota RP_OFFSET je 0. Můžete použít "Průvodce polohováním", "Control panel pro EM253" nebo instrukci POSx_LDOFF (Nahrát offset), abyste změnili hodnotu RP_OFFSET.

Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je instrukce ukončena.

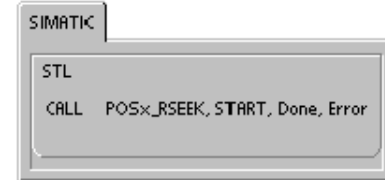
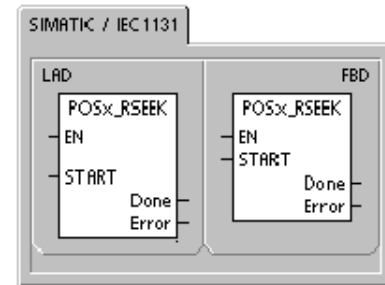
Zapnutí parametru START vyše příkaz RSEEK polohovacímu modulu. V každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul není právě v činnosti, vyše instrukce polohovacímu modulu příkaz RSEEK. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr "Done" (Dokončeno) se zapne, když modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Tabulka 9-6 Parametry instrukce POSx_RSEEK

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Done	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Instrukce POSx_LDOFF

Instrukce POSx_LDOFF (Nahrát offset referenčního bodu) stanovuje novou nulovou polohu, která leží na jiném místě, než je poloha referenčního bodu.

Před provedením této instrukce musíte nejprve určit polohu referenčního bodu. Musíte také stroj uvést do počáteční polohy. Když instrukce vyšle příkaz LDOFF, vypočítá polohovací modul offset mezi počáteční polohou (aktuální poloha) a polohou referenčního bodu. Pak polohovací modul uloží vypočítaný offset do parametru RP_OFFSET a nastaví aktuální polohu na 0. Tím je počáteční poloha určena jako nulová poloha.

V případě, že motor ztratí přehled o své poloze (například při ztrátě napájení nebo když je poloha motoru změněna ručně), můžete použít pro automatické opětovné stanovení nulové polohy instrukci POSx_RSEEK.

Zapnutí výstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je provádění instrukce ukončeno.

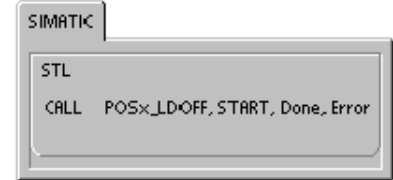
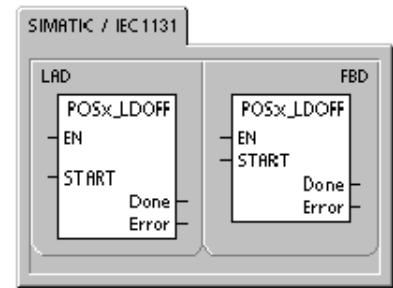
Zapnutí parametru START vyšle příkaz LDOFF polohovacímu modulu. V každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul právě není v činnosti, vyšle instrukce polohovacímu modulu příkaz LDOFF. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr "Done" (Dokončeno) se zapne, když modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Tabulka 9-7 Parametry instrukce POSx_LDOFF

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Done	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Instrukce POSx_LDPOS

Instrukce POSx_LDPOS (Nahrát polohu) mění hodnotu aktuální polohy v polohovacím modulu na novou. Tuto instrukci můžete také použít pro stanovení nové nulové polohy pro kterýkoliv příkaz absolutního pohybu.

Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je instrukce ukončena.

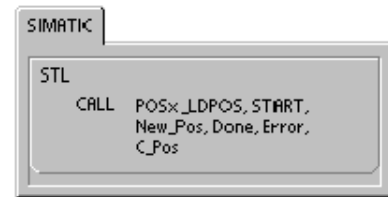
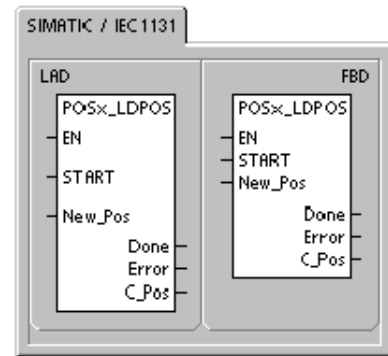
Zapnutí parametru START vyše příkaz LDPOS polohovacímu modulu. V každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul není právě v činnosti, vyše instrukce polohovacímu modulu příkaz LDPOS. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr New_Pos dává novou hodnotu, která nahrazuje hodnotu aktuální polohy a kterou polohovací modul hlásí a používá pro absolutní pohyby. V závislosti na jednotkách odměřování je tato hodnota buď počet pulzů (DINT), nebo inženýrských jednotek (REAL).

Parametr "Done" (Dokončeno) se zapne ve chvíli, kdy modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Parametr C_Pos obsahuje aktuální polohu modulu. V závislosti na jednotkách měření je tato hodnota buď počet pulzů (DINT), nebo počet inženýrských jednotek (REAL).



Tabulka 9-8 Parametry instrukce POSx_LDPOS

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
New_Pos, C_Pos	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD
Done	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Instrukce POSx_SRATE

Instrukce POSx_SRATE (Nastavit rychlost) dává polohovacímu modulu příkaz změnit dobu zrychlení, zpomalení a zaoblení.

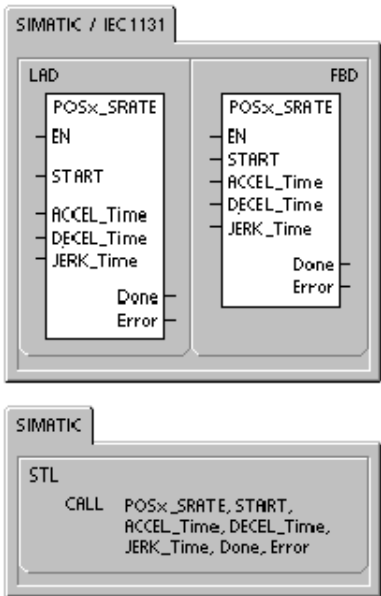
Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je provádění instrukce ukončeno.

Zapnutí parametru START zkopíruje nové časové hodnoty do tabulky konfigurace profilů a vyšle příkaz SRATE polohovacímu modulu. V každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul právě není v činnosti, vyšle instrukce polohovacímu modulu příkaz SRATE. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametry ACCEL_Time, DECEL_Time a JERK_Time určují novou dobu zrychlení, dobu zpomalení a dobu zaoblení v milisekundách (ms).

Parametr "Done" (Dokončeno) se zapne, když modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.



Tabulka 9-9 Parametry instrukce POSx_SRATE

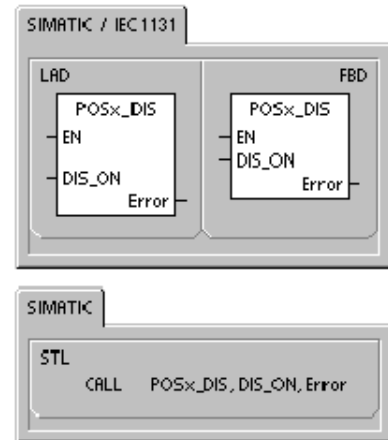
Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L,
ACCEL_Time, DECEL_Time, JERK_Time	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, konstanta
Done	BOOL	I, Q, V, M, SM, S, T, C, L,
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Instrukce POSx_DIS

Instrukce POSx_DIS zapíná nebo vypíná výstup DIS polohovacího modulu. To umožňuje použít výstup DIS pro zablokování nebo povolení řídicí jednotky motoru. Pokud použijete výstup DIS na polohovacím modulu, je možné volat tuto instrukci v každém programovém cyklu nebo pouze tehdy, je-li třeba změnit hodnotu výstupu DIS.

Když se zapne bit EN, aby byla instrukce povolena, řídí parametr DIS_ON výstup DIS polohovacího modulu. Více informací o výstupu DIS najdete v tabulce 9-1 na straně 244 nebo nahlédněte do specifikace polohovacího modulu v příloze A.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.



Tabulka 9-10 Parametry instrukce POSx_DIS

Vstupy/Výstupy	Typ dat	Operandy
DIS_ON	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, konstanta
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Instrukce POSx_CLR

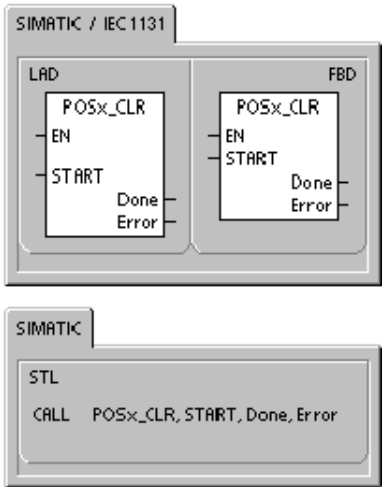
Instrukce POSx_CLR (Pulz na výstupu CLR) dá polohovacímu modulu příkaz, aby na výstupu CLR generoval 50ms pulz.

Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je provádění instrukce ukončeno.

Zapnutí parametru START vyšle příkaz CLR polohovacímu modulu. Při každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul právě není v činnosti, vyšle instrukce polohovacímu modulu příkaz CLR. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr "Done" (Dokončeno) se zapne ve chvíli, kdy modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.



Tabulka 9-11 Parametry instrukce POSx_CLR

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Instrukce POSx_CFG

Instrukce POSx_CFG (Znovu nahrát konfiguraci) dá polohovacímu modulu příkaz, aby načel konfiguraci z místa specifikovaného ukazatelem tabulky konfigurace profilů. Pak polohovací modul porovná novou konfiguraci se stávající a provede všechny potřebné změny nastavení nebo nové výpočty.

Zapnutí vstupu EN instrukci povoluje. Zajistěte, aby bit EN zůstal zapnutý, dokud nebude bit "Done" signalizovat, že je provádění instrukce ukončeno.

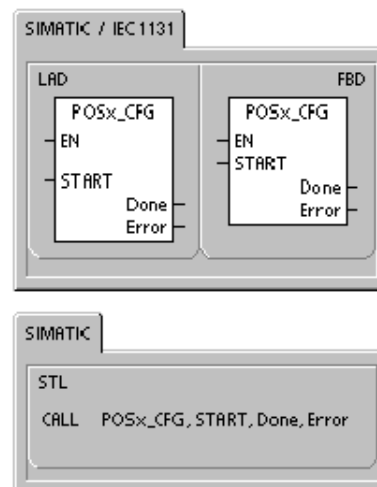
Zapnutí parametru START vyše příkaz CFG polohovacímu modulu. V každém programovém cyklu, kdy je parametr START zapnutý a polohovací modul právě není v činnosti, vyše instrukce polohovacímu modulu příkaz CFG. Aby bylo jisté, že je vyslán pouze jediný příkaz, použijte pro vyslání pulzu k zapnutí parametru START detekci hrany.

Parametr "Done" (Dokončeno) se zapne ve chvíli, kdy modul dokončí tuto instrukci.

Parametr "Error" (Chyba) obsahuje výsledek této instrukce. Chybové kódy jsou definovány v tabulce 9-13 na straně 276.

Tabulka 9-12 Parametry instrukce POSx_CFG

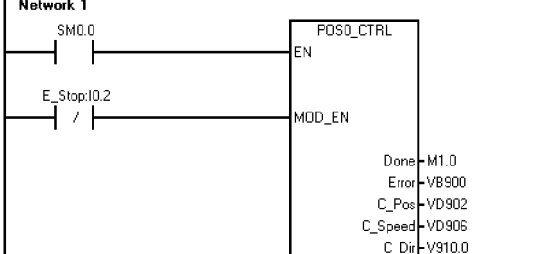
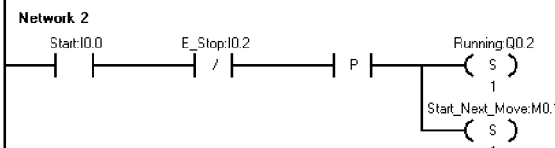
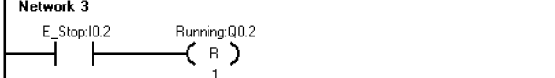
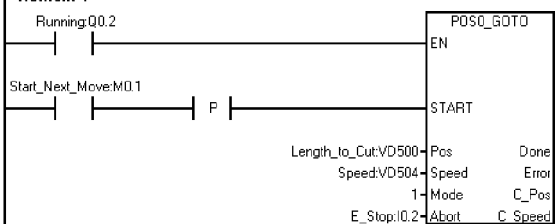
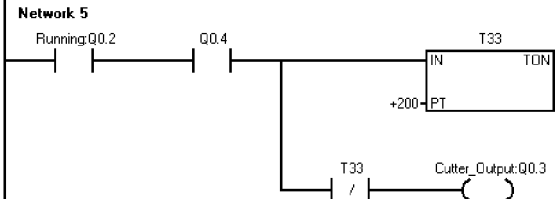
Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, V, M, SM, S, T, C, L, signálový tok
Done	BOOL	I, Q, V, M, SM, S, T, C, L
Error	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD



Vzorové programy

První vzorový program ukazuje jednoduchý relativní pohyb, který používá instrukce POSx_CTRL a POSx_GOTO pro provedení operace řezání materiálu o konstantní délce. Tento program nepotřebuje režim hledání RP nebo profil pohybu a délku lze měřit buď na pulzy, nebo na inženýrské jednotky. Vložte délku (VD500) a cílovou rychlost (VD504). Když se zapne I0.0 (Start), začne stroj pracovat. Když se zapne I0.1 (Stop), stroj dokončí operaci a zastaví se. Když se zapne I0.2 (E_Stop), zruší stroj jakýkoliv pohyb a okamžitě se zastaví.

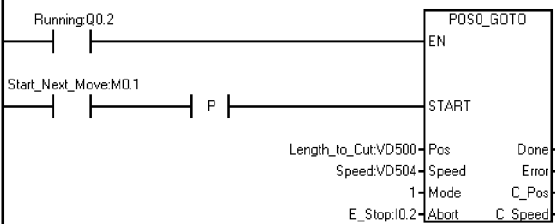
Ve druhém vzorovém programu jsou použity instrukce POSx_CTRL, POSx_RUN, POSx_RSEEK a POSx_MAN. Musíte nakonfigurovat režim hledání RP a profil pohybu.

Vzorový program 1: Jednoduchý relativní pohyb (použití pro řezání materiálu)	
Network 1 	Network 1 //Řídící instrukce (modul v místě 0). <pre>LD SM0.0 = L60,0 LDN I0.2 = L63,7 LD L60.0 CALL POS0_CTRL, L63.7, M1.0, VB900, VD902, VD906, V910.0</pre>
Network 2 	Network 2 //Spuštění uvede stroj do automatického režimu <pre>LD I0.0 AN I0.2 EU S Q0.2, 1 S M0.1, 1</pre>
Network 3 	Network 3 //E_Stop: provede okamžité zastavení a vypne automatický režim. <pre>LD I0.2 R Q0.2, 1</pre>
Network 4 	Network 4 //Pohyb do určitého bodu: //Vložíte délku řezu. //Vložíte cílovou rychlost do "Speed". //Nastavte režim na 1 (Relativní režim). <pre>LD Q0.2 = L60,0 LD M0.1 EU = L63,7 LD L60.0 CALL POS0_GOTO, L63.7, VD500, VD504, 1, I0.2, Q0.4, VB920, VD922, VD926</pre>
Network 5 	Network 5 //Dosáhne-li polohy, zapne řezací stroj //na 2 sekundy pro dokončení řezu. <pre>LD Q0.2 A Q0.4 TON T33, +200 AN T33 = Q0,3</pre>

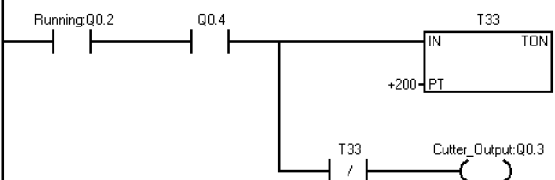
Network 3



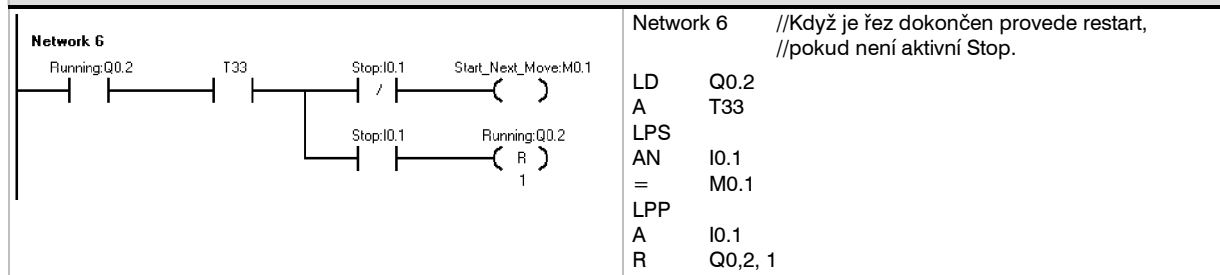
Network 4



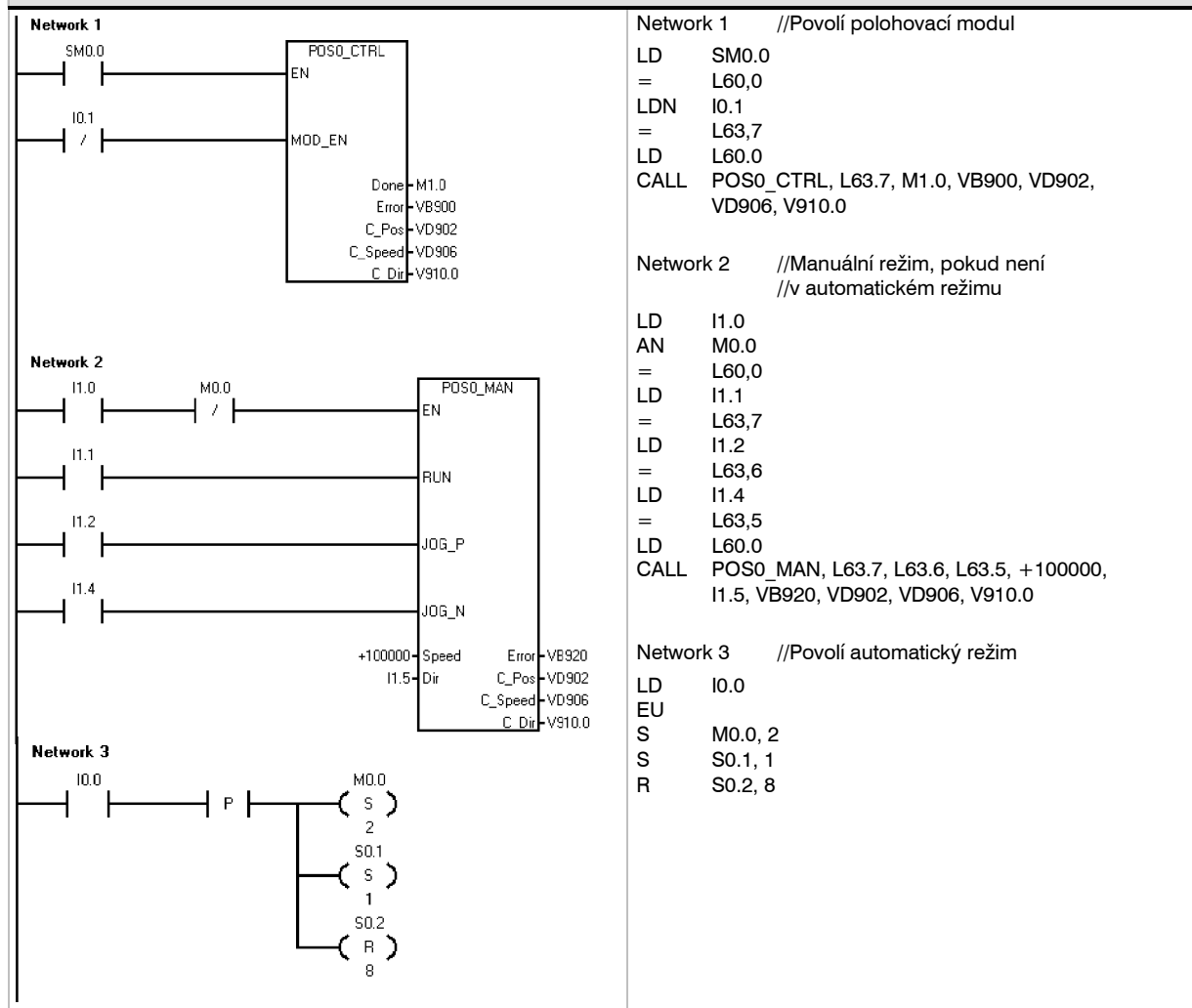
Network 5

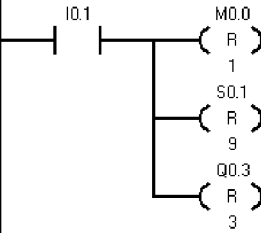
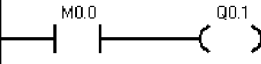
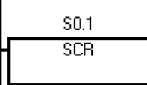
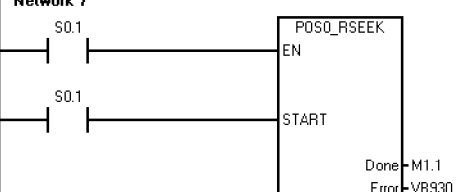
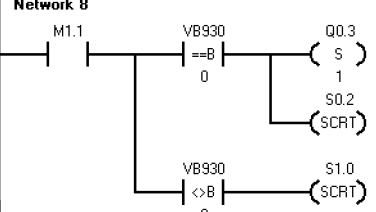
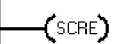
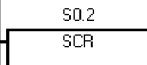


Vzorový program 1: Jednoduchý relativní pohyb (použití pro řezání materiálu) , pokračování



Vzorový program 2: Program s instrukcemi POSx_CTRL, POSx_RUN, POSx_SEEK a POSx_MAN



Vzorový program 2: Program s instrukcemi POSx_CTRL, POSx_RUN, POSx_SEEK a POSx_MAN, pokračování	
Network 4  Network 5  Network 6  Network 7  Network 8  Network 9  Network 10 	Network 4 //Nouzové zastavení //Zablokuje modul a automatický režim <pre>LD I0.1 R M0.0, 1 R S0.1, 9 R Q0.3, 3</pre> Network 5 //Když je v automatickém režimu: //Rozsvítí signálku pro signalizaci chodu <pre>LD M0.0 = Q0,1</pre> Network 6 <pre>LSCR S0.1</pre> Network 7 //Vyhledá referenční bod (RP) <pre>LD S0.1 = L60,0 LD S0.1 = L63,7 LD L60.0 CALL POS0_RSEEK, L63.7, M1.1, VB930</pre> Network 8 //Když je dosaženo referenčního bodu (RP): //Upne materiál a //přejde na další krok. <pre>LD M1.1 LPS AB= VB930, 0 S Q0,3, 1 SCRT S0.2 LPP AB<> VB930, 0 SCRT S1.0</pre> Network 9 <pre>LSCR S0.2</pre> Network 10 <pre>LSCR S0.2</pre>

Network 11

S0.2

S0.2

POS0_RUN

EN

START

VB228 Profile Done M1.2

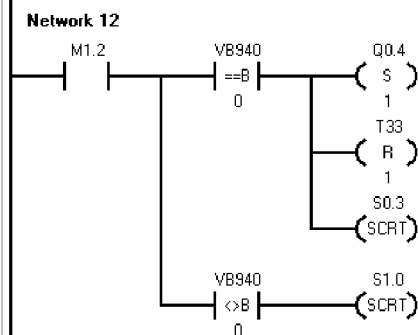
I0.1 Abort Error VB940

C_Profile VB941

C_Step VB942

C_Pos VD944

C_Speed VD948



—(SCRE)

S0.3
SCR

Diagram 1: A ladder logic network with two parallel branches. The top branch contains a normally open contact labeled S0.3 in series with a coil labeled T33. The bottom branch contains a normally open contact labeled PT in series with a coil labeled TON. Both branches are connected to a common return line labeled +200V.

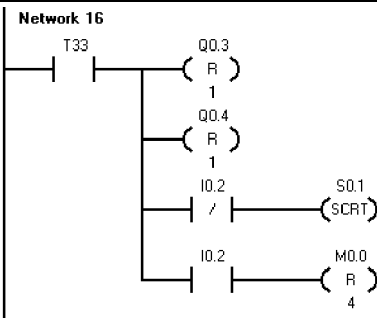
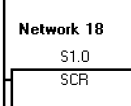
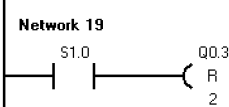
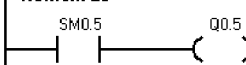
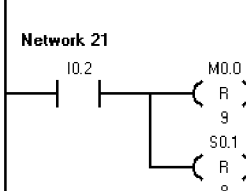
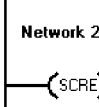
```
LD      S0.2
=       L60,0
LD      S0.2
=       L63,7
LD      L60.0
CALL    POS0_RUN, L63.7, VB228, IO.1, M1.2, VB940,
        VB941, VB942, VD944, VD948
```

LD	M1.2
LPS	
AB=	VB940, 0
S	Q0,4, 1
R	T33, 1
SCRT	S0.3
LPP	
AB<>	VB940, 0
SCRT	S1.0

SCRE

LSCR S0.3

LD	S0.3
TON	T33, +200

Vzorový program 2: Program s instrukcemi POSx_CTRL, POSx_RUN, POSx_SEEK a POSx_MAN, pokračování	
Network 16  Network 17  Network 18  Network 19  Network 20  Network 21  Network 22 	Network 16 //Pokud není zapnutý STOP, provede po //dokončení řezu restart. <pre>LD T33 LPS R Q0.3, 1 R Q0.4, 1 AN I0.2 SCRT S0.1 LPP A I0.2 R M0.0, 4</pre> Network 17 <pre>SCRE</pre> Network 18 <pre>LSCR S1.0</pre> Network 19 //Resetuje výstupy. <pre>LD S1.0 R Q0.3, 2</pre> Network 20 //Bliká signálkou chyby. <pre>LD SM0.5 = Q0.5</pre> Network 21 //Ukončí chybový podprogram, //je-li zapnutý STOP. <pre>LD I0.2 R M0.0, 9 R S0.1, 8</pre> Network 22 <pre>SCRE</pre>

Monitorování polohovacího modulu "Control panelem pro EM 253"

Jako pomoc při vypracování vašeho řešení aplikace řízení polohy nabízí STEP 7 – Micro/WIN "Control panel pro EM 253". Díky záložkám "Ovládání", "Konfigurace" a "Diagnostika" je snadné monitorovat a řídit práci polohovacího modulu během spuštění a zkušebních fází vývojového procesu.

"Control panel pro EM 253" použijte pro ověření toho, že je polohovací modul správně zapojen, pro úpravu konfiguračních dat a pro odzkoušení všech pohybových profilů.

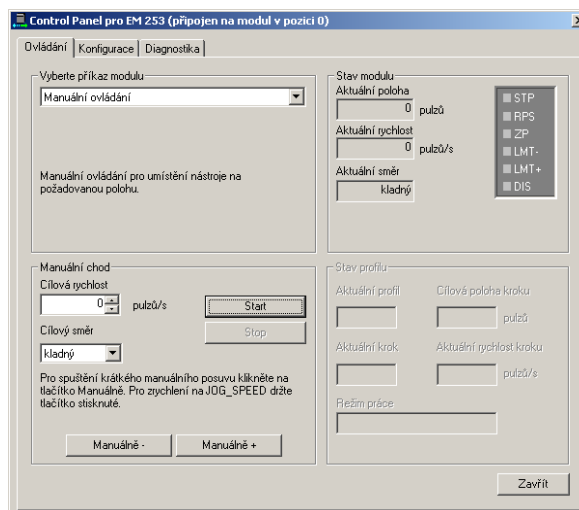
Zobrazení stavu a řízení polohovacího modulu

Záložka "Ovládání" na "Control panelu pro EM 253" umožňuje přímo ovládat polohovací modul. "Control panel pro EM 253" ukazuje aktuální rychlost, polohu a směr polohovacího modulu. Můžete také sledovat stav vstupních a výstupních LED diod (s výjimkou pulzních LED).

"Control panel pro EM 253" umožňuje interakci s polohovacím modulem tím, že měníte jeho rychlost a směr, zastavujete a spouštíte pohyb nebo manuálně posouváte nástroj (pokud je pohyb zastaven).

Můžete také generovat následující pohybové příkazy:

- ☐ "Manuální ovládání". Tento příkaz umožňuje použít pro polohování nástroje ruční ovládací prvky.
- ☐ "Spustit profil pohybu". Tento příkaz umožňuje vybrat profil, který bude prováděn. "Control panel pro EM 253" zobrazuje stav profilu, který polohovací modul právě provádí.
- ☐ "Vyhledat referenční bod". Tento příkaz najde referenční bod pomocí nakonfigurovaného způsobu hledání.
- ☐ "Načíst offset referenčního bodu". Poté, co jste použili ruční ovládací prvky pro offset nástroje do nové nulové polohy, nahrajte offset referenčního bodu.
- ☐ "Znovu nahrát aktuální polohu". Tento příkaz aktualizuje polohu a stanoví novou nulovou polohu.
- ☐ "Aktivovat výstup DIS" a "Deaktivovat výstup DIS". Tyto příkazy zapínají nebo vypínají výstup DIS polohovacího modulu.
- ☐ "Pulz na výstupu CLR". Tento příkaz generuje 50ms pulz na výstupu CLR polohovacího modulu.
- ☐ "Naučit profil pohybu". Tento příkaz umožňuje uložit cílovou polohu a rychlost profilu pohybu a kroku, když ručně polohujete nástroj. "Control panel pro EM 253" zobrazuje stav profilu, který polohovací modul právě provádí.
- ☐ "Nahrát konfiguraci modulu". Tento příkaz nahraje novou konfiguraci tak, že přikáže polohovacímu modulu načíst konfigurační blok z paměti V automatu S7-200.
- ☐ "Posun na absolutní polohu". Tento příkaz umožňuje posun na zadanou polohu při cílové rychlosti. Dříve než použijete tento příkaz, musíte mít stanovenou nulovou polohu.
- ☐ "Posun o relativní vzdálenost". Tento příkaz umožňuje posun o zadanou vzdálenost z aktuální polohy při cílové rychlosti. Můžete vložit kladnou i zápornou vzdálenost.
- ☐ "Resetovat příkazy modulu". Tento příkaz vymaže příkazový byte polohovacího modulu a nastaví bit "Done" (Dokončeno). Použijte tento příkaz, pokud se zdá, že polohovací modul nereaguje na příkazy.

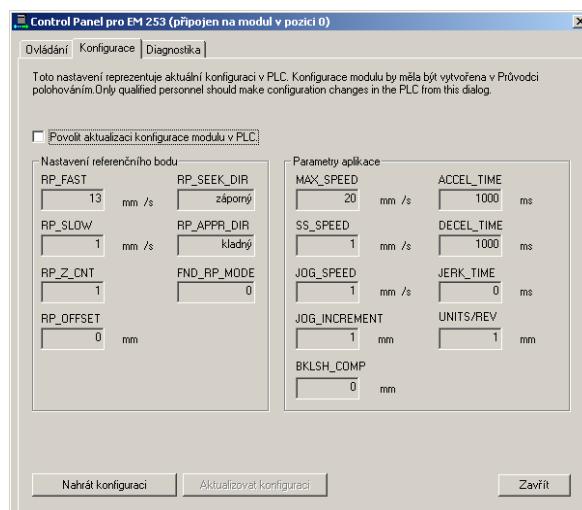


Obr. 9-19 Záložka "Ovládání" "Control panelu pro EM 253"

Zobrazení a úprava konfigurace polohovacího modulu

Záložka "Konfigurace" na "Control panelu pro EM 253" umožňuje prohlížet a modifikovat nastavení konfigurace polohovacího modulu, která jsou uložena v datovém bloku automatu S7-200.

Po změně nastavení konfigurace jednoduše klikněte na tlačítko, čímž aktualizujete jak nastavení v projektu STEP 7 – Micro/Win, tak i v datovém bloku S7-200.



Obr. 9-20 Záložka "Konfigurace" "Control panelu pro EM 253"

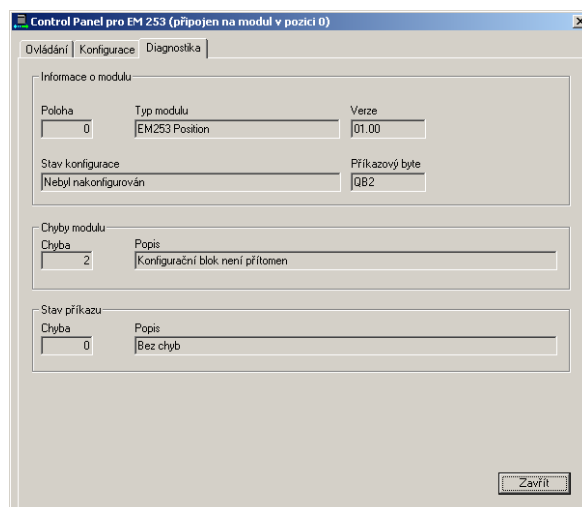
Zobrazení diagnostických informací polohovacího modulu

Záložka "Diagnostika" "Control panelu pro EM 253" umožňuje prohlížet diagnostické informace, týkající se polohovacího modulu.

Je možné prohlížet konkrétní informace o polohovacím modulu, jako je umístění modulu, typ modulu a verze firmware i výstupní byte, používaný jako příkazový byte pro modul.

"Control panel pro EM 253" zobrazí jakýkoli chybový stav, který vyplývá z příkázané operace. Chybové stavy pro jednotlivé instrukce najdete v tabulce 9-13 na straně 276.

Můžete také prohlížet jakýkoliv chybový stav, hlášený polohovacím modulem. Chybové stavy modulu najdete v tabulce 9-14 na straně 277.



Obr. 9-21 Záložka "Diagnostika" "Control panelu pro EM 253"

Chybové kódy pro polohovací modul a polohovací instrukce

Tabulka 9-13 Chybové kódy instrukcí

Chybové kódy	Popis
0	Bez chyby
1	Zrušeno uživatelem
2	Chyba konfigurace Pro prohlížení chybových kódů použijte záložku "Diagnostika" na "Control panelu pro EM 253"
3	Neplatný příkaz
4	Zrušeno pro neexistenci platné konfigurace Pro prohlížení chybových kódů použijte záložku "Diagnostika" na "Control panelu pro EM 253"
5	Zrušeno vzhledem k nepřítomnosti napájení ze strany uživatele
6	Zrušeno pro neexistenci definovaného referenčního bodu
7	Zrušeno kvůli aktivaci vstupu STP
8	Zrušeno kvůli aktivaci vstupu LMT –
9	Zrušeno kvůli aktivaci vstupu LMT +
10	Zrušeno kvůli problému s prováděním pohybu
11	Pro zadaný profil není nakonfigurován profilový blok
12	Nepovolený pracovní režim
13	Pracovní režim není pro tento příkaz podporován
14	Nepovolený počet kroků v profilovém bloku
15	Nepovolená změna směru
16	Nepovolená vzdálenost
17	Ke spuštění RPS došlo před dosažením cílové rychlosti
18	Nedostatečná šířka aktivní oblasti RPS
19	Rychlost mimo rozsah
20	Nedostatečná vzdálenost pro provedení požadované změny rychlosti
21	Nepovolená poloha
22	Není známa nulová poloha
23 až 127	Rezervováno
128	Polohovací modul nemůže zpracovat tuto instrukci: polohovací modul zpracovává jinou instrukci, nebo pro tuto instrukci nebyl pulz "Start"
129	Chyba polohovacího modulu: ■ Umístění polohovacího modulu nebo adresa v paměti Q, která byla nakonfigurována pomocí "Průvodce polohováním", neodpovídá skutečnému umístění nebo adrese v paměti ■ Ostatní chybové stavy viz SMB8 až SMB21 (ID vstupního/výstupního modulu a chybový registr)
130	
131	Polohovací modul není dostupný kvůli chybě modulu nebo není povolen (Viz stav POSx_CTRL)

Tabulka 9-14 Chybové kódy modulu

Chybové kódy	Popis
0	Bez chyby
1	Není napájení ze strany uživatele
2	Není přítomen konfigurační blok
3	Chyba ukazatele konfiguračního bloku
4	Velikost konfiguračního bloku větší než dostupná paměť V
5	Nepovolený formát konfiguračního bloku
6	Zadáno příliš mnoho profilů
7	Nepovolená specifikace STP_RSP
8	Nepovolená specifikace LMT- _RPS
9	Nepovolená specifikace LMT+ _RPS
10	Nepovolená specifikace FILTER_TIME
11	Nepovolená specifikace MEAS_SYS
12	Nepovolená specifikace RP_CFG
13	Nepovolená hodnota PLS/REV
14	Nepovolená hodnota UNITS/REV
15	Nepovolená hodnota RP_ZP_CNT
16	Nepovolená hodnota JOG_INCREMENT
17	Nepovolená hodnota MAX_SPEED
18	Nepovolená hodnota SS_SPD
19	Nepovolená hodnota RP_FAST
20	Nepovolená hodnota RP_SLOW
21	Nepovolená hodnota JOG_SPEED
22	Nepovolená hodnota ACCEL_TIME
23	Nepovolená hodnota DECEL_TIME
24	Nepovolená hodnota JERK_TIME
25	Nepovolená hodnota BKLSH_COMP

Témata pro pokročilé

Vysvětlení tabulky konfigurace profilů

"Průvodce polohováním" byl vyvinut, aby usnadnil polohovací aplikace tím, že automaticky generuje konfigurační a profilové hodnoty na základě odpovědí, které o svém systému dáváte řízení polohy. Informace o tabulce konfigurace profilů jsou pro pokročilé uživatele, kteří chtějí vytvořit své vlastní podprogramy pro řízení polohy.

Tabulka konfigurace profilů je umístěna v oblasti paměti V automatu S7-200. Jak ukazuje tabulka 9-15, hodnoty jsou uloženy v následujících informacích:

- ☐ Konfigurační blok obsahuje informace použité pro nastavení modulu při přípravě na provádění pohybových příkazů.
- ☐ Interaktivní blok podporuje přímé nastavení pohybových parametrů uživatelským programem.
- ☐ Každý profilový blok popisuje předdefinovanou pohybovou operaci, kterou má provádět polohovací modul. Můžete nakonfigurovat až 25 profilových bloků.



Tip

Chcete-li vytvořit více než 25 pohybových profilů, můžete vyměnit tabulky konfigurace profilů změnou hodnoty uložené v ukazateli na tabulku konfigurace profilu.

Tabulka 9-15 Tabulka konfigurace profilu

Offset	Název	Popis funkce	Typ								
Konfigurační blok											
0	MOD_ID	Identifikátor modulu	--								
5	CB_LEN	Délka konfiguračního bloku v bytech (1 byte)	--								
6	IB_LEN	Délka interaktivního bloku v bytech (1 byte)	--								
7	PF_LEN	Délka jednoho profilu v bytech (1 byte)	--								
8	STP_LEN	Délka jednoho kroku v bytech (1 byte)	--								
9	STEPS	Počet kroků na profil (1 byte)	--								
10	PROFILES	Počet profilů od 0 do 25 (1 byte)	--								
11	Rezervováno	Nastavit na 0x0000	--								
13	IN_OUT_CFG	Specifikuje použití vstupů a výstupů modulu (1 byte) MSB 7 6 5 4 3 2 1 LSB 0 <table><tr><td>P/D</td><td>POL</td><td>0</td><td>0</td><td>STP</td><td>RPS</td><td>LMT-</td><td>LMT+</td></tr></table> P/D Tento bit specifikuje použití P0 a P1. Kladná polarita (POL=0): 0 – Pulzy P0 pro kladný pohyb Pulzy P1 pro záporný pohyb 1 – Pulzy P0 pro pohyb P1 řídí směr pohybu (0 – kladné, 1 – záporné) Záporná polarita (POL=1): 0 – Pulzy P0 pro kladný pohyb Pulzy P1 pro záporný pohyb 1 – Pulzy P0 pro pohyb P1 řídí směr pohybu (0 – kladné, 1 – záporné) POL Tento bit volí polaritu pro P0 a P1. (0 – kladná polarita, 1 – záporná polarita) STP Tento bit řídí aktivní úroveň vstupu stop. RPS Tento bit řídí aktivní úroveň vstupu RPS. LMT- Tento bit řídí aktivní úroveň vstupu meze záporného posunu. LMT+ Tento bit řídí aktivní úroveň vstupu meze kladného posunu. 0 – Aktivní vysoká úroveň 1 – Aktivní nízká úroveň	P/D	POL	0	0	STP	RPS	LMT-	LMT+	--
P/D	POL	0	0	STP	RPS	LMT-	LMT+				

Tabulka 9-15 Tabulka konfigurace profilu, pokračování

Offset	Název	Popis funkce	Typ																				
14	STP_RSP	Specifikuje odezvu pohonu na vstup STP (1 byte) 0 Nereagovat. Ignorovat stav na vstupu. 1 Zastavit se zpomalením a indikovat aktivní stav vstupu STP. 2 Ukončit pulzy a indikovat vstup STP 3 až 255 Rezervováno (zadání této hodnoty znamená chybu)	--																				
15	LMT- _RSP	Specifikuje odezvu pohonu na vstup záporné meze (1 byte) 0 Nereagovat. Ignorovat stav na vstupu. 1 Zastavit se zpomalením a indikovat, že bylo dosaženo meze. 2 Ukončit pulzy a indikovat, že bylo dosaženo meze. 3 až 255 Rezervováno (zadání této hodnoty znamená chybu)	--																				
16	LMT+ _RSP	Specifikuje odezvu pohonu na vstup kladné meze (1 byte) 0 Nereagovat. Ignorovat stav na vstupu. 1 Zastavit se zpomalením a indikovat, že bylo dosaženo meze. 2 Ukončit pulzy a indikovat, že bylo dosaženo meze. 3 až 255 Rezervováno (zadání této hodnoty znamená chybu)	--																				
17	FILTER_TIME	Specifikuje čas filtrů použitých pro vstupy STP, LMT-, LMT+ a RPS (1 byte) MSB 7654321LSB STP, LMT-, LMT+RPS <table style="margin-top: 10px; border-collapse: collapse;"> <tr> <td>'0000'</td><td>200 μs</td><td>'0101'</td><td>3200 μs</td></tr> <tr> <td>'0001'</td><td>400 μs</td><td>'0110'</td><td>6400 μs</td></tr> <tr> <td>'0010'</td><td>800 μs</td><td>'0111'</td><td>12800 μs</td></tr> <tr> <td>'0011'</td><td>1600 μs</td><td>'1000'</td><td>Bez filtru</td></tr> <tr> <td>'0100'</td><td>1600 μs</td><td>'1001' až '1111'</td><td>Rezervováno (zadání této hodnoty znamená chybu)</td></tr> </table>	'0000'	200 μs	'0101'	3200 μs	'0001'	400 μs	'0110'	6400 μs	'0010'	800 μs	'0111'	12800 μs	'0011'	1600 μs	'1000'	Bez filtru	'0100'	1600 μs	'1001' až '1111'	Rezervováno (zadání této hodnoty znamená chybu)	--
'0000'	200 μs	'0101'	3200 μs																				
'0001'	400 μs	'0110'	6400 μs																				
'0010'	800 μs	'0111'	12800 μs																				
'0011'	1600 μs	'1000'	Bez filtru																				
'0100'	1600 μs	'1001' až '1111'	Rezervováno (zadání této hodnoty znamená chybu)																				
18	MEAS_SYS	Specifikuje systém odměřování (1 byte) 0 Počet pulzů (rychlost je měřena v pulzech/sekundu, hodnoty polohy jsou měřeny v pulzech). Hodnoty jsou uchovávány jako DINT. 1 Inženýrské jednotky (rychlost je měřena v jednotkách/sekundu, hodnoty polohy jsou měřeny v jednotkách). Hodnoty jsou ukládány jako hodnota REAL s jednoduchou přesností. 2 až 255 Rezervováno (zadání této hodnoty znamená chybu)	--																				
19	--	Rezervováno (Nastaveno na 0)	--																				
20	PLS/REV	Specifikuje počet pulzů na rychlost motoru (4 byty) Použitelné pouze tehdy, když je MEAS_SYS nastaven na 1.	DINT																				
24	UNITS/REV	Specifikuje počet inženýrských jednotek na rychlost motoru (4 byty) Použitelné pouze tehdy, když je MEAS_SYS nastaven na 1.	REAL																				
28	UNITS	Rezervováno pro STEP 7–Micro/WIN na uložení řetězce uživatelských jednotek (4 byty)	--																				
32	RP_CFG	Specifikuje sekvenci hledání referenčního bodu (1 byte) MSB 7654321LSB 00MODE RP_ADDR_DIR RP_SEEK_DIR RP_SEEK_DIR Tento bit specifikuje počáteční směr hledání referenčního bodu. (0 – kladný směr, 1 – záporný směr) RP_APPR_DIR Tento bit specifikuje směr přístupu k referenčnímu bodu. (0 – kladný směr, 1 – záporný směr) MODE Specifikuje způsob hledání referenčního bodu. '0000' Hledání referenčního bodu blokováno. '0001' Referenční bod je tam, kde vstup RPS přejde do aktivního stavu. '0010' Referenční bod leží ve středu aktivní oblasti vstupu RPS. '0011' Referenční bod leží mimo aktivní oblast vstupu RPS. '0100' Referenční bod leží v aktivní oblasti vstupu RPS. '0101' až '1111' Rezervováno (zadání této hodnoty znamená chybu)	--																				
33	--	Rezervováno (Nastaveno na 0)	--																				
34	RP_Z_CNT	Počet pulzů vstupu ZP použitý pro definování referenčního bodu (4 byty)	DINT																				

Tabulka 9-15 Tabulka konfigurace profilu, pokračování

Offset	Název	Popis funkce	Typ
38	RP_FAST	Vysoká rychlost pro operaci hledání RP (RP Seek): MAX_SPD nebo méně (4 byty)	DINT REAL
42	RP_SLOW	Nízká rychlost pro operaci hledání RP (RP Seek): maximální rychlost, ze které může motor okamžitě zastavit, nebo menší (4 bytes)	DINT REAL
46	SS_SPEED	Rychlost spuštění/zastavení (4 byty) Rychlost spuštění je maximální rychlost, na kterou může motor okamžitě přejít ze zastavení, a maximální rychlost, ze které může motor okamžitě zastavit. Provoz je pod touto rychlostí povolený, neplatí ale doba zrychlení a zpomalení.	DINT REAL
50	MAX_SPEED	Maximální provozní rychlost motoru (4 byty)	DINT REAL
54	JOG_SPEED	Rychlost manuálního posuvu. MAX_SPEED nebo méně (4 byty)	
58	JOG_INCREMENT	Hodnota přírůstku manuálního posunu je vzdálenost (nebo počet pulzů), o kterou je proveden posun jako reakce na jeden pulz manuálního posunu. (4 byty)	DINT REAL
62	ACCEL_TIME	Čas potřebný pro zrychlení z minimální na maximální rychlost v milisekundách (4 byty)	DINT
66	DECEL_TIME	Čas potřebný pro zpomalení z maximální na minimální rychlost v milisekundách (4 byty)	DINT
70	BKLSH_COMP	Kompenzace vůle: vzdálenost použitá na kompenzaci mrtvého chodu systému při změně směru (4 byty)	DINT REAL
74	JERK_TIME	Doba, po kterou je kompenzace rázů aplikována na počáteční a konečnou část křivky zrychlení/zpomalení (S křivka). Zadání hodnoty 0 zakáže kompenzaci rázů. Doba rázu je uvedena v milisekundách. (4 byty)	DINT
Interaktivní blok			
78	MOVE_CMD	Vybírá pracovní režim (1 byte) 0 Absolutní poloha 1 Relativní poloha 2 Kontinuální kladný pohyb s jednou rychlostí 3 Kontinuální záporný pohyb s jednou rychlostí 4 Manuální řízení rychlostí, kladný pohyb 5 Manuální řízení rychlostí, záporný pohyb 6 Kontinuální kladný pohyb s jednou rychlostí a se spouštěným zastavením (vstup RPS signalizuje zastavení) 7 Kontinuální záporný pohyb s jednou rychlostí a se spouštěným zastavením (vstup RPS signalizuje zastavení) 8 až 255 – Rezervováno (zadání této hodnoty znamená chybu)	--
79	--	Rezervováno. Nastaveno na 0	--
80	TARGET_POS	Cílová poloha, která se má dosáhnout v tomto posunu (4 byty)	DINT REAL
84	TARGET_SPEED	Cílová rychlost pro tento posun (4 byty)	DINT REAL
88	RP_OFFSET	Absolutní poloha referenčního bodu (4 byty)	DINT REAL
Profilový blok 0			
92 (+0)	STEPS	Počet kroků v této pohybové sekvenci (1 byte)	--
93 (+1)	MODE	Volí pracovní režim pro tento profilový blok (1 byte) 0 Absolutní poloha 1 Relativní poloha 2 Kontinuální kladný pohyb s jednou rychlostí 3 Kontinuální záporný pohyb s jednou rychlostí 4 Rezervováno (zadání této hodnoty znamená chybu) 5 Rezervováno (zadání této hodnoty znamená chybu) 6 Kontinuální kladný pohyb s jednou rychlostí a se spouštěným zastavením (RPS volí rychlost) 7 Kontinuální záporný pohyb s jednou rychlostí a se spouštěným zastavením (vstup RPS signalizuje stop) 8 Kontinuální kladný pohyb se dvěma rychlostmi (RPS volí rychlost) 9 Kontinuální záporný pohyb se dvěma rychlostmi (RPS volí rychlost) 10 až 255 – Rezervováno (zadání této hodnoty znamená chybu)	--

Tabulka 9-15 Tabulka konfigurace profilu, pokračování

Offset	Název		Popis funkce	Typ
94 (+2)	0	POS	Poloha, které se má dosáhnout v kroku 0 (4 byty)	DINT REAL
98 (+6)		SPEED	Cílová rychlost pro krok 0 (4 byty)	DINT REAL
102 (+10)	1	POS	Poloha, které se má dosáhnout v kroku 1 (4 byty)	DINT REAL
106 (+14)		SPEED	Cílová rychlost pro krok 1 (4 byty)	DINT REAL
110 (+18)	2	POS	Poloha, která se má dosáhnout v kroku 2 (4 byty)	DINT REAL
114 (+22)		SPEED	Cílová rychlost pro krok 2 (4 byty)	DINT REAL
118 (+26)	3	POS	Poloha, která se má dosáhnout v kroku 3 (4 byty)	DINT REAL
122 (+30)		SPEED	Cílová rychlost pro krok 3 (4 byty)	DINT REAL
Profilový blok 1				
126 (+34)	STEPS		Počet kroku v této pohybové sekvenci (1 byte)	--
127 (+35)	MODE		Volí pracovní režim pro tento profilový blok (1 byte)	--
128 (+36)	0	POS	Poloha, která se má dosáhnout v kroku 0 (4 byty)	DINT REAL
132 (+40)		SPEED	Cílová rychlost pro krok 0 (4 byty)	DINT REAL
...	...	...	...	...

Byty speciální paměti

Automat S7-200 přiděluje 50 bytů speciální paměti (SM) každému inteligentnímu modulu na základě fyzické polohy modulu v systému vstupů a výstupů. Viz tabulku 9-16. Když modul detekuje chybu nebo změnu stavu dat, aktualizuje tato místa v paměti SM. Nejprve modul aktualizuje SMB200 až SMB249, které jsou nutné pro hlášení chybového stavu, za druhé modul aktualizuje SMB250 až SMB299 apod.

Tabulka 9-16 Byty speciální paměti SMB200 až SMB549

Byty SM pro inteligentní modul na:						
místě 0	místě 1	místě 2	místě 3	místě 4	místě 5	místě 6
SMB200 až SMB249	SMB250 až SMB299	SMB300 až SMB349	SMB350 až SMB399	SMB400 až SMB449	SMB450 až SMB499	SMB500 až SMB549

Tabulka 9-17 ukazuje strukturu datové oblasti SM pro inteligentní modul. Tabulka předpokládá umístění inteligentního modulu na pozici 0 v systému rozšiřovacích modulů.

Tabulka 9-17 Definice oblasti speciální paměti pro polohovací modul EM 253

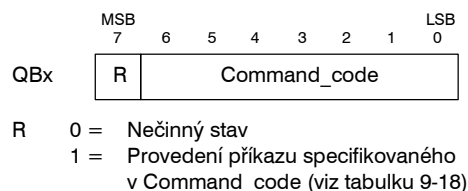
Adresa SM	Popis																									
SMB200 až SMB215	Název modulu (16 znaků ASCII). SMB200 začíná: "EM 253 Position"																									
SMB216 až SMB219	Číslo revize softwaru (4 ASCII znaky). SMB216 je první znak.																									
SMW220	Chybový kód modulu. Chybové kódy jsou popsány v tabulce 9-14.																									
SMB222	Stav vstupů a výstupů. Odráží stav vstupů a výstupů modulu. <table><tr><th colspan="7">MSB</th><th colspan="2">LSB</th></tr><tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr><tr><td>DIS</td><td>0</td><td>0</td><td>STP</td><td>LMT–</td><td>LMT+</td><td>RPS</td><td>ZP</td></tr></table> DIS Blokování výstupů STP Vstup stop LMT– Vstup meze záporného posunu LMT+ Vstup meze kladného posunu RPS Vstup spínače referenčního bodu ZP Vstup nulového pulzu 0 = Proud neprotéká 1 = Proud protéká 0 = Proud neprotéká 1 = Proud protéká 0 = Proud neprotéká 1 = Proud protéká 0 = Proud neprotéká 1 = Proud protéká 0 = Proud neprotéká 1 = Proud protéká 0 = Proud neprotéká 1 = Proud protéká	MSB							LSB		7	6	5	4	3	2	1	0	DIS	0	0	STP	LMT–	LMT+	RPS	ZP
MSB							LSB																			
7	6	5	4	3	2	1	0																			
DIS	0	0	STP	LMT–	LMT+	RPS	ZP																			
SMB223	Okamžitý stav modulu. Odráží stav konfigurace modulu a stav směru otáčení. <table><tr><th colspan="7">MSB</th><th colspan="2">LSB</th></tr><tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>OR</td><td>R</td><td>CFG</td></tr></table> OR Cílová rychlost mimo rozsah R Směr pohybu CFG Modul konfigurován 0 = V rozsahu 1 = Mimo rozsah 0 = Kladný pohyb 1 = Záporný pohyb 0 = Nekonfigurován 1 = Konfigurován	MSB							LSB		7	6	5	4	3	2	1	0	0	0	0	0	0	OR	R	CFG
MSB							LSB																			
7	6	5	4	3	2	1	0																			
0	0	0	0	0	OR	R	CFG																			
SMB224	CUR_PF je byte, který ukazuje na právě prováděný profil.																									
SMB225	CUR_STP je byte, který ukazuje na právě prováděný krok v profilu.																									
SMD226	CUR_POS je hodnota double word, která ukazuje na aktuální polohu modulu.																									
SMD230	CUR_SPD je hodnota double word, která ukazuje aktuální rychlost modulu.																									
SMB234	Výsledek instrukce. Chybové kódy jsou popsány v tabulce 9-13. Chyby s číslem vyšším než 127 jsou generovány instrukčními podprogramy vytvořenými průvodcem. <table><tr><th colspan="2">MSB</th><th colspan="2">LSB</th></tr><tr><th>7</th><th>6</th><th>7</th><th>0</th></tr><tr><td>D</td><td colspan="3">ERROR</td></tr></table> D Bit "Done" (Dokončeno) 0 = Operace probíhá 1 = Operace dokončena (nastaveno modulem během inicializace)	MSB		LSB		7	6	7	0	D	ERROR															
MSB		LSB																								
7	6	7	0																							
D	ERROR																									
SMB235 až SMB244	Rezervováno																									
SMB245	Offset prvního bytu Q použitého jako příkazové rozhraní tohoto modulu. Offset dodává automat S7-200 automaticky pro pohodlí uživatele; modul ho nepotřebuje.																									
SMD246	Ukazatel na místo tabulky konfigurace profilu v paměti V. Ukazatel na jinou oblast než paměť V není platný. Polohovací modul toto místo monitoruje, dokud neobdrží jinou než nulovou hodnotu ukazatele.																									

Vysvětlení příkazového bytu polohovacího modulu

Polohovací modul má jeden byte digitálních výstupů, který se používá jako příkazový byte. Obrázek 9-22 a tabulka 9-18 ukazují definici příkazového bytu.

Zápis do příkazového bytu, kdy se bit R změní z 0 na 1, interpretuje modul jako nový příkaz.

Pokud modul detekuje přechod na nečinnost (bit R změní stav na 0) v době, kdy je příkaz aktivní, je prováděná operace zrušena. Pokud je prováděn pohyb, zastaví se zpomalením.



Obr. 9-22 Definice příkazového bytu

Po dokončení operace musí modul zaregistrovat přechod do nečinného stavu dříve, než přijme nový příkaz. Pokud je operace zrušena, musí modul dokončit jakékoliv zpomalení dříve, než přijme nový příkaz. Jakákoliv změna hodnoty Command_code v době, kdy je nějaký příkaz aktivní, je ignorována.

Odezva polohovacího modulu na změnu pracovního režimu automatu S7-200 nebo na chybu se řídí tím, jak S7-200 ovlivní digitální výstupy podle stávající definice funkce programovatelného automatu S7-200:

- ☐ Pokud S7-200 přejde z režimu STOP do režimu RUN: Program v automatu S7-200 řídí práci polohovacího modulu.
- ☐ Pokud S7-200 přejde z režimu RUN do režimu STOP: Můžete vybrat stav, do kterého mají digitální výstupy přejít při přechodu do režimu STOP, nebo to, že si uchovají svůj poslední stav.
 - Pokud se při přechodu do režimu STOP vypne bit R: Zastaví polohovací modul jakýkoliv probíhající pohyb se zpomalením.
 - Pokud se při přechodu do režimu STOP zapne bit R: Polohovací modul dokončí jakýkoliv probíhající příkaz. Pokud žádný příkaz neprobíhá, provede polohovací modul příkaz specifikovaný bity v Command_code.
 - Pokud je bit R uchován ve svém posledním stavu: Polohovací modul dokončí jakýkoliv probíhající příkaz.
- ☐ Pokud automat S7-200 detekuje kritickou chybu a vypne všechny digitální výstupy: Polohovací modul zastaví jakýkoliv probíhající pohyb se zpomalením.

Tabulka 9-18 Definice Command_code

Command_code	Příkaz
000 0000 až 000 1111	0 až 24 Provést pohyb specifikovaný v profilových blocích 0 až 24
100 0000 až 111 0101	25 až 117 Rezervováno (zadání této hodnoty znamená chybu)
111 0110	118 Aktivovat výstup DIS
111 0111	119 Deaktivovat výstup DIS
111 1000	120 Pulz na výstupu CLR
111 1001	121 Znovu nahrát aktuální polohu
111 1010	122 Provést pohyb specifikovaný v interaktivním bloku
111 1011	123 Zachytit offset referenčního bodu
111 1100	124 Kladný manuální pohyb chodu
111 1101	125 Záporný manuální pohyb chodu
111 1110	126 Vyhledat polohu referenčního bodu
111 1111	127 Znovu nahrát konfiguraci

Při ztrátě komunikace s automatem S7-200 aktivuje polohovací modul obvod sledování funkce, který vypne výstupy. Pokud vyprší doba obvodu sledování funkce, polohovací modul zastaví jakýkoliv probíhající pohyb se zpomalením.

Při zjištění kritické chyby hardwaru nebo firmware modulu uvede polohovací modul výstupy P0, P1, DIS a CLR do neaktivního stavu.

Tabulka 9-19 Pohybové příkazy

Příkaz	Popis
Příkazy 0 až 24: <i>Provede pohyb specifikovaný v profilových blocích 0 až 24</i>	Při uskutečňování tohoto příkazu provádí polohovací modul pohybovou operaci specifikovanou v poli MODE profilového bloku, označeného částí příkazu Command_code. ■ V režimu 0 (absolutní poloha) definuje blok pohybového profilu jeden až čtyři kroky, z nichž každý obsahuje polohu (POS) i rychlost (SPEED), které popisují daný segment pohybu. Specifikace POS přestavuje absolutní místo, které se počítá podle místa označeného jako referenční bod. Směr pohybu je určen vztahem mezi aktuální polohou a polohou prvního kroku profilu. U pohybu o více krocích je obrácení směru posunu zakázáno a má za následek ohlášení chyby. ■ V režimu 1 (relativní poloha) definuje blok pohybového profilu jeden až čtyři kroky, z nichž každý obsahuje polohu (POS) i rychlost (SPEED), které popisují daný segment pohybu. Znaménko hodnoty polohy (POS) určuje směr pohybu. U pohybu o více krocích je obrácení směru posunu zakázáno a má za následek ohlášení chyby. ■ V režimech 2 a 3 (režimy s jednou rychlostí a kontinuálním pohybem) je specifikace polohy (POS) ignorována a modul zrychluje na rychlost specifikovanou v poli SPEED prvního kroku. Režim 2 se používá pro kladný pohyb, režim 3 pro záporný. Pohyb se zastaví, když příkazový byte přejde do nečinného stavu). ■ V režimech 6 a 7 (režimy s jednou rychlostí a nepřetržitým provozem se spouštěným zastavením) modul zrychluje na rychlost specifikovanou v poli SPEED prvního kroku. Jestliže vstup RPS přejde do aktivního stavu, pak se pohyb zastaví po dokončení vzdálenosti specifikované v poli POS prvního kroku. (Vzdálenost specifikovaná v poli POS musí zahrnovat vzdálenost pro zpomalování.) Pokud má pole POS hodnotu nula v době, kdy vstup RPS přejde do aktivního stavu, polohovací modul zastaví se zpomalením. Režim 6 se používá pro kladný pohyb, režim 7 pro záporný. ■ V režimech 8 a 9 vybere binární hodnota vstupu RPS jednu ze dvou hodnot rychlostí specifikovaných prvními dvěma kroky profilového bloku. – Pokud není RPS aktivní: Krok 0 řídí rychlost pohonu. – Pokud je RPS aktivní: Krok 1 řídí rychlost pohonu. Režim 8 se používá pro kladný pohyb, režim 9 pro záporný. Hodnota SPEED řídí rychlost pohybu. V tomto režimu jsou hodnoty POS ignorovány.
Příkaz 118 <i>Aktivuje výstup DIS</i>	Je-li prováděn tento příkaz, aktivuje polohovací modul výstup DIS.
Příkaz 119 <i>Deaktivuje výstup DIS</i>	Je-li prováděn tento příkaz, deaktivuje polohovací modul výstup DIS.
Příkaz 120 <i>Pulz na výstupu CLR</i>	Je-li prováděn tento příkaz, generuje polohovací modul 50milisekundový pulz na výstupu CLR.
Příkaz 121 <i>Znovu nahraje současnou polohu</i>	Je-li prováděn tento příkaz, nastaví polohovací modul aktuální polohu na hodnotu nalezenou v poli TARGET_POS interaktivního bloku.
Příkaz 122 <i>Provede pohyb specifikovaný v interaktivním bloku</i>	Je-li prováděn tento příkaz, provede polohovací modul pohybovou operaci specifikovanou v poli MOVE_CMD interaktivního bloku. ■ V režimech 0 a 1 (režimy absolutního a relativního pohybu) je pohyb o jednom kroku prováděn na základě informací o cílové rychlosti a poloze uvedených v polích TARGET_SPEED a TARGET_POS interaktivního bloku. ■ V režimech 2 a 3 (režimy s jednou rychlostí a nepřetržitým provozem) je specifikace polohy ignorována a polohovací modul zrychluje na rychlost specifikovanou v poli TARGET_SPEED interaktivního bloku. Pohyb se zastaví, když příkazový byte přejde do nečinného stavu. ■ V režimech 4 a 5 (režimy manuálního řízení rychlosti) je specifikace polohy ignorována a uživatelský program nahraje hodnotu změn rychlosti do pole TARGET_SPEED interaktivního bloku. Polohovací modul toto místo trvale monitoruje a příslušně reaguje na změny hodnoty rychlosti.

Tabulka 9-19 Pohybové příkazy, pokračování

Příkaz	Popis
Příkaz 123 <i>Zachycuje posunutí referenčního bodu</i>	Je-li prováděn tento příkaz, stanovuje polohovací modul nulovou polohu, která je na jiném místě, než je poloha referenčního bodu. Před vydáním tohoto příkazu musíte mít stanovenou polohu referenčního bodu a stroj musí být manuálním posunem připraven v počáteční pracovní poloze. Po obdržení tohoto příkazu vypočítá polohovací modul offset mezi počáteční pracovní polohou (aktuální poloha) a polohou referenčního bodu a vypočítaný offset zapíše do pole RP_OFFSET interaktivního bloku. Pak je aktuální poloha nastavena na 0, čímž je počáteční pracovní poloha stanovena jako nulová poloha. V případě, že krokový motor ztratí přehled o své poloze (například při ztrátě napájení nebo když je poloha krokového motoru změněna ručně), je možné vydat příkaz "Vyhledat polohu referenčního bodu" pro automatické opětovné stanovení nulové polohy.
Příkaz 124 <i>Kladný manuální chod</i>	Tento příkaz umožňuje ručně vysílat pulzy pro posun krokového motoru v kladném směru. Pokud zůstane příkaz aktivní po dobu kratší než 0,5 sekundy, vysílá polohovací modul pulzy pro posun o vzdálenost specifikovanou v JOG_INCREMENT. Pokud zůstane příkaz aktivní po dobu 0,5 sekundy nebo déle, začne polohovací modul zrychlovat na rychlost specifikovanou v JOG_SPEED. Je-li detekován přechod do nečinného stavu, zastaví polohovací modul se zpomalením.
Příkaz 125 <i>Záporný manuální chod</i>	Tento příkaz umožňuje ručně vysílat pulzy pro posun krokového motoru v záporném směru. Pokud zůstane příkaz aktivní po dobu kratší než 0,5 sekundy, vysílá polohovací modul pulzy pro posun o vzdálenost specifikovanou v JOG_INCREMENT. Pokud zůstane příkaz aktivní po dobu 0,5 sekundy nebo déle, začne polohovací modul zrychlovat na rychlost specifikovanou v JOG_SPEED. Je-li detekován přechod do nečinného stavu, zastaví polohovací modul se zpomalením.
Příkaz 126 <i>Vyhledat polohu referenčního bodu</i>	Je-li prováděn tento příkaz, iniciuje polohovací modul operaci vyhledání referenčního bodu pomocí specifikovaného způsobu hledání. Když je referenční bod lokalizován a pohyb se zastavil, nahraje polohovací modul hodnotu načtenou z pole RP_OFFSET interaktivního bloku na aktuální polohu a generuje na výstupu CLR pulz o délce 50 milisekund.
Příkaz 127 <i>Opětovně nahrát konfiguraci</i>	Je-li prováděn tento příkaz, načte polohovací modul ukazatel tabulky konfigurace profilu z příslušného místa paměti SM a potom načte konfigurační blok z místa specifikovaného ukazatelem tabulky konfigurace profilu. Polohovací modul porovná právě získaná konfigurační data se stávající konfigurací modulu a provede nutné změny nastavení nebo nové výpočty. Všechny profily uložené v rychlé vyrovnávací paměti jsou smazány.

Vysvětlení rychlé vyrovnávací paměti polohovacího modulu pro profily

Polohovací modul uchovává prováděcí data až pro 4 profily v rychlé vyrovnávací paměti. Když polohovací modul dostane příkaz k provedení nějakého profilu, kontroluje, zda je požadovaný profil uložen v rychlé vyrovnávací paměti. Pokud jsou prováděcí data profilu v této paměti přítomna, polohovací modul profil okamžitě provede. Pokud prováděcí data profilu v této paměti přítomna nejsou, načte polohovací modul informace profilového bloku z tabulky konfigurace profilů v automatu S7-200 a před jeho provedením vypočítá prováděcí data profilu.

Příkaz 122 (Provést pohyb specifikovaný v interaktivním bloku) nepoužívá pro uchovávání prováděcích dat rychlou vyrovnávací paměť, ale vždy načítá interaktivní blok z tabulky konfigurace profilů v automatu S7-200 a vypočítává prováděcí data pohybu.

Opětovná konfigurace polohovacího modulu vymaže všechna prováděcí data uložená v rychlé vyrovnávací paměti.

Tvorba vlastních instrukcí pro řízení polohy

Polohovací instrukce pro řízení práce polohovacího modulu vytváří "Průvodce polohováním"; můžete ale také vytvořit své vlastní instrukce. Následující segment kódu STL je příklad toho, jak můžete vytvořit své vlastní řídicí instrukce pro polohovací modul.

V tomto příkladu je použit automat S7-200 CPU 224 s polohovacím modulem na místě 0. Polohovací modul je konfigurován při studeném startu. CMD_STAT je symbol pro SMB234, CMD je symbol pro QB2 a NEW_CMD je symbol pro profil.

Vzorový program: Řízení polohovacího modulu

```

Network 1      //Stav nového pohybového příkazu
LSCR          State_0

Network 2      //CMD_STAT je symbol pro SMB234
               //CMD je symbol pro QB2
               //NEW_CMD je symbol pro profil.
               //
               //1. Vymaže bit "Done" (Dokončeno) polohovacího modulu.
               //2. Vymaže příkazový byte polohovacího modulu.
               //3. Vydá nový příkaz.
               //4. Čeká na provedení příkazu.

LD            SM0.0
MOVB          0, CMD_STAT
BIW           0, CMD_
BIW           NEW_CMD, CMD
SCRT          State_1

Network 3
SCRE

Network 4      //Čeká na dokončení příkazu.
LSCR          State_1

Network 5      //Pokud je příkaz dokončen bez chyby, přejde do nečinného stavu.
LDB=          CMD_STAT, 16#80
SCRT          Idle_State

Network 6      //Pokud je příkaz dokončen s chybou, přejde do stavu správy chyb.
LDB>          CMD_STAT, 16#80
SCRT          Error_State

Network 7
SCRE

```

Tvorba programu pro modemový modul

Modemový modul EM 241 umožňuje přímé připojení automatu S7-200 k analogové telefonní lince a podporuje komunikaci mezi S7-200 a STEP 7 – Micro/WIN. Modemový modul také podporuje protokol Modbus slave RTU. Komunikace mezi modemovým modulem a automatem S7-200 probíhá přes rozšiřovací sběrnici.

STEP 7 – Micro/WIN obsahuje průvodce, který pomáhá nastavit vzdálený modem nebo modemový modul pro připojení lokálního S7-200 ke vzdálenému zařízení.

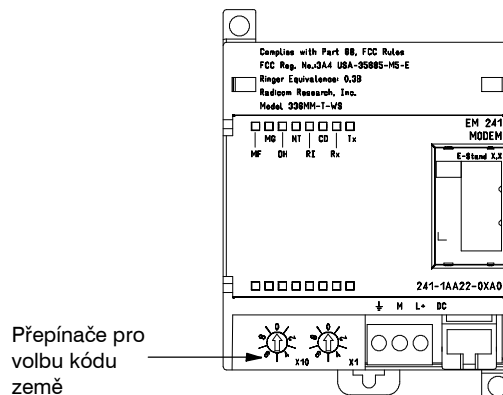
Přehled kapitoly

Funkce modemového modulu	288
Použití průvodce pro konfiguraci modemového modulu	294
Přehled instrukcí a omezení pro modem	298
Instrukce pro modemový modul	299
Vzorový program	303
CPU S7-200 podporující inteligentní moduly	303
Byty speciální paměti pro modemový modul	304
Témata pro pokročilé	306
Formát telefonního čísla pro zprávu	308
Formát textové zprávy	309
Formát zprávy pro přenos dat CPU	310

Funkce modemového modulu

Modemový modul umožňuje připojení S7-200 přímo k analogové telefonní lince; má následující funkce:

- ☐ Připojení mezinárodní telefonní linky
- ☐ Modemové spojení se STEP 7 – Micro/WIN pro programování a hledání chyb (teleservis)
- ☐ Podporuje protokol Modbus RTU
- ☐ Podporuje přenos numerických a textových zpráv na pager
- ☐ Podporuje SMS zprávy
- ☐ Umožňuje přesun dat z CPU do CPU nebo z CPU do Modbus
- ☐ Umožňuje ochranu heslem
- ☐ Zabezpečení zpětným voláním
- ☐ Konfigurace modemového modulu je uložena v CPU

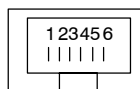


Obr. 10-1 Modemový modul EM 241

Pro konfiguraci modemového modulu můžete použít STEP 7 – Micro/WIN a jeho funkci "Průvodce modemem". Specifikaci modemového modulu najdete v příloze A.

Připojení mezinárodní telefonní linky

Modemový modul je standardní 10bitový modem V.34 (33,6 kBaud); je kompatibilní s většinou interních a externích modemů pro PC. Modemový modul nekomunikuje s 11bitovými modemy.



Vývod	Popis	
3	Vyzvánění	Je povoleno obrácené zapojení.
4	Společný	

Obr. 10-2 Pohled na jack RJ11

Modemový modul se připojuje k telefonní lince pomocí šestipólového čtyřdrátového konektoru RJ11, instalovaného na přední straně modulu. Viz obr. 10-2.

Pro připojení konektoru RJ11 ke standardní telefonní zásuvce může být v různých zemích nutný adaptér. Více informací najdete v dokumentaci přechodového konektoru.

Rozhraní mezi modemem a telefonní linkou je napájeno z vnějšího zdroje 24 V DC. Ten může být připojen ke zdroji snímačů v CPU nebo k vnějšímu zdroji. Zemnicí svorku modemového modulu spojte s uzemněním systému.

Pokud je modemový modul připojen ke zdroji, nakonfiguruje automaticky telefonní připojení pro provoz v dané zemi. Kód země se volí dvěma otočnými přepínači na přední straně modulu. Dříve než modemový modul připojíte ke zdroji, musíte přepínači nastavit potřebnou volbu země. Nastavení přepínačů pro podporované země najdete v tabulce 10-1.

Tabulka 10-1 Země podporované EM 241

Nastavení přepínačů	Země
00	Česká republika
01	Rakousko
02	Belgie
05	Kanada
08	Dánsko
09	Finsko
10	Francie
11	Německo
12	Řecko
16	Irsko
18	Itálie
22	Lucembursko
25	Nizozemí
27	Norsko
30	Portugalsko
34	Španělsko
35	Švédsko
36	Švýcarsko
38	Velká Británie
39	USA

Připojení ke STEP 7–Micro/WIN

Modemový modul umožňuje komunikaci se STEP 7–Micro/WIN prostřednictvím telefonní linky (teleservis). Pokud používáte STEP 7–Micro/WIN, nemusíte CPU S7-200 na použití modemového modulu jako vzdáleného modemu ani konfigurovat, ani programovat.

Při použití modemového modulu spolu se STEP 7–Micro/WIN proveďte následující kroky:

1. Vypněte napájení CPU S7-200 a připojte modemový modul na rozšiřovací vstupní/výstupní sběrnici. Nepřipojujte žádné moduly, pokud je CPU S7-200 pod napětím.
2. Připojte modemový modul k telefonní lince. V případě potřeby použijte adaptér.
3. Připojte 24 V DC ke svorkovnici modemového modulu.
4. Připojte uzemnění svorkovnice modemového modulu k uzemnění systému.
5. Nastavte přepínače pro kód země.
6. Připojte CPU S7-200 a modemový modul ke zdroji napájecího napětí.
7. Nakonfigurujte STEP 7–Micro/WIN pro komunikaci s 10bitovým modemem.

Protokol Modbus RTU

Modemový modul můžete nakonfigurovat tak, aby komunikoval jako Modbus RTU slave. Modemový modul přijímá požadavky Modbus přes modemové rozhraní, interpretuje tyto požadavky a přenáší data do CPU nebo z něj. Modul pak generuje odpověď Modbus a odešle ji přes modemové rozhraní.



Tip

Pokud je modemový modul nakonfigurován, aby komunikoval jako Modbus RTU slave, není STEP 7–Micro/WIN schopen komunikovat s modemovým modulem přes telefonní linku.

Modemový modul podporuje funkce Modbus, uvedené v tabulce 10-2.

Funkce Modbus 4 a 16 umožňují čtení nebo zápis maximálně 125 paměťových registrů (250 bytů paměti V) v jednom požadavku. Funkce 5 a 15 zapisují hodnoty do registru obrazu výstupů CPU. Tyto hodnoty mohou být přepsány uživatelským programem.

Adresy Modbus se běžně zapisují jako 5 nebo 6 znakových hodnot, které obsahují typ dat a offset. První jeden nebo dva znaky určují typ dat a poslední čtyři znaky volí příslušnou hodnotu v daném typu dat. Zařízení Modbus master mapuje adresy na správné funkce Modbus.

Tabulka 10-2 Funkce Modbus podporované modemovým modulem

Funkce	Popis
Funkce 01	Čtení stavu výstupu
Funkce 02	Čtení stavu vstupu
Funkce 03	Čtení paměťových registrů
Funkce 04	Čtení registrů (analogových vstupů)
Funkce 05	Zápis na jeden výstup
Funkce 06	Předvolení jednoho registru
Funkce 15	Zápis na více výstupů
Funkce 16	Předvolení více registrů

Tabulka 10-3 uvádí adresy Modbus podporované modemovým modulem a mapování adres Modbus na adresy CPU S7-200.

Pro vytvoření konfiguračního bloku pro protokol Modbus RTU pomocí modemového modulu použijte "Průvodce modemem". Dříve než můžete použít protokol Modbus, musíte provést download konfiguračního bloku modemového modulu do datového bloku CPU.

Tabulka 10-3 Mapování adres Modbus na CPU S7-200

Adresy Modbus	Adresy CPU S7-200
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	VW0
040002	VW2
040003	VW4
...	...
04xxxx	VW 2*(xxxx-1)

Přenos zpráv pro pager a SMS zpráv

Modemový modul podporuje posílání numerických a textových zpráv pro pager a SMS zpráv (Short Message Service – služba krátkých textových zpráv) na mobilní telefony (pokud je podporuje operátor). Zprávy a telefonní čísla jsou uloženy v konfiguračním bloku modemového modulu, který musí být nahrán do datového bloku CPU S7-200. Pro vytvoření zpráv a telefonních čísel pro konfigurační blok modemového modulu můžete použít průvodce. Ten také vytváří programový kód, který v uživatelském programu umožňuje posílat zprávy.

Přenos numerických zpráv

Přenos numerických zpráv používá tóny tlačítkového tónového telefonu pro posílání numerických hodnot na pager. Modemový modul vytočí požadovanou službu přenosu zpráv, vyčká ukončení hlasové zprávy a pak odešle tóny odpovídající číslicím ve zprávě. V ní jsou povoleny číslice 0 až 9, hvězdička (*), A, B, C a D. Skutečné znaky zobrazené na pageru pro hvězdičku a znaky A, B, C a D nejsou standardizované; určuje je však pager a operátor.

Přenos textových zpráv

Přenos textových zpráv umožňuje posílání alfanumerických zpráv operátorovi a odtud na pager. Poskytovatelé přenosu textových zpráv mají běžně modemovou linku, která přijímá textové zprávy. Modemový modul používá pro přenos textových zpráv na operátora protokol "Telelocator Alphanumeric Protocol" (TAP). Tento protokol používá pro příjem zpráv mnoho operátorů.

SMS – služba krátkých textových zpráv

Přenos SMS zpráv podporují některé mobilní telefony; obecně ty, které jsou GSM kompatibilní. SMS umožňuje modemovému modulu zaslat SMS zprávu poskytovateli prostřednictvím analogové telefonní linky. Poskytovatel SMS pak zprávu vyšle na mobilní telefon. Pro posílání zpráv poskytovateli SMS používá modemový modul protokoly "Telelocator Alphanumeric Protocol" (TAP) a "Universal Computer Protocol" (UCP). SMS zprávy můžete posílat pouze těm poskytovatelům SMS, kteří na modemové lince tyto protokoly podporují.

Vložené proměnné v textových zprávách pro pager a SMS zprávách

Modemový modul umí vkládat hodnoty dat z CPU do textových zpráv a formátovat hodnoty dat na základě specifikace ve zprávě. Můžete zadat počet číslic vlevo a vpravo od desetinné čárky a také to, zda použít desetinnou čárku, nebo tečku. Pokud uživatelský program přikáže modemovému modulu, aby odeslal textovou zprávu, vyhledá modemový modul zprávu v CPU, určí, které hodnoty z CPU jsou ve zprávě potřeba, vyhledá tyto hodnoty v CPU a pak tyto hodnoty zformátuje a vloží je do textové zprávy dříve, než ji odešle operátorovi.

Telefonní číslo operátora zpráv, zpráva a proměnné vložené do zprávy jsou načteny z CPU během několika programových cyklů CPU. V průběhu posílání zprávy by uživatelský program neměl měnit telefonní čísla nebo text zprávy. Aktualizace proměnných vložených do zprávy může v průběhu odesílání pokračovat. Pokud zpráva obsahuje více proměnných, jsou tyto proměnné načítány během více programových cyklů CPU. Pokud chcete, aby byly všechny vložené proměnné ve zprávě jednotné, nesmíte po odeslání zprávy žádnou z vložených proměnných měnit.

Přenos dat

Modemový modul umožňuje pomocí uživatelského programu přesouvat data do jiného CPU nebo zařízení Modbus po telefonní lince. Přenos dat a telefonní čísla se konfiguruji pomocí "Průvodce modemem" a jsou uloženy v konfiguračním bloku modemového modulu. Pak je proveden download konfiguračního bloku do datového bloku CPU S7-200. Průvodce vytváří také programový kód, který umožňuje, aby uživatelský program spustil přenos dat.

Přenos dat může znamenat jednak požadavek načtení dat ze vzdáleného zařízení, jednak požadavek zápisu dat do vzdáleného zařízení. Při datovém přenosu je možné načíst nebo zapsat 1 až 100 wordů dat. Data se přesouvají do paměti V připojeného CPU nebo z něj.

Průvodce umožňuje vytvořit přenos dat, skládající se z jednoho čtení ze vzdáleného zařízení, jednoho zápisu do vzdáleného zařízení nebo kombinace čtení i zápisu vzdáleného zařízení.

Přenos dat používá nakonfigurovaný protokol modemového modulu. Jestliže je modemový modul nakonfigurován, aby podporoval protokol PPI (tam, kde odpovídá STEP 7 – Micro/WIN), použije protokol PPI i pro přenos dat. Pokud je modemový modul nakonfigurován, aby podporoval protokol Modbus RTU, jsou přenosy dat uskutečněny s použitím protokolu Modbus.

Telefonní číslo vzdáleného zařízení, požadavek na přenos dat a přenášená data jsou načtena z CPU během několika programových cyklů CPU. V průběhu posílání zprávy by uživatelský program neměl měnit telefonní čísla nebo text zprávy. Také byste v době posílání zprávy neměli měnit přenášená data.

Pokud je vzdáleným zařízením jiný modemový modul, je možné při přesunu dat použít funkci ochrany heslem vložením hesla vzdáleného modemového modulu do konfigurace telefonního čísla. Při přesunu dat není možné použít funkci zpětného volání.

Ochrana heslem

Zabezpečení modemového modulu heslem je volitelné; aktivuje se pomocí průvodce. Heslo používané modemovým modulem není stejné jako heslo pro CPU. Heslo modemového modulu je samostatné heslo o 8 znacích, které musí volající zadat do modemového modulu dříve, než je mu umožněn přístup k připojenému CPU. Heslo je uloženo v paměti V CPU jako součást konfiguračního bloku modemového modulu. Konfigurační blok modemového modulu musí být nahrán do datového bloku připojeného CPU.

Pokud má CPU aktivováno v systémovém bloku zabezpečení heslem, musí volající zadat heslo pro CPU, aby získal přístup k jakémukoli funkci chráněné heslem.

Zabezpečení zpětným voláním

Zabezpečení modemového modulu zpětným voláním je volitelné a konfiguruje se pomocí průvodce. Funkce zpětného volání poskytuje další zabezpečení připojeného CPU tím, že umožňuje přístup k CPU pouze z předdefinovaných telefonních čísel. Když je povolena funkce zpětného volání, odpoví modemový modul na jakékoliv příchozí volání, ověří volajícího a pak linku odpojí. Jestliže má volající oprávnění, vytočí potom modemový modul předdefinované telefonní číslo volajícího a povolí přístup k CPU.

Modemový modul podporuje tři režimy zpětného volání:

- ☐ Zpětné volání na jedno předdefinované telefonní číslo
- ☐ Zpětné volání na více předdefinovaných telefonních čísel
- ☐ Zpětné volání na libovolné telefonní číslo

Režim zpětného volání se volí zaškrtnutím příslušné volby v průvodci a následným definováním telefonních čísel pro zpětné volání. Telefonní čísla pro zpětné volání jsou uložena v konfiguračním bloku modemového modulu uloženém v datovém bloku připojeného CPU.

Nejjednodušší forma zpětného volání je na jediné předdefinované telefonní číslo. Pokud je v konfiguračním bloku uloženo pouze jedno číslo pro zpětné volání, odpovídá modemový modul na příchozí volání kdykoliv, uvědomí volajícího, že je aktivováno zpětné volání, odpojí volajícího a potom vytočí číslo pro zpětné volání, specifikované v konfiguračním bloku.

Modemový modul také podporuje zpětné volání na více předdefinovaných telefonních čísel. V tomto režimu je volající požádán o telefonní číslo. Pokud zadané číslo odpovídá jednomu z předdefinovaných telefonních čísel v konfiguračním bloku modemového modulu, odpojí modul volajícího a zavolá zpět s použitím odpovídajícího telefonního čísla z konfiguračního bloku. Uživatel může nakonfigurovat až 250 čísel pro zpětné volání.

Tam, kde existuje více předdefinovaných čísel pro zpětné volání, musí číslo pro zpětné volání zadané při připojování k modemovému modulu přesně odpovídat číslu v konfiguračním bloku modemového modulu s výjimkou prvních dvou číslic. Jestliže je nakonfigurované číslo pro zpětné volání například 91(123)4569999 vzhledem k nutnosti vytočit vnější linku (9) a meziměsto (1), může být číslo zadané pro zpětné volání kterékoli z následujících:

- ☐ 91(123)4569999
- ☐ 1(123)4569999
- ☐ (123)4569999

Všechna výše uvedená telefonní čísla jsou považována za odpovídající číslo pro zpětné volání. Modul použije při provádění zpětného volání telefonní číslo ze svého konfiguračního bloku, v tomto příkladu 91(123)4569999. Při konfigurování více čísel pro zpětné volání se ujistěte, zda jsou všechna telefonní čísla jedinečná s výjimkou prvních dvou číslic. Při porovnávání telefonních čísel pro zpětné volání se používají pouze číselné znaky telefonního čísla. Při tomto porovnávání jsou speciální znaky (jako čárky nebo závorky) ignorovány.

Zpětné volání na jakékoliv telefonní číslo se v průvodci nastavuje pomocí volby "Použít zpětné volání pro všechna čísla" během konfigurace zpětného volání. Pokud je tato volba vybrána, odpovídá modemový modul na příchozí volání a požaduje telefonní číslo pro zpětné volání. Poté, co volající zadá telefonní číslo, odpojí ho modemový modul a vytočí toto telefonní číslo. Tento režim zpětného volání umožňuje účtovat telefonní poplatky pouze za telefonní připojení modulu a nezabezpečuje CPU S7-200. Pokud použijete tento režim zpětného volání, měl by být modul zabezpečen heslem.

Funkce zabezpečení heslem a zpětného volání mohou být pro modemový modul aktivovány zároveň. Modul požaduje po volajícím, aby zadal správné heslo dříve, než začne zpracovávat zpětné volání.

Tabulka konfigurace modemového modulu

Všechny textové zprávy, telefonní čísla, informace o přesunu dat, čísla pro zpětné volání a ostatní nastavení jsou uloženy v tabulce konfigurace modemového modulu, která musí být nahrána do V paměti CPU S7-200. Při tvorbě tabulky konfigurace modemového modulu vás provede průvodce. Potom STEP 7—Micro/WIN umístí tabulku konfigurace modemového modulu do datového bloku, který je následně nahrán do CPU S7-200.

Modemový modul načte tuto tabulku konfigurace z CPU při spuštění a také do pěti sekund po jakémkoliv přechodu CPU z režimu STOP do režimu RUN. Modemový modul nenačte novou tabulku konfigurace z CPU, dokud je modul spojen online se STEP 7–Micro/WIN. Pokud je proveden download nové tabulky konfigurace, když je modemový modul online, načte modul novou tabulku konfigurace až po skončení online připojení.

Pokud modemový modul zjistí v tabulce konfigurace chybu, začne na přední straně modulu blikat LED "Modul v pořádku" (MG). Zkontrolujte informační dialogové okno STEP 7–Micro/WIN programovatelného automatu nebo přečtete hodnotu v SMW220 (modul na pozici 0), abyste získali informace o chybě konfigurace. Chyby konfigurace modemového modulu jsou uvedeny v tabulce 10-4. Pokud pro vytvoření tabulky konfigurace modemového modulu použijete průvodce, STEP 7–Micro/WIN data před vytvořením tabulky konfigurace zkontroluje.

Tabulka 10-4 Chyby konfigurace EM 241 (hexadecimální)

Chyba	Popis
0000	Bez chyby
0001	Není vnější napájení 24 V DC
0002	Porucha modemu
0003	Chybí ID konfiguračního bloku – identifikace EM 241 na začátku tabulky konfigurace pro tento modul neplatí.
0004	Konfigurační blok mimo rozsah – ukazatel tabulky konfigurace neukazuje na paměť V nebo je některá část tabulky mimo rozsah paměti V připojeného CPU.
0005	Chyba konfigurace – je povoleno zpětné volání, počet telefonních čísel pro zpětné volání se rovná 0 nebo je větší než 250. Počet zpráv je větší než 250. Počet telefonních čísel pro přenos zpráv je větší než 250 nebo je délka telefonních čísel pro přenos zpráv větší než 120 bytů.
0006	Chyba výběru kódu země – výběr země pomocí dvou otočných přepínačů není platný.
0007	Příliš dlouhé telefonní číslo – je povoleno zpětné volání a délka čísla pro něj je větší než maximální.
0008 až 00FF	Rezervováno
01xx	Chyba v čísle xx pro zpětné volání – v telefonním čísle pro zpětné volání xx jsou nepřipustné znaky. Hodnota xx je 1 pro první číslo pro zpětné volání, 2 pro druhé atd.
02xx	Chyba v telefonním čísle xx – jedno z polí v čísle xx telefonu pro přenos zpráv nebo číslo xx telefonu pro přesun dat obsahuje nepřipustnou hodnotu. Hodnota xx je 1 pro první telefonní číslo, 2 pro druhé atd.
03xx	Chyba ve zprávě xx – zpráva nebo přesun dat číslo xx překračuje maximální délku. Hodnota xx je 1 pro první zprávu, 2 pro druhou...
0400 až FFFF	Rezervováno

Stavové LED modemového modulu

Modemový modul má na předním panelu 8 LED diod pro zobrazení stavu. Tabulka 10-5 tyto LED popisuje.

Tabulka 10-5 Stavové LED modulu EM 241

LED	Popis
MF	Porucha modulu – tato kontrolka svítí, pokud modul zjistí poruchový stav jako například: ■ Chybí vnější napájení 24 V DC ■ Vypršel čas obvodu sledování funkce vstupů a výstupů ■ Porucha modemu ■ Chyba komunikace s lokálním CPU
MG	Modul v pořádku – tato kontrolka svítí, pokud modul nehlásí žádnou chybu. LED "Modul v pořádku" bliká, pokud je chyba v tabulce konfigurace nebo uživatel zvolil nepřípustné nastavení kódu země pro rozhraní telefonní linky. Zkontrolujte u programovatelného automatu informační dialogové okno STEP 7–Micro/WIN nebo přečtěte hodnotu v SMW220 (pro modul na pozici 0), abyste získali informace o chybě konfigurace.
OH	Hovor – tato LED svítí, pokud EM 241 aktivně používá telefonní linku.
NT	Není oznamovací tón (No Dial Tone) – LED indikuje chybu a rozsvítí se, pokud dostal EM 241 příkaz odeslat zprávu a na telefonní lince není oznamovací tón. Chyba je to pouze tehdy, když byl EM 241 nakonfigurován, aby před vytáčením kontroloval přítomnost oznamovacího tónu. LED zůstane po neúspěšném pokusu o vytvoření čísla svítit přibližně 5 sekund.
RI	Indikátor vyzvánění – tato LED znamená, že EM 241 přijímá příchozí hovor.
CD	Detekce nosné – tato LED znamená, že bylo navázáno spojení se vzdáleným modemem.
Rx	Příjem dat – tato LED bliká, když modem přijímá data.
Tx	Odesílání dat – tato LED bliká, když modem odesílá data.

Použití průvodce pro konfiguraci modemového modulu

Spusťte průvodce z menu STEP 7–Micro/WIN nebo z části "Nástroje" v navigační liště.

Abyste mohli tohoto průvodce použít, musí být projekt zkompileován a nastaven na režim symbolického adresování. Pokud jste svůj program ještě nezkompileovali, udělejte to.

1. V prvním dialogovém okně "Průvodce modemem" vyberte "Konfigurovat modemový modul E 241" a klikněte na "Další >".
2. Průvodce vyžaduje zadat pozici modemového modulu vzhledem k CPU S7-200, aby generoval správný programový kód. Pro automatické načtení inteligentních modulů připojených k CPU klikněte na tlačítko "Přečíst manuály". Rozšiřovací moduly jsou číslovány postupně počínaje nulou. Dvakrát klikněte na modemový modul, který chcete konfigurovat, nebo nastavte hodnotu do pole pozice modemového modulu. Klikněte na "Další >".

U CPU S7-200 s verzí firmware nižší než 1.2 musíte instalovat inteligentní modul přímo vedle CPU, aby průvodce modul konfiguroval.

3. Dialogové okno pro ochranu heslem umožňuje povolit tuto ochranu a přiřadit modulu heslo o 1 až 8 znacích. Toto heslo je nezávislé na hesle pro CPU S7-200. Když je modul chráněn heslem, je od každého, kdo se pokouší spojit s CPU S7-200 přes modem, požadováno zadání správného hesla. Pokud chcete, vyberte ochranu heslem a vložte heslo. Klikněte na "Další >".
4. Modemový modul podporuje dva komunikační protokoly: protokol PPI (pro komunikaci se STEP 7–Micro/WIN) a protokol Modbus RTU. Výběr protokolu závisí na typu zařízení, které se používá jako vzdálený komunikační partner. Toto nastavení řídí použitý komunikační protokol, pokud modemový modul odpovídá na volání a pokud iniciuje přesun dat. Vyberte příslušný protokol a klikněte na "Další >".

5. Modul můžete nakonfigurovat pro posílání numerických a textových zpráv na pager nebo SMS zpráv mobilním telefonům. Zaškrtněte políčko "Povolit v této konfiguraci zprávy" a klikněte na tlačítko "Konfigurovat zprávy" pro definování zpráv a telefonních čísel příjemců.
6. Když konfigurujete zprávu, která má být odeslána na pager nebo mobilnímu telefonu, musíte definovat text zprávy a telefonní číslo. Vyberte záložku "Zprávy" v dialogovém okně "Konfigurace zprávy" a klikněte na tlačítko "Nová zpráva". Vložte text zprávy a specifikujte hodnoty všech dat CPU, která mají být vložena do zprávy. Chcete-li vložit do zprávy nějakou hodnotu dat CPU, klikněte na tlačítko "Vložit data...". Zadejte adresu hodnoty dat CPU (např. VW100), formát zobrazení (např. integer se znaménkem) a počet číslic vlevo a vpravo od desetinné čárky. Můžete také specifikovat, zda chcete použít desetinnou čárku, nebo desetinnou tečku.
 - Numerické zprávy pro pager jsou omezeny na číslice 0 až 9, písmena A, B, C a D a hvězdičku (*). Maximální povolená délka numerické zprávy pro pager se mění podle operátora.
 - Textové zprávy mohou mít délku až 119 znaků a obsahovat libovolné alfanumerické znaky.
 - Textové zprávy mohou obsahovat libovolný počet vložených proměnných.
 - Proměnné se mohou vkládat z paměti V, M, SM, I, Q, S, T, C nebo AI připojené CPU.
 - Před hexadecimálními daty je zobrazeno "16#". Počet znaků v hodnotě je závislý na velikosti proměnné. Například VW100 se zobrazí jako 16#0123.
 - Počet číslic vlevo od desetinné čárky musí být dostatečně velký na to, aby zobrazil předpokládaný rozsah hodnot včetně záporného znaménka, pokud je hodnota dat integer se znaménkem nebo hodnota s plovoucí desetinnou čárkou.
 - Pokud je formát dat integer a počet číslic vpravo od desetinné čárky není nulový, je hodnota integer zobrazena jako integer s upraveným měřítkem. Pokud například VW100 = 1234 a vy zadáte dvě číslice vpravo od desetinné čárky, jsou data zobrazena jako "12,34".
 - Pokud je hodnota dat větší, než je možné zobrazit ve specifikovaném poli, uvede modemový modul znak # na všech místech pro znaky hodnoty dat.
7. Telefonní čísla se konfigurují vybráním záložky "Telefonní čísla" v dialogovém okně "Konfigurace zprávy". Chcete-li přidat nové telefonní číslo, klikněte na tlačítko "Nové telefonní číslo...". Když je telefonní číslo nakonfigurováno, musí být přidáno do projektu. Vyberte telefonní číslo ve sloupci "Dostupné telefonní číslo" a klikněte na okénko se šipkou vpravo, abyste telefonní číslo přidali do aktuálního projektu. Jakmile jste jednou telefonní číslo přidali do aktuálního projektu, můžete toto číslo vybrat a přidat k němu symbolický název pro používání ve vašem programu.

Telefonní číslo se skládá z několika polí, která závisí na typu přenosu zpráv vybraném uživatelem.

- Volba protokolu pro přenos zpráv říká modulu, který protokol má použít při odesílání zprávy operátorovi. Numerický pager podporuje pouze numerický protokol. Textové služby obvykle vyžadují TAP (Telelocator Alphanumeric Protocol). Operátoři SMS zpráv podporují TAP nebo UCP (Universal Computer Protocol). Pro přenos SMS zpráv se normálně používají tři různé služby UCP. Většina poskytovatelů podporuje příkaz 1 nebo 51. Ověřte si u operátora SMS zpráv, jaký protokol a příkazy vyžaduje.
- Pole "Popis tohoto kontaktu" umožňuje přidat textový popis telefonního čísla.

- Pole "Telefonní číslo" je telefonní číslo operátora služby přenosu zpráv. U textových zpráv je to telefonní číslo modemové linky, kterou operátor používá pro příjem těchto zpráv. U přenosu numerických zpráv je to přímo telefonní číslo pageru. Modemový modul povoluje maximálně 40 znaků pro pole telefonního čísla. V telefonních číslech, která modemový modul vytáčí, jsou povoleny následující znaky:

0 až 9	zadání z klávesnice telefonu
A, B, C, D, *, #	znaky DTMF (pouze tónová tlačítková volba)
,	pozastaví vytáčení na dobu 2 sekund
!	dá modemu příkaz generovat rychlé zavěšení
@	počkat 5 sekund
W	před pokračováním počkat na oznamovací tón
()	ignoruje se (je možné použít pro formátování telefonního čísla)

- Pole "Telefonní číslo nebo ID pageru" je určeno pro vložení čísla pageru nebo čísla mobilního telefonu příjemce zprávy. Toto číslo nesmí obsahovat jiné znaky než číslice 0 až 9. Je přípustných maximálně 20 znaků.
 - Pole "Heslo" je volitelné pro zprávu TAP. Někteří operátoři požadují heslo, ale normálně by toto pole mělo být ponecháno prázdné. Modemový modul povoluje heslo až o 15 znacích.
 - Pole "Číslo odchozího telefonu" umožňuje identifikaci modemového modulu v SMS zprávě. Toto pole požadují někteří operátoři, kteří používají příkazy UCP. Někteří operátoři mohou v tomto poli požadovat minimální počet znaků. Modemový modul povoluje až 15 znaků.
 - Pole "Standard modemu" se používá v případech, kdy se modemový modul a modem operátora nemohou dohodnout na standardu modemu. Implicitní hodnota je V.34 (33,6 kBaud).
 - Pole "Datový formát" umožňuje nastavit počet datových bitů a paritu použité modemem při odesílání zprávy operátorovi. TAP běžně používá 7 datových bitů a sudou paritu, ale někteří operátoři používají 8 datových bitů a žádnou paritu. UCP vždy používá 8 datových bitů a žádnou paritu. Ověřte si u operátora, jaké nastavení použít.
8. Modemový modul můžete nakonfigurovat, aby přenášel data do jiného CPU S7-200 (pokud byl vybrán protokol PPI) nebo aby přenášel data zařízení Modbus (pokud byl vybrán protokol Modbus). Zaškrtněte políčko "Povolit v této konfiguraci přesun dat CPU – CPU" a klikněte na tlačítko "Konfigurovat CPU – CPU", abyste mohli definovat přesuny dat a telefonní čísla vzdálených zařízení.
 9. Když nastavujete přesun dat z CPU na CPU nebo z CPU na Modbus, musíte definovat data, která se mají přesouvat, a telefonní číslo vzdáleného zařízení. Vyberte záložku "Přenosy dat" v dialogovém okně "Konfigurovat datových přenosů" a klikněte na tlačítko "Nový přenos". Přenos dat se skládá ze čtení dat ze vzdáleného zařízení, zápisu dat do vzdáleného zařízení nebo ze čtení i zápisu do vzdáleného zařízení. Pokud je vybráno čtení i zápis, je nejprve provedeno čtení a potom zápis.

Při každém čtení nebo zápisu může být přesunuto až 100 wordů. Data se musí přesunovat do paměti V lokálního CPU nebo z něj. Průvodce vždy popisuje paměťová místa ve vzdáleném zařízení, jako kdyby bylo vzdálené zařízení CPU S7-200. Pokud je vzdálené zařízení Modbus, probíhá přenos do paměťových registrů v zařízení Modbus nebo z nich (adresa 04xxxx). Ekvivalentní adresa Modbus (xxxx) se zjistí následujícím způsobem:

$$\begin{aligned} \text{Adresa Modbus} &= 1 + (\text{adresa v paměti V} / 2) \\ \text{Adresa v paměti V} &= (\text{adresa Modbus} - 1) * 2 \end{aligned}$$

10. Záložka "Telefonní čísla" v dialogovém okně "Konfigurace datových přenosů" umožňuje definovat telefonní čísla pro přenosy dat z CPU do CPU nebo z CPU do Modbus. Chcete-li přidat nové telefonní číslo, klikněte na tlačítko "Nové telefonní číslo...". Když je telefonní číslo nakonfigurováno, musí být přidáno do projektu. Vyberte telefonní číslo ve sloupci "Dostupná telefonní čísla" a klikněte na okénko se šipkou vpravo, abyste telefonní číslo přidali do aktuálního projektu. Jakmile jste jednou telefonní číslo přidali do aktuálního projektu, můžete toto číslo vybrat a přidat k němu symbolický název pro použití ve vašem programu.

Pole "Popis" a "Telefonní číslo" jsou stejná jako ta, která byla popsána výše pro přenos zpráv. Pole "Heslo" se vyžaduje, jestliže je vzdálené zařízení modemový modul a byla aktivována ochrana heslem. Pole "Heslo" v lokálním modemovém modulu musí být nastaveno na heslo vzdáleného modemového modulu. Lokální modemový modul toto heslo zadá, když to požaduje vzdálený modemový modul.

11. Zpětné volání způsobí, že modemový modul automaticky odpojí a vytočí předdefinované telefonní číslo po tom, co přijme příchozí volání ze vzdáleného STEP 7–Micro/WIN. Abyste mohli konfigurovat telefonní čísla pro zpětné volání, zaškrtněte políčko "Povolit v této konfiguraci zpětné volání" a klikněte na tlačítko "Konfigurace zpětného volání...". Klikněte na "Další >".

12. Dialogové okno "Konfigurace zpětného volání..." umožňuje vkládat telefonní čísla, která modemový modul používá, když odpovídá na příchozí volání. Zaškrtněte "Povolit zpětná volání pouze pro některá čísla", jestliže mají být čísla pro zpětné volání předem definována. Pokud má modemový modul zpětně volat na kterékoliv číslo (kvůli hovorům na účet volaného), zaškrtněte volbu "Povolit zpětné volání pro všechna čísla".

Pokud chcete pro zpětné volání použít pouze specifikovaná telefonní čísla, klikněte na tlačítko "Nové telefonní číslo", požadujete-li přidat telefonní čísla pro zpětné volání. Dialogové okno "Vlastnosti zpětného volání" umožňuje vložit předdefinovaná telefonní čísla pro zpětné volání a popis těchto čísel. Číslo pro zpětné volání je takové telefonní číslo, které modemový modul vytáčí při zpětném volání. Toto telefonní číslo by mělo obsahovat všechny číslice potřebné pro připojení k vnější lince, pauzu pro čekání na vnější linku, připojení na meziměsto atd.

Po vložení nového telefonního čísla pro zpětné volání musíte toto číslo přidat do projektu. Vyberte telefonní číslo ve sloupci "Dostupná čísla pro zpětné volání" a klikněte na okénko se šipkou vpravo, abyste telefonní číslo přidali do aktuálního projektu.

13. Můžete nastavit počet pokusů o vytočení čísla, které má modemový modul uskutečnit při odesílání zprávy nebo při přesunu dat. Modemový modul hlásí uživatelskému programu chybu pouze tehdy, jsou-li všechny pokusy o vytočení čísla a odeslání zprávy neúspěšné.

U některých telefonních linek není slyšet oznamovací tón, když se zvedne sluchátko. Běžně modemový modul hlásí uživatelskému programu chybu, pokud není přítomen oznamovací tón, když modemový modul dostane příkaz odeslat zprávu nebo provést zpětné volání. Abyste umožnili vytáčení čísla na lince bez oznamovacího tónu, zaškrtněte okno "Povolit vytáčení bez oznamovacího tónu".

14. "Průvodce modemem" vytvoří pro modemový modul konfigurační blok a požaduje po uživateli, aby vložil počáteční adresu, kde jsou uložena konfigurační data modemového modulu. Konfigurační blok modemového modulu je uložen v paměti V CPU. STEP 7–Micro/WIN zapíše konfigurační blok do datového bloku projektu. Velikost konfiguračního bloku se mění v závislosti na počtu konfigurovaných zpráv a telefonních čísel. Můžete vybrat adresu v paměti V, kam chcete konfigurační blok uložit, nebo klikněte na tlačítko "Navrhnout adresu", chcete-li, aby průvodce navrhl adresu volného bloku paměti o správné velikosti v paměti V. Klikněte na "Další >".

15. Závěrečným krokem při konfiguraci modemového modulu je zadání adresy v paměti Q pro příkazový byte modemového modulu. Adresu v paměti Q můžete stanovit spočítáním výstupních bytů použitých digitálními výstupními moduly připojenými k S7-200 před modemovým modulem. Klikněte na "Další >".

16. Nyní průvodce generuje projektové komponenty pro vaši zvolenou konfiguraci (programový blok a datový blok) a zpřístupní tento kód pro použití ve vašem programu. Poslední dialogové okno průvodce zobrazí komponenty vámi požadované konfigurace projektu. Musíte provést download konfiguračního bloku modemového modulu (datový blok) a programového bloku do CPU S7-200.

Přehled instrukcí a omezení pro modem

Průvodce usnadňuje řízení modemového modulu tak, že vytváří jednoznačné instrukční podprogramy na základě pozice modulu a nastavení, které jste si vybrali. Každá instrukce má prefix "MODx_", v němž x je pozice modulu.

Požadavky na použití instrukcí modemového modulu EM 241

Pokud používáte instrukce modemového modulu, vezměte v úvahu tyto požadavky:

- ☐ Instrukce modemového modulu používají tři podprogramy.
- ☐ Díky instrukcím modemového modulu se zvětšuje část paměti potřebná pro uživatelský program až o 370 bytů. Pokud vymažete podprogram, který nepoužíváte, můžete v případě potřeby opět spustit průvodce a instrukci znovu vytvořit.
- ☐ Musíte zajistit, aby byla v danou dobu aktivní pouze jedna instrukce.
- ☐ Instrukce není možné používat v podprogramu přerušení.
- ☐ Modemový modul načte informace z tabulky konfigurace při prvním studeném startu a po přechodu z režimu STOP do režimu RUN. Jakoukoliv změnu provedenou uživatelským programem v tabulce konfigurace modul neregistruje až do změny režimu nebo příštího vypnutí a obnovení napájení.

Použití instrukcí modemového modulu EM 241

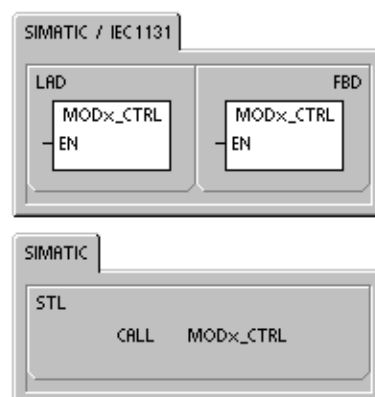
Pokud chcete použít instrukce pro modemový modul, proveďte v programu S7-200 tyto kroky:

1. Použijte průvodce a vytvořte tabulku konfigurace modemového modulu.
2. Vložte do svého programu instrukci MODx_CTRL a použijte kontakt SM0.0 pro její provedení v každém programovém cyklu.
3. Vložte instrukci MODx_MSG pro každou zprávu, kterou potřebujete odeslat.
4. Vložte instrukci MODx_XFR pro každý přenos dat.

Instrukce pro modemový modul

Instrukce MODx_CTRL

Instrukce MODx_CTRL (Řízení) se používá pro aktivaci a inicializaci modemového modulu. Tuto instrukci je třeba volat v každém programovém cyklu; musí být v projektu použita pouze jednou.



Instrukce MODx_XFR

Instrukce MODx_XFR (Přenos dat) se používá jako příkaz pro modemový modul, aby načítal data z jiného CPU S7-200 či zařízení Modbus nebo je do nich zapisoval. Tato instrukce vyžaduje 20 až 30 sekund od doby, kdy je spuštěn vstup START do nastavení bitu "Done" (Dokončeno).

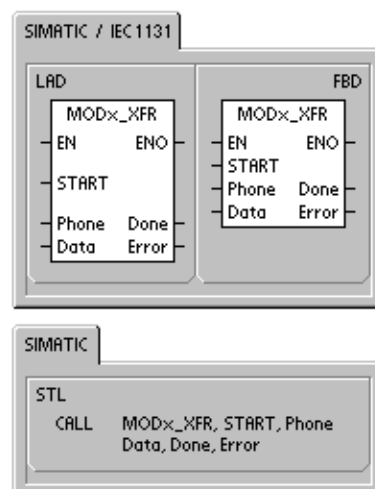
Bit EN musí být zapnutý, aby byl povolen příkaz pro modul, a musí zůstat zapnutý, dokud se nenastaví bit "Done", který signalizuje ukončení procesu. Příkaz XFR je poslán modemovému modulu v každém programovém cyklu, když je zapnutý vstup START a modul právě nepracuje. Pulzy na vstupu START by měly být generovány pomocí detekce hrany, která aktivuje odeslání pouze jednoho příkazu.

"Phone" (Telefon) je jedno z telefonních čísel pro přenos dat. Můžete použít symbolický název, který jste přiřadili každému telefonnímu číslu pro přenos dat, když bylo číslo definováno pomocí průvodce.

"Data" je číslo jednoho z definovaných přenosů dat. Můžete použít symbolický název, který jste přiřadili přenosu dat, když byl požadavek definován pomocí průvodce.

"Done" (Dokončeno) je bit, který se zapne, když modul ukončí přenos dat.

"Error" (Chyba) je byte, který obsahuje výsledek přenosu dat. Tabulka 10-4 definuje možné chyby, které mohou vyplynout z provedení této instrukce.



Tabulka 10-6 Parametry instrukce MODx_XFR

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, M, S, SM, T, C, V, L, signálový tok
Phone, Data	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, konstanta, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Instrukce MODx_MSG

Instrukce MODx_MSG (Odeslat zprávu) se používá pro odeslání zprávy z modemového modulu. Tato instrukce vyžaduje 20 až 30 sekund od doby, kdy je spuštěn vstup START do nastavení bitu "Done" (dokončeno).

Bit EN musí být zapnutý, aby byl příkaz aktivován, a musí zůstat zapnutý, dokud se nenastaví bit "Done", který signalizuje ukončení procesu. Příkaz MSG je poslán modemovému modulu v každém programovém cyklu, když je zapnutý vstup START a modul právě nepracuje. Pulzy by měly být na vstupu START generovány pomocí detekce hrany, která aktivuje odeslání pouze jednoho příkazu.

"Phone" je jedno z telefonních čísel pro přenos zpráv. Můžete použít symbolický název, který jste přiřadili každému telefonnímu číslu pro přenos zpráv, když bylo číslo definováno pomocí průvodce.

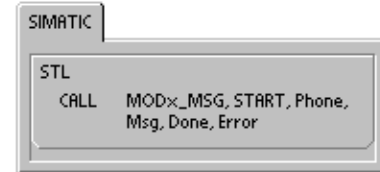
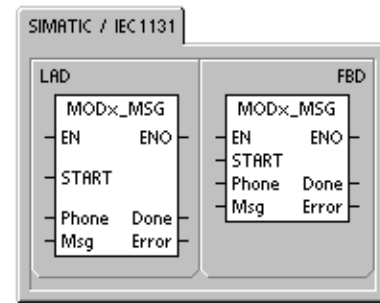
"Msg" je číslo jedné z definovaných zpráv. Můžete použít symbolický název, který jste zprávě přiřadili, když byla definována pomocí průvodce.

"Done" (Dokončeno) je bit, který se zapne, když modemový modul ukončí odesílání zprávy operátorovi.

"Error" (chyba) je byte, který obsahuje výsledek tohoto požadavku na modul. Tabulka 10-8 definuje možné chyby, které mohou vyplynout z provedení této instrukce.

Tabulka 10-7 Parametry instrukce MODx_MSG

Vstupy/Výstupy	Typ dat	Operandy
START	BOOL	I, Q, M, S, SM, T, C, V, L, signálový tok
Phone, Msg	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, konstanta, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



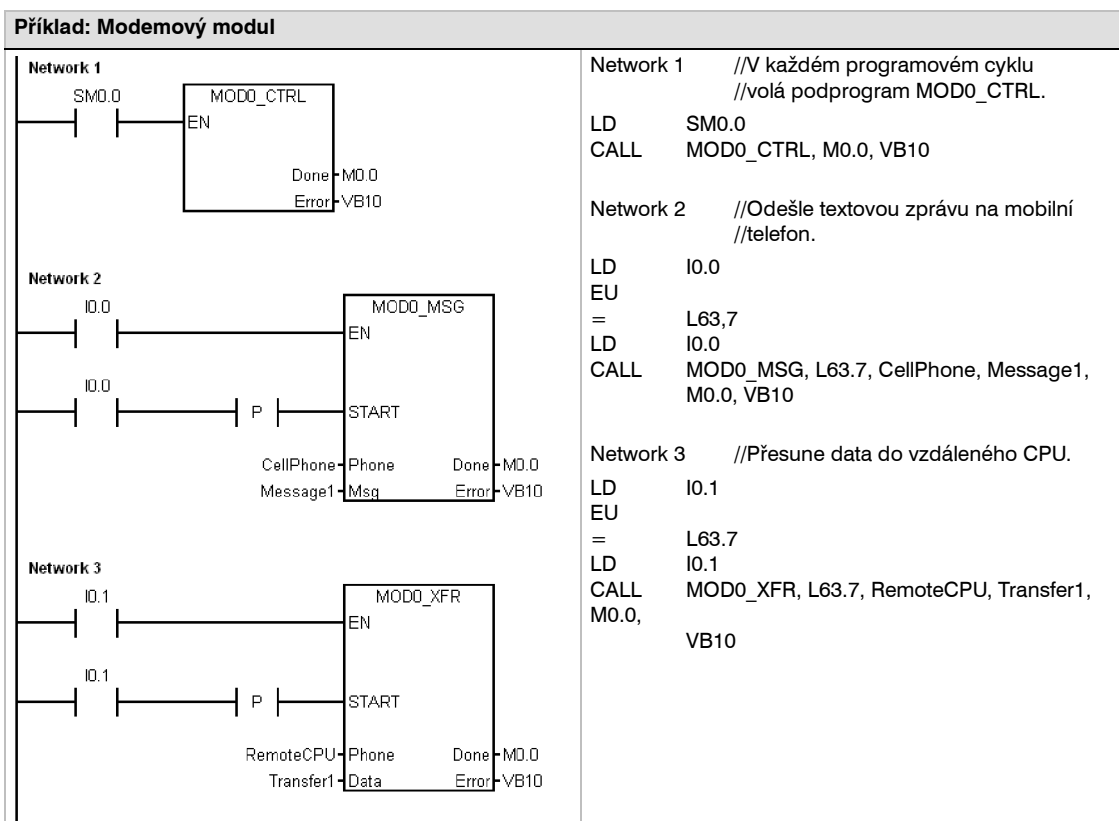
Tabulka 10-8 Chybové kódy hlášené instrukcemi MODx_MSG a MODx_XFR

Chyba	Popis
0	Bez chyby
Chyby telefonní linky	
1	Není přítomen oznamovací tón
2	Linka obsazena
3	Chyba vytáčení
4	Volané číslo neodpovídá
5	Vypršení spojovací doby (ke spojení nedošlo během 1 minuty)
6	Spojení přerušeno nebo neznámá odezva
Chyby v příkazu	
7	Numerická zpráva obsahuje nepřípustné znaky
8	Telefonní číslo (vstup Phone) mimo rozsah
9	Zpráva nebo přesun dat (vstup Msg nebo Data) mimo rozsah
10	Chyba v textové zprávě nebo zprávě přenosu dat
11	Chyba v telefonním čísle pro přenos zpráv nebo dat
12	Nepřípustná operace (např. počet pokusů o vytočení nastaven na nulu)
Chyby operátora	
13	Bez odezvy (vypršení času) od služby přenosu zpráv
14	Služba přenosu zpráv odpojena z neznámého důvodu
15	Zpráva zrušena uživatelem (zablokován příkazový bit)
TAP – chyby zpráv ohlášené operátorem	
16	Příjem vzdáleného odpojení (operátor zrušil relaci)
17	Přihlášení nebylo přijato službou přenosu zpráv (nesprávné heslo)
18	Blok nebyl přijat službou přenosu zpráv (chyba kontrolního součtu nebo přenosu)
19	Blok nebyl přijat službou přenosu zpráv (neznámý důvod)
UCP – chyby zpráv ohlášené operátorem	
20	Neznámá chyba
21	Chyba kontrolního součtu
22	Syntaktická chyba
23	Operace není systémem podporována (nepřípustný příkaz)
24	V této době není operace povolena
25	Aktivní blokování účastníka (černá listina)
26	Neplatná adresa volajícího
27	Neúspěšné prokazování totožnosti
28	Neúspěšný legitimovací kód
29	Neplatný GA
30	Opakování není povoleno
31	Legitimovací kód pro opakování, neúspěch
32	Prioritní spojení nepovoleno
33	Legitimovací kód pro prioritní spojení, neúspěch
34	Urgentní zpráva nepovolena
35	Legitimovací kód pro urgentní zprávu, neúspěch
36	Spojení na účet volaného nepovoleno
37	Legitimovací kód pro spojení na účet volaného, neúspěch

Tabulka 10-8 Chybové kódy hlášené instrukcemi MODx_MSG a MODx_XFR, pokračování

Chyba	Popis
UCP – operátor hlásí chyby SMS zprávy (pokračování)	
38	Odložené doručení nepovoleno
39	Nový AC neplatný
40	Nový legitimovací kód nepovolen
41	Neplatný standardní text
42	Neplatná délka časového úseku
43	Systém nepodporuje typ zprávy
44	Zpráva příliš dlouhá
45	Požadovaný standardní text neplatný
46	Typ zprávy neplatný pro typ pageru
47	Zpráva nenalezena v SMSC
48	Rezervováno
49	Rezervováno
50	Účastník zavěsil
51	Nepodporovaná skupina faxu
52	Nepodporován typ faxové zprávy
Chyby přenosu dat	
53	Vypršení doby pro zprávu (není odezva od vzdáleného zařízení)
54	Vzdálené CPU provádí upload nebo download
55	Chyba přístupu (paměť mimo rozsah, nepřipustný typ dat)
56	Chyba komunikace (neznámá odpověď)
57	Chyba kontrolního součtu nebo CRC v odpovědi
58	Vzdálený EM 241 nastaven na zpětné volání (nepovoleno)
59	Vzdálený EM 241 zamítl zadané heslo
60 až 127	Rezervováno
Chyby v používání instrukcí	
128	Tento požadavek nelze zpracovat. Modemový modul zpracovává jiný požadavek, nebo pro tento požadavek nebyl impuls START.
129	Chyba modemového modulu: ■ Umístění modemového modulu nebo adresa v paměti Q, která byla nakonfigurována pomocí "Průvodce modemem", neodpovídá skutečnému umístění nebo adrese v paměti. ■ Viz SMB8 až SMB21 (ID vstupního/výstupního modulu a chybový registr)

Vzorový program



CPU S7-200 podporující inteligentní moduly

Modemový modul je inteligentní rozšiřovací modul určený pro spolupráci s CPU S7-200 uvedenými v tabulce 10-9.

Tabulka 10-9 Kompatibilita modemového modulu EM 241 s CPU S7-200

CPU	Popis
Model CPU 222 verze 1.10 nebo vyšší	CPU 222 DC/DC/DC
	CPU 222 AC/DC/Relé
Model CPU 224 verze 1.10 nebo vyšší	CPU 224 DC/DC/DC
	CPU 224 AC/DC/Relé
Model CPU 226 verze 1.00 nebo vyšší	CPU 226 DC/DC/DC
	CPU 226 AC/DC/Relé
Model CPU 226XM verze 1.00 nebo vyšší	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/Relé

Byty speciální paměti pro modemový modul

Každému inteligentnímu modulu je na základě jeho fyzického umístění v rozšiřovací sběrnici přiděleno padesát bytů speciální paměti (SM). Když je zjištěna chyba nebo změna stavu, modul na to reaguje aktualizací míst v paměti SM odpovídajících umístění modulu. Pokud je to první modul, aktualizuje SMB200 až SMB249 tak, jak je to potřeba pro hlášení stavových a chybových informací. Pokud je to druhý modul, aktualizuje SMB250 až SMB299 a tak dále. Viz tabulku 10-10.

Tabulka 10-10 Byty speciální paměti SMB200 až SMB549

Byty speciální paměti SMB200 až SMB549						
Inteligentní modul na pozici 0	Inteligentní modul na pozici 1	Inteligentní modul na pozici 2	Inteligentní modul na pozici 3	Inteligentní modul na pozici 4	Inteligentní modul na pozici 5	Inteligentní modul na pozici 6
SMB200 až SMB249	SMB250 až SMB299	SMB300 až SMB349	SMB350 až SMB399	SMB400 až SMB449	SMB450 až SMB499	SMB500 až SMB549

Tabulka 10-11 ukazuje datovou oblast speciální paměti přidělenou modemovému modulu. Tato oblast je definována, jako kdyby byl inteligentní modul umístěn na pozici 0 rozšiřovacího systému.

Tabulka 10-11 Byty v paměti SM pro modemový modul EM 241

Adresa SM	Popis								
SMB200 až SMB215	Název modulu (16 znaků ASCII), SMB200 je první znak. “Modem EM241”								
SMB216 až SMB219	Číslo verze firmwaru (4 znaky ASCII), SMB216 je první znak.								
SMW220	Kód chyby 0000 – Bez chyby 0001 – Není napájení ze strany uživatele 0002 – Porucha modemu 0003 – Chybí ID konfiguračního bloku 0004 – Konfigurační blok mimo rozsah 0005 – Chyba konfigurace 0006 – Chyba ve volbě kódu země 0007 – Telefonní číslo příliš dlouhé 0008 – Zpráva příliš dlouhá 0009 až 00FF – Rezervováno 01xx – Chyba v čísle xx pro zpětné volání 02xx – Chyba v čísle xx pro pager 02xx – Chyba čísla zprávy xx 0400 až FFFF – Rezervováno								
SMB222	Stav modulu – odráží stav LED MSB 76543210LSB <table><tr><td>F</td><td>G</td><td>H</td><td>T</td><td>R</td><td>C</td><td>0</td><td>0</td></tr></table> F – EM_FAULT0 – bez poruchy1 – porucha G – EM_GOOD0 – není v pořádku1 – v pořádku H – OFF_HOOK0 – sluchátko zavěšené,1 – sluchátko zvednuté T – NO_DIALTONE0 – oznamovací tón1 – není oznamovací tón R – RING0 – nevyzvání1 – telefon vyzvání C – CONNECT0 – nespojen1 – spojen	F	G	H	T	R	C	0	0
F	G	H	T	R	C	0	0		
SMB223	Kód země, jak je nastaven na přepínačích (desítková hodnota)								
SMW224	Přenosová rychlost stanovená pro spojení (desítková hodnota bez znaménka).								

Tabulka 10-11 Byty v paměti SM pro modemový modul EM 241, pokračování

Adresa SM	Popis			
SMB226	Výsledek příkazu uživatele MSB 7 6 5 LSB 0 <table><tr><td>D</td><td>0</td><td>ERROR</td></tr></table> D – Bit "Done"; 0 – operace probíhá 1 – operace ukončena ERROR : Popis kódu chyby, viz tabulku 10-8	D	0	ERROR
D	0	ERROR		
SMB227	Volič telefonního čísla – tento byte specifikuje, které telefonní číslo pro přenos zpráv se má při odeslání zprávy použít. Platné hodnoty jsou 1 až 250.			
SMB228	Volič zprávy – tento byte specifikuje, kterou zprávu odeslat. Platné hodnoty jsou 1 až 250.			
SMB229 až SMB244	Rezervováno			
SMB245	Offset od prvního bytu Q použitého jako příkazové rozhraní tohoto modulu. Offset dodává CPU pro pohodlí uživatele a modul ho nepotřebuje.			
SMD246	Ukazatel na tabulku konfigurace modulu v paměti V. Hodnota ukazatele na oblast jinou než paměť V není přijata a modul nadále kontroluje toto místo a čeká na jinou než nulovou hodnotu ukazatele.			

Tabulka 10-12 Tabulka konfigurace modemového modulu, pokračování

Blok telefonních čísel pro zpětné volání (volitelný)	
Offset bytu	Popis
24	Telefonní číslo pro zpětné volání 1 – řetězec představující první telefonní číslo, které má oprávnění pro přístup ke zpětnému volání z modemového modulu EM 241. Každému telefonnímu číslu pro zpětné volání musí být přidělen takový prostor, jaký specifikuje pole pro délku telefonního čísla pro zpětné volání (offset 6 v konfiguračním bloku).
24 + číslo pro zpětné volání	Telefonní číslo pro zpětné volání 2
:	:
:	Telefonní číslo pro zpětné volání n
Blok telefonních čísel pro přenos zpráv (volitelný)	
Offset bytu	Popis
M	Telefonní číslo pro přenos zpráv 1 – řetězec představující telefonní číslo pro přenos zpráv, které obsahuje volby protokolu a tónové pulzní volby. Každému telefonnímu číslu musí být přidělena taková oblast paměti, jaká specifikuje pole pro délku telefonního čísla zpráv (offset 7 v konfiguračním bloku). Následuje popis formátu telefonního čísla pro přenos zpráv
M + délka tel. čísla pro přenos zpráv	Telefonní číslo pro přenos zpráv 2
:	:
:	Telefonní číslo pro přenos zpráv n
Blok zpráv (volitelný)	
Offset bytu	Popis
N	Offset paměti V (vzhledem k VB0) pro první zprávu (2 byty)
N+2	Délka zprávy 1
N+3	Délka zprávy 2
:	:
:	Délka zprávy n
P	Zpráva 1 – řetězec (max. 120 bytů) představující první zprávu. Tento řetězec obsahuje specifikaci textu a vložené proměnné nebo může zadat přenos dat CPU. Viz formát textové zprávy a formát přenosu dat CPU níže.
P + délka zprávy 1	Zpráva 2
:	:
:	Zpráva n

Modul znovu načítá tabulku konfigurace, jestliže dojde k následujícím událostem:

- ☐ Do pěti vteřin po každém přechodu z režimu STOP do režimu RUN CPU S7-200 (pokud modem právě nepracuje online)
- ☐ Každých pět vteřin, dokud není nalezena platná konfigurace (pokud modem právě nepracuje online)
- ☐ Pokaždé, když modem přejde z práce online do práce offline

Formát telefonního čísla pro zprávu

Telefonní číslo pro přenos zpráv je struktura obsahující informace, které modemový modul potřebuje pro odeslání zprávy. Telefonní číslo pro přenos zpráv je řetězec ASCII znaků, v němž je na prvním místě byte obsahující délku a za ním následují ASCII znaky. Maximální délka telefonního čísla pro přenos zpráv je 120 bytů (včetně bytu délky).

Telefonní číslo pro přenos zpráv obsahuje až 6 polí oddělených lomítkem (/). Dvě lomítka vedle sebe označují prázdné (nulové) pole. V modemovém modulu jsou nulová pole nastavena na implicitní hodnoty.

Formát: <Telefonní číslo>/<ID>/<Heslo>/<Protokol>/<Standard>/<Formát>

Pole "Telefonní číslo" je telefonní číslo, které modemový modul vytáčí, když odesílá zprávu. Pokud je odesílanou zprávou textová zpráva pro pager nebo SMS zpráva, jde o telefonní číslo operátora. Pokud je zprávou numerická zpráva pro pager, obsahuje toto pole telefonní číslo pageru. Pokud je zprávou přenos dat CPU, jde o telefonní číslo vzdáleného zařízení. Maximální počet znaků v tomto poli je 40.

ID je číslo pageru nebo číslo mobilního telefonu. Toto pole může obsahovat pouze číslice od 0 do 9. Pokud je protokol nastaven na přenos dat CPU, použije se toto pole pro zadání adresy vzdáleného zařízení. V tomto poli je povoleno až 20 znaků.

Pole "Heslo" se používá pro zadání hesla u zpráv odesílaných prostřednictvím TAP v případě, že heslo vyžaduje operátor. U zpráv odesílaných prostřednictvím UCP se toto pole používá jako adresa nebo telefonní číslo odesílatele. Pokud zpráva znamená přenos dat CPU do jiného modemového modulu, může být toto pole použito pro zadání hesla pro vzdálený modul. Heslo může mít délku až 15 znaků.

Pole "Protokol" obsahuje jeden ASCII znak, který říká modemovému modulu, jak má formátovat a vysílat zprávu. Příпустné jsou následující hodnoty:

- 1 – Numerický protokol pro pager (implicitně)
- 2 – TAP
- 3 – UCP příkaz 1
- 4 – UCP příkaz 30
- 5 – UCP příkaz 51
- 6 – přesun dat CPU

Pole "Standard" nutí modemový modul používat konkrétní standard pro modem. Pole "Standard" obsahuje jeden ASCII znak. Příпустné jsou následující hodnoty:

- 1 – Bell 103
- 2 – Bell 212
- 3 – V.21
- 4 – V.22
- 5 – bit V.22
- 6 – V.23c
- 7 – V.32
- 8 – bit V.32
- 9 – V.34 (implicitní)

Pole "Formát" obsahuje tři ASCII znaky, které specifikují počet datových bitů a paritu použité při odesílání zprávy. Toto pole neplatí, pokud je protokol nastaven na numerické zprávy na pager. Příпустná jsou pouze dvě nastavení:

- 8N1 – 8 datových bitů, žádná parita, jeden stop bit (implicitní)
- 7E1 – 7 datových bitů, sudá parita, jeden stop bit

Formát textové zprávy

Formát textové zprávy definuje formát textových zpráv na pager nebo SMS. Tyto zprávy mohou obsahovat text a vložené proměnné. Textová zpráva je řetězec ASCII znaků, v němž je na prvním místě byte obsahující délku a za ním následují další ASCII znaky. Maximální délka textové zprávy je 120 bytů (včetně bytu délky).

Formát: <Text> <Proměnná> <Text> <Proměnná> ...

Pole "Text" se skládá z ASCII znaků.

Pole "Proměnná" definuje hodnotu vložených dat, které modemový modul čte z lokálního CPU, formátuje a vkládá do zprávy. Znak procenta (%) se používá pro označení začátku a konce pole proměnné. Adresa a pole "Vlevo" jsou odděleny dvojtečkou. Oddělovač mezi poli vlevo a vpravo může být buď tečka, nebo čárka; používá se jako oddělení desetinných míst ve formátované proměnné. Syntaxe pro pole proměnné je:

%Adresa:Formát%Vlevo.Vpravo%

Pole "Adresa" specifikuje adresu, typ dat a velikost vložených dat (např. VD100, VW50, MB20 nebo T10). Přípustné jsou následující typy dat: I, Q, M, S, SM, V, T, C a AI. Přípustné jsou hodnoty byte, word a double word.

Pole "Vlevo" definuje počet číslic, které se mají zobrazit nalevo od desetinné čárky. Tato hodnota musí být dostatečně velká, aby obsáhla předpokládaný rozsah vložené proměnné včetně případného znaménka minus. Pokud je pole "Vlevo" nulové, je hodnota zobrazena s nulou nalevo od desetinné čárky. Platný rozsah pro "Vlevo" je od 0 do 10.

Pole "Vpravo" definuje počet číslic, které se mají zobrazit napravo od desetinné čárky. Nuly napravo od desetinné čárky se vždy zobrazují. Pokud je pole "Vpravo" nulové, je číslo zobrazeno bez desetinné čárky. Platný rozsah pro "Vpravo" je od 0 do 10.

Pole "Formát" specifikuje formát zobrazení vložené proměnné. V poli "Formát" jsou přípustné následující znaky:

- i – integer se znaménkem
- u – integer bez znaménka
- h – hexadecimální
- f – plovoucí desetinná čárka/reálné

Příklad: "Teplota = %VW100:3.1i% Tlak = %VD200:4.3f%"

Formát zprávy pro přenos dat CPU

Přenos dat CPU, ať už z CPU do CPU nebo z CPU do Modbus, se specifikuje pomocí formátu zprávy. Zpráva s přenosem dat CPU je řetězec ASCII znaků, který může specifikovat libovolný počet přesunů dat mezi zařízeními; tento počet je omezen maximální délkou zprávy 120 bytů (119 znaků plus byte délky zprávy). Pro oddělení specifikací přenosu dat může být použit ASCII znak mezery, ale není vyžadován. Všechny specifikace přenosu dat jsou provedeny v rámci jednoho spojení. Přenosy dat jsou prováděny v pořadí definovaném ve zprávě. Pokud je v přenosu dat zjištěna chyba, je spojení se vzdáleným zařízením ukončeno a následující transakce nejsou zpracovány.

Pokud je operace specifikována jako čtení, je počet wordů "Count" načten ze vzdáleného zařízení počínaje Remote_address a pak zapsán do paměti V lokální CPU počínaje Local_address.

Pokud je operace specifikována jako zápis, je počet wordů "Count" načten z lokálního CPU počínaje Local_address a pak zapsán do vzdáleného zařízení počínaje Remote_address.

Formát: <Operace>=<Count>,<Local_address>,<Remote_address>

Pole "Operace" se skládá z jednoho ASCII znaku a definuje typ přenosu.

R – Načíst data ze vzdáleného zařízení
W – Zapsat data do vzdáleného zařízení

Pole "Count" specifikuje počet hodnot wordů, které mají být přesunuty. Platný rozsah pro pole "Count" je 1 až 100 wordů.

Pole Local_address specifikuje adresu pro přenos dat v paměti V lokální CPU (např. VW100).

Pole Remote_address specifikuje adresu pro přesun dat ve vzdáleném zařízení (např. VW500). Tato adresa se vždy specifikuje jako adresa v paměti V, dokonce i když přesun dat probíhá do zařízení Modbus. Pokud je vzdálené zařízení Modbus, konverze mezi adresou v paměti V a adresou Modbus vypadá takto:

Adresa Modbus = 1 + (adresa v paměti V / 2)
Adresa v paměti V = (adresa Modbus – 1) * 2

Příklad: R=20,VW100, VW200 W=50,VW500,VW1000 R=100,VW1000,VW2000

Použití knihovny USS protokolu pro řízení pohonu MicroMaster

Knihovny instrukcí STEP 7 – Micro/WIN usnadňují řízení pohonů MicroMaster, protože obsahují již nakonfigurované podprogramy přerušení a jiné podprogramy, které jsou speciálně navrženy pro použití USS protokolu při komunikaci s pohonem. Pomocí instrukcí USS můžete řídit fyzický pohon a číst, popř. zapisovat jeho parametry.

Tyto instrukce najdete ve složce "Knihovny" instrukčního stromu STEP 7 – Micro/WIN. Když vyberete nějakou instrukci USS, automaticky se k ní přidá jeden nebo více příslušných podprogramů (USS1 až USS7).

Přehled kapitoly

Požadavky použití USS protokolu	312
Výpočet doby, potřebné pro komunikaci s pohonem	313
Použití instrukcí USS	314
Instrukce USS protokolu	315
Vzorové programy USS protokolu	322
Chybové kódy při běhu USS	323
Připojení a nastavení pohonu MicroMaster řady 3	324
Připojení a nastavení pohonu MicroMaster řady 4	327

Požadavky použití USS protokolu

Knihovny instrukcí ve STEP 7–Micro/WIN obsahují 14 podprogramů, 3 podprogramy přerušení a 8 instrukcí, které podporují USS protokol. Instrukce USS používají v automatu S7-200 následující prostředky:

- ☐ Inicializace USS protokolu vyhradí port 0 pro komunikace USS.

Instrukce USS_INIT se používá pro volbu USS nebo PPI pro port 0. (USS se vztahuje na USS protokol pro pohony SIMOTION MicroMaster.) Poté, co zvolíte pro komunikaci s pohony použití USS protokolu, nemůžete port 0 používat pro žádný jiný účel včetně komunikace se STEP 7–Micro/WIN.

Během programování aplikace používající USS protokol byste měli používat CPU 226, CPU 226XM nebo modul EM 277 PROFIBUS–DP s připojením na kartu PROFIBUS CP. Tento druhý komunikační port umožňuje STEP 7–Micro/WIN monitorovat aplikaci během chodu USS protokolu.

- ☐ Instrukce USS ovlivňují všechna místa paměti SM spojená s komunikací Freeport na portu 0.
- ☐ Instrukce USS používají 14 podprogramů a 3 podprogramy přerušení.
- ☐ Díky instrukcím USS se zvětšuje část paměti potřebná pro uživatelský program až o 3450 bytů. V závislosti na konkrétních použitých instrukcích USS mohou podpůrné podprogramy pro tyto instrukce zvýšit režii pro řídicí program nejméně o 2150 bytů až na 3450 bytů.
- ☐ Proměnné pro instrukce USS potřebují 400bytový blok v paměti V. Počáteční adresa tohoto bloku je přidělena uživatelem a je rezervována pro proměnné USS.
- ☐ Některé z instrukcí USS také vyžadují 16bytový komunikační zásobník. Uživatel přiděluje počáteční adresu v paměti V pro tento zásobník jako parametr instrukce. Doporučuje se přiřadit vlastní zásobník každé instrukci USS.
- ☐ Při provádění výpočtů používají instrukce USS akumulátory AC0 až AC3. Ve vašem programu můžete také použít akumulátory, ale hodnoty v akumulátorech budou instrukcemi USS změněny.
- ☐ Instrukce USS není možné používat v podprogramu přerušení.



Tip

Chcete-li změnit činnost portu 0 zpět na PPI, abyste mohli komunikovat se STEP 7–Micro/WIN, použijte další instrukci USS_INIT, která změní přiřazení portu 0.

Můžete také přepnout přepínač režimu na S7-200 na režim STOP. Tím se resetují parametry portu 0. Uvědomte si ale, že zastavení komunikace s pohony zastaví i pohony samotné.

Výpočet doby, potřebné pro komunikaci s pohonem

Komunikace s pohonem je asynchronní s programovým cyklem automatu S7-200. Běžně S7-200 dokončí několik programových cyklů dříve, než je ukončena jedna komunikační transakce. Následující faktory pomáhají při stanovení potřebné délky času: počet připojených pohonů, přenosová rychlost a doba programového cyklu S7-200.

Některé pohony potřebují při použití instrukcí pro přístup k parametrům delší čas. Čas, potřebný pro přístup k parametrům, závisí na typu pohonu a na parametru, ke kterému se přistupuje.

Poté, co instrukce USS_INIT přikáže portu 0 používat USS protokol, provádí S7-200 pravidelné dotazování všech aktivních pohonů v intervalech uvedených v tabulce 11-1. Pro každý pohon musíte nastavit parametr časové prodlevy s přihlédnutím k této skutečnosti.

Tabulka 11-1 Časy komunikace

Přenosová rychlost	Doba mezi dotazováním aktivních pohonů (když není aktivní žádná instrukce pro přístup k parametrům)
1200	240 ms (maximálně) x počet pohonů
2400	130 ms (maximálně) x počet pohonů
4800	75 ms (maximálně) x počet pohonů
9600	50 ms (maximálně) x počet pohonů
19200	35 ms (maximálně) x počet pohonů
38400	30 ms (maximálně) x počet pohonů
57600	25 ms (maximálně) x počet pohonů
115200	25 ms (maximálně) x počet pohonů



Tip

V jednu dobu může být aktivní pouze jedna instrukce USS_RPM_x nebo USS_WPM_x. Výstup "Done" (Dokončeno) každé instrukce musí signalizovat ukončení dříve, než uživatelský program spustí novou instrukci.

Pro každý pohon použijte pouze jednu instrukci USS_CTRL.

Použití instrukcí USS

Pro použití instrukcí USS protokolu v programu programovatelného automatu S7-200 proveďte tyto kroky:

1. Vložte do svého programu instrukci USS_INIT a provádějte ji pouze po dobu jednoho programového cyklu. Instrukci USS_INIT můžete použít buď pro inicializaci, nebo pro změnění parametrů komunikace USS.

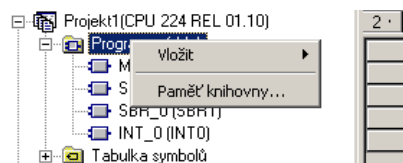
Když vložíte instrukci USS_INIT, přidá se k vašemu programu automaticky několik skrytých podprogramů a podprogramů přerušení.

2. Pro každý aktivní pohon vložte do svého programu pouze jednu instrukci USS_CTRL.

Instrukci USS_RPM_x a USS_WPM_x můžete vložit tolik, kolik potřebujete, ale pouze jedna z nich může být v určitou dobu aktivní.

3. Paměť V pro knihovnu instrukcí přidělíte tak, že kliknete pravým tlačítkem myši (abyste rozbílili menu) na uzel programového bloku v instrukčním stromě.

Aby se zobrazilo dialogové okno "Přidělení paměti pro knihovnu", vyberte "Paměť knihovny".



Obr. 11-1 Přidělení paměti V knihovně instrukcí

4. Nakonfigurujte parametry pohonu tak, aby odpovídaly přenosové rychlosti a adrese použité programem.

5. Zapojte komunikační kabel mezi automatem S7-200 a pohony.

Zajistěte, aby všechna řídicí zařízení, jako je například S7-200, která jsou propojena s pohonem, byla spojena krátkým silným kabelem se stejným uzemněním nebo nulovým bodem jako pohon.

Upozornění

Jestliže propojíte zařízení s různými referenčními potenciály, mohou spojovacím kabelem protékat nežádoucí proudy. Tyto proudy mohou způsobit chyby v komunikaci nebo poškodit zařízení.

Zkontrolujte si, že všechna zařízení, která jsou propojena komunikačním kabelem, mají buď společný referenční potenciál, nebo jsou napěťově izolována, aby se zabránilo průtoku nežádoucích proudů.

Stínění musí být zapojeno na uzemnění rozvaděče nebo k pinu 1 v 9pinovém konektoru. Doporučuje se připojit svorku 2–0V na pohonu MicroMaster k uzemnění rozvaděče.

Instrukce USS protokolu

Instrukce USS_INIT

Instrukce USS_INIT se používá pro povolení a inicializaci nebo pro ukončení komunikace s pohony MicroMaster. Instrukce USS_INIT musí být dokončena bez chyb před použitím jakékoliv jiné instrukce USS. Dříve než může být provedena další instrukce, musí být prováděná instrukce ukončena a nastaven bit "Done".

Tato instrukce je provedena v každém programovém cyklu, když je zapnutý vstup EN.

Instrukci USS_INIT proved'te pro každou změnu ve stavu komunikace pouze jedenkrát. Proto by vstup EN měl být zapojený pulzně přes vyhodnocující náběžnou hranu. Chcete-li změnit parametry inicializace, proved'te novou instrukci USS_INIT.

Parametr "Mode" volí komunikační protokol: vstupní hodnota 1 přiřazuje port 0 USS protokolu a aktivuje protokol, vstupní hodnota 0 přiřazuje port 0 protokolu PPI a blokuje USS protokol.

"Baud" nastavuje přenosovou rychlost na 1200, 2400, 4800, 9600, 19200, 38400, 57600 nebo 115200.

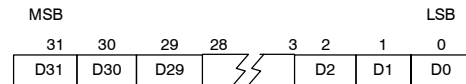
"Active" indikuje, které pohony jsou aktivní. Některé pohony podporují pouze adresy 0 až 30.

Tabulka 11-2 Parametry instrukce USS_INIT

Vstupy/Výstupy	Typ dat	Operandy
Mode	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, konstanta, *VD, *AC, *LD
Baud, Active	DWORD	VD, ID, QD, MD, SD, SMD, LD, konstanta, AC *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Obrázek 11-2 ukazuje popis a formát vstupu aktivního pohonu. Kterýkoliv pohon označený jako "Active" je automaticky dotazován na pozadí tak, aby řídil pohon, shromažďoval stav a zabránil časovému odpojení sériové linky pohonu.

Pokud chcete vypočítat časový interval mezi dotazy na stav, nahlédněte do tabulky 11-1.



D0 Bit aktivity pro pohon 0; 0 – pohon není aktivní, 1 – pohon aktivní
D1 Bit aktivity pro pohon 1; 0 – pohon není aktivní, 1 – pohon aktivní
...

Obr. 11-2 Formát parametru aktivního pohonu

Když je dokončena instrukce USS_INIT, zapne se výstup "Done". Výstupní byte "Error" obsahuje výsledek provádění této instrukce. Tabulka 11-6 definuje chyby, které mohou vyplynout z provedení této instrukce.

Příklad: Podprogram USS_INIT	
Network 1	Network 1 <pre> LD IO.0 EU CALL USS_INIT, 1, 9600, 16#00000001, M0.0, VB10 </pre>

Instrukce USS_CTRL

Instrukce USS_CTRL se používá na řízení aktivního pohonu MicroMaster. Tato instrukce uloží vybrané příkazy do komunikačního zásobníku, který je pak odeslán adresovanému pohonu (parametr "Drive" – pohon), pokud byl tento pohon vybrán v parametru "Active" instrukce USS_INIT.

Každému pohonu by měla být přiřazena pouze jedna instrukce USS_CTRL.

Některé pohony hlásí rychlost pouze jako kladnou hodnotu. Pokud je rychlost záporná, hlásí ji pohon jako kladnou, ale invertuje se bit D_Dir (směr).

Aby byla povolena instrukce USS_CTRL, musí být zapnutý bit EN. Tato instrukce by měla být vždy povolena.

RUN (RUN/STOP) indikuje, zda je pohon zapnutý (1), nebo vypnutý (0). Když je bit RUN zapnutý, dostane pohon MicroMaster příkaz, aby se rozběhl se zadanou rychlostí a směrem. Aby byl pohon uveden do chodu, musí platit následující:

- ☐ Pohon musí být vybrán jako "Active" v USS_INIT.
- ☐ OFF2 a OFF3 musí být nastaveny na 0.
- ☐ "Fault" (Porucha) a "Inhibit" (Blokování) musí být 0.

Když je bit RUN vypnutý, je pohonu MicroMaster poslán příkaz, aby zastavil se zpomalením. Bit OFF2 se používá, aby umožnil pohonu MicroMaster volně doběhnout až do zastavení. Bit OFF3 se používá, jestliže chceme pohonu MicroMaster přikázat, aby zastavil okamžitě.

Bit Resp_R (odpověď přijata) potvrzuje odpověď pohonu. Všechny aktivní pohony jsou cyklicky dotazovány na aktuální informace o stavu pohonu. Pokaždé, když automat S7-200 dostane odpověď od pohonu, zapne se bit Resp_R po dobu jednoho programového cyklu a všechny hodnoty jsou aktualizovány.

Bit F_ACK (potvrzení poruchy) se používá pro potvrzení poruchy pohonu. Pohon vymaže poruchu (Fault), když F_ACK přejde z 0 na 1.

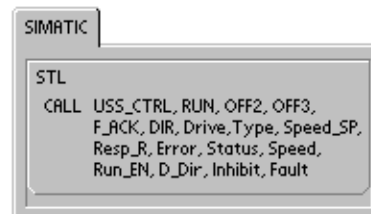
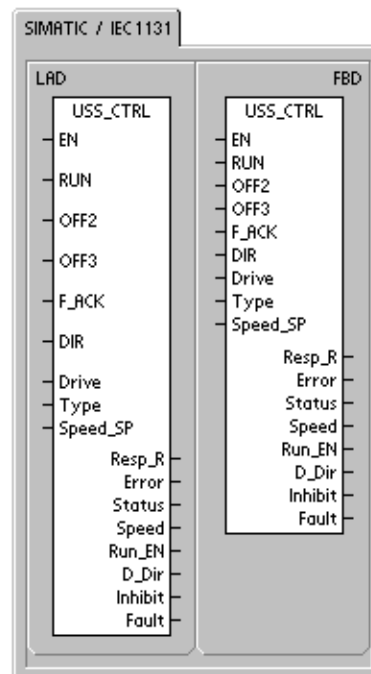
Bit DIR (směr) ukazuje, kterým směrem by se měl pohon pohybovat.

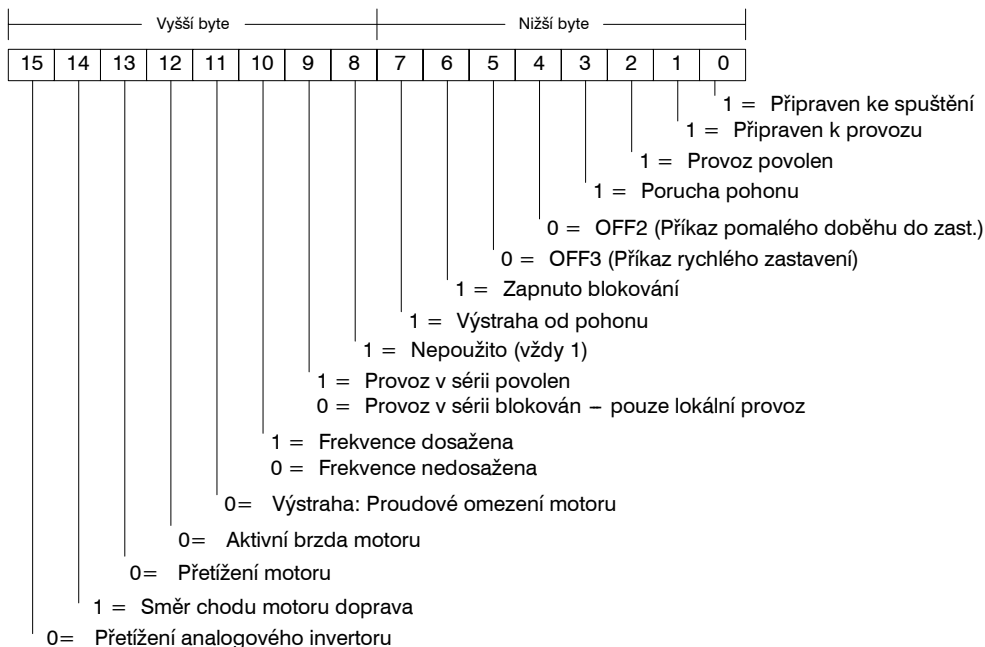
Vstup "Drive" (adresa pohonu) je adresa pohonu MicroMaster, na kterou má být poslán příkaz USS_CTRL. Platné adresy: 0 až 31

Vstup "Type" (typ pohonu) volí typ pohonu. Pro pohon MicroMaster 3 (nebo předchozí verzi) nastavte "Type" na 0. Pro pohon MicroMaster 4 nastavte "Type" na 1.

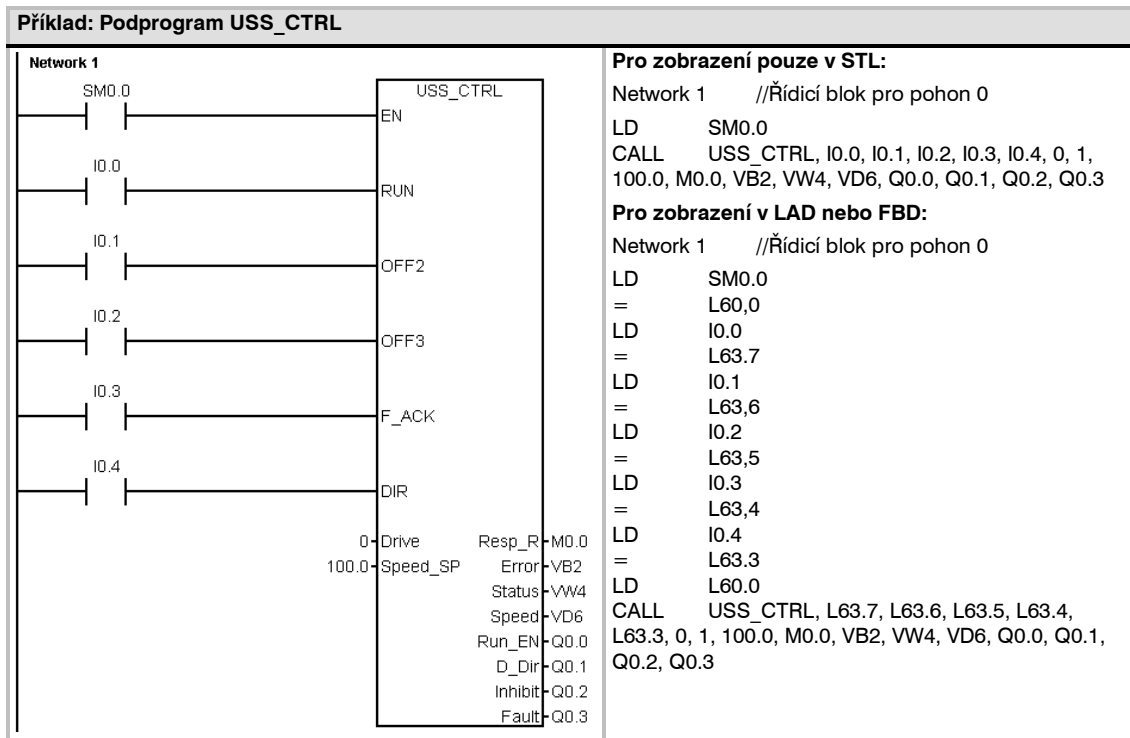
Tabulka 11-3 Parametry instrukce USS_CTRL

Vstupy/Výstupy	Typy dat	Operandy
RUN, OFF 2, OFF 3, F_ACK, DIR	BOOL	I, Q, M, S, SM, T, C, V, L, signálový tok
Resp_R, Run_EN, D_Dir, Inhibit, Fault	BOOL	I, Q, M, S, SM, T, C, V, L
Drive, Type	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, konstanta
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD
Stav	WORD	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
Speed_SP	REAL	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD, konstanta
Speed	REAL	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD





Obr. 11-4 Stavové bity pro standardní stavové word pro MicroMaster 4 a hlavní zpětná vazba



Instrukce USS_RPM_x

V USS protokolu existují tři instrukce pro čtení:

- ☐ Instrukce USS_RPM_W čte parametr word bez znaménka.
- ☐ Instrukce USS_RPM_D čte parametr double word bez znaménka.
- ☐ Instrukce USS_RPM_R čte parametr s plovoucí desetinnou čárkou.

V jednu dobu může být aktivní pouze jedna instrukce pro čtení (USS_RPM_x) nebo zápis (USS_WPM_x).

Transakce USS_RPM_x je ukončena, když pohon MicroMaster potvrdí příjem příkazu nebo když je ohlášena chyba. Programový cyklus se nadále provádí i v době, kdy tento proces čeká na odezvu.

Aby byl povolen přenos požadavku, musí být Bit EN zapnutý a musí zůstat zapnutý do nastavení bitu "Done", které signalizuje ukončení procesu. Například požadavek USS_RPM_x je vysílán pohonu MicroMaster v každém programovém cyklu, kdy je zapnutý vstup XMT_REQ. Proto by měl být vstup XMT_REQ spouštěn pulzem náběžné hrany, aby byl požadavek vyslán pouze jednou v každém kladném přechodu vstupu EN.

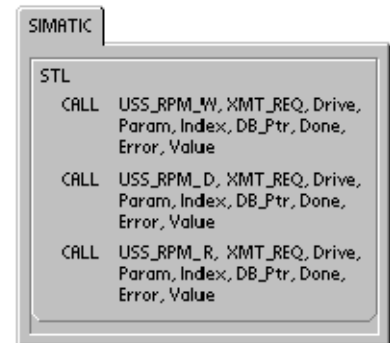
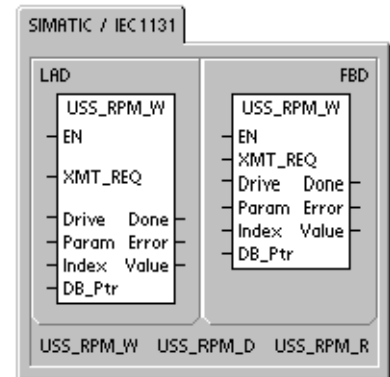
Vstup "Drive" je adresa pohonu MicroMaster, na kterou má být poslán příkaz USS_RPM. Platné adresy jednotlivých pohonů jsou 0 až 31.

"Param" je číslo parametru. "Index" je indexová hodnota parametru, který má být načten. "Value" je vrácená hodnota parametru. Adresa 16bytového zásobníku musí být zadána na vstupu DB_Ptr. Tento zásobník používá instrukce USS_RPM_x na uložení výsledků příkazu daného pohonu MicroMaster.

Po ukončení instrukce USS_RPM_x se zapne výstup "Done" a výstupní byte "Error" a výstup "Value" obsahují výsledky provedení instrukce. Tabulka 11-6 definuje chyby, které mohou nastat při provádění této instrukce. Výstupy "Error" a "Value" nejsou platné, dokud se nezapne výstup "Done".

Tabulka 11-4 Operandy platné pro USS_RPM_x

Vstupy/Výstupy	Typ dat	Operandy
XMT_REQ	BOOL	I, Q, M, S, SM, T, C, V, L, signálový tok podmíněný detekcí náběžné hrany
Drive	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, konstanta
Param, Index	WORD	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, *VD, *AC, *LD, konstanta
DB_Ptr	DWORD	&VB
Value	WORD DWORD, REAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



Instrukce USS_WPM_x

V USS protokolu existují tři instrukce pro zápis:

- ☐ Instrukce USS_WPM_W zapisuje parametr word bez znaménka.
- ☐ Instrukce USS_WPM_D zapisuje parametr double word bez znaménka.
- ☐ Instrukce USS_WPM_R zapisuje parametr s plovoucí desetinnou čárkou.

V jednu dobu může být aktivní pouze jedna instrukce pro čtení (USS_RPM_x) nebo zápis (USS_WPM_x).

Transakce USS_WPM_x je ukončena, když pohon MicroMaster potvrdí příjem příkazu nebo když je ohlášena chyba. Programový cyklus se nadále provádí i v době, kdy tento proces čeká na odezvu.

Aby byl povolen přenos požadavku, musí být Bit EN zapnutý a musí zůstat zapnutý do nastavení bitu "Done", které signalizuje ukončení procesu. Například požadavek USS_WPM_x je vyslán pohonu MicroMaster v každém programovém cyklu, kdy je zapnutý vstup XMT_REQ. Proto by měl být vstup XMT_REQ spouštěn pulzem náběžné hrany, aby byl požadavek vyslán pouze jednou v každém kladném přechodu vstupu EN.

Vstup "Drive" je adresa pohonu MicroMaster, na kterou má být poslán příkaz USS_WPM. Platné adresy jednotlivých pohonů jsou 0 až 31.

"Param" je číslo parametru. "Index" je indexová hodnota parametru, který má být zapsán. "Value" je hodnota parametru, která má být zapsána do RAM v pohonu. U pohonů MicroMaster 3 můžete tuto hodnotu také zapsat do EEPROM pohonu; záleží na tom, jak jste nakonfigurovali P971 (řízení ukládání do paměti EEPROM).

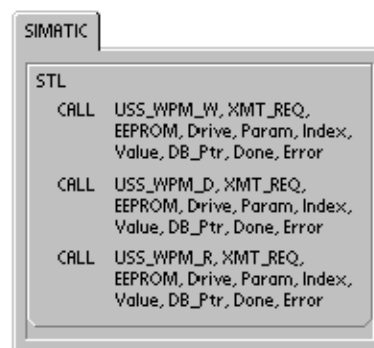
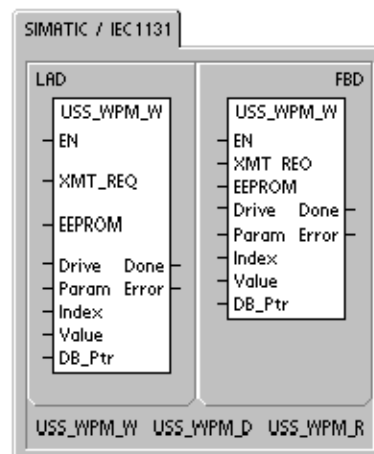
Adresa 16bytového zásobníku musí být zadána na vstupu DB_Ptr. Tento zásobník instrukce USS_WPM_x používá na uložení výsledku provádění příkazu daného pohonu MicroMaster.

Po ukončení instrukce USS_WPM_x se zapne výstup "Done" a výstupní byte "Error" obsahuje výsledky provedení instrukce. Tabulka 11-6 definuje chyby, které mohou vyplynout z provedení této instrukce.

Pokud je vstup EEPROM zapnutý, zapisuje instrukce do RAM i EEPROM pohonu. Když je vstup vypnutý, zapisuje instrukce pouze do RAM pohonu. Abyste mohli tuto instrukci s pohonem MicroMaster 3 používat, musíte zajistit vypnutí tohoto vstupu, protože pohon MicroMaster 3 tuto funkci nepodporuje.

Tabulka 11-5 Operandy platné pro instrukce USS_WPM_x

Vstupy/Výstupy	Typ dat	Operandy
XMT_REQ	BOOL	I, Q, M, S, SM, T, C, V, L, signálový tok podmíněný detekcí náběžné hrany
EEPROM	BOOL	I, Q, M, S, SM, T, C, V, L, signálový tok
Drive	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, konstanta
Param, Index	WORD	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, *VD, *AC, *LD, konstanta
DB_Ptr	DWORD	&VB
Value	WORD DWORD, REAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



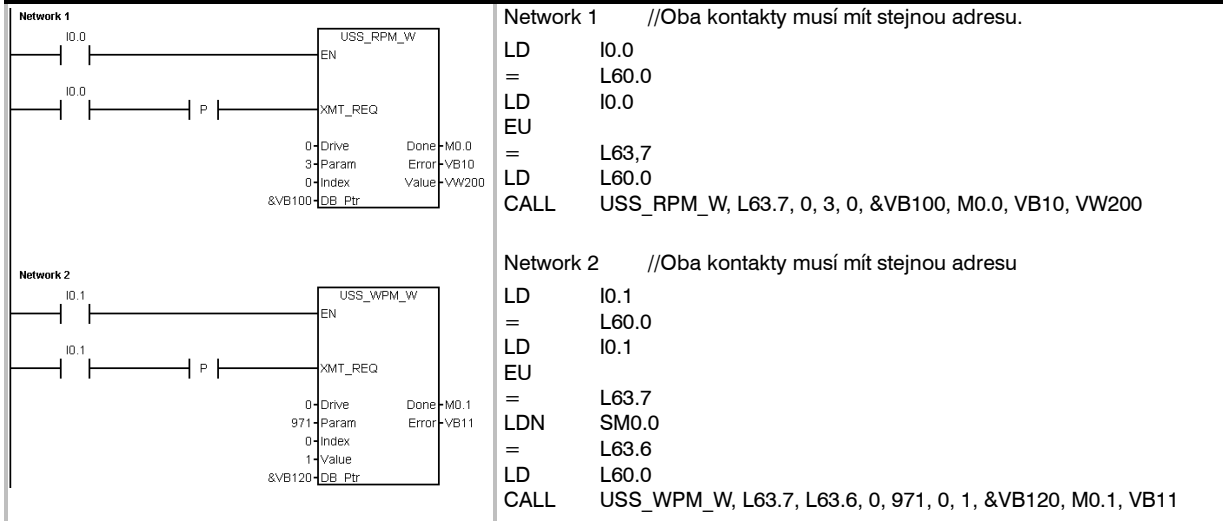
Upozornění

Pokud použijete instrukci USS_WPM_x na aktualizaci sady parametrů uložených v paměti EEPROM pohonu, musíte zajistit, aby nebyl překročen maximální počet cyklů zápisu do EEPROM (přibližně 50.000).

Překročení maximálního počtu zapisovacích cyklů bude mít za následek poškození uložených dat a jejich následnou ztrátu. Počet cyklů čtení není omezen.

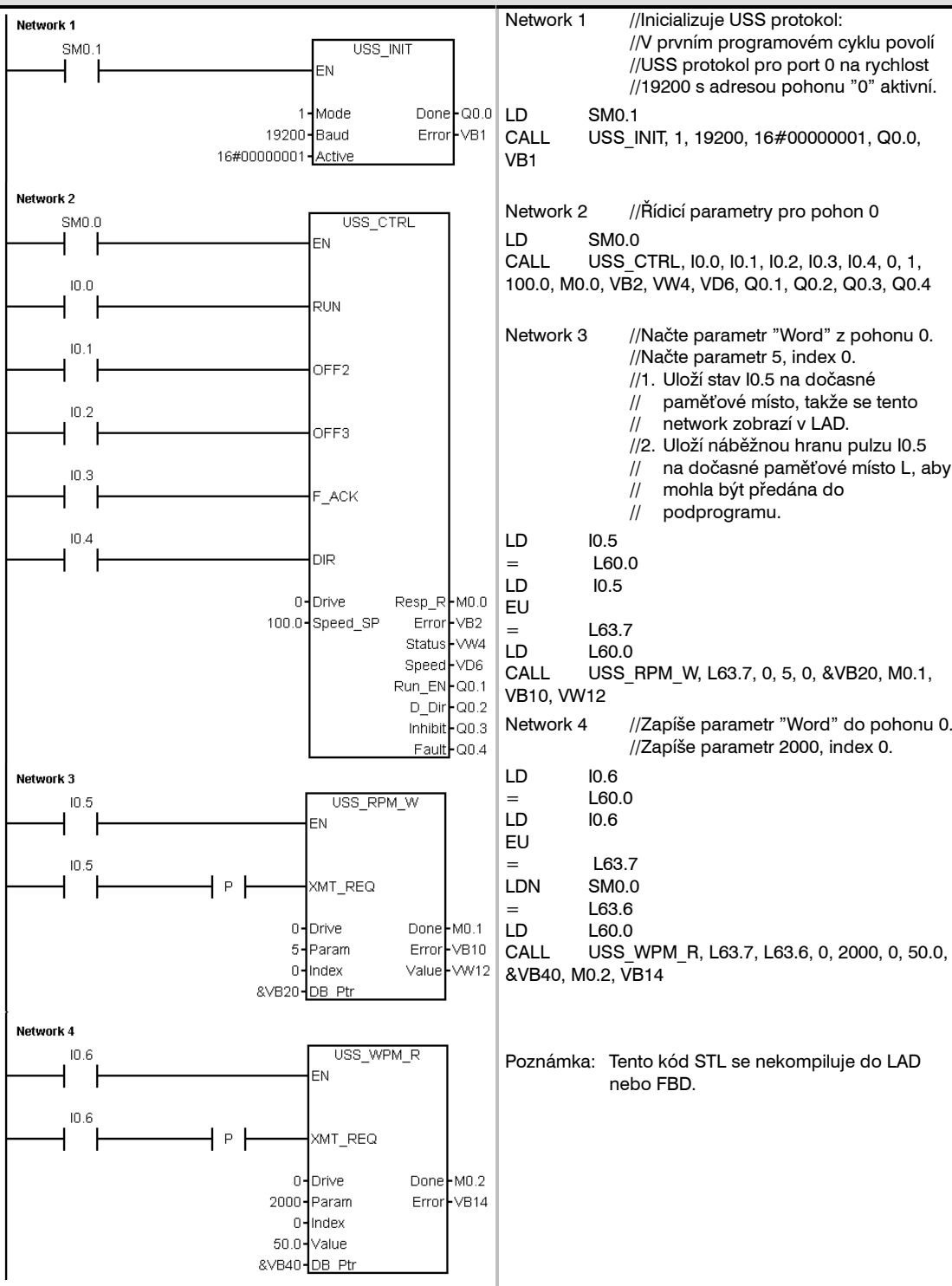
Pokud požadujete častý zápis do parametrů pohonů, měli byste u pohonů MicroMaster 4 nejprve nastavit řídicí parametr paměti EEPROM v pohonu na nulu (u pohonů MicroMaster 3) nebo vypnout vstup EEPROM.

Příklad: USS_RPM_x a USS_WPM_x



Vzorové programy USS protokolu

Příklad: Vzorový program instrukcí USS, který se správně zobrazí v STL



Chybové kódy při běhu USS

Tabulka 11-6 Chybové kódy při běhu instrukcí USS

Chybové kódy	Popis
0	Bez chyby
1	Pohon neodpověděl
2	Byla zjištěna chyba kontrolního součtu v odpovědi pohonu
3	Byla zjištěna chyba parity v odpovědi pohonu
4	Chyba způsobená zásahem uživatelského programu
5	Pokus o nepřipustný příkaz
6	Zadána nepřipustná adresa pohonu
7	Komunikační port nebyl nastaven pro USS protokol
8	Komunikační port je zaneprázdněn zpracováváním instrukce
9	Vstupní rychlost pohonu je mimo rozsah
10	Nesprávná délka odpovědi pohonu
11	Nesprávný první znak v odpovědi pohonu
12	Znak délky v odpovědi pohonu není podporován instrukcemi USS
13	Odpověděl nesprávný pohon
14	Zadaná adresa DB_Ptr je nesprávná
15	Zadané číslo parametru je nesprávné
16	Byl vybrán neplatný protokol
17	USS je aktivní; změna není povolena
18	Byla specifikována neplatná přenosová rychlost
19	Nekomunikuje se: pohon není AKTIVNÍ
20	Parametr nebo hodnota v odpovědi pohonu jsou nesprávné nebo obsahují chybový kód
21	Namísto požadované hodnoty word byla vrácena hodnota double word
22	Namísto požadované hodnoty double word byla vrácena hodnota word

Připojení a nastavení pohonu MicroMaster řady 3

Připojení pohonu MicroMaster 3

Pro spojení automatu S7-200 s pohonem MicroMaster řady 3 (MM3) můžete použít standardní kabel a konektory PROFIBUS. Správnou symetrizaci a zakončení propojovacího kabelu najdete na obrázku 11-5.

Upozornění

Jestliže propojíte zařízení s různými referenčními potenciály, mohou spojovacím kabelem protékat nežádoucí proudy.

Tyto nežádoucí proudy mohou způsobit chyby v komunikaci nebo poškodit zařízení.

Aby se zabránilo průtoku nežádoucích proudů, zkontrolujte, že všechny části zařízení, které chcete propojit komunikačním kabelem, mají buď společný referenční potenciál, nebo jsou napěťově odděleny.

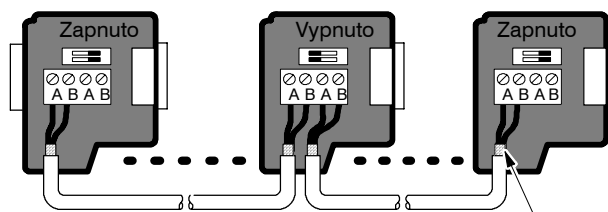
Stínění musí být zapojeno na uzemnění rozvaděče nebo k pinu 1 v 9pinovém konektoru. Doporučuje se připojit svorku 2–0V na pohonu MicroMaster k uzemnění rozvaděče.

Kabel musí být zakončen a symetrizován na obou koncích.

Poloha přepínače = On
Zakončen a symetrizován

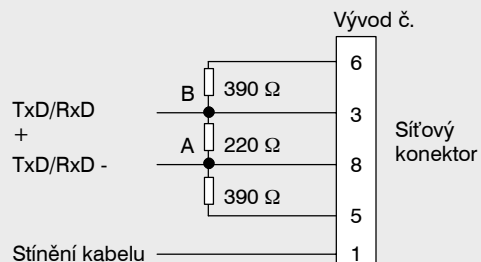
Poloha přepínače = Off
Bez zakončení a symetrizace

Poloha přepínače = On
Zakončen a symetrizován

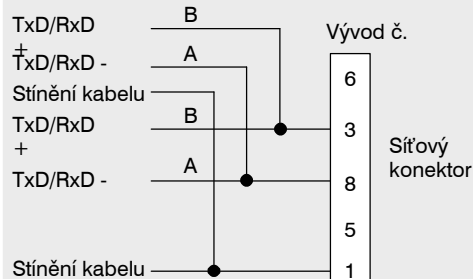


Holé stínění: přibližně 12 mm (1/2 palce) se musí dotýkat kovových vodičů lišt na všech místech.

Poloha přepínače = On:
Zakončen a symetrizován



Poloha přepínače = Off:
Bez zakončení a symetrizace



Obr. 11-5 Symetrizace a zakončení síťového kabelu

Nastavení pohonu MicroMaster 3

Dříve, než pohon připojíte k automatu S7-200, musíte zkontrolovat, že má pohon následující systémové parametry. Pro nastavení parametrů použijte číslcovou klávesnici na pohonu:

1. Resetujte pohon na tovární nastavení. Stiskněte klávesu P. Zobrazí se P000. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí P944. Stiskněte P, abyste vložili parametr.
P944 = 1
2. Povolte přístup ke čtení/zápisu všech parametrů. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí P009. Stiskněte P, abyste vložili parametr.
P009 = 3
3. Zkontrolujte nastavení motoru vašeho pohonu. Nastavení se budou měnit podle použitého motoru. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí nastavení motoru vašeho pohonu. Stiskněte P, abyste vložili parametr.
P081 = Jmenovitá frekvence motoru (Hz)
P082 = Jmenovitá rychlost motoru (RPM)
P083 = Jmenovitý proud motoru (A)
P084 = Jmenovité napětí motoru (V)
P085 = Jmenovitý výkon motoru (kW/HP)
4. Nastavte režim řízení. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí P910. Stiskněte P, abyste vložili parametr.
P910 = 1 Režim řízení "Remote" (vzdáleně)
5. Nastavte přenosovou rychlost sériového rozhraní RS–485. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P092. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí číslo, které odpovídá přenosové rychlosti sériového rozhraní RS–485. Stiskněte P, abyste ji vložili.

P092	3	(1200 baud)
	4	(2400 baud)
	5	(4800 baud)
	6	(9600 baud – implicitní nastavení)
	7	(19200 baud)
6. Vložte adresu zařízení typu slave. Každý pohon (maximálně 31) může být řízen přes sběrnici. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P091. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí adresa zařízení typu slave. Stiskněte P, abyste ji vložili.
P091 = 0 až 31.
7. Doba rozběhu (volitelné). Je to doba v sekundách, kterou motor potřebuje na to, aby zrychlil na maximální frekvenci. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P002. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí doba rozběhu. Stiskněte P, abyste ji vložili.
P002 = 0–650,00
8. Doba doběhu (volitelné). Je to doba v sekundách, kterou motor potřebuje na to, aby zpomalil až do úplného zastavení. Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P003. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí doba doběhu. Stiskněte P, abyste ji vložili.
P003 = 0–650,00

9. Časová prodleva sériové linky. Je to maximální povolený interval mezi dvěma přichozími datovými zprávami. Tato funkce se používá pro vypnutí invertoru v případě poruchy komunikace.

Odpočítávání času začne od příjmu platné datové zprávy. Pokud ve specifikovaném časovém intervalu není přijata další datová zpráva, invertor se vypne a zobrazí se chybový kód F008. Nastavení hodnoty na nulu vypne řízení. Pro výpočet doby mezi dotazy na stav pohonu použijte tabulku 11-1.

Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P093. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí časová prodleva sériové linky. Stiskněte P, abyste ji vložili.

P093 = 0–240 (0 je implicitní nastavení; čas je v sekundách)

10. Nominální systémová žádaná hodnota sériové linky. Tato hodnota se může měnit, ale běžně odpovídá 50 Hz nebo 60 Hz, což je 100% hodnota PV (procesní hodnota) nebo SP (žádaná hodnota). Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P094. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí nominální systémová žádaná hodnota sériové linky. Stiskněte P, abyste ji vložili.

P094 = 0–400,00

11. Kompatibilita s USS (volitelné). Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P095. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí číslo odpovídající kompatibilitě USS. Stiskněte P, abyste ji vložili.

P095 = 0 rozlišení 0,1 Hz (implicitně nastaveno)
 1 rozlišení 0,01 Hz

12. Řízení ukládání do paměti EEPROM (volitelné). Stiskněte klávesu P. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se neobjeví P971. Stiskněte P, abyste vložili parametr. Stiskněte klávesu se šipkou nahoru nebo dolů, dokud se na displeji nezobrazí číslo odpovídající řízení paměti EEPROM. Stiskněte P, abyste je vložili.

P971 = 0 Změny v nastavení parametrů (včetně P971) se při vypnutí napájení ztratí.
 1 Změny nastavení parametrů jsou uchovány i v době, kdy je napájení odpojeno. (implicitní nastavení)

13. Stiskněte P, abyste ukončili režim nastavování parametrů. Vraťte se zpět do pracovního displeje.

Připojení a nastavení pohonu MicroMaster řady 4

Připojení pohonu MicroMaster 4

Chcete-li připojit pohon MicroMaster řady 4 (MM4), zasuněte konce kabelu RS-485 do dvou speciálních svorek bez šroubů určených pro USS. Pro připojení automatu S7-200 je možné použít standardní kabel a konektory PROFIBUS.

Upozornění

Jestliže propojíte zařízení s různými referenčními potenciály, mohou spojovacím kabelem protékat nežádoucí proudy.

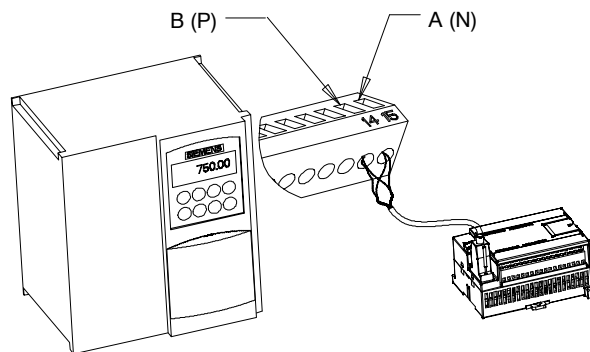
Tyto nežádoucí proudy mohou způsobit chyby v komunikaci nebo poškodit zařízení.

Aby se zabránilo průtoku nežádoucích proudů, zkontrolujte, zda všechny části zařízení, které chcete propojit komunikačním kabelem, mají buď společný referenční potenciál, nebo jsou napětově odděleny.

Stínění musí být zapojeno na uzemnění rozvaděče nebo k pinu 1 v 9pinovém konektoru. Doporučuje se připojit svorku 2–0V na pohonu MicroMaster k uzemnění rozvaděče.

Jak je vidět na obrázku 11-6, dva vodiče na opačném konci kabelu RS-485 musí být zasunuty do svorkovnic pohonu MM4. Chcete-li připojit pohon MM4 kabelem, sejměte kryt pohonu, abyste získali přístup ke svorkovnicím. Podrobnosti o tom, jak demontovat kryt vašeho konkrétního pohonu, najdete v uživatelské příručce MM4.

Připojné body jsou ve svorkovnici označeny čísly. Jestliže použijete konektor PROFIBUS na straně automatu S7-200, připojte kolík A kabelu do svorky 15 pohonu (pro MM420) nebo svorky 30 pohonu (MM440). Kolík B kabelového konektoru zapojte do svorky 14 (MM420) nebo svorky 29 (MM440).



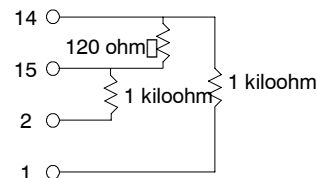
Obr. 11-6 Připojení ke svorkovnici MM420

Pokud je automat koncovým uzlem v síti nebo pokud je spojení dvoubodové (point-to-point), je třeba použít piny A1 a B1 (nikoliv A2 a B2) konektoru, protože umožňují použít zakončení (například u konektoru DP s obj. č. 6ES7 972-0BA40-0X40).

Upozornění

Před připojením napětí k pohonu zkontrolujte, zda jsou kryty správně připojeny zpět na své místo.

Pokud je pohon nakonfigurován jako koncový uzel sítě, musí být zakončovací a symetrizační rezistory také zapojeny k příslušným svorkám svorkovnice. Obrázek 11-7 ukazuje příklad zapojení nutných pro zakončení a symetrizaci pohonu MM4, typ 6SE6420.



Obr. 11-7 Příklad zakončení a symetrizace

Nastavení pohonu MM4

Dříve než pohon připojíte k automatu S7-200, musíte zkontrolovat, zda má pohon následující systémové parametry. Pro nastavení parametrů použijte číslcovou klávesnici na pohonu:

1. Resetujte pohon na tovární nastavení: P0010 = 30
P0970 = 1

Pokud tento krok přeskočíte, zajistěte, aby byly následující parametry nastaveny na tyto hodnoty:
Délka USS PZD: P2012, Index 0 = 2
Délka USS PKW: P2013, Index 0 = 127
2. Povolte čtení/zápis všech parametrů (režim "Expert"): P0003 = 3
3. Zkontrolujte nastavení motoru vašeho pohonu: P0304 = Jmenovité napětí motoru (V)
P0305 = Jmenovitý proud motoru (A)
P0307 = Jmenovitý výkon motoru (W)
P0310 = Jmenovitá frekvence motoru (Hz)
P0311 = Jmenovitá rychlost motoru (RPM)

Nastavení se budou měnit podle použitého motoru.

Abyste mohli nastavovat parametry P304, P305, P307, P310 a P311, musíte nejprve nastavit parametr P010 na 1 (režim rychlého uvedení do provozu). Pokud jste dokončili nastavování parametrů, nastavte parametr P010 na 0. Parametry P304, P305, P307, P310 a P311 mohou být měněny pouze v režimu rychlého uvedení do provozu.

4. Nastavte režim řízení: P0700 Index 0 = 5
5. Nastavte volbu požadované hodnoty frekvence na USS na lince COM: P0700 Index 0 = 5
6. Doba rozběhu (volitelné): P1120 = 0 až 650,00
Je to doba v sekundách, kterou motor potřebuje na to, aby zrychlil na maximální frekvenci.
7. Doba doběhu (volitelné): P1121 = 0 až 650,00
Je to doba v sekundách, kterou motor potřebuje na to, aby zpomalil až do úplného zastavení.
8. Nastavte referenční frekvenci sériové linky: P2000 = 1 až 650 Hz
9. Nastavte normalizaci USS: P2009 Index 0 = 0
10. Nastavte přenosovou rychlost sériového rozhraní RS 485:
P2010 Index 0 = 4 (2400 baud)
5 (4800 baud)
6 (9600 baud)
7 (19200 baud)
8 (38400 baud)
9 (57600 baud)
12 (115200 baud)

11. Vložte adresu zařízení typu slave: P2011 Index 0 = 0 až 31

Každý pohon (maximálně 31) může být řízen přes sběrnici.

12. Nastavte časovou prodlevu sériové linky: P2014 Index 0 = 0 až 65.535 ms
(0 = časová prodleva blokována)

Je to maximální povolený interval mezi dvěma příchozími datovými zprávami. Tato funkce se používá pro vypnutí invertoru v případě poruchy komunikace. Odpočítávání času začne od příjmu platné datové zprávy. Pokud není ve specifikovaném časovém intervalu přijata další datová zpráva, invertor se vypne a zobrazí se chybový kód F0070. Nastavení hodnoty na nulu vypne řízení. Pro výpočet doby mezi dotazy na stav pohonu použijte tabulku 11-1.

13. Přesuňte data z RAM do EEPROM:

P0971 = 1 (Začátek přesunu) Uložte změny nastavení parametrů do EEPROM.

Použití knihovny protokolu Modbus

Knihovny instrukcí STEP 7–Micro/WIN usnadňují komunikaci se zařízeními Modbus master, protože obsahují předkonfigurované podprogramy přerušení i jiné podprogramy, které jsou speciálně navrženy pro komunikaci Modbus. Pomocí instrukcí protokolu Modbus slave můžete nakonfigurovat automat S7-200, aby se choval jako zařízení Modbus RTU slave a komunikoval se zařízeními Modbus master.

Tyto instrukce najdete ve složce "Knihovny" instrukčního stromu STEP 7–Micro/WIN. S těmito novými instrukcemi přimějete automat S7-200, aby se choval jako Modbus slave. Když vyberete instrukci Modbus Slave, je k projektu automaticky připojen jeden nebo více přidružených podprogramů.

Přehled kapitoly

Požadavky použití protokolu Modbus	330
Inicializace a doba provádění protokolu Modbus	330
Adresování Modbus	331
Použití instrukcí protokolu Modbus slave	332
Instrukce protokolu Modbus slave	333

Požadavky použití protokolu Modbus

Instrukce protokolu Modbus slave používají v automatu S7-200 následující prostředky:

- ☐ Inicializace protokolu Modbus slave vyhradí port 0 pro komunikaci protokolu Modbus slave.
Pokud je port 0 použit pro komunikaci protokolu Modbus slave, nemůže být použit pro žádný jiný účel včetně komunikace se STEP 7–Micro/WIN. Instrukce MBUS_INIT řídí přiřazení portu 0 protokolu Modbus slave nebo PPI.
- ☐ Instrukce protokolu Modbus slave ovlivňují všechna místa paměti SM spojená s komunikací Freeport na portu 0.
- ☐ Instrukce protokolu Modbus slave používají 3 podprogramy a 2 podprogramy přerušení.
- ☐ Instrukce protokolu Modbus slave potřebují volnou paměť o velikosti 1857 bytů pro dvě instrukce Modbus slave a podpůrné podprogramy.
- ☐ Proměnné pro instrukce protokolu Modbus slave potřebují 779bytový blok v paměti V. Počáteční adresa tohoto bloku je přidělena uživatelem a je rezervována pro proměnné Modbus.



Tip

Chcete – li změnit režim práce portu 0 zpět na PPI, abyste mohli komunikovat se STEP 7-Micro/WIN, použijte další instrukci MBUS_INIT, která změní přiřazení portu 0.

Můžete také přepnout přepínač režimu na S7-200 na režim STOP. Tím se resetují parametry portu 0.

Inicializace a doba provádění protokolu Modbus

Komunikace Modbus používá pro zajištění integrity přenášených zpráv kontrolu CRC (kontrolu cyklickým kódem). Protokol Modbus slave používá tabulku předem vypočítaných hodnot, aby se zkrátila doba potřebná pro zpracování zprávy. Inicializace této tabulky kódu CRC vyžaduje přibližně 425 milisekund. Inicializace se provádí v podprogramu MBUS_INIT. Běžně je provedena v prvním programovém cyklu uživatelského programu po přechodu do režimu RUN. Uživatel odpovídá za reset obvodu sledování funkce (watchdog) a udržování výstupů v povoleném stavu (pokud to vyžadují rozšiřovací moduly), jestliže doba potřebná pro podprogram MBUS_INIT a jakoukoliv jinou uživatelskou inicializaci překročí o 500 milisekund dobu cyklu obvodu sledování funkce. Obvod sledování funkce výstupního modulu se resetuje zápisem na výstupy modulu. Viz instrukce pro reset obvodu sledování funkce v kapitole 6.

Doba cyklu se prodlužuje, když podprogram MBUS_SLAVE zpracovává požadavek. Protože se většina času spotřebuje na výpočet CRC, je doba cyklu prodloužena o cca 650 mikrosekund na každý byte v požadavku a v odpovědi. Nejdelší požadavek/odpověď (čtení nebo zápis 120 wordů) prodlužuje dobu cyklu přibližně o 165 milisekund.

Adresování Modbus

Adresy Modbus se běžně zapisují jako 5 nebo 6 znakových hodnot, které obsahují typ dat a offset. První jeden nebo dva znaky určují typ dat, poslední čtyři znaky volí příslušnou hodnotu v daném typu dat. Zařízení Modbus master pak mapuje adresy na příslušné funkce. Instrukce Modbus slave podporují následující adresy:

- ☐ 000001 až 000128 jsou digitální výstupy mapované na Q0.0 - Q15.7
- ☐ 010001 až 010128 jsou digitální vstupy mapované na I0.0 - I15.7
- ☐ 030001 až 030032 jsou analogové vstupní registry mapované na AIW0 až AIW62
- ☐ 040001 až 04xxxx jsou úložné registry mapované do paměti V.

Všechny adresy Modbus pracují stejně. Tabulka 12-1 ukazuje mapování adres Modbus na adresy S7-200.

Protokol Modbus slave umožňuje omezit počet vstupů, výstupů, analogových vstupů a úložných registrů (paměť V) přístupných pro Modbus master.

Parametr MaxIQ instrukce MBUS_INIT specifikuje maximální počet digitálních vstupů nebo výstupů (I nebo Q), ke kterým má Modbus master povolený přístup.

Parametr MaxAI instrukce MBUS_INIT specifikuje maximální počet vstupních registrů (AIW), ke kterým má Modbus master povolený přístup.

Parametr MaxHold instrukce MBUS_INIT specifikuje maximální počet úložných registrů (word paměti V), ke kterým má Modbus master povolený přístup.

Více informací o nastavování paměťových omezení pro Modbus slave viz popis instrukce MBUS_INIT.

Tabulka 12-1 Mapování adresy Modbus do S7-200

Adresy Modbus	Adresa S7-200
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
...	...
04xxxx	HoldStart+2 x (xxxx-1)

Konfigurace tabulky symbolů

Když vložíte adresu prvního symbolu, tabulka automaticky vypočítá a přiřadí zbytek symbolů v tabulce.

Tabulce, která zabírá 779 bytů, musíte přiřadit počáteční místo v paměti V. Ujistěte se, že se přiřazení symbolů Modbus slave nepřekrývá s částí paměti V, přiřazenou úložným registrům s parametry HoldStart a MaxHold u instrukce MBUS_INIT. Pokud by došlo k jakémukoliv překrytí paměťových oblastí, instrukce MBUS_INIT ohlásí chybu.

Použití instrukcí protokolu Modbus slave

Pro použití instrukcí protokolu Modbus slave v programu S7-200 proveďte tyto kroky:

1. Vložte do svého programu instrukci MBUS_INIT a provádějte ji pouze po dobu jednoho programového cyklu. Instrukci MBUS_INIT můžete použít buď pro inicializaci, nebo pro změnu parametrů komunikace Modbus.
Když vložíte instrukci MBUS_INIT, přidá se k vašemu programu automaticky několik skrytých podprogramů a podprogramů přerušení.
2. Přidělte počáteční adresu pro 779 za sebou jdoucích bytů paměti V, potřebných pro instrukce protokolu Modbus slave.
3. Do svého programu vložte pouze jednu instrukci MBUS_SLAVE. Tato instrukce je volána při každém programovém cyklu, aby zpracovala všechny přijaté požadavky.
4. Port 0 automatu S7-200 a zařízení Modbus master propojte komunikačním kabelem.

Upozornění

Jestliže propojíte zařízení s různými referenčními potenciály, mohou spojovacím kabelem protékat nežádoucí proudy, které mohou způsobit chyby v komunikaci nebo poškodit zařízení.

Zkontrolujte, zda všechna zařízení, která jsou propojena komunikačním kabelem, mají buď společný referenční potenciál, nebo jsou napěťově oddělena, aby se zabránilo průtoku nežádoucích proudů.

Akumulátory (AC0, AC1, AC2, AC3) jsou využívány instrukcemi Modbus slave a objevují se na seznamu křížových odkazů. Dříve, než je provedena instrukce Modbus slave, jsou hodnoty uloženy v akumulátorech a pak do akumulátorů obnoveny; tím je zajištěno, že jsou v akumulátorech při provádění instrukce Modbus slave zachována všechna uživatelská data.

Instrukce protokolu Modbus slave podporují protokol Modbus RTU. Tyto instrukce využívají pro podporu nejběžnějších funkcí Modbus obslužné instrukce Freeport automatu S7-200. Jsou podporovány následující funkce Modbus:

Tabulka 12-2 Podporované funkce protokolu Modbus slave

Funkce	Popis
1	Načtení stavu jednoho nebo více digitálních výstupů. Funkce 1 hlásí stav zapnuto/vypnuto libovolného počtu výstupů (Q).
2	Načtení stavu jednoho nebo více digitálních vstupů. Funkce 2 hlásí stav zapnuto/vypnuto libovolného počtu vstupů (I).
3	Načtení jednoho nebo více úložných registrů. Funkce 3 hlásí obsah paměti V. Úložné registry jsou hodnoty word pod Modbus a umožňují načtení až 120 wordů v jednom požadavku.
4	Načtení jednoho nebo více vstupních registrů. Funkce 4 hlásí hodnoty analogových vstupů.
5	Zapíše jeden digitální výstup. Funkce 5 nastavuje digitální výstup na zadanou hodnotu. Výstup není vnucený (forced) a program může přepsat hodnotu zapsanou požadavkem Modbus.
6	Zapíše jeden úložný registr. Funkce 6 zapisuje hodnotu jednoho úložného registru do paměti V automatu S7-200.
15	Zapíše více digitálních výstupů. Funkce 15 zapisuje hodnoty více digitálních výstupů do registru obrazu Q automatu S7-200. Počáteční výstupní bod musí začínat na hranici bytu (například Q0.0 nebo Q2.0); počet výstupů, na které se zapisuje, musí být násobkem osmi. To je omezení instrukcí protokolu Modbus slave. Výstupy nejsou vnucené (forced) a program může přepsat hodnoty zapsané požadavkem Modbus.
16	Zapíše více úložných registrů. Funkce 16 zapisuje více úložných registrů do paměti V automatu S7-200. V jednom požadavku může být zapsáno až 120 wordů.

Instrukce protokolu Modbus slave

Instrukce MBUS_INIT

Instrukce MBUS_INIT se používá pro povolení a inicializaci nebo pro ukončení komunikace Modbus. Instrukce MBUS_INIT musí být provedena bez chyb dříve, než je možné použít instrukci MBUS_SLAVE. Dříve než může být provedena další instrukce, musí být prováděná instrukce dokončena a nastaven bit "Done" (Dokončeno).

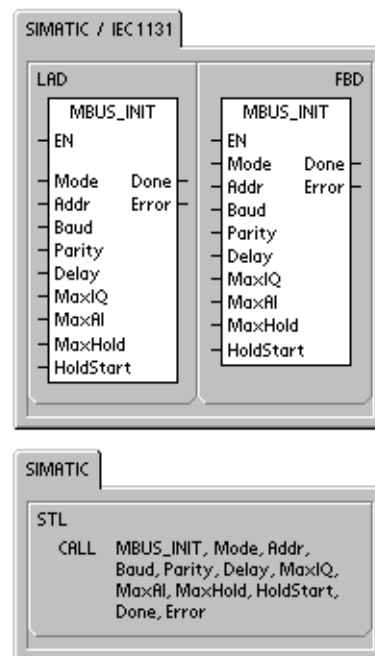
Tato instrukce je prováděna v každém programovém cyklu, pokud je zapnutý vstup EN.

Instrukce MBUS_INIT musí být pro každou změnu stavu komunikace provedena právě jednou. Proto by vstup EN měl být zapojen pomocí bloku vyhodnocujícího náběžnou hranu, nebo by měl být prováděn pouze v prvním programovém cyklu.

Hodnota vstupu "Mode" (režim) volí komunikační protokol: vstupní hodnota 1 přiřazuje portu 0 protokol Modbus a povoluje protokol, vstupní hodnota 0 přiřazuje portu 0 protokol PPI a blokuje protokol Modbus.

Parametr "Baud" (přenosová rychlost) nastavuje přenosovou rychlost na 1200, 2400, 4800, 9600, 19200, 38400, 57600 nebo 115200.

Parametr "Addr" (adresa) nastavuje adresu na hodnoty mezi 1 a 247.



Tabulka 12-3 Parametry instrukce MBUS_INIT

Vstupy/Výstupy	Typ dat	Operandy
Mode, Addr, Parity	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, konstanta, *VD, *AC, *LD
Baud, HoldStart	DWORD	VD, ID, QD, MD, SD, SMD, LD, AC, konstanta, *VD, *AC, *LD
Delay, MaxIQ, MaxAI, MaxHold	WORD	VW, IW, QW, MW, SW, SMW, LW, AC, konstanta, *VD, *AC, *LD
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Parametr "Parity" (parita) je nastaven tak, aby odpovídal paritě zařízení Modbus master. Přijímané hodnoty jsou:

- ☐ 0 bez parity
- ☐ 1 lichá parita
- ☐ 2 sudá parita

Parametr "Delay" (zpoždění) prodlužuje standardní časovou prodlevu na konci zprávy Modbus přidáním specifikovaného počtu milisekund ke standardní časové prodlevě zprávy Modbus. Typickou hodnotou tohoto parametru při provozu v pevně zapojené síti by měla být 0. Pokud používáte modemy s opravou chyb, nastavte zpoždění na hodnotu 50 až 100 milisekund. Pokud používáte rádiové modemy s rozptřeným pásmem, nastavte zpoždění na hodnotu 10 až 100 milisekund. Hodnota "Delay" může být 0 až 32767 milisekund.

Parametr MaxIQ nastavuje počet bodů I a Q, které jsou k dispozici pro adresy Modbus 00xxxx a 01xxxx při hodnotách 0 až 128. Hodnota 0 zablokuje všechny čtení a zápisy na vstupy a výstupy. Navrhovaná hodnota MaxIQ je 128, což umožňuje přístup na všechny body I a Q automatu S7-200.

Parametr MaxAI nastavuje počet vstupních registrů o velikosti word (AI), které jsou k dispozici adrese Modbus 03xxx na hodnotách 0 až 32. Hodnota 0 zablokuje čtení analogových vstupů. Navrhovaná hodnota MaxAI, která umožňuje přístup ke všem analogovým vstupům automatu S7-200, je následující:

- ☐ 0 pro CPU 221
- ☐ 16 pro CPU 222
- ☐ 32 pro CPU 224, 226 a 226XM

Parametr MaxHold nastavuje počet úložných registrů word v paměti V, které jsou k dispozici adrese Modbus 04xxx. Aby mělo například zařízení typu master přístup k 2000 bytům paměti V, nastavte MaxHold na hodnotu 1000 wordů (úložných registrů).

Parametr HoldStart je adresa začátku úložných registrů v paměti V. Tato hodnota se obvykle nastavuje na VB0, takže parametr HoldStart je nastaven na &VB0 (adresa VB0). Jako počáteční adresu úložných registrů je možné specifikovat jiné adresy v paměti V, aby bylo možné použít VB0 jinde v projektu. Zařízení Modbus master má přístup k počtu wordů uvedeném v MaxHold počínaje na HoldStart.

Když je instrukce MBUS_INIT dokončena, zapne se výstup "Done". Výstupní byte "Error" obsahuje výsledek provedení této instrukce. Tabulka 12-5 definuje chyby, které mohou nastat při provádění této instrukce.

Instrukce MBUS_SLAVE

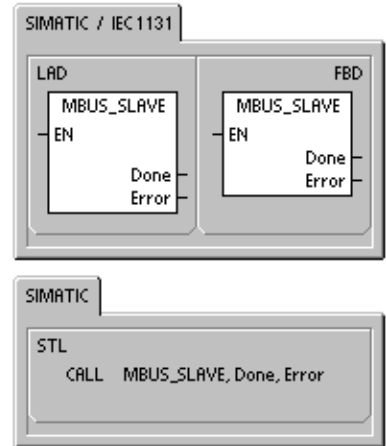
Instrukce MBUS_SLAVE se používá pro zpracování požadavku ze zařízení Modbus master; musí být prováděna v každém programovém cyklu, aby mu umožnila kontrolovat požadavky Modbus a odpovídat na ně.

Tato instrukce je prováděna v každém programovém cyklu, pokud je zapnutý vstup EN.

Instrukce MBUS_SLAVE nemá žádné vstupní parametry.

Výstup "Done" (Dokončeno) je zapnutý, pokud instrukce MBUS_SLAVE odpovídá na požadavek Modbus. Výstup "Done" je vypnutý, pokud se nezpracovává žádný požadavek.

Výstup "Error" (chyba) obsahuje výsledek provedení této instrukce. Tento výstup je platný pouze tehdy, je-li "Done" zapnutý. Pokud je vypnutý, chybový parametr se nemění. Tabulka 12-5 definuje chyby, které mohou nastat při provádění této instrukce.

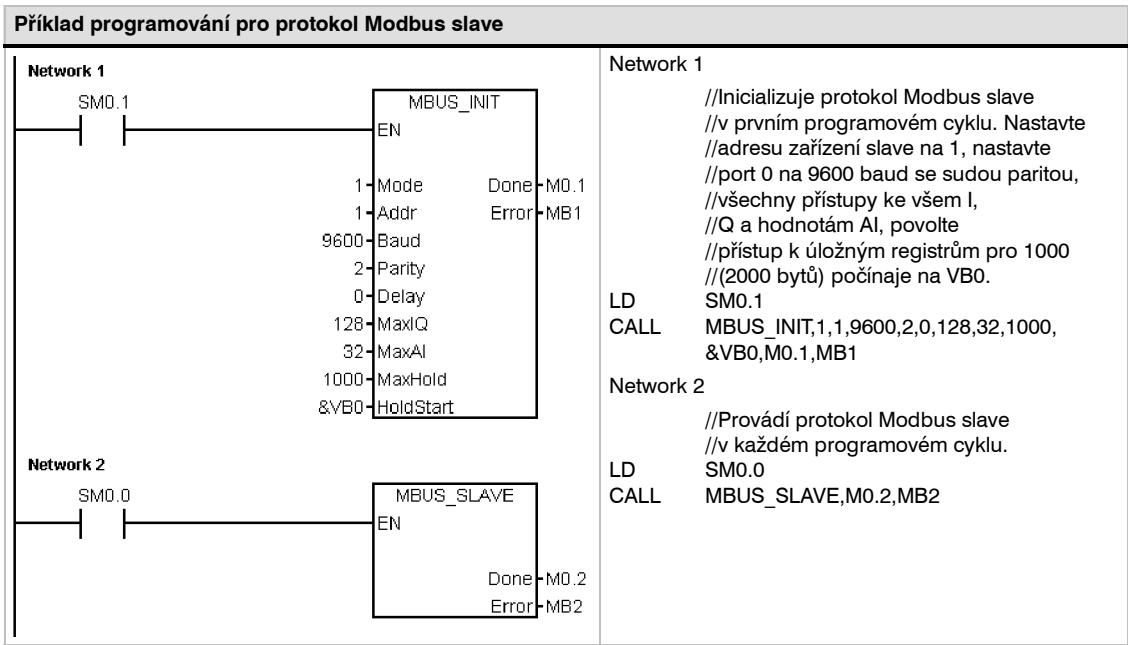


Tabulka 12-4 Parametry instrukce MBUS_SLAVE

Parametr	Typ dat	Operandy
Done	BOOL	I, Q, M, S, SM, T, C, V, L
Error	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Tabulka 12-5 Chybové kódy provádění protokolu Modbus Slave

Chybové kódy	Popis
0	Bez chyby
1	Chyba rozsahu paměti
2	Nepřípustná přenosová rychlost nebo parita
3	Nepřípustná adresa zařízení slave
4	Nepřípustná hodnota parametru Modbus
5	Úložné registry překrývají symboly Modbus Slave
6	Chyba parity příjmu
7	Chyba CRC příjmu
8	Nepřípustný funkční požadavek nebo funkce není podporována
9	Nepřípustná adresa paměti v požadavku
10	Funkce slave není povolena



Technické specifikace



Přehled kapitoly

Obecné technické specifikace	338
Specifikace CPU	340
Specifikace digitálních rozšiřovacích modulů	346
Specifikace analogových rozšiřovacích modulů	351
Specifikace termočlánekových a RTD rozšiřovacích modulů	361
Specifikace modulu EM 277 PROFIBUS–DP	373
Specifikace modemového modulu EM 241	385
Specifikace polohovacího modulu EM 253	387
Specifikace modulu AS–Interface (CP 243–2)	393
Volitelné zásuvné moduly	395
Kabel pro rozšiřovací moduly	395
PC/PPI kabel	396
Simulátory vstupu	398

Obecné technické specifikace

Normy, kterým výrobek odpovídá

Níže uvedené státní a mezinárodní normy byly použity pro stanovení příslušné specifikace vlastností a zkoušek pro skupinu výrobků S7-200. Tabulka A-1 definuje konkrétní zkušební organizace.

- ☐ Směrnice Evropského společenství (CE) pro nízké napětí 73/23/EEC
EN 61131–2: Programovatelné automaty – Požadavky na zařízení
- ☐ Směrnice Evropského společenství (CE) pro EMC (elektromagnetickou kompatibilitu) 89/336/EEC
Norma pro elektromagnetické emise
EN 50081–1: obyvatelstvo, komerční a lehký průmysl
EN 50081–2: průmyslové prostředí
Normy elektromagnetické odolnosti
EN 61000–6–2: průmyslové prostředí
- ☐ Underwriters Laboratories, Inc.: UL 508 (Průmyslová řídicí zařízení)
Registrační číslo E75310
- ☐ Canadian Standards Association (Kanadské sdružení pro normy): CSA C22.2 číslo 142 schváleno
(Zařízení pro řízení procesu)
- ☐ Factory Mutual Research: FM třída I, oddělení 2, nebezpečná místa skupiny A, B, C a D, T4A a třída I, pásmo 2, IIC, T4.

Schválení pro lodní dopravu

V následující tabulce jsou uvedeny námořní agentury a příslušná čísla certifikátů platná v době tisku této příručky.

Chcete-li zjistit nejnovější certifikáty výrobku, kontaktujte místního distributora nebo prodejnu Siemens.

Agentura	Číslo certifikátu
Lloyds Register of Shipping (LRS)	99 / 20018(E1)
American Bureau of Shipping (ABS)	01 – HG20020–PDA
Germanischer Lloyd (GL)	12 045 – 98 HH
Det Norske Veritas (DNV)	A–8071
Bureau Veritas (BV)	09051 / A2 BV
Nippon Kaiji Kyokai (NK)	A–534

Technické specifikace

Všechny CPU S7-200 a rozšiřovací moduly odpovídají technickým specifikacím uvedeným v tabulce A-1.

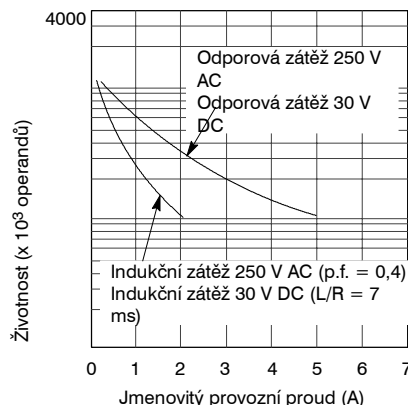
Poznámka

Po zapnutí napájení CPU S7-200 nebo kteréhokoliv digitálního rozšiřovacího modulu je vyslán signál log. 1 digitálním výstupům pod dobu cca 50 mikrosekund. S tím musíte počítat obzvláště v případě, když používáte zařízení, která reagují na krátké pulzy.

Elektrická životnost relé

Typické údaje o vlastnostech relé jsou uvedeny na obrázku A-1. Skutečné vlastnosti se mohou lišit v závislosti na konkrétním použití.

Externí ochranný obvod přizpůsobený zatěži prodlouží životnost kontaktů.



Obr. A-1 Elektrická životnost relé

Tabulka A-1 Technická specifikace

Podmínky prostředí — Doprava a skladování	
IEC 68–2–2, zkouška Bb, suché teplo IEC 68–2–1, zkouška Ab, chlad	–40 °C až +70 °C
IEC 68–2–30, zkouška Db, vlhké teplo	25 °C až 55 °C, vlhkost 95 %
IEC 68–2–31, Převržení	100 mm, 4 pády, nebaleno
IEC 68–2–32, Volný pád	1 m, 5krát, baleno pro dopravu
Podmínky prostředí — Provoz	
Rozsah teplot okolí (Přívod vzduchu 25 mm pod přístrojem)	0 °C až 55 °C při horizontální montáži, 0 °C až 45 °C při vertikální montáži nekondenzující vlhkost 95 %
IEC 68–2–14, zkouška Nb	5 °C až 55 °C, 3 °C/min
IEC 68–2–27 Mechanický ráz	15 G, pulz 11 ms, 6 rázů ve směru všech 3 os
IEC 68–2–6 Sinusové vibrace	Montáž na panel: 0,30 mm od 10 do 57 Hz; 2 G od 57 do 150 Hz Montáž na DIN lištu: 0,15 mm od 10 do 57 Hz; 1 G od 57 do 150 Hz 10 vychýlení v každé ose, 1 oktáva/min
EN 60529, IP20 Mechanická ochrana	Ochrana před nebezpečným dotykovým napětím (prstem), jak bylo odzkoušeno standardními sondami. Vnější ochrana je nutná proti prachu, nečistotě, vodě a cizím předmětům o průměru menším než 12,5 mm.
Elektromagnetická kompatibilita — odolnost podle EN61000–6–2 ¹	
EN 61000–4–2 Elektrostatický výboj	8 kV výboj vzduchem na všechny povrchy a komunikační port 4kV výboj dotykem na všechny exponované vodivé povrchy
EN 61000–4–3 Vyzářované elektromagnetické pole	80 MHz až 1 GHz 10 V/m, 80% modulace signálem 1 kHz
EN 61000–4–4 Rychlé přechodové vysokofrekvenční pulzy	2 kV, 5 kHz s párovou sítí k AC a DC napájení systému 2 kV, 5 kHz s párovou svorkou k digitálním vstupům a výstupům 1 kV, 5 kHz s párovou svorkou ke komunikaci
EN 61000–4–5 Odolnost vůči rázům	Napájení: 2 kV asymetrické, 1 kV symetrické vstupy/výstupy 1 kV symetrické (obvody 24 V DC vyžadují externí ochranu proti rázu)
EN 61000–4–6 Poruchy z vedení	0,15 až 80 MHz 10 V efektivní hodnota 80% amplitudová modulace při 1kHz
EN 61000–4–11 Poklesy napětí, krátká přerušení a kolísání napětí	>95% redukce pro 8,3 ms, 83 ms, 833 ms a 4167 ms
VDE 0160 - Neperiodické přepětí	Při lince 85 V AC, 90° fázový úhel, platí špička 390 V, pulz 1,3 ms Při lince 180 V AC, 90° fázový úhel, platí špička 750 V, pulz 1,3 ms
Elektromagnetická kompatibilita — vedené a vyzářované emise podle EN50081 – 1 ² a – 2	
EN 55011, třída A, skupina 1, vedené ¹ 0,15 MHz až 0,5 MHz 0,5 MHz až 5 MHz 5 MHz až 30 MHz	< 79 dB (μV) - kvazi–špička; < 66 dB (μV) střední hodnota < 73 dB (μV) - kvazi–špička; < 60 dB (μV) střední hodnota < 73 dB (μV) - kvazi–špička; < 60 dB (μV) střední hodnota
EN 55011, třída A, skupina 1, vyzářované ¹ 30 MHz až 230 MHz 230 MHz až 1 GHz	30 dB (μV/m) - kvazi–špička; měřeno ve 30 m 37 dB (μV/m) - kvazi–špička; měřeno ve 30 m
EN 55011, třída B, skupina 1, vedené ² 0,15 až 0,5 MHz 0,5 MHz až 5 MHz 5 MHz až 30 MHz	< 66 dB (μV) - kvazi–špička; klesá s log frekvencí na 56 dB (μV); < 56 dB (μV) - průměrná hodnota klesá s log frekvencí na 46 dB (μV); < 56 dB (μV) - kvazi–špička; < 46 dB (μV) střední hodnota < 60 dB (μV) - kvazi–špička; < 50 dB (μV) střední hodnota
EN 55011, třída B, skupina 1, vyzářované ² 30 MHz až 230 kHz 230 MHz až 1 GHz	30 dB (μV/m) - měřeno v 10 m 37 dB (μV/m) - měřeno v 10 m
Vysokonapěťová izolační zkouška	
obvody 24 V/5 V	500 V AC (hranice optické izolace)
obvody 115/230 V vůči zemi	1.500 V AC
obvody 115/230 V vůči obvodům 115/230 V	1.500 V AC
obvody 230 V vůči obvodům 24 V/5 V	1.500 V AC
obvody 115 V vůči obvodům 24 V/5 V	1.500 V AC

- ¹ Přístroj musí být instalován na uzemněném kovovém rámu; uzemnění S7-200 musí být provedeno přímo na instalační kov. Kabely jsou vedeny po kovových podpěrách.
- ² Přístroj musí být instalován v uzemněném kovovém krytu. Napájecí vedení AC musí být vybaveno filtrem EPCOS B84115–E–A30 nebo ekvivalentním; délka vodiče od filtru k automatu S7-200 může být maximálně 25 cm. Kabeláž napájení 24 V DC a napájení snímače musí být stíněná.

Specifikace CPU

Tabulka A-2 Objednací čísla CPU

Objednací číslo	Model CPU	Napájecí napětí CPU (jmenovité)	Vstupy CPU	Výstupy CPU	Odnímatelná svorkovnice
6ES7 211-0AA22-0XB0	CPU 221	24 V DC	6 x 24 V DC	4 x 24 V DC	Ne
6ES7 211-0BA22-0XB0	CPU 221	120 až 240 V AC	6 x 24 V DC	4 x Relé	Ne
6ES7 212-1AB22-0XB0	CPU 222	24 V DC	8 x 24 V DC	6 x 24 V DC	Ne
6ES7 212-1BB22-0XB0	CPU 222	120 až 240 V AC	8 x 24 V DC	6 x Relé	Ne
6ES7 214-1AD22-0XB0	CPU 224	24 V DC	14 x 24 V DC	10 x 24 V DC	Ano
6ES7 214-1BD22-0XB0	CPU 224	120 až 240 V AC	14 x 24 V DC	10 x Relé	Ano
6ES7 216-2AD22-0XB0	CPU 226	24 V DC	24 x 24 V DC	16 x 24 V DC	Ano
6ES7 216-2BD22-0XB0	CPU 226	120 až 240 V AC	24 x 24 V DC	16 x Relé	Ano
6ES7 216-2AF22-0XB0	CPU 226XM	24 V DC	24 x 24 V DC	16 x 24 V DC	Ano
6ES7 216-2BF22-0XB0	CPU 226XM	120 až 240 V AC	24 x 24 V DC	16 x Relé	Ano

Tabulka A-3 Obecná specifikace CPU

Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Ztráty	K dispozici	
					+5 V DC	+24 V DC
6ES7 211-0AA22-0XB0	CPU 221 DC/DC/DC 6 vstupů/ 4 výstupy	90 x 80 x 62	270 g	3 W	0 mA	180 mA
6ES7 211-0BA22-0XB0	CPU 221 AC/DC/relé 6 vstupů/ 4 relé	90 x 80 x 62	310 g	6 W	0 mA	180 mA
6ES7 212-1AB22-0XB0	CPU 222 DC/DC/DC 8 vstupů/ 6 výstupy	90 x 80 x 62	270 g	5 W	340 mA	180 mA
6ES7 212-1BB22-0XB0	CPU 222 AC/DC/relé 8 vstupů/ 6 relé	90 x 80 x 62	310 g	7 W	340 mA	180 mA
6ES7 214-1AD22-0XB0	CPU 224 DC/DC/DC 14 vstupů/ 10 výstupy	120,5 x 80 x 62	360 g	7 W	660 mA	280 mA
6ES7 214-1BD22-0XB0	CPU 224 AC/DC/relé 14 vstupů/ 10 relé	120,5 x 80 x 62	410 g	10 W	660 mA	280 mA
6ES7 216-2AD22-0XB0	CPU 226 DC/DC/DC 24 vstupů/ 16 výstupy	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BD22-0XB0	CPU 226 AC/DC/relé 24 vstupů/ 16 relé	196 x 80 x 62	660 g	17 W	1000 mA	400 mA
6ES7 216-2AF22-0XB0	CPU 226XM DC/DC/DC 24 vstupů/16 výstupů	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BF22-0XB0	CPU 226XM AC/DC/relé 24 vstupů/16 relé	196 x 80 x 62	660 g	17 W	1000 mA	400 mA

Tabulka A-4 Specifikace CPU

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Paměť					
Velikost uživ. programu (EEPROM)	2048 word		4096 word	4096 word	8192 word
Uživatelská data (EEPROM)	1024 word (trvale uložena)		2560 word (trvale uložena)	2560 word (trvale uložena)	5120 word (trvale uložena)
Záloha (velkokapacitní kondenzátor) (volitelně baterie)	běžně 50 hodin (8 hodin min. při 40°C) běžně 200 dnů		běžně 190 hodin (120 hodin min. při 40°C) běžně 200 dnů		
Vstupy a výstupy					
Integrované digitální I/O	6 vstupů/4 výstupy	8 vstupů/6 výstupů	14 vstupů/10 výstupů	24 vstupů/16 výstupů	
Velikost obrazu digitálních I/O	256 (128 vstupů/128 výstupů)				
Velikost obrazu analogových I/O	Nemá	32 (16 vst./16 výst.)	64 (32 vstupů/32 výstupů)		
Max. počet rozšiřovacích modulů	Žádný	2 moduly	7 modulů		
Max. počet inteligentních modulů	Žádný	2 moduly	7 modulů		
Vstupy pro zachycení pulzů	6	8	14		
Vysokorychlostní čítače Jednofázové Dvoufázové	celkem 4 čítače 4 při 30 kHz 2 při 20 kHz		celkem 6 čítačů 6 při 30 kHz 4 při 20 kHz		
Pulzní výstupy	2 při 20 kHz (pouze DC výstupy)				

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Obecné					
Časovače	256 časovačů celkem; 4 časovače (1 ms); 16 časovačů (10 ms); 236 časovačů (100 ms)				
Čítače	256 (zálohovaných velkokapacitním kondenzátorem nebo baterií)				
Bits vnitřní paměti Ukládány při výpadku napájení	256 (zálohovaných velkokapacitním kondenzátorem nebo baterií) 112 (uložených do EEPROM)				
Časovaná přerušení	2 s rozlišením 1 ms				
Přerušení od náběž./sestupné hrany	4 od náběžné a 4 od sestupné				
Analogové potenciometry	1 s rozlišením 8 bitů		2 s rozlišením 8 bitů		
Rychlost Booleovských instrukcí	0.37 μs na instrukci				
Hodiny	Zásuvný modul		Integrované		
Volitelné zásuvné moduly	Paměťový, bateriový a hodiny		Paměťový a bateriový		
Zabudovaná komunikace					
Porty	1 port RS – 485			2 porty RS – 485	
Přenosové rychlosti PPI, DP/T	9,6; 19,2; 187,5 kbaud				
Přenosové rychlosti Freeport	1,2 kbaud až 115,2 kbaud				
Max. délka kabelu na segment	S napětově odděleným repeaterem: 1000 m až do 187,5 kbaud, 1200 m až do 38,4 kbaud Bez napětově odděleného repeateru: 50 m				
Max. počet stanic	32 na segment, 126 na síť				
Max. počet zařízení typu master	32				
Síť s rovnoc. uzly (režim PPI Master)	Ano (NETR/NETW)				
MPI připojení	celkem 4, 2 vyhrazeny (1 pro PG a 1 pro OP)				

Tabulka A-5 Specifikace napájení CPU

DC			AC	
Vstupní napájení				
Vstupní napětí	20,4 až 28,8 V DC		85 až 264 V AC (47 až 63 Hz)	
Vstupní proud	CPU pouze při 24 V DC	Max. zatížení při 24 V DC	Pouze CPU	Max. zatížení
CPU 221	80 mA	450 mA	30/15 mA při 120/240 V AC	120/60 mA při 120/240 V AC
CPU 222	85 mA	500 mA	40/20 mA při 120/240 V AC	140/70 mA při 120/240 V AC
CPU 224	110 mA	700 mA	60/30 mA při 120/240 V AC	200/100 mA při 120/240 V AC
CPU 226/CPU 226XM	150 mA	1050 mA	80/40 mA při 120/240 V AC	320/160 mA při 120/240 V AC
Nárazový proud	10 A při 28,8 V DC		20 A při 264 V AC	
Napětové oddělení (okolí od logiky)	Není odděleno		1500 V AC	
Doba uchování (ztráta napájení)	10 ms při 24 V DC		20/80 ms při 120/240 V AC	
Pojistka (nevyměňuje se)	3 A, 250 V pomalá		2 A, 250 V pomalá	
Napájení snímačů 24 V DC				
Napětí snímačů	L+ minus 5 V		20,4 až 28,8 V DC	
Proudové omezení	Vrchol 1,5 A, tepelná nedestruktivní mez			
Zvlnění / rušení	Odvozeno od vstupního napájení		Méně než 1 V špička-špička	
Napětové odd. (snímač od logiky)	Není odděleno			

Tabulka A-6 Specifikace vstupů CPU

Obecné	Vstup 24 V DC
Typ	Aktivní/pasivní (pasivní IEC typ 1)
Jmenovité napětí	Typické 24 V DC při 4 mA
Max. trvalé dovolené napětí	30 V DC
Rázové napětí	35 V DC po 0,5 s
Logická 1 (min.)	15 V DC při 2,5 mA
Logická 0 (max.)	5 V DC při 1 mA
Zpoždění vstupu	Volitelné (0,2 až 12,8 ms) CPU 226, CPU 226XM: vstupy I1.6 až I2.7 mají pevné zpoždění (4,5 ms)

Tabulka A-6 Specifikace vstupů CPU, pokračování

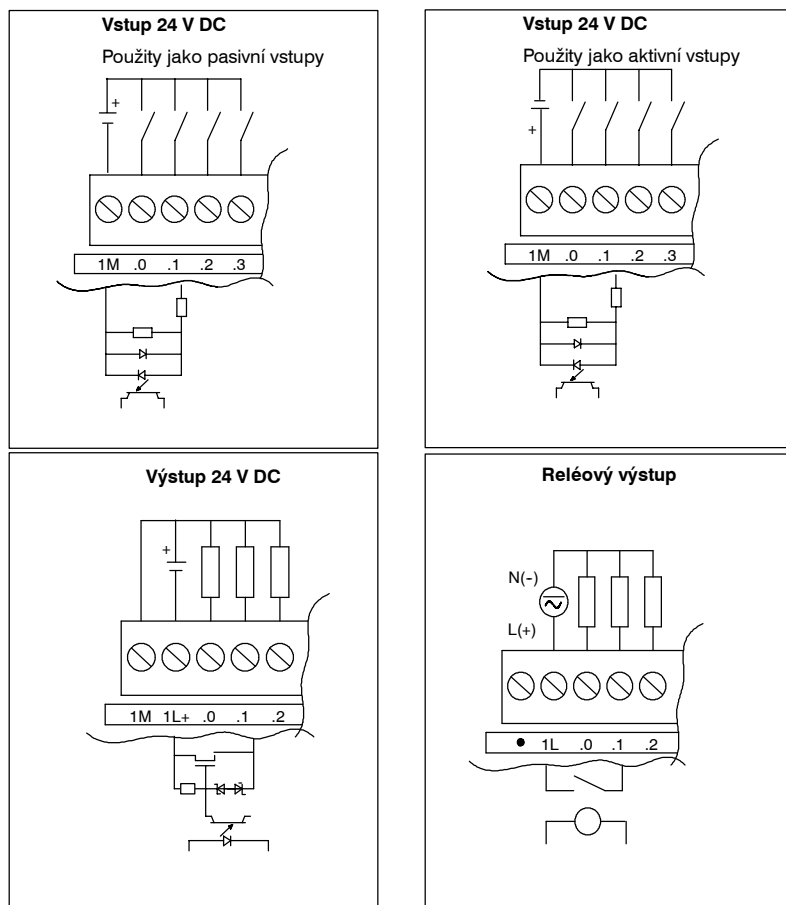
Obecné	Vstup 24 V DC	
Připojení dvou vodičového snímače přiblížení (Bero) Povolený svodový proud (max.)	1 mA	
Napěťové oddělení (okolí od logiky) Optické (galvanické) Oddělení po skupinách	Ano 500 V AC po dobu 1 minuty Viz schéma zapojení	
Vysokorychlostní čítač (max.)	Jednofázový	Dvoufázový
Logická 1 = 15 až 30 V DC	20 kHz	10 kHz
Logická 1 = 15 až 26 V DC	30 kHz	20 kHz
Současné sepnuté vstupy	Všechny při 55 °C	
Délka kabelu (max.) Stíněný Nestíněný	500 m normální vstupy, 50 m vstupy HSC 300 m normální vstupy	

Tabulka A-7 Specifikace výstupů CPU

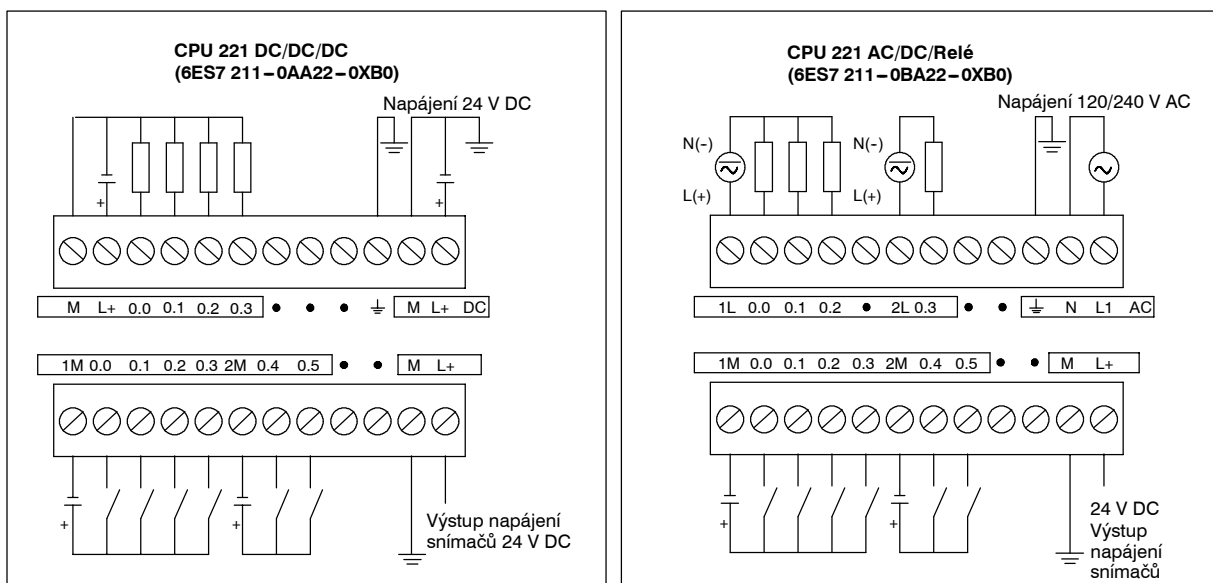
Obecné	Výstup 24 V DC	Reléový výstup
Typ	Polovodičový MOSFET ¹	Kontakt (suchý)
Jmenovité napětí	24 V DC	24 V DC nebo 250 V AC
Rozsah napětí	20,4 až 28,8 V DC	5 až 30 V DC nebo 5 až 250 V AC
Nárazový proud (max.)	8 A po dobu 100 ms	7 A při sepnutých kontaktech
Logická 1 (min.)	20 V DC při maximálním proudu	–
Logická 0 (max.)	0,1 V DC při zatížení 10 K Ω	–
Jmenovitý proud na výstup (max.)	0,75 A	2,0 A
Jmenovitý proud společný (max.)	6 A	10 A
Svodový proud (max.)	10 μ A	–
Zatěž (max.)	5 W	30 W DC; 200 W AC
Induktivní napětí na svorkách	L+ minus 48 V DC, rozptyl 1 W	–
Odpor v sepnutém stavu (kontakt)	0,3 Ω max.	0,2 Ω (maximální u nového)
Napěťové oddělení Optické (galvanické, okolí od logiky) Logika od kontaktu Kontakt od kontaktu Odpor (logika od kontaktu) Oddělení po skupinách	500 V AC po dobu 1 minuty – – – Viz schéma zapojení	– 1500 V AC po dobu 1 minuty 750 V AC po dobu 1 minuty 100 M Ω Viz schéma zapojení
Zpoždění vypnutí, zapnutí/zapnutí vypnutí (max.) Zapnutí (max.)	2/10 μ s (Q0.0 a Q0.1) 15/100 μ s (všechny ostatní) –	– 10 ms
Opakovací kmitočet impulzů (max.) Q0.0 a Q0.1	20 kHz	1 Hz
Mechanická životnost v cyklech	–	10.000.000 (bez zatížení)
Životnost kontaktů	–	100.000 (jmenovitá zátěž)
Současné zapnuté výstupy	Všechny při 55° C	Všechny při 55 °C
Paralelní spojení dvou výstupů	Ano	Není
Délka kabelu (max.) Stíněný Nestíněný	500 m 150 m	500 m 150 m

¹ Když se zapne napájení CPU S7-200 nebo kteréhokoliv digitálního rozšiřovacího modulu, vysílá signál log. 1 digitálním výstupům pod dobu cca 50 mikrosekund. S tím musíte počítat obzvláště tehdy je-li používáte zařízení, která reagují na krátké pulzy.

Schémata zapojení

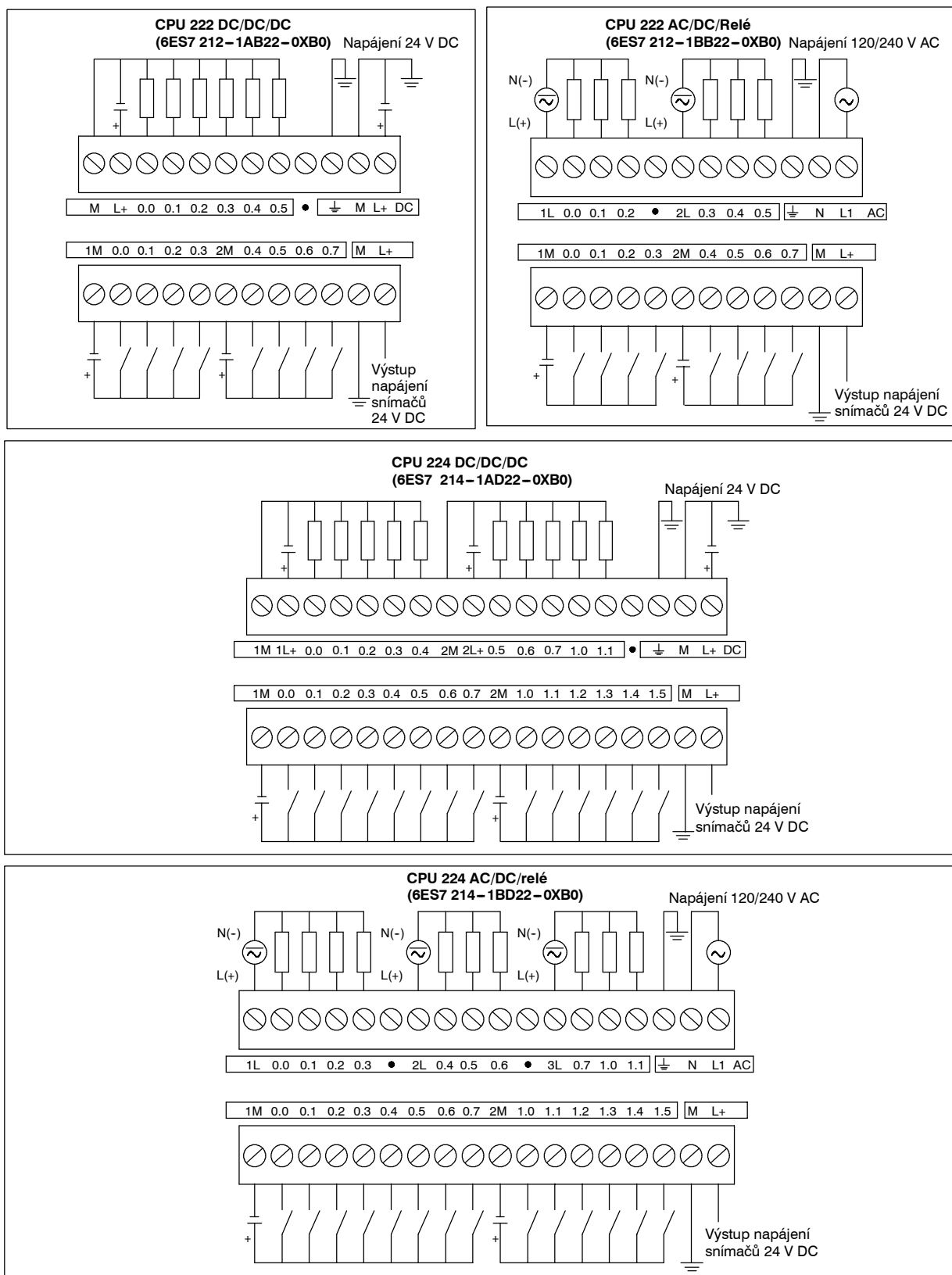


Obr. A-2 Vstupy a výstupy CPU

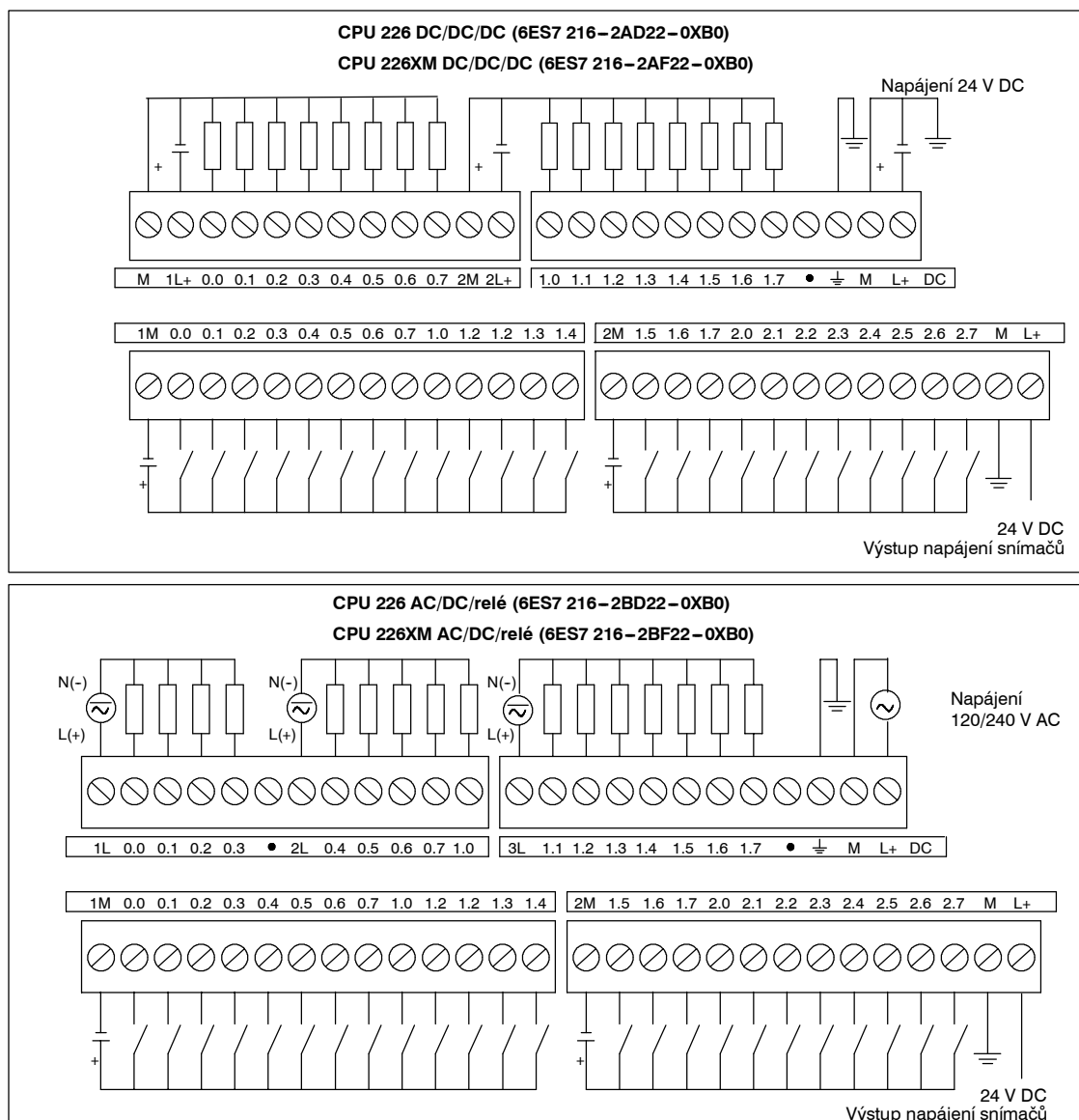


Obr. A-3 Schéma zapojení CPU 221

A



Obr. A-4 Schéma zapojení CPU 222 a CPU 224



Obr. A-5 Schéma zapojení CPU 226 a CPU 226XM

A

Tabulka A-8 Rozmístění vývodů v komunikačním portu automatu S7-200

Konektor	Číslo vývodu	PROFIBUS Signal	Port 0/Port 1
	1	Stínění	Uzemnění rozváděče
	2	24 V zpětný	Logický společný
	3	RS-485 signál B	RS-485 signál B
	4	Výzva k vysílání	RTS (TTL)
	5	5 V zpětný	Logický společný
	6	+5 V	+5 V, 100 Ω sériový odpor
	7	+24 V	+24 V
	8	RS-485 signál A	RS-485 signál A
	9	Nepoužito	10bitová volba protokolu (vstup)
	Pouzdro konektoru	Stínění	Uzemnění rozváděče

Specifikace digitálních rozšiřovacích modulů

Tabulka A-9 Objednací čísla digitálních rozšiřovacích modulů

Objednací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Snímatelný konektor
6ES7 221-1BF22-0XA0	EM 221 digitální vstup 8 x 24 V DC	8 x 24 V DC	–	Ano
6ES7 221-1EF22-0XA0	EM 221 digitální vstup 8 x AC 120/230 V	8 x AC 120/230 V	–	Ano
6ES7 222-1BF22-0XA0	EM 222 digitální výstup 8 x 24 V DC	–	8 x 24 V DC	Ano
6ES7 222-1HF22-0XA0	EM 222 digitální výstup 8 x relé	–	8 x relé	Ano
6ES7 222-1EF22-0XA0	EM 222 digitální výstup 8 x AC 120/230 V	–	8 x AC 120/230 V	Ano
6ES7 223-1BF22-0XA0	EM 223 24 V DC digitální kombinace 4 vstupy/4 výstupy	4 x 24 V DC	4 x 24 V DC	Ano
6ES7 223-1HF22-0XA0	EM 223 24 V DC digitální kombinace 4 vstupy/4 reléových výstupů	4 x 24 V DC	4 x relé	Ano
6ES7 223-1BH22-0XA0	EM 223 24 V DC digitální kombinace 8 vstupů/8 výstupů	8 x 24 V DC	8 x 24 V DC	Ano
6ES7 223-1PH22-0XA0	EM 223 24 V DC digitální kombinace 8 vstupů/8 reléových výstupů	8 x 24 V DC	8 x relé	Ano
6ES7 223-1BL22-0XA0	EM 223 24 V DC digitální kombinace 16 vstupů/16 výstupů	16 x 24 V DC	16 x 24 V DC	Ano
6ES7 223-1PL22-0XA0	EM 223 24 V DC digitální kombinace 16 vstupů/16 reléových výstupů	16 x 24 V DC	16 x relé	Ano

Tabulka A-10 Obecné specifikace digitálních rozšiřovacích modulů

Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Ztráty	Požadavky V DC	
					+5 V DC	+24 V DC
6ES7 221-1BF22-0XA0	EM 221 DI 8 x 24 V DC	46 x 80 x 62	150 g	2 W	30 mA	–
6ES7 221-1EF22-0XA0	EM 221 DI 8 x AC 120/230 V	71,2 x 80 x 62	160 g	3 W	30 mA	–
6ES7 222-1BF22-0XA0	EM 222 DO 8 x 24 V DC	46 x 80 x 62	150 g	2 W	50 mA	–
6ES7 222-1HF22-0XA0	EM 222 DO 8 x relé	46 x 80 x 62	170 g	2 W	40 mA	ZAP: 9 mA/výstup, 20,4 až 28,8 V DC
6ES7 222-1EF22-0XA0	EM 222 DO 8 x AC 120/230 V	71,2 x 80 x 62	165 g	4 W	110 mA	–
6ES7 223-1BF22-0XA0	EM 223 24 V DC 4 vst./4 výst.	46 x 80 x 62	160 g	2 W	40 mA	–
6ES7 223-1HF22-0XA0	EM 223 24 V DC 4 vst./4 relé	46 x 80 x 62	170 g	2 W	40 mA	ZAP: 9 mA/výstup, 20,4 až 28,8 V DC
6ES7 223-1BH22-0XA0	EM 223 24 V DC 8 vst./8 výst.	71,2 x 80 x 62	200 g	3 W	80 mA	–
6ES7 223-1PH22-0XA0	EM 223 24 V DC 8 vst./8 relé	71,2 x 80 x 62	300 g	3 W	80 mA	ZAP: 9 mA/výstup, 20,4 až 28,8 V DC
6ES7 223-1BL22-0XA0	EM 223 24 V DC 16 vst./16 výst.	137,3 x 80 x 62	360 g	6 W	160 mA	–
6ES7 223-1PL22-0XA0	EM 223 24 V DC 16 vst./16 relé	137,3 x 80 x 62	400 g	6 W	150 mA	ZAP: 9 mA/výstup, 20,4 až 28,8 V DC

Tabulka A-11 Specifikace vstupů digitálních rozšiřovacích modulů

Obecné	Vstup 24 V DC	Vstup 120/230 V AC (47 až 63 Hz)
Typ	Aktivní/pasivní (pasivní IEC typ 1)	IEC typ I
Jmenovité napětí	24 V DC při 4 mA	120 V AC při 6 mA nebo 230 V AC při 9 mA
Maximální trvalé dovolené napětí	30 V DC	264 V AC
Rázové napětí (max.)	35 V DC po 0,5 s	–
Logická 1 (min.)	15 V DC při 2,5 mA	79 V AC při 2,5 mA
Logická 0 (max.)	5 V DC při 1 mA	20 V AC nebo 1 mA AC
Zpoždění na vstupu (max.)	4,5 ms	15 ms
Připojení 2vodič. sním. přiblížení Povolený svodový proud (max.)	1 mA	1 mA AC
Napěťové oddělení Optické (galvanické, pole od logiky) Oddělení po skupinách	500 V AC po dobu 1 minuty Viz schéma zapojení	1500 V AC po dobu 1 minuty 1 bod

Tabulka A-11 Specifikace vstupů digitálních rozšiřovacích modulů, pokračování

Obecné	Vstup 24 V DC	Vstup 120/230 V AC (47 až 63 Hz)
Současné sepnuté vstupy	Všechny při 55 °C	Všechny při 55 °C
Délka kabelu (max.)		
Stíněný	500 m	500 m
Nestíněný	300 m	300 m

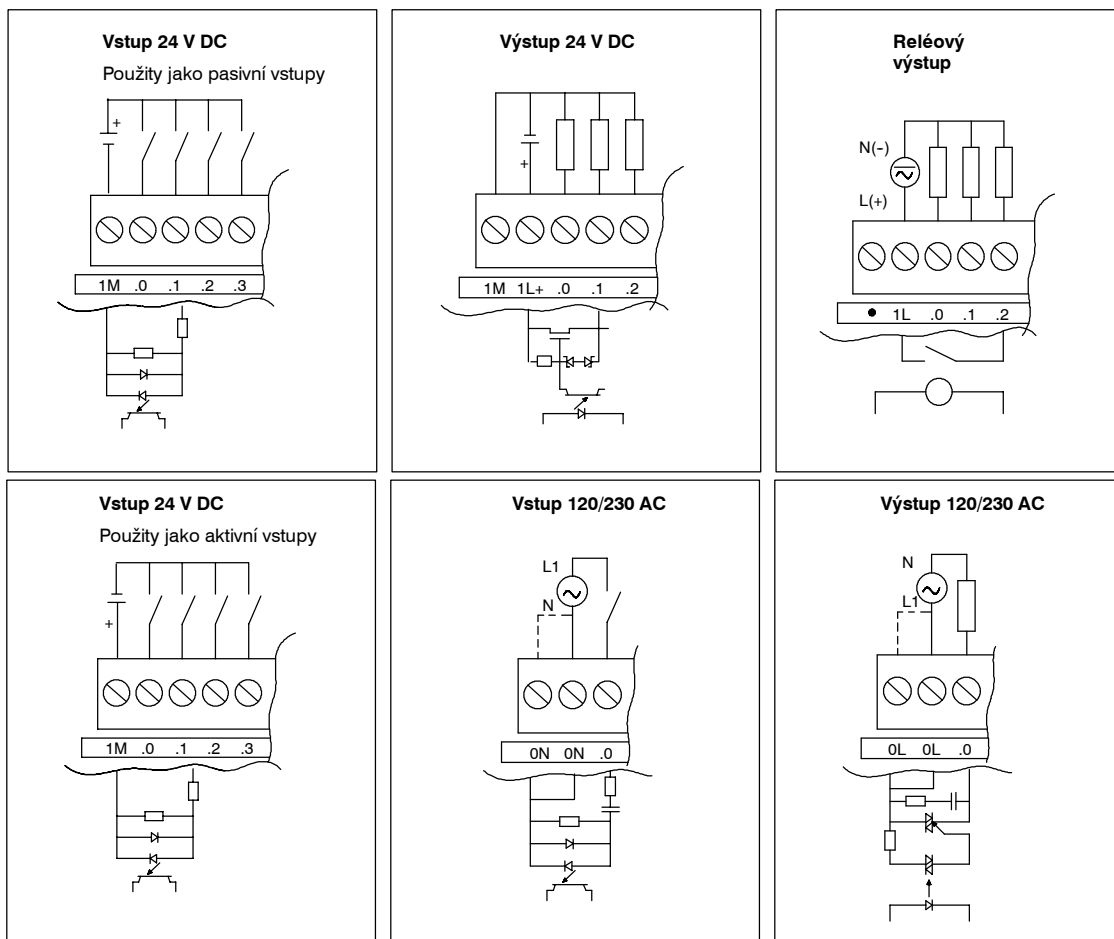
Tabulka A-12 Specifikace výstupů digitálních rozšiřovacích modulů

Obecné	Výstup 24 V DC	Reléový výstup	Výstup 120/230 V AC
Typ	Polovodičový MOSFET ¹	Kontakt (suchý)	Triak, spínací ²
Jmenovité napětí	24 V DC	24 V DC nebo 250 V AC	120/230 V AC
Rozsah napětí	20,4 až 28,8 V DC	5 až 30 V DC nebo 5 až 250 V AC	40 až 264 V AC (47 až 63 Hz)
Rozsah napětí cívky 24 V DC	–	20,4 až 28,8 V DC	–
Nárazový proud (max.)	8 A po dobu 100 ms	7 A při sepnutých kontaktech	ef. hodn. 5 A po dobu 2 AC cyklů
Logická 1 (min.)	20 V DC	–	L1 (–0,9 V efektivní hodnota)
Logická 0 (max.)	0,1 V DC	–	–
Jmenovitý proud na výstup (max.)	0,75 A	2,00 A	0,5 A AC ³
Jmenovitý proud společný (max.)	6 A	8 A	0,5 A AC
Svodový proud (max.)	10 µA	–	1,1 mA efektivní při 132 V AC a 1,8 mA RH faktor při 264 V AC
Zatěž (max.)	5 W	30 W DC/200 W AC	60 W
Indukční napětí na svorkách	L+ minus 48 V	–	–
Odpor v sepnutém stavu (kontakt)	0,3 Ω (maximální)	0,2 Ω maximální u nového	410 Ω maximální při zátěžovém proudu menším než 0,05 A
Napěťové oddělení			
Optické (galvanické, okolí od logiky)	500 V AC po dobu 1 minuty	–	1500 V AC po dobu 1 minuty
Cívka od logiky	–	Žádný	–
Cívka od kontaktu	–	1500 V AC po dobu 1 minuty	–
Kontakt od kontaktu	–	750 V AC po dobu 1 minuty	–
Odpor (cívka vůči kontaktu)	–	100 M Ω min. v novém stavu	–
Oddělení po skupinách	Viz schéma zapojení	4 výstupy	1 výstup
Zpoždění vypnutí, zapnutí/zapnutí, vypnutí (max.)	50 µs max./200 µs	–	0,2 ms + 1/2 AC cyklu
Sepnutí (max.)	–	10 ms	–
Spínací kmitočet (max.)	–	1 Hz	10 Hz
Mechanická životnost v cyklech	–	10.000.000 (bez zatížení)	–
Životnost kontaktů	–	100.000 (jmenovitá zátěž)	–
Současné sepnuté výstupy	Všechny při 55 °C	Všechny při 55 °C	Všechny při 55 °C
Paralelní spojení dvou výstupů	Ano	Ne	Ne
Délka kabelu (max.)			
Stíněný	500 m	500 m	500 m
Nestíněný	150 m	150 m	150 m

1 Po připojení napájení CPU S7-200 nebo kteréhokoliv digitálního rozšiřovacího modulu vysílá "1" signál digitálním výstupům pod dobu cca 50 mikrosekund. S tím musíte počítat obzvláště tehdy používáte – li zařízení, která reagují na krátké pulzy.

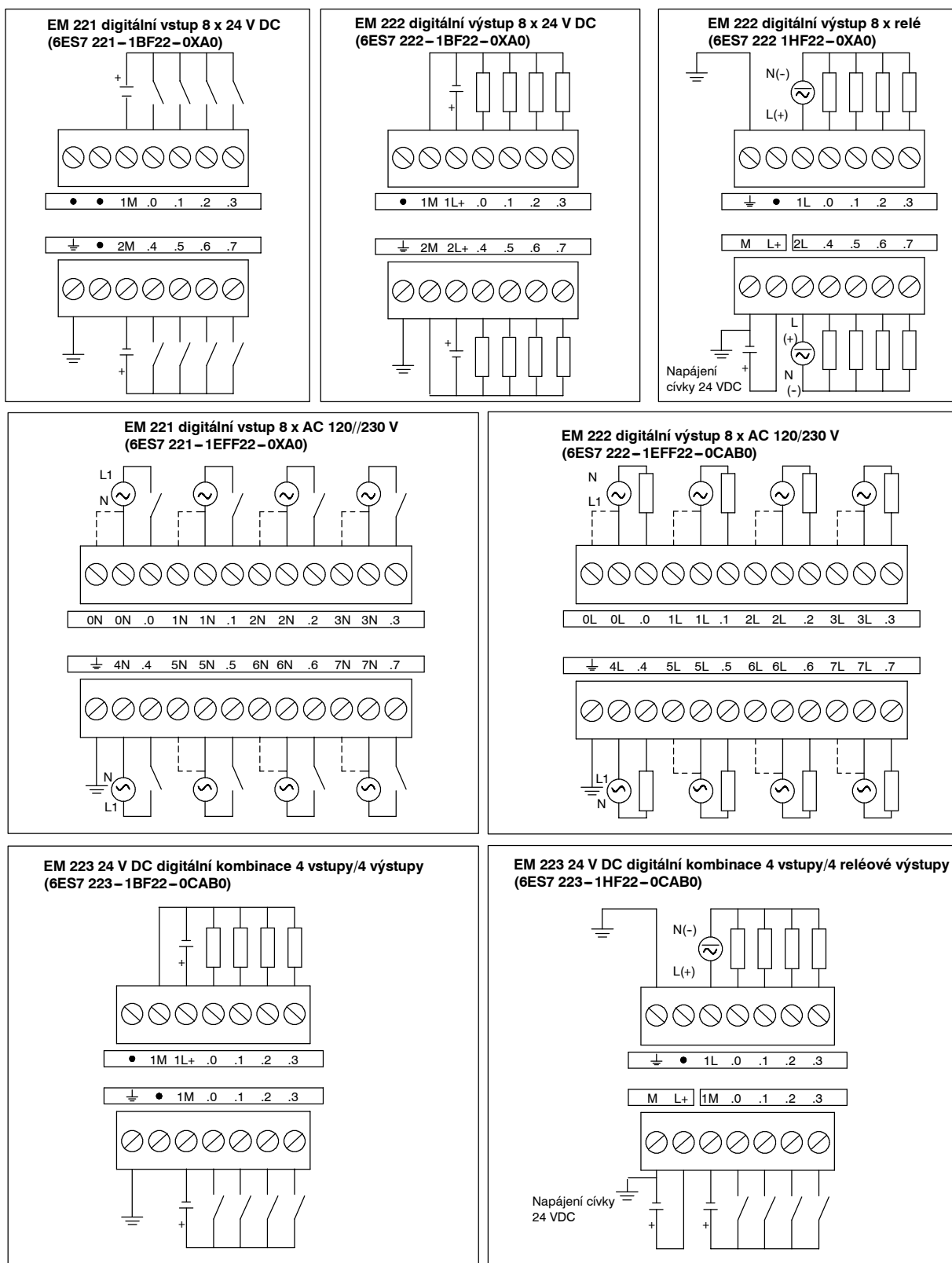
2 Po připojení napájení na AC rozšiřovací modul, vysílá signál "1" na AC výstupy po dobu přibližně 1/2 periody AC. S tím je třeba počítat.

3 Zátěžový proud musí mít celou periodu AC vlnu, nesmí být pouze půlperioda, kvůli obvodům "zero – cross". Minimální zátěžový proud je 0,05 A AC. Při zátěžovacím proudu mezi 5 mA a 50 mA AC je možné proud regulovat, ale dojde k dalšímu poklesu napětí kvůli sériovému odporu 410 ohmů.

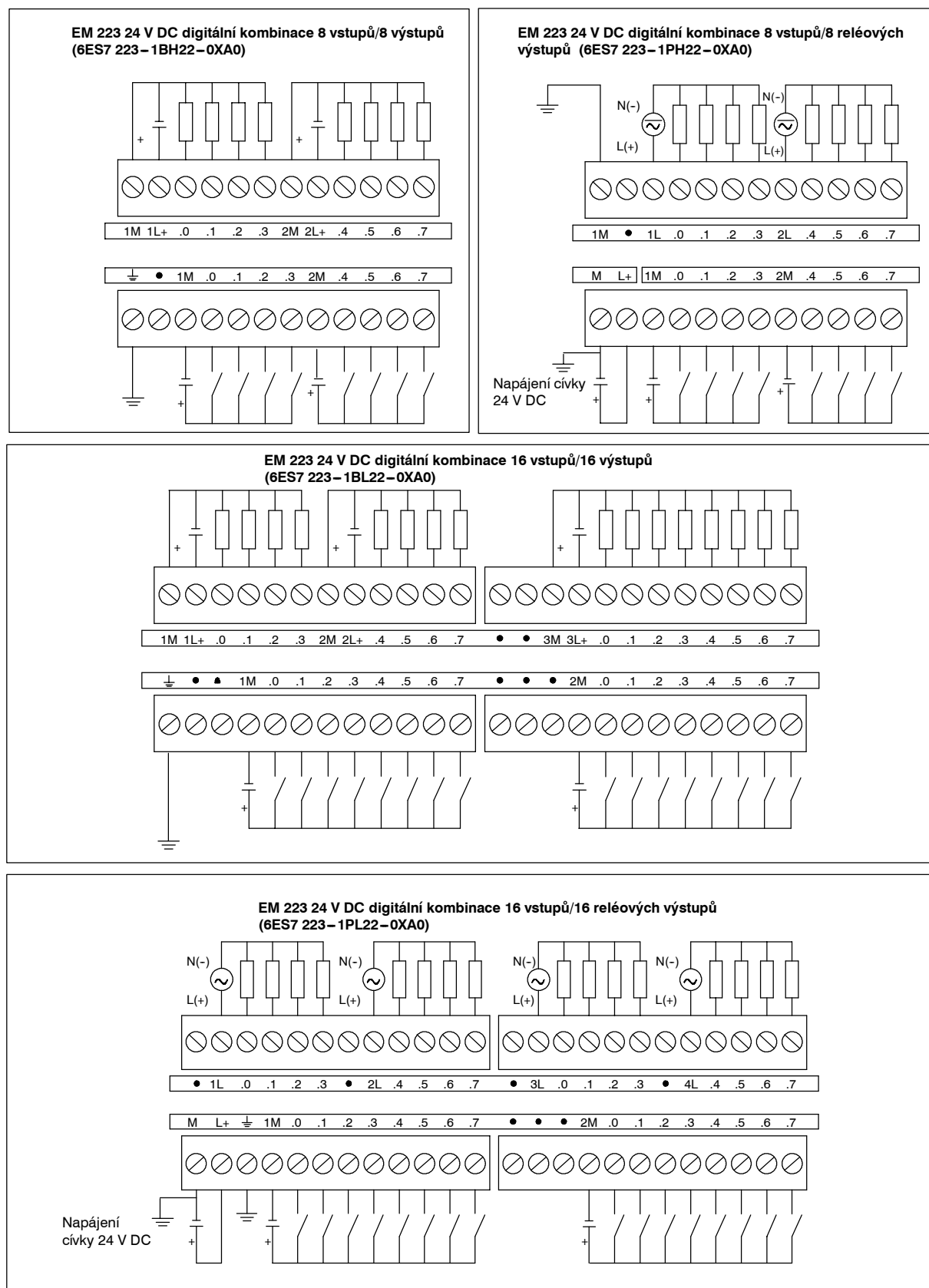


Obr. A-6 Vstupy a výstupy digitálních rozšiřovacích modulů automatu S7-200

Schéma zapojení



Obr. A-7 Schéma zapojení pro rozšiřovací moduly EM 221, EM 222 a EM 223



Obr. A-8 Schéma zapojení pro rozšiřovací moduly EM 223

Specifikace analogových rozšiřovacích modulů

Tabulka A-13 Objednací čísla analogových rozšiřovacích modulů

Objednací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Odnímatelná svorkovnice
6ES7 231-0HC22-0XA0	EM 231 analogový vstup, 4 vstupy	4	–	Ne
6ES7 232-0HB22-0XA0	EM 232 analogový výstup, 2 výstupy	–	2	Ne
6ES7 235-0KD22-0XA0	EM 235 analogová kombinace 4 vstupy/1 výstup	4	1 ¹	Ne

1 CPU rezervuje pro tento modul 2 analogové výstupy.

Tabulka A-14 Obecná specifikace analogových rozšiřovacích modulů

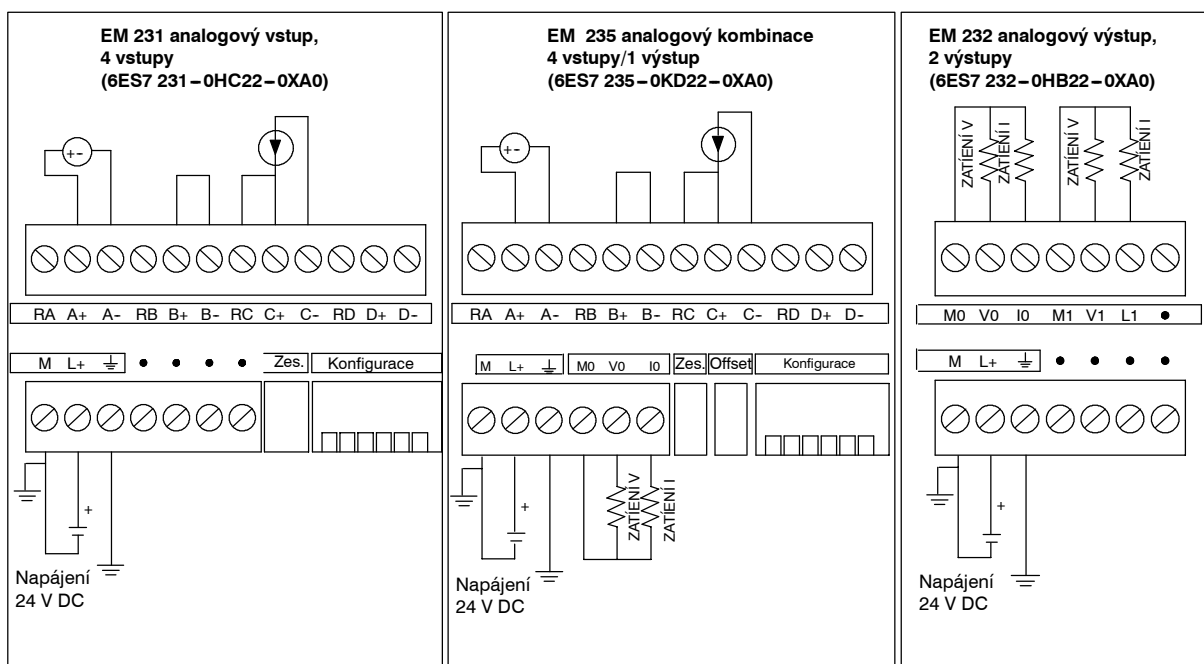
Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Ztráty	Požadavky V DC	
					+5 V DC	+24 V DC
6ES7 231-0HC22-0XA0	EM 231 analogový vstup, 4 vstupy	71,2 x 80 x 62	183 g	2 W	20 mA	60 mA
6ES7 232-0HB22-0XA0	EM 232 analogový výstup, 2 výstupy	46 x 80 x 62	148 g	2 W	20 mA	70 mA (s oběma výstupy na 20 mA)
6ES7 235-0KD22-0XA0	EM 235 analogová kombinace 4 vstupy/1 výstup	71,2 x 80 x 62	186 g	2 W	30 mA	60 mA (s výstupem na 20 mA)

Tabulka A-15 Specifikace vstupů analogových rozšiřovacích modulů

Obecné	6ES7 231-0HC22-0XA0	6ES7 235-0KD22-0XA0
Formát datového word	(Viz obrázek A-11)	(Viz obrázek A-11)
Bipolární, rozsah celé stupnice	–32000 až +32000	–32000 až +32000
Unipolární, rozsah celé stupnice	0 až 32000	0 až 32000
Impedance DC vstupu	≥10 MΩ napěťový vstup, 250 Ω proudový vstup	≥10 MΩ napěťový vstup, 250 Ω proudový vstup
Útlum vstupního filtru	–3 db při 3,1 kHz	–3 db při 3,1 kHz
Maximální vstupní napětí	30 V DC	30 V DC
Maximální vstupní proud	32 mA	32 mA
Rozlišení	12bitový A/D převodník	12bitový A/D převodník
Napěťové oddělení (okolí od logiky)	Nemá	Nemá
Typ vstupu	Diferenční	Diferenční
Vstupní rozsahy		
Napětí (unipolární)	0 až 10 V, 0 až 5 V	0 až 10 V, 0 až 5 V 0 až 1 V, 0 až 500 mV 0 až 100 mV, 0 až 50 mV
Napětí (bipolární)	±5 V, ±2,5 V	±10 V, ±5 V, ±2,5 V, ±1 V, ±500 mV, ±250 mV, ±100 mV, ±50 mV, ±25 mV
Proud	0 až 20 mA	0 až 20 mA
Vstupní rozlišení	Viz tabulka A-18	Viz tabulku A-19
Napětí (unipolární)		
Napětí (bipolární)		
Proud		
Doba konverze analogového na digitální	< 250 μs	< 250 μs
Skoková odezva analogového vstupu	1,5 ms na 95 %	1,5 ms na 95 %
Potlačení souhlasného signálu	40 dB, DC na 60 Hz	40 dB, DC na 60 Hz
Souhlasné napětí	Signálové napětí plus souhlasné napětí musí být ≤ ±12 V	Signálové napětí plus souhlasné napětí musí být ≤ ±12 V
Rozsah napětí napájení 24 V DC	20,4 až 28,8	20,4 až 28,8

Tabulka A-16 Specifikace výstupů analogových rozšiřovacích modulů

Obecné	6ES7 232-0HB22-0XA0	6ES7 235-0KD22-0XA0
Napěťové oddělení (okolí od logiky)	Nemá	Nemá
Rozsah signálu		
Napěťový výstup	$\pm 10 \text{ V}$	$\pm 10 \text{ V}$
Proudový výstup	0 až 20 mA	0 až 20 mA
Rozlišení při plné výchylce		
Napětí	12 bitů	12 bitů
Proud	11 bitů	11 bitů
Formát datového word		
Napětí	-32000 až +32000	-32000 až +32000
Proud	0 až +32000	0 až +32000
Přesnost		
Nejhorší případ, 0° až 55 °C		
Napěťový výstup	$\pm 2 \%$ plné stupnice	$\pm 2 \%$ plné stupnice
Proudový výstup	$\pm 2 \%$ plné stupnice	$\pm 2 \%$ plné stupnice
Běžně, 25 °C		
Napěťový výstup	$\pm 0,5 \%$ plné stupnice	$\pm 0,5 \%$ plné stupnice
Proudový výstup	$\pm 0,5 \%$ plné stupnice	$\pm 0,5 \%$ plné stupnice
Doba nastavení		
Napěťový výstup	100 μs	100 μs
Proudový výstup	2 ms	2 ms
Maximální buzení		
Napěťový výstup	5000 Ω minimálně	5000 Ω minimálně
Proudový výstup	500 Ω maximálně	500 Ω maximálně



Obr. A-9 Schéma zapojení analogových rozšiřovacích modulů

LED diody analogových modulů

LED diody analogových modulů jsou uvedeny v tabulce A-17.

Tabulka A-17 LED diody analogových modulů

LED	ZAPNUTO	VYPNUTO
Napájení 24 VDC v pořádku	Bez poruch	Chybí napájení 24 V DC

Kalibrace vstupů

Nastavení kalibrace ovlivňuje zesílení přístrojového zesilovače, který se nachází za analogovým multiplexorem (viz blokové schéma vstupů pro EM 231 na obrázku A-12 a EM 235 na obrázku A-13). Proto ovlivňuje kalibrace všechny uživatelské vstupní kanály. Dokonce i po kalibraci způsobí odchylky hodnot komponent každého vstupního obvodu, který se nachází před analogovým multiplexorem, drobné rozdíly v odečtech kanálů připojených na stejný vstupní signál.

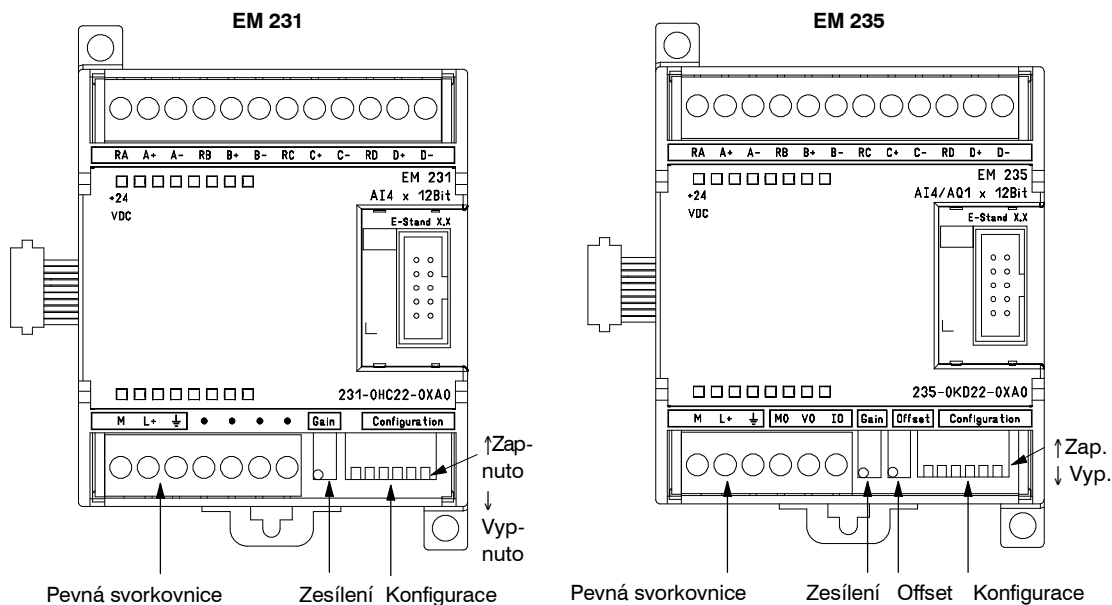
Aby hodnoty odpovídaly specifikaci, musíte aktivovat analogové vstupní filtry pro všechny vstupy modulu. Vezměte 64 nebo více vzorků a vypočítejte průměrnou hodnotu.

Chcete-li kalibrovat vstup, proveďte následující kroky.

1. Vypněte napájení modulu. Vyberte požadovaný vstupní rozsah.
2. Zapněte napájení CPU a modulu. Nechejte modul stabilizovat po dobu 15 minut.
3. Pomocí vysílače, zdroje napětí nebo zdroje proudu připojte nulový signál na jeden ze vstupních konektorů.
4. Přečtěte hodnotu poslanou do CPU příslušným vstupním kanálem.
5. Otáčejte potenciometrem OFFSET (posunutí), dokud se odečet nerovná nule nebo požadované hodnotě číselného údaje.
6. Na jeden ze vstupních konektorů připojte signál s maximální hodnotou. Přečtěte hodnotu poslanou do CPU.
7. Otáčejte potenciometrem GAIN (zesílení), dokud není odečet 32000 nebo požadovaná hodnota.
8. Kalibraci OFFSET a GAIN opakujte podle potřeby.

Umístění kalibračních potenciometrů u EM 231 a EM 235

Obrázek A-10 ukazuje kalibrační potenciometr a konfigurační DIP přepínače umístěné vpravo od dolní svorkovnice modulu.



Obr. A-10 Umístění kalibračního potenciometru a konfiguračního DIP přepínače u EM 231 a EM 235

Konfigurace EM 231

Tabulka A-18 ukazuje, jak nakonfigurovat modul EM 231 pomocí konfiguračních DIP přepínačů. Přepínače 1, 2 a 3 volí rozsah analogového vstupu. Všechny vstupy se nastavují na stejný rozsah analogového vstupu. V této tabulce znamená ON sepnutý, OFF rozepnutý.

Tabulka A-18 EM 231 Tabulka konfiguračních přepínačů pro volbu rozsahu analogového vstupu

Unipolární			Plný rozsah vstupu	Rozlišení
SW1	SW2	SW3		
ON	OFF	ON	0 až 10 V	2,5 mV
	ON	OFF	0 až 5 V	1,25 mV
			0 až 20 mA	5 μ A
Bipolární			Plný rozsah vstupu	Rozlišení
SW1	SW2	SW3		
OFF	OFF	ON	± 5 V	2,5 mV
	ON	OFF	± 2,5 V	1,25 mV

Konfigurace EM 235

Tabulka A-19 ukazuje, jak nakonfigurovat modul EM 235 pomocí konfiguračních DIP přepínačů. Přepínače 1 až 6 volí rozsah a rozlišení analogového vstupu. Všechny vstupy se nastavují na stejný rozsah a formát analogového vstupu. Tabulka A-20 ukazuje, jak volit typ signálu unipolární/bipolární (přepínač 6), zesílení (přepínače 4 a 5) a útlum (přepínače 1, 2 a 3). V těchto tabulkách znamená ON sepnutý, OFF rozepnutý.

Tabulka A-19 Tabulka konfiguračních přepínačů modulu EM 235 pro volbu analogového rozsahu a rozlišení

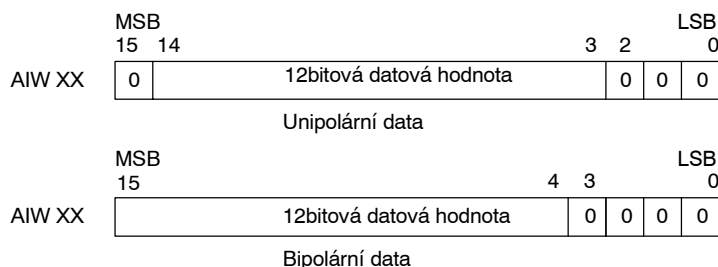
Unipolární						Plný rozsah vstupu	Rozlišení
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	0 až 50 mV	12.5 μ V
OFF	ON	OFF	ON	OFF	ON	0 až 100 mV	25 μ V
ON	OFF	OFF	OFF	ON	ON	0 až 500 mV	125 μ V
OFF	ON	OFF	OFF	ON	ON	0 až 1 V	250 μ V
ON	OFF	OFF	OFF	OFF	ON	0 až 5 V	1,25 mV
ON	OFF	OFF	OFF	OFF	ON	0 až 20 mA	5 μ A
OFF	ON	OFF	OFF	OFF	ON	0 až 10 V	2,5 mV
Bipolární						Plný rozsah vstupu	Rozlišení
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	$\pm$ 25 mV	12.5 μ V
OFF	ON	OFF	ON	OFF	OFF	$\pm$ 50 mV	25 μ V
OFF	OFF	ON	ON	OFF	OFF	$\pm$ 100 mV	50 μ V
ON	OFF	OFF	OFF	ON	OFF	$\pm$ 250 mV	125 μ V
OFF	ON	OFF	OFF	ON	OFF	$\pm$ 500 mV	250 μ V
OFF	OFF	ON	OFF	ON	OFF	$\pm$ 1 V	500 μ V
ON	OFF	OFF	OFF	OFF	OFF	$\pm$ 2,5 V	1,25 mV
OFF	ON	OFF	OFF	OFF	OFF	$\pm$ 5 V	2,5 mV
OFF	OFF	ON	OFF	OFF	OFF	$\pm$ 10 V	5 mV

Tabulka A-20 Tabulka konfiguračních přepínačů modulu EM 235 pro volbu typu signálu, zesílení a útlumu

EM 235 Konfigurační přepínače						Nastavení unipolární/bipolární	Nastavení zesílení	Nastavení útlumu
SW1	SW2	SW3	SW4	SW5	SW6			
					ON	Unipolární	x1 x10 x100 neplatí	
					OFF	Bipolární		
			OFF	OFF				
			OFF	ON				
			ON	OFF				
ON	OFF	OFF						0.8
OFF	ON	OFF						0.4
OFF	OFF	ON						0.2

Formát vstupního datového word pro EM 231 a EM 235

Obrázek A-11 ukazuje, kde je 12bitová hodnota dat umístěna v analogovém vstupním word CPU.



Obr. A-11 Formát vstupního datového word pro EM 231 a EM 235



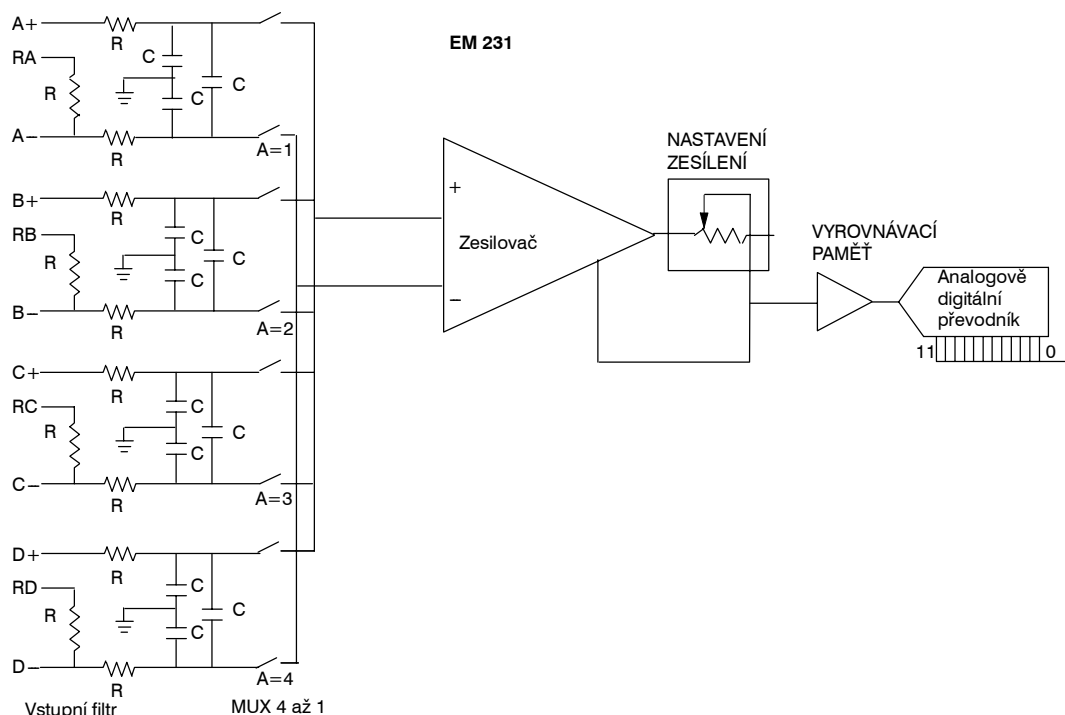
Tip

12 bitů načtených z analogově číslicového převodníku je v datovém formátu word zarovnaného vlevo. MSB je znaménkový bit: nula znamená kladnou hodnotu datového word.

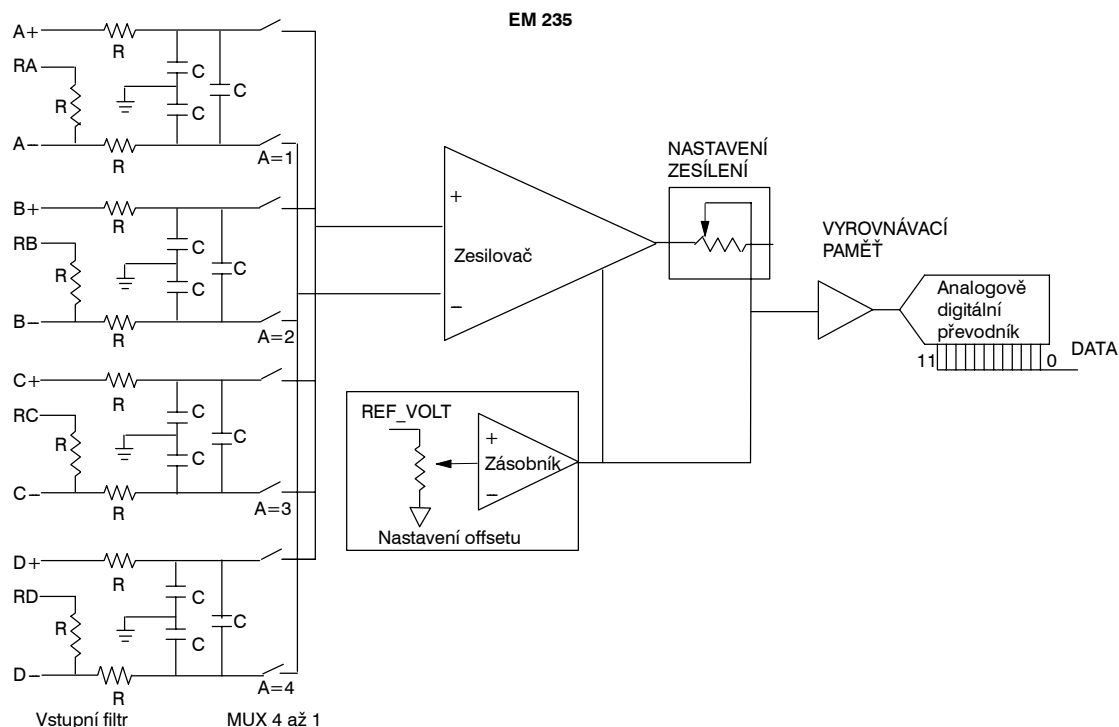
V unipolárním formátu mají tři za sebou jdoucí nuly za následek změnu datového word o osm pro každou jednotkovou změnu v hodnotě analogově číslicového převodníku.

V bipolárním formátu mají čtyři za sebou jdoucí nuly za následek změnu datového word o šestnáct pro každou jednotkovou změnu v hodnotě analogově číslicového převodníku.

Blokové schéma vstupů EM 231 a EM 235



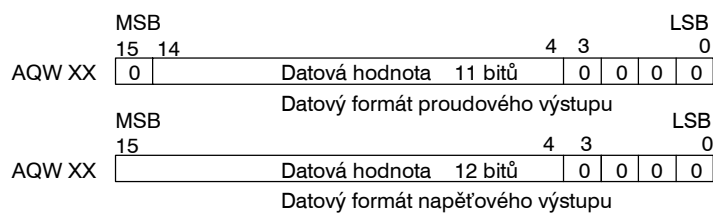
Obr. A-12 Blokové schéma vstupů EM 231



Obr. A-13 Blokové schéma vstupů EM 235

Formát výstupního datového word pro EM 232 a EM 235

Obrázek A-14 ukazuje, kde je 12bitová hodnota dat umístěna v analogovém výstupním word CPU.



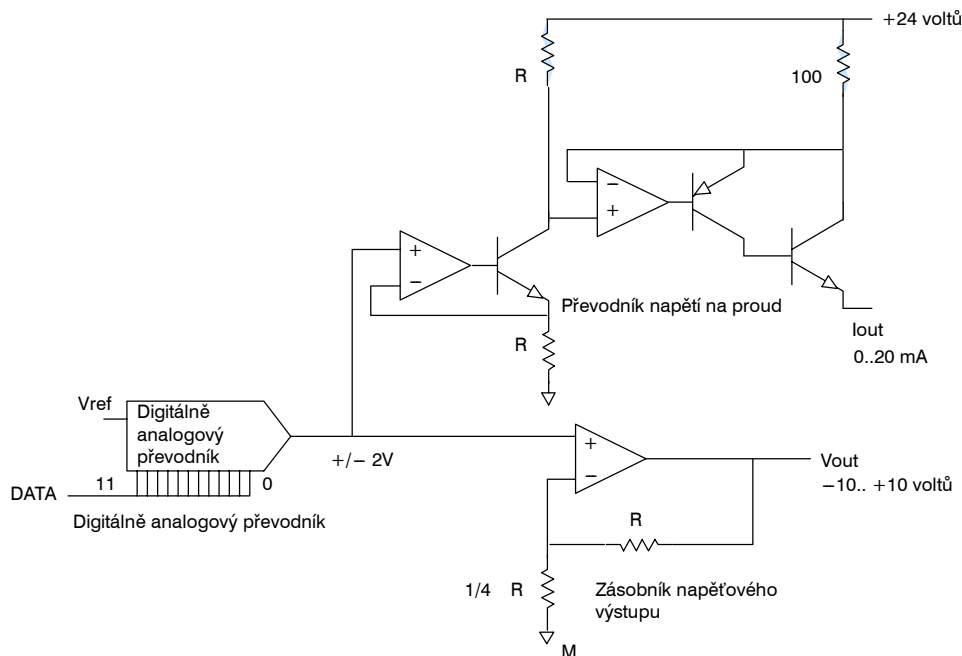
Obr. A-14 Formát výstupního datového word pro EM 232 a EM 235



Tip

12 bitů načtených z číslicově analogového převodníku je v datovém formátu word zarovnáno vlevo. MSB je znaménkový bit: nula znamená kladnou hodnotu datového word. Čtyři za sebou jdoucí nuly se před nahráním do registrů převodníku vypustí. Tyto bity nemají žádný vliv na hodnotu výstupního signálu.

Blokové schéma výstupů EM 232 a EM 235



Obr. A-15 Blokové schéma výstupů EM 232 a EM 235

Návod pro instalaci

Abyste zajistili přesnost a opakovatelnost, postupujte podle následujícího návodu:

- ☐ Zajistěte, aby napájení snímačů 24 V DC bylo bez šumu a stabilizované.
- ☐ Pro připojení snímače použijte co možná nejkratší kabely.
- ☐ Pro kabeláž snímače použijte stíněnou kroucenou dvojlinku.
- ☐ Stínění se zakončuje pouze na straně snímače.
- ☐ Pro nepoužité kanály zkratujte vstupy podle obrázku A-9.
- ☐ Vyhněte se ohýbání vodičů do ostrých úhlů.
- ☐ Vodiče ved'te kabelovými kanály.
- ☐ Vyhněte se paralelnímu vedení signálních vodičů se silovými vodiči. Pokud se dva vodiče musí křížit, umístěte je kolmo na sebe.
- ☐ Oddělením vstupních signálů nebo zavedením analogového referenčního společného vodiče 24 V zajistěte, aby byly vstupní signály ve specifikovaném rozsahu souhlasného napětí.



Tip

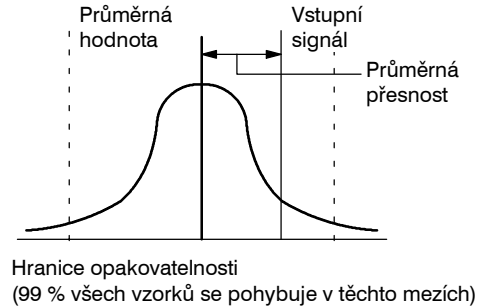
Nedoporučuje se používat rozšiřovací moduly EM 231 a EM 235 s termočlánky.

Vysvětlení analogového vstupního modulu: Přesnost a opakovatelnost

Moduly EM 231 a EM 235 jsou levné a rychlé 12bitové analogové vstupní moduly. Dokáží převést vstupní analogový signál na odpovídající digitální hodnotu za 149 μsec . Vstupní analogový signál se převádí pokaždé, když uživatelský program přistupuje k analogovému vstupu. Tyto doby převodu se musí přičíst k základní době provádění instrukce použité pro přístup k analogovému vstupu.

Moduly EM 231 a EM 235 poskytují nezpracovanou číselnou hodnotu (bez linearizace nebo filtrování), která odpovídá analogovému napětí nebo proudu na vstupních svorkách modulu. Protože jsou moduly vysokorychlostní, mohou sledovat rychlé změny analogového vstupního signálu (včetně interního a externího šumu).

Rozdíly mezi dvěma hodnotami vzorků zapříčiněné šumem v případě konstantního nebo pomalu se měnícího analogového vstupního signálu je možné minimalizovat vypočítáním průměru z určitého počtu vzorků. Uvědomte si, že zvýšení počtu vzorků použitých při výpočtu průměrné hodnoty má za následek úměrně pomalejší dobu odezvy na změny vstupního signálu.



Obr. A-16 Definice přesnosti

Obrázek A-16 ukazuje 99% hranice opakovatelnosti, střední neboli průměrnou hodnotu jednotlivých vzorků a průměrnou přesnost v grafické formě.

Specifikace opakovatelnosti popisuje odchylky mezi jednotlivými vzorky modulu pro vstupní signál, který se nemění. Specifikace opakovatelnosti definuje hranice, mezi kterými se pohybuje 99 % všech vzorků. Opakovatelnost je na tomto obrázku znázorněna křivkou ve tvaru zvonu.

Specifikace průměrné přesnosti popisuje průměrnou hodnotu chyby (rozdílu mezi průměrnou hodnotou jednotlivých vzorků a přesnou hodnotou skutečného analogového vstupního signálu).

Tabulka A-21 uvádí specifikaci opakovatelnosti a střední přesnost ve vztahu ke každému z konfigurovatelných rozsahů.

Definice vlastností

- ☐ Přesnost: odchylka od předpokládané hodnoty v daném bodě
- ☐ Rozlišení: výsledek změny LSB, jak se odráží na výstupu.

Tabulka A-21 Specifikace EM 231 a EM 235

Plný rozsah vstupu	Opakovatelnost ¹		Průměrná přesnost ^{1,2,3,4}	
	% plného rozsahu	Jednotky	% plného rozsahu	Jednotky
Specifikace EM 231				
0 až 5 V	± 0.075 %	± 24	± 0.1 %	± 32
0 až 20 mA				
0 až 10 V		± 48		
± 2,5 V				
± 5 V				
Specifikace EM 235				
0 až 50 mV	± 0.075 %	± 24	± 0.25 %	± 80
0 až 100 mV			± 0.2 %	± 64
0 až 500 mV			± 0.05 %	± 16
0 až 1 V				
0 až 5 V				
0 až 20 mA				
0 až 10 V		± 0.075 %	± 48	± 0.25 %
± 25 mV	± 0.2 %			± 128
± 50 mV	± 0.1 %			± 64
± 100 mV	± 0.05 %		± 32	
± 250 mV				
± 500 mV				
± 1 V				
± 2,5 V				
± 5 V				
± 10 V				

¹ Měření byla provedena po kalibraci zvoleného vstupního rozsahu.

² Chyba posunutí u signálu blízkého nulovému analogovému vstupu není korigována a není zahrnuta do specifikace přesnosti.

³ Při přechodu z kanálu na kanál dochází k chybě převodu způsobené konečnou dobou ustálení analogového multiplexeru. Maximální chyba při přechodu je 0,1 % rozdílu mezi kanály.

⁴ Průměrná přesnost zahrnuje účinky nelinearity a driftu při 0 až 55 °C.

Specifikace termočlávkových a RTD rozšiřovacích modulů

Tabulka A-22 Objednací čísla termočlávkových a RTD modulů

Objednací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Odnímatelná svorkovnice
6ES7 231-7PD22-0XA0	EM 231 analogové vstupy pro termočlásky, 4 vstupy	4 termočlásky	–	Nemá
6ES7 231-7PB22-0XA0	EM 231 analogové vstupy RTD, 2 vstupy	2 RTD	–	Nemá

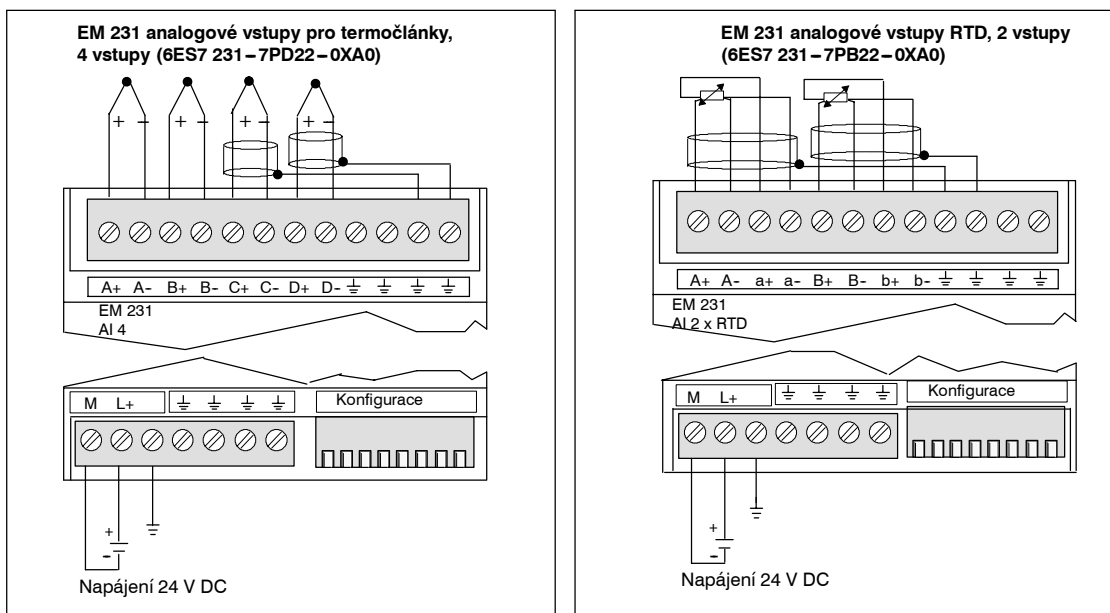
Tabulka A-23 Obecná specifikace termočlávkových a RTD modulů

Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Výkonové ztráty	Požadavky V DC +5 V DC +24 V DC
6ES7 231-7PD22-0XA0	EM 231 analogové vstupy pro termočlásky, 4 vstupy	71,2 x 80 x 62	210 g	1,8 W	87mA 60 mA
6ES7 231-7PB22-0XA0	EM 231 analogové vstupy RTD, 2 vstupy	71,2 x 80 x 62	210 g	1,8 W	87 mA 60 mA

Tabulka A-24 Specifikace termočlávkových a RTD modulů

Obecné	6ES7 231-7PD22-0XA0 Termočlánek	6ES7 231-7PB22-0XA0 RTD
Napětové oddělení Okolí od logiky Okolí od 24 V DC 24 V DC od logiky	500 V AC 500 V AC 500 V AC	500 V AC 500 V AC 500 V AC
Rozsah vstupního souhlasného napětí (vstupní kanál ke vstupnímu kanálu)	120 V AC	0
Potlačení souhlasného signálu	> 120 dB při 120 V AC	> 120 dB při 120 V AC
Typ vstupu	Plovoucí termočlánek	RTD vztažený k uzemnění modulu
Vstupní rozsahy ¹	Typy termočláneků (vyberte jeden na modul) S, T, R, E, N, K, J Rozsah napětí: +/- 80 mV	Typy RTD (vyberte jeden na modul): Pt – 100Ω, 200Ω, 500Ω, 1000Ω (s α = 3850 ppm, 3920 ppm, 3850,55 ppm, 3916 ppm, 3902 ppm) Pt – 10000Ω (α = 3850 ppm) Cu – 9,035Ω (α = 4720 ppm) Ni – 10Ω, 120Ω, 1000Ω (s α = 6720 ppm, 6178 ppm) R – 150Ω, 300Ω, 600Ω FS
Vstupní rozlišení Teplota Napětí Odpor	0.1 °C / 0.1 °F 15 bitů plus znaménko –	0.1 °C / 0.1 °F – 15 bitů plus znaménko
Princip měření	Sigma-delta	Sigma-delta
Doba aktualizace modulu: Všechny kanály	405 ms	405 ms (700 ms pro Pt10000)
Délka vodičů	100 metrů ke snímači max.	100 metrů ke snímači max.
Odpor smyčky vodičů	100Ω max.	20Ω, 2,7Ω pro Cu max.
Potlačení rušení	85 dB při 50 Hz/60 Hz/ 400 Hz	85 dB při 50 Hz/60 Hz/ 400 Hz
Formát datového word	Napětí: –27648 až +27648	Odpor: –27648 až +27648
Maximální ztráty snímače	–	1 mW
Vstupní impedance	≥1 MΩ	≥ 10 MΩ
Maximální vstupní napětí	30 V DC	30 V DC (snímání), 5 V DC (napájení)
Útlum vstupního filtru	–3 db při 21 kHz	–3 db při 3,6 kHz
Základní chyba	0,1 % plného rozsahu (napětí)	0,1 % plného rozsahu (odpor)
Opakovatelnost	0,05 % plného rozsahu	0,05 % plného rozsahu
Chyba studeného spoje	±1,5 °C	–
Rozsah napětí napájení 24 V DC	20,4 až 28,8 V DC	20,4 až 28,8 V DC

¹ Volba vstupního rozsahu (teplota, napětí na odporu) platí pro všechny kanály modulu.



Obr. A-17 Identifikace připojovacích svorek termočláňkového modulu EM 231 a RTD modulu EM 231

Kompatibilita

Modul RTD a termočláňkový modul jsou určeny pro práci s CPU 222, CPU 224, CPU 226 a CPU 226XM.



Tip

Modul RTD a termočláňkový modul podávají maximální výkon, jsou-li instalovány v prostředí se stálou teplotou.

Například termočláňkový modul EM 231 má speciální obvod pro kompenzaci studeného spoje, který měří teplotu na svorkách modulu a provádí nutné změny, aby kompenzoval teplotní rozdíly mezi referenční teplotou a teplotou modulu. Pokud se rychle mění teplota okolí v prostoru, kde je instalován termočláňkový modul EM 231, se zanášejí do měření další chyby.

Aby se dosáhlo maximální přesnosti a opakovatelnosti, doporučuje Siemens instalovat RTD moduly a termočláňkové moduly programovatelného automatu S7-200 na místech se stabilní teplotou okolí.

Odolnost proti rušení

Pro dosažení nejlepší odolnosti proti rušení použijte stíněné kabely. Pokud není termočláňkový vstupní kanál použit, zkratujte nepoužité kanálové vstupy nebo je zapojte paralelně s jiným kanálem.

Termočlánkový modul EM 231

Termočlánkový modul EM 231 je praktické, napětově oddělené rozhraní řady S7-200 pro sedm typů termočlánků: J, K, E, N, S, T a R. Umožňuje připojení automatu S7-200 na nízkourovňové analogové signály v rozsahu ± 80 mV. Všechny termočlánky připojené k modulu musí být stejného typu.

Základní informace o termočláncích

Termočlánky se vytvoří, kdykoliv jsou elektricky vzájemně spojeny dva různé kovy. Vzniká napětí, které je přímo úměrné teplotě spoje. Jde o malé napětí; jeden mikrovolt může představovat mnoho stupňů. Měření napětí termočlánku, kompenzace nadbytečných spojů a následující linearizace výsledků tvoří základ měření teploty pomocí termočlánků.

Když připojujete termočlánek k termočlánkovému modulu EM 231, jsou dva vodiče z různých kovů připojeny k signálovému konektoru modulu. Místo, kde jsou dva různé vodiče vzájemně spojeny, tvoří termočlánek.

Dva další termočlánky jsou vytvořeny tam, kde jsou dva různé vodiče připojeny k signálovému konektoru. Teplota konektoru dává vzniknout napětí, které se přičítá k napětí termočlánku. Pokud toto napětí není korigováno, bude se hlášená teplota lišit od měřené teploty.

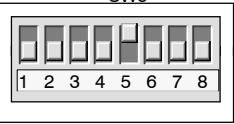
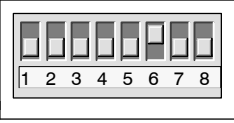
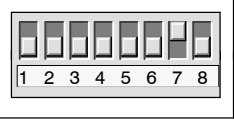
Kompenzace studených spojů se používá pro kompenzaci termočlánku vzniklého v místě konektoru. Tabulky pro termočlánky jsou založeny na referenční teplotě spoje, obvykle je to nula stupňů Celsia. Kompenzace studeného spoje kompenzuje konektor na 0 °C. Kompenzace studeného spoje redukuje napětí přidané konektorovými termočlánci. Měří se vnitřní teplota modulu, která je převedena na hodnotu, která se přičte k převedené hodnotě snímače. Korigovaná převedená hodnota snímače je pak linearizována s použitím tabulek pro termočlánky.

Konfigurace termočlánkového modulu EM 231

Konfigurační DIP přepínače umístěné na spodní straně modulu umožňují výběr typu termočlánku, detekce přerušného vodiče, teplotní stupnice a kompenzace studeného spoje. Aby nastavení DIP přepínače nabyla platnosti, musíte vypnout a znovu zapnout programovatelný automat nebo odpojit uživatelské napájení 24 V.

DIP přepínač č. 4 je rezervován pro použití v budoucnosti. Nastavte DIP přepínač č. 4 na polohu 0 (dole neboli vypnuto). Tabulka A-25 ukazuje nastavení ostatních DIP přepínačů.

Tabulka A-25 Nastavení DIP přepínačů termočláňkového modulu

Přepínače 1,2,3	Typ termočláňku	Nastavení	Popis
SW1, 2, 3  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto * Nastavte DIP přepínač č. 4 do polohy 0 (dole).	J (Implicitní nastavení)	000	Přepínači 1 až 3 se volí typ termočláňku (nebo provoz s mV) pro všechny kanály modulu. Například pro termočlánek typu E SW1 = 0, SW2 = 1, SW3 = 1.
	K	001	
	T	010	
	E	011	
	R	100	
	S	101	
	N	110	
	+/-80 mV	111	
Přepínač č. 5	Směr při detekci přerušného vodiče	Nastavení	Popis
SW5  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	Nahoru po stupnici (+3276,7 stupňů)	0	0 indikuje kladnou hodnotu na přerušném vodiči 1 indikuje zápornou hodnotu na přerušném vodiči
	Dolů po stupnici (-3276,8 stupňů)	1	
Přepínač č. 6	Povolení detekce přerušného vodiče	Nastavení	Popis
SW6  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	Povoleno	0	Detekce přerušného vodiče se provádí přivedením proudu 25 µA na vstupní svorky. Přepínač pro povolení detekce přerušného vodiče povoluje nebo blokuje proudový zdroj. Kontrola rozsahu přerušného vodiče se provádí vždy, i když je proudový zdroj zablokován. Termočláňkový modul EM 231 detekuje přerušný vodič, jestliže vstupní signál překročí přibližně ±200 mV. Při zjištění přerušného vodiče je odečet modulu nastaven na přednastavenou hodnotu.
	Blokovat	1	
Přepínač č. 7	Teplotní stupnice	Nastavení	Popis
SW7  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	Celsius (°C)	0	Termočláňkový modul EM 231 může udávat teplotu ve stupních Celsia nebo Fahrenheita. Převod stupňů Celsia na Fahrenheita se provádí v modulu.
	Fahrenheit (°F)	1	
Přepínač č. 8	Studený spoj	Nastavení	Popis
SW8  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	Kompenzace studeného spoje povolena	0	Jestliže používáte termočláňky, musí být povolena kompenzace studeného spoje. Pokud není povolena kompenzace studeného spoje, budou převody z modulu chybné vzhledem k napětí, které vzniká při připojení vodiče termočláňku ke konektoru modulu. Tato kompenzace je automaticky blokována, jestliže zvolíte rozsah ±80 mV.
	Kompenzace studeného spoje blokována	1	

**Tip**

- Zdroj proudu pro detekci přerušného vodiče by mohl rušit signály některých nízkoúrovňových zdrojů, například termočlávkové simulátory.
- Vstupní napětí vyšší než asi ± 200 mV spustí detekci přerušného vodiče, i když je proudový zdroj přerušného vodiče blokován.

**Tip**

- Pokud dochází ke změnám teploty okolí, může být chyba modulu vyšší, než je uvedeno ve specifikaci.
- Překročení rozsahu teplot okolí specifikovaného pro modul může způsobit chybu kompenzace studeného spoje modulu.

Použití termočlátku: Stavové indikátory

Termočlávkový modul EM 231 dodává programovatelnému automatu hodnoty word, které představují teploty nebo chyby. Stavové bity indikují chybu rozsahu a poruchu uživatelského napájení/modulu. LED diody ukazují stav modulu. Uživatelský program by měl mít logiku, která by detekovala chyby a reagovala tak, jak to daná aplikace vyžaduje. Tabulka A-26 ukazuje stavové indikátory termočlávkového modulu EM 231.

Tabulka A-26 Stavové indikátory termočlávkového modulu EM 231

Chyba	Data v kanálu	SF LED Červená	24 V LED Zelená	Stavový bit rozsahu ¹	Chyba uživatelského napájení 24 V DC ²
Bez chyby	Data z převodu	NESVÍTÍ	SVÍTÍ	0	0
Chybí 24 V	32766	NESVÍTÍ	NESVÍTÍ	0	1
Aktivována detekce přerušného vodiče a proudový zdroj	-32768/32767	BLIKÁ	SVÍTÍ	1	0
Vstup mimo rozsah	-32768/32767	BLIKÁ	SVÍTÍ	1	0
Diagnostická chyba ³	0000	SVÍTÍ	NESVÍTÍ	0	poznámka ³

¹ Stavový bit rozsahu je bit 3 v bytu chybového registru modulu (SMB9 pro modul 1, SMB11 pro modul 2, atd.)

² Stavový bit chyby uživatelského napájení je bit 2 v bytu chybového registru modulu (SMB 9, SMB 11, atd., viz přílohu D)

³ Diagnostické chyby způsobují chybu konfigurace modulu. Stavový bit chyby napájení může, ale nemusí být nastaven před chybou konfigurace modulu.

**Tip**

Datový formát kanálu je dvojkový doplněk 16bitových word. Teplota je uváděna v jednotkách 0,1 stupně. Je-li například naměřená teplota 100,2 stupňů, modul uvádí 1002. Napěťová data jsou vynásobena koeficientem 27648. Například -60,0 mV je hlášeno jako -20736 ($= -60 \text{ mV} / 80 \text{ mV} * 27648$).

Všechny čtyři kanály jsou aktualizovány každých 405 milisekund, pokud programovatelný automat načte data. Pokud automat data nenačte během doby jedné aktualizace, uvádí modul stará data až do příští aktualizace modulu. Aby data v kanálu zůstávala aktuální, doporučuje se, aby program v automatu načítal data nejméně tak často, jaká je rychlost aktualizace modulu.

**Tip**

Při používání termočlávkového modulu EM 231 byste měli zablokovat analogovou filtraci programovatelného automatu. Analogová filtrace může zabránit včasnému rozpoznání chyby.

Tabulka A-27 Rozsahy teploty (°C) a přesnost pro různé typy termočlánků

Dat. word (1 čís. = 0,1 °C)		Typ J	Typ K	Typ T	Typ E	Typ R, S	Typ N	±80mV	
DEC	HEX								
32767	7FFF	>1200,0 °C	>1372,0 °C	>400,0 °C	>1000,0 °C	>1768,0 °C	>1300,0 °C	>94,071 mV	OF
↑	↑							↑	↑
32511	7EFF							94,071 mV	
:	:								OR
27649	6C01							80,0029 mV	
27648	6C00					↑		80 mV	
:	:								
17680	4510		↑			1768,0 °C			
:	:								
13720	3598		1372,0 °C nad rozs.				↑		
:	:								
13000	32C8	↑	1300,0 °C				1300,0 °C		
:	:								
12000	2EE0	1200,0 °C			↑				
:	:								
10000	2710			↑	1000,0 °C				
:	:								
4000	0FA0			400,0 °C		400,0 °C			
:	:								
1	0001	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,0029 mV	
0	0000	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,0 mV	
-1	FFFF	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	-0,0029 mV	
:	:								
-500	FE0C					pod rozsahem -50,0 °C			
-1500	FA24	-150,0 °C				↓			
:	:								
-2000	F830	pod rozs.	-200,0 °C						
:	:								
-2100	F7CC	-210,0 °C							
:	:		pod rozs.						
-2550	F60A			-255,0	-255,0 °C				
:	:			pod rozs.	pod rozs.				
-2700	F574	↓	-270,0 °C	-270,0 °C	-270,0 °C		-270,0 °C		
:	:								
-27648	9400		↓	↓	↓		↓	-80 mV	
-27649	93FF							-80,0029mV	
:	:								
-32512	8100							-94,071 mV	UR
↓	↓							↓	↓
-32768	8000	<-210,0 °C	<-270,0 °C	<-270,0 °C	<-270,0 °C	<-50,0 °C	<-270,0 °C	<-94,071 mV	UF
Přesnost v celém int.		±0,1 %	±0,3 %	±0,6 %	±0,1 %	±0,6 %	±0,1 %	±0,1 %	
Přesnost (jm. rozsah bez komp. studeného spoje)		±1,5 °C	±1,7 °C	±1,4 °C	±1,3 °C	±3,7 °C	±1,6 °C	±0,10 %	
Chyba studeného spoje		±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	N/A	
*OF = Přetečení; OR = Nad rozsahem; NR = Jmenovitý rozsah; UR = Pod rozsahem; UF = Podtečení									
↑ znamená, že všechny analog. hodn. větší než tato a pod prahem přeruš. vodiče mají za výsledek datovou hodn. přetečení, 32767 (0x7FFF).									
↓ znamená, že všechny analog. hodn. menší než tato a vyšší než práh přeruš. vodiče mají za výsl. datovou hodn. podtečení, -32768 (0x8000).									

Tabulka A-28 Rozsahy teplot (°F) pro různé typy termočlánků

Datové word (1 číslice = 0,1 °F)		Typ J	Typ K	Typ T	Typ E	Typ R, S	Typ N	±80 mV	
DEC	HEX								
32767	7FFF	>2192,0 °F	>2502,0 °F	>752,0 °F	>1832,0 °F	>3214,0 °F	>2372,0 °F	>94,071 mV	OF
↑	↑					↑		↑	↑
32511	7EFF							94,071 mV	OR
32140	7D90					3214,0 °F		80,0029 mV	
27649	6C01					2764,8 °F		80 mV	
27648	6C00		↑						NR
:	:								
25020	61B8		2502,0 °F nad rozs.				↑		
:	:								
23720	5CA8	↑	2372,0 °F				2372,0 °F		
:	:								
21920	55A0	2192,0 °F			↑				
:	:								
18320	4790			↑	1832,0 °F				
:	:								
7520	1D60			752,0 °F		752,0 °F			
:	:								
320	0140					pod rozsahem	32,0 °F		
:	:								
1	0001	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,0029 mV	
0	0000	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 mV	
-1	FFFF	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,0029 mV	
:	:								
-580	FDBC					-58,0 °F			
:	:								
-2380	F6B4	-238,0 °F							
:	:								
-3280	F330	pod rozs.	-328,0 °F				pod rozs.		
:	:								
-3460	F27C	-346,0 °F				↓			
:	:		pod rozs.						
-4270	EF52			-427,0 °F	-427,0 °F				
:	:			pod rozs.	pod rozs.				
-4540	EE44	↓	-454,0 °F	-454,0 °F	-454,0 °F		-454,0 °F		
:	:								
-27648	9400		↓	↓	↓		↓	-80 mV	
-27649	93FF							-80,0029 mV	
:	:								
-32512	8100							-94,071 mV	
↓	↓							↓	↓
-3268	8000	<-346,0 °F	<-454,0 °F	<-454,0 °F	<-454,0 °F	<-58,0 °F	<-454,0 °F	<-94,07 mV	UF

*OF = Přetečení; OR = Nad rozsahem; NR = Normální rozsah; UR = Pod rozsahem; UF = Podtečení
↑ znamená, že všechny analog. hodn. větší než tato a pod prahem přeruš. vodiče mají za výsledek datovou hodn. přetečení, 32767 (0x7FFF).
↓ znamená, že všechny analog. hodn. menší než tato a vyšší než práh přeruš. vodiče mají za výsl. datovou hodn. podtečení, -32768 (0x8000).

A

RTD modul EM 231

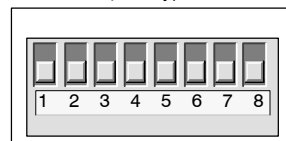
RTD modul EM 231 je praktické rozhraní řady S7-200 pro několik různých RTD (odporových teploměrů měření teploty). Zároveň umožňuje automatu S7-200 měřit tři různé rozsahy odporu. Oba RTD připojené k modulu musí být stejného typu.

Konfigurace RTD modulu EM 231

DIP přepínače umožňují zvolit typ RTD, zapojení, teplotní stupnici a směr při poruše čidla. DIP přepínače jsou umístěny na spodní straně modulu a znázorněny na obrázku A-18. Aby nastavení DIP přepínače nabyla platnosti, musíte vypnout a znovu zapnout programovatelný automat nebo odpojit napájení 24 V.

Zvolte typ RTD nastavením DIP přepínačů č. 1, 2, 3, 4 a 5 tak, aby odpovídaly danému RTD, jak je to uvedeno v tabulce A-29. Nastavení ostatních DIP přepínačů viz tabulku A-30.

Konfigurace
↑1 – zapnuto
↓0 – vypnuto



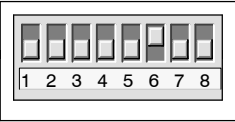
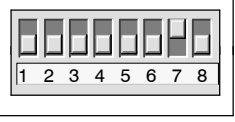
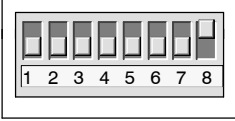
Obr. A-18

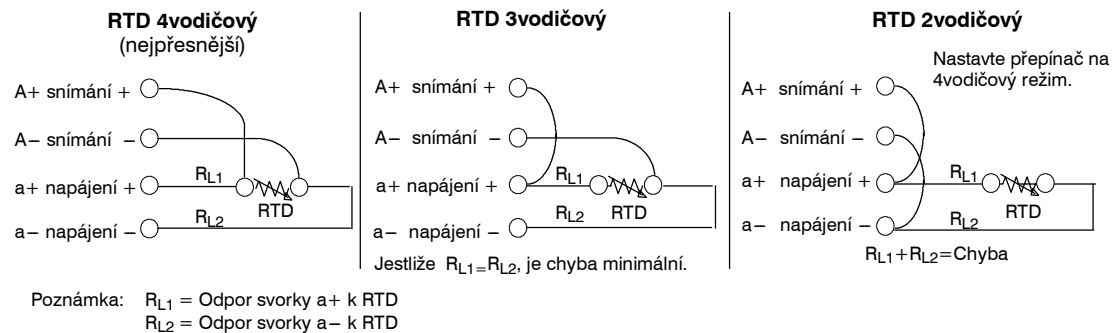
DIP přepínače pro RTD modul EM 231

Tabulka A-29 Volba typu RTD: DIP přepínače 1 až 5

Typ a alfa RTD	SW1	SW2	SW3	SW4	SW5	Typ a alfa RTD	SW1	SW2	SW3	SW4	SW5
100Ω Pt 0.003850 (Implicitní)	0	0	0	0	0	100Ω Pt 0.00302	1	0	0	0	0
200Ω Pt 0.003850	0	0	0	0	1	200Ω Pt 0.003902	1	0	0	0	1
500Ω Pt 0.003850	0	0	0	1	0	500Ω Pt 0.003902	1	0	0	1	0
1000Ω Pt 0.003850	0	0	0	1	1	1000Ω Pt 0.003902	1	0	0	1	1
100Ω Pt 0.003920	0	0	1	0	0	REZERVA	1	0	1	0	0
200Ω Pt 0.003920	0	0	1	0	1	100Ω Ni 0.00672	1	0	1	0	1
500Ω Pt 0.003920	0	0	1	1	0	120Ω Ni 0.00672	1	0	1	1	0
1000Ω Pt 0.003920	0	0	1	1	1	1000Ω Ni 0.00672	1	0	1	1	1
100Ω Pt 0.00385055	0	1	0	0	0	100Ω Ni 0.006178	1	1	0	0	0
200Ω Pt 0.00385055	0	1	0	0	1	120Ω Ni 0.006178	1	1	0	0	1
500Ω Pt 0.00385055	0	1	0	1	0	1000Ω Ni 0.006178	1	1	0	1	0
1000Ω Pt 0.00385055	0	1	0	1	1	10000Ω Pt 0.003850	1	1	0	1	1
100Ω Pt 0.003916	0	1	1	0	0	10Ω Cu 0.004270	1	1	1	0	0
200Ω Pt 0.003916	0	1	1	0	1	150Ω plný rozsah, odpor	1	1	1	0	1
500Ω Pt 0.003916	0	1	1	1	0	300Ω plný rozsah, odpor	1	1	1	1	0
1000Ω Pt 0.003916	0	1	1	1	1	600Ω PHYS odpor	1	1	1	1	1

Tabulka A-30 Nastavení DIP přepínačů pro RTD

Přepínač č. 6	Detekce přerušného vodiče	Nastavení	Popis
SW6  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	Nahoru po stupnici (+3276,7 stupňů)	0	Indikuje kladnou hodnotu na přerušném vodiči
	Dolů po stupnici (–3276,8 stupňů)	1	Indikuje zápornou hodnotu na přerušném vodiči
Přepínač č. 7	Teplotní stupnice	Nastavení	Popis
SW7  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	Celsius (°C)	0	RTD modul EM 231 může udávat teplotu ve stupních Celsia nebo Fahrenheita. Převod stupňů Celsia na Fahrenheita se provádí v modulu.
	Fahrenheit (°F)	1	
Přepínač č. 8	Schéma zapojení	Nastavení	Popis
SW8  Nastavení ↑ 1 – zapnuto ↓ 0 – vypnuto	3vodičový	0	RTD modul můžete připojit ke snímači třemi způsoby (ukázáno na obrázku). (Nejpřesnější je 4vodičový.) Nejméně přesný je 2vodičový, který lze doporučit pouze tehdy, pokud je ve vaší aplikaci možné ignorovat chyby způsobené zapojením.
	2vodičový nebo 4vodičový	1	



Obr. A-19 Připojení RTD ke snímači 4, 3 a 2 vodiči

Stavové indikátory RTD modulu EM 231

RTD modul dodává programovatelnému automatu hodnoty word, které představují teploty nebo chyby. Stavové bity indikují chybu rozsahu a poruchu napájení modulu. LED diody ukazují stav modulu. Uživatelský program by měl mít logiku, která by detekovala chyby a reagovala tak, jak to dané použití vyžaduje. Tabulka A-31 uvádí stavové indikátory RTD modulu EM 231.



Tip

Datový formát kanálu je dvojkový doplněk 16bitových word. Teplota je uváděna v jednotkách 0,1 stupně. (Jestliže je například naměřená teplota 100,2 stupňů, udává modul 1002.) Hodnoty odporů jsou vynásobeny koeficientem 27648. Například 75 % plného rozsahu uvádí modul jako 20736.
($225\Omega / 300\Omega * 27648 = 20736$)

Tabulka A-31 Stavové indikátory RTD modulu EM 231

Chyba	Data v kanálu	SF LED Červená	24 V LED Zelená	Stavový bit rozsahu ¹	Chyba uživatelského napájení 24 V DC ²
Bez chyby	Data z převodu	NESVÍTÍ	SVÍTÍ	0	0
Chybí 24 V	32766	NESVÍTÍ	NESVÍTÍ	0	1
SW zjistil přerušný vodič	-32768/32767	BLIKÁ	SVÍTÍ	1	0
Vstup mimo rozsah	-32768/32767	BLIKÁ	SVÍTÍ	1	0
Diagnostická chyba ³	0000	SVÍTÍ	NESVÍTÍ	0	poznámka ³

¹ Stavový bit rozsahu je bit 3 v bytu chybového registru modulu (SMB9 pro modul 1, SMB11 pro modul 2, atd.)

² Stavový bit chyby napájení je bit 2 v bytu chybového registru modulu (například SMB 9, SMB 11, viz přílohu D)

³ Diagnostické chyby způsobují chybu konfigurace modulu. Stavový bit chyby napájení může, ale nemusí být nastaven před chybou konfigurace modulu.

Data v kanálech se aktualizují každých 405 milisekund, pokud načte programovatelný automat data. Pokud automat data nenačte během doby jedné aktualizace, uvádí modul stará data až do příští aktualizace modulu. Aby data v kanálu zůstávala aktuální, doporučuje se, aby program automatu načítal data nejméně tak často, jako je rychlost aktualizace modulu.



Tip

Při používání RTD modulu určitě zablokujte analogovou filtraci programovatelného automatu. Analogová filtrace může zabránit včasnému rozpoznání chyby.

Detekce přerušného vodiče se provádí interním softwarem RTD modulu. Vstupy mimo rozsah jsou signalizovány a přerušný vodič je indikován jako porucha snímače. Detekce přerušného vodiče trvá minimálně tři programové cykly modulu a může trvat i déle, podle toho, který vodič je přerušný. Přerušné vodiče napájení+ nebo napájení- snímače jsou detekované v minimální době. Přerušné vodiče snímání+ a/nebo snímání- z čidla mohou být detekované za 5 i více sekund. Přerušné vodiče snímání mohou náhodně dávat platná data, s občasnou detekcí přerušného vodiče, obzvláště v prostředí s elektrickým šumem. Elektrický šum může také prodloužit dobu potřebnou pro detekci přerušného vodiče. Po ohlášení platných dat se doporučuje zablokovat indikaci přerušného vodiče a hodnoty mimo rozsah v uživatelském programu.

Rozsahy RTD modulu EM 231

Rozsahy teplot a přesnost pro každý typ RTD modulu EM 231 jsou uvedeny v tabulkách A-32 a A-33.

Tabulka A-32 Rozsahy teploty (°C) a přesnost pro různé typy RTD

Systémové word (1 číslice = 0,1 °C)		Pt10000	Pt100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000	Cu9.035	0 – 150Ω	0 – 300Ω	0 – 600Ω	
Desítkové	Šetnáctkové								
32767	7FF.								
32766	7FFE					↑	↑	↑	
32511	7EFF					176.383 Ω	352.767 Ω	705.534 Ω	
29649	6C01					150.005 Ω	300.011 Ω	600.022 Ω	
27648	6C00					150.000 Ω	300.000 Ω	600.000 Ω	
25000	61A8								↑
18000	4650								OR
15000	3A98								
13000	32C8	↑	↑						
10000	2710	1000.0 °C	1000.0 °C						
8500	2134		850.0 °C						
6000	1770	600.0 °C							
3120	0C30			↑	312.0 °C				
2950	0B86			295.0 °C					
2600	0A28				260.0 °C				
2500	09C4			250.0 °C					
1	0001	0.1 °C	0.1 °C	0.1 °C	0.1 °C	0.005 Ω	0.011 Ω	0.022 Ω	
0	0000	0.0 °C	0.0 °C	0.0 °C	0.0 °C	0.000 Ω	0.000 Ω	0.000 Ω	
-1	FFFF	-0.1 °C	-0.1 °C	-0.1 °C	-0.1 °C	(záporné hodnoty nejsou přípustné)			
-600	FDA8			-60.0 °C		↓	↓	↓	N R
-1050	FBE6			-105.0 °C					
-2000	F830	-200.0 °C	-200.0 °C	↓	-200.0 °C				
-2400	F6A0				-240.0 °C				
-2430	F682	-243.0 °C	-243.0 °C		↓				
		↓	↓						
-5000	EC78								
-6000	E890								UR
-10500	D6FC								↓
-12000	D120								
-20000	4E20								
-32767	8001								
-32768	8000								
Přesnost v celém intervalu		±0.4 %	±0.1 %	±0.2 %	±0.5 %	±0.1 %	±0.1 %	±0.1 %	
Přesnost (jmenovitý rozsah)		±4 °C	±1 °C	±0.6 °C	±2.8 °C	±0.15 Ω	±0.3 Ω	±0.6 Ω	
*OF = Přetečení; OR = Přesah; KNURL = Jmenovitý rozsah; OUR = Pod rozsahem; OUI = Podtečení									
↑ nebo ↓ indikuje, že všechny analogové hodn. větší než mezní hlásí zvolenou hodn. pro poruchu čidla, 32767 (0x7FF.) nebo -32768 (0x8000).									

A

Tabulka A-33 Rozsahy teplot (°F) pro různé typy RTD

Systémové word (1 číslice = 0,1 °F)		PT1000	PT100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000	Cu 9.035	
Desítkové	Šestnáctkové					
32767	7FF.					
32766	7PHAGE					
18320	4790	1832.0 °F	1832.0 °F			
						↑ Přesah
15620	3D04	1562.0 °F				
11120	2B70	1112.0 °F				
5936	1730					↑ 593.6 °F
5630	15FE	563.0 °F				500.0 °F
5000	1388					
4820	12D4	482.0 °F				
						Normální rozsah
1	0001	0.1 °F	0.1 °F	0.1 °F	0.1 °F	
0	0000	0.0 °F	0.0 °F	0.0 °F	0.0 °F	
−1	FFFF	−0.1 °F	−0.1 °F	−0.1 °F	−0.1 °F	
						−76.0 °F
−760	FD08					
−1570	F9DE	−157.0 °F				−328.0 °F
−3280	F330	−328.0 °F	−328.0 °F			
						−400.0 °F
−4000	F060					
−4054	F02A	−405.4 °F	−405.4 °F			
		↓	↓	↓		
−5000	EC78					Pod rozsahem
−6000	E890					
−10500	D6FC					
−32767	8001					↓
−32768	8000					

↑ nebo ↓ indikuje, že všechny analogové hodn. větší než mezní hlásí zvolenou hodn. pro poruchu čidla, 32767 (0x7FFF) nebo −32768 (0x8000).

Specifikace modulu EM 277 PROFIBUS–DP

Tabulka A-34 Objednací číslo modulu EM 277 PROFIBUS–DP

Objednací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Odnímatelná svorkovnice
6ES7 277–0AA22–0XA0	EM 277 PROFIBUS–DP	–	–	Ne

Tabulka A-35 Obecná specifikace modulu EM 277 PROFIBUS–DP

Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Ztráty	Požadavky V DC +5 V DC	Požadavky V DC +24 V DC
6ES7 277–0AA22–0XA0	EM 277 PROFIBUS–DP	71 x 80 x 62	175 g	2,5 W	150 mA	Viz níže

Tabulka A-36 Specifikace modulu EM 277 PROFIBUS–DP

Obecné	6ES7 277–0AA22–0XA0
Počet portů	1
Elektrické rozhraní	RS–485
Přenosové rychlosti PROFIBUS–DP/MPI (nastaveno automaticky)	9,6; 19,2; 45,45; 93,75; 187,5 a 500 kbaud; 1; 1,5; 3; 6 a 12 Mbaud
Protokoly	PROFIBUS–DP slave a MPI slave
Délka kabelu	
Až do 93,75 kbaud	1200 m
187,5 kbaud	1000 m
500 kbaud	400 m
1 až 1,5 Mbaud	200 m
3 až 12 Mbaud	100 m
Možnosti sítě	
Nastavení adresy stanice	0 až 99 (nastavuje se otočnými přepínači)
Maximální počet stanic na segment	32
Maximální počet stanic v síti	126, až 99 stanic EM 277
Připojení MPI	celkem 6, 2 vyhrazeny (1 pro PG a 1 pro OP)
Požadavky na zdroj 24 V DC	
Rozsah napětí	20,4 až 28,8 V DC (třída 2 nebo napájení ze snímačů programovatelného automatu)
Maximální proud Pouze modul s aktivním portem Přidáno 90 mA zátěže na 5V portu Přidáno 120 mA zátěže na 24V portu	30 mA 60 mA 180 mA
Zvlnění šumu (<10 MHz)	<1 V špička–špička (maximálně)
Napětové oddělení (okolí od logiky) ¹	500 V AC po dobu 1 minuty
Napájení 5 V DC na komunikačním portu	
Maximální proud na port	90 mA
Napětové oddělení (24 V DC od logiky)	500 V AC po dobu 1 minuty
Napájení 24 V DC na komunikačním portu	
Rozsah napětí	20,4 až 28,8 V DC
Maximální proud na port	120 mA
Proudové omezení	0,7 až 2,4 A
Napětové oddělení	Není odděleno, stejný obvod jako vstupní 24 V DC

¹ Logika modulu není napájena ze zdroje 24 V DC. 24 V DC napájí komunikační port.

CPU S7-200, které podporují inteligentní moduly

Modul EM 277 PROFIBUS – DP slave je inteligentní rozšiřovací modul určený pro spolupráci s CPU S7-200 uvedenými v tabulce A-37.

Tabulka A-37 Kompatibilita modulu EM 277 PROFIBUS – DP s CPU S7-200

CPU	Popis
Model CPU 222 verze 1.10 nebo vyšší	CPU 222 DC/DC/DC
	CPU 222 AC/DC/Relé
Model CPU 224 verze 1.10 nebo vyšší	CPU 224 DC/DC/DC
	CPU 224 AC/DC/Relé
Model CPU 226 verze 1.00 nebo vyšší	CPU 226 DC/DC/DC
	CPU 226 AC/DC/Relé
Model CPU 226XM verze 1.00 nebo vyšší	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/Relé

Přepínače adres a stavové LED

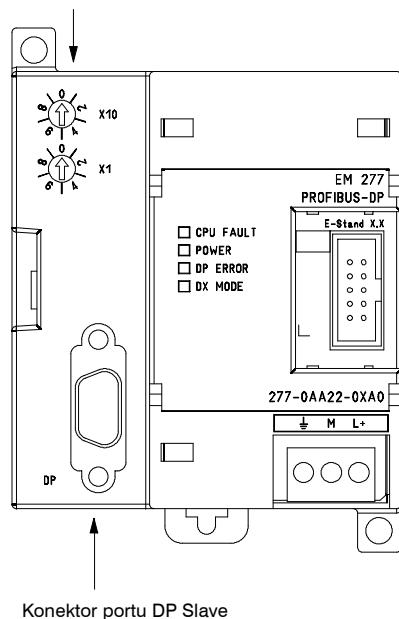
Přepínače adres a LED diody jsou umístěny na přední straně modulu a jsou znázorněny na obrázku A-20, kde je zobrazeno i rozmístění vývodů konektoru portu DP slave.

Pohled na EM 277 PROFIBUS – DP zepředu

Přepínače adres:

x10 = nastavuje nejvyšší platnou číslici adresy

x1 = nastavuje nejnižší platnou číslici adresy



9kolíkový subminiaturní konektor typu D – rozmístění vývodů

Vývod č.	Popis
1	Ochranné uzemnění rozváděče propojené s rámem konektoru
2	Střední vodič 24 V (stejně jako M na svorkovnici)
3	Napětově oddělený signál B (RxD/TxD+)
4	Napětově oddělená výzva k vysílání (úroveň TTL)
5	Napětově oddělený střední vodič +5 V
6	Napětově oddělené +5 V (maximálně 90 mA)
7	+24 V (maximálně 120 mA, s ochrannou diodou se závěrným napětím)
8	Napětově oddělený signál A (RxD/TxD-)
9	Nezapojeno

Poznámka: Napětově odděleno znamená izolaci 500 V od digitální logiky a napájecího vstupu.

Konektor portu DP Slave

Obr. A-20 EM 277 PROFIBUS – DP

Komunikace se standardními distribuovanými periferními zařízeními (DP)

PROFIBUS – DP (neboli standard DP) je protokol pro dálkovou I/O komunikaci definovaný evropskou normou EN 50170. Zařízení, která dodržují tuto normu, jsou kompatibilní, i když jsou vyráběna různými firmami. DP znamená distribuovaná periferní zařízení, tj. dálkové vstupy a výstupy. PROFIBUS je zkráceno z Process Field Bus.

Modul EM 277 PROFIBUS–DP má implementovaný standardní protokol DP tak, jak je definován pro zařízení typu slave v následujících normách pro komunikační protokoly:

- ☐ EN 50 170 (PROFIBUS) popisuje přístup po sběrnici, protokol přenosu a udává vlastnosti přenosového média.
- ☐ EN 50 170 (Standard DP) popisuje vysokorychlostní cyklickou výměnu dat mezi zařízeními DP master a DP slave. Tato norma definuje postupy pro konfiguraci a přiřazení parametrů, vysvětluje fungování cyklické výměny dat s distribuovanými vstupy a výstupy a obsahuje seznam podporovaných diagnostických funkcí.

DP master je nakonfigurován tak, aby poznal adresy a typy zařízení slave a jakoukoliv informaci o přiřazení parametrů, které zařízení slave vyžadují. Master obsahuje i informace o tom, kam umístit data, která čte ze slave (vstupy), a odkud získat data, která zapisuje na slave (výstupy). DP master zakládá síť a pak inicializuje svá zařízení DP slave. Master zapisuje informace o přiřazení parametrů a konfiguraci vstupů a výstupů na zařízení slave. Potom master přečte diagnostiku ze zařízení DP slave, aby ověřil, že toto zařízení přijalo parametry a konfiguraci vstupů a výstupů. Následně master zahájí výměnu I/O dat se zařízením slave. Při každé transakci se zařízením slave zapisuje výstupy a čte vstupy. Režim výměny dat pokračuje po neomezenou dobu. Zařízení slave mohou uvědomit master, když dojde k výjimečnému stavu; master pak čte diagnostické informace ze zařízení slave.

Po tom, co DP master zapsal parametry a konfiguraci vstupů a výstupů na DP slave a slave akceptoval parametry a konfiguraci od masteru, který vlastní toto zařízení slave. Slave přijímá požadavky na zápis pouze od svého masteru. Ostatní master v dané síti mohou číst vstupy a výstupy tohoto zařízení slave, ale nemohou do něj zapisovat.

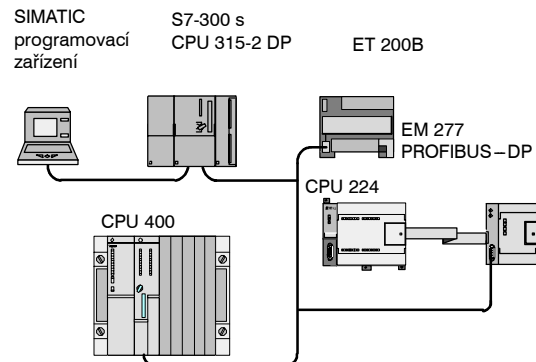
Použití EM 277 pro připojení S7-200 jako zařízení DP slave

CPU S7-200 může být připojen k síti PROFIBUS–DP prostřednictvím rozšiřovacího modulu EM 277 PROFIBUS–DP slave. Modul EM 277 je připojen k CPU S7-200 přes sériovou sběrnici. Síť PROFIBUS je k modulu EM 277 PROFIBUS–DP připojena přes jeho DP komunikační port. Tento port pracuje při kterékoliv přenosové rychlosti PROFIBUS v rozsahu od 9600 baudů do 12 Mbaudů. Podporované přenosové rychlosti najdete ve specifikaci modulu EM 277 PROFIBUS–DP.

Jako zařízení DP slave přijímá modul EM 277 od masteru několik různých konfigurací vstupů a výstupů, což umožňuje přizpůsobit množství přenášených dat tak, aby to odpovídalo požadavkům daného použití. Na rozdíl od mnoha DP zařízení nepřenáší modul EM 277 pouze I/O data. Vstupy, hodnoty čítače, hodnoty časovače nebo jiné vypočítané hodnoty je možné přenést do zařízení master tak, že se data nejprve přesunou do variabilní paměti CPU S7-200. Obdobně jsou data ze zařízení master ukládána do variabilní paměti CPU S7-200 a mohou se přesouvat do jiných datových oblastí.

DP port modulu EM 277 PROFIBUS–DP může být připojen k zařízení DP master v síti a přitom ve stejné síti komunikovat jako MPI slave s jinými zařízeními typu master, jako jsou programovací zařízení SIMATIC nebo CPU S7-300/S7-400. Obrázek A-21 znázorňuje síť PROFIBUS s CPU 224 a modulem EM 277 PROFIBUS–DP.

- ☐ CPU 315–2 je DP master a byl nakonfigurován programovacím zařízením SIMATIC s programovacím softwarem STEP 7.
- ☐ CPU 224 je DP slave a vlastní ho CPU 315–2. Modul ET 200 je také zařízení slave, které vlastní CPU 315–2.
- ☐ CPU S7-400 je připojena k síti PROFIBUS a čte data z CPU 224 pomocí instrukcí XGET v uživatelském programu CPU S7-400.

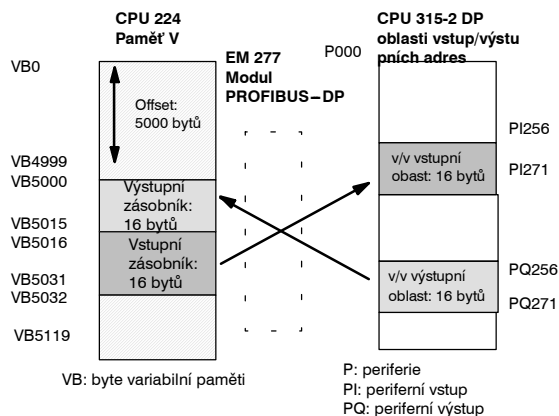


Obr. A-21 Modul EM 277 PROFIBUS–DP a CPU 224 v síti PROFIBUS

Konfigurace

Chcete-li používat EM 277 PROFIBUS–DP jako DP slave, musíte nastavit adresu stanice DP portu tak, aby odpovídala adrese v konfiguraci zařízení master. Adresa stanice se nastavuje otočnými přepínači na modulu EM 277. Po provedení změny přepínačem je třeba CPU vypnout a opět zapnout, aby nová adresa zařízení slave nabyla platnosti.

Master si vyměňuje data s každým svým slave tak, že posílá informace ze své výstupní oblasti do výstupního zásobníku zařízení slave (nazývaného “Receive mailbox” (Schránka pro příjem)). Zařízení slave odpovídá na zprávu vrácením vstupního zásobníku (nazývaného “Send mailbox” (Schránka pro odesílání)), který master uloží ve vstupní oblasti.



Obr. A-22 Paměť V a oblast adres vstupů a výstupů

Obrázek A-22 ukazuje příklad paměti V a oblastí adres vstupů a výstupů zařízení PROFIBUS–DP master.

EM 277 PROFIBUS–DP může být nakonfigurován DP master tak, aby přijímal výstupní data od zařízení master a vrátil mu data vstupní. Zásobníky pro výstupní a vstupní data jsou umístěny ve variabilní paměti (paměť V) CPU S7-200. Když konfiguruje zařízení DP master, definujete místo bytu v paměti V, kde má začínat zásobník pro výstupní data, jako součást informace o přiřazení parametrů pro EM 277. Definujete i konfiguraci vstupů a výstupů jako množství výstupních dat, které se zapisuje do CPU S7-200, a jako množství vstupních dat, které se vrací z CPU S7-200. EM 277 určuje velikost vstupního a výstupního zásobníku z konfigurace vstupů a výstupů. DP master zapisuje informace o přiřazení parametrů a konfiguraci vstupů a výstupů do modulu EM 277 PROFIBUS DP. Potom EM 277 přenáší do CPU S7-200 adresu a délku vstupních a výstupních dat v paměti V.

Obrázek A-22 ukazuje paměťový model paměti V v CPU 224 a oblast adres vstupů a výstupů CPU DP mastera. V tomto příkladu DP master definoval vstupní/výstupní konfiguraci 16 výstupních bytů a 16 vstupních bytů a offset V paměti 5000. Výstupní a vstupní zásobník v CPU 224 jsou oba 16 bytů dlouhé (dáno v konfiguraci vstupů a výstupů). Zásobník pro výstupní data začíná na V5000; vstupní zásobník následuje bezprostředně za výstupním a začíná na V5016. Výstupní data (ze zařízení master) se zapisují do paměti V od V5000. Vstupní data (do zařízení master) se čtou z paměti V od V5016.



Tip

Pokud pracujete s datovou jednotkou (konzistentní data) tři byty nebo s datovými jednotkami většími než čtyři byty, je nutné použít SFC14 na čtení vstupů DP slave a SFC15 na adresaci výstupů DP slave. Více informací najdete v *Referenčním manuálu System software for S7-300 and S7-400 a System and Standard Function*.

Tabulka A-38 uvádí konfigurace, které jsou podporovány modulem EM 277 PROFIBUS–DP. Implicitní konfigurace modulu EM 277 je dva word pro vstup a dva word pro výstup.

Tabulka A-38 Volby konfigurace EM 277

Konfigurace	Vstupy masteru	Výstupy masteru	Konzistence dat
1	1 word	1 word	Konzistence po wordech
2	2 word	2 word	
3	4 word	4 word	
4	8 word	8 word	
5	16 word	16 word	
6	32 word	32 word	
7	8 word	2 word	
8	16 word	4 word	
9	32 word	8 word	
10	2 word	8 word	
11	4 word	16 word	
12	8 word	32 word	
13	2 byty	2 byty	Konzistence po bytech
14	8 bytů	8 bytů	
15	32 bytů	32 bytů	
16	64 bytů	64 bytů	
17	4 byty	4 byty	Zásobníková konzistence
18	8 bytů	8 bytů	
19	12 bytů	12 bytů	
20	16 bytů	16 bytů	

Umístění vstupního a výstupního zásobníku v paměti V CPU S7-200 můžete nakonfigurovat libovolně. Implicitně nastavená adresa vstupního a výstupního zásobníku je VB0. Umístění vstupního a výstupního zásobníku je součástí informací o přiřazení parametrů, které master zapisuje do CPU S7-200. Master musí být konfigurován tak, aby rozpoznal svoje zařízení slave a do každého zapisoval požadované parametry a konfiguraci vstupů a výstupů.

Pro konfiguraci zařízení DP master použijte následující nástroje:

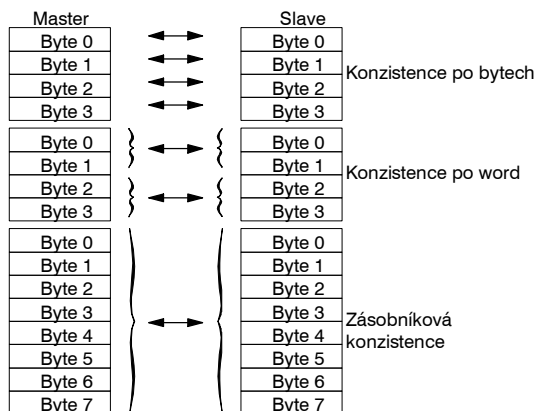
- ☐ Pro SIMATIC S5 master použijte software COM PROFIBUS Windows
- ☐ Pro SIMATIC S7 master použijte programovací software STEP 7
- ☐ Pro SIMATIC 505 master použijte COM PROFIBUS a TISOFT2 nebo SoftShop

Podrobné informace o používání těchto konfiguračních a programovacích balíčků najdete v příručkách pro daný software. Podrobné informace o síti PROFIBUS a jejích komponentách najdete v *Manuálu ET200 Distributed I/O System*.

Konzistence dat

PROFIBUS podporuje tři typy konzistence dat:

- ☐ Konzistence po bytech zajišťuje, že jsou byty přenášeny jako celé jednotky.
- ☐ Konzistence po wordech zajišťuje, že přenosy word nemohou být přerušeny jinými procesy v CPU (dva byty, ze kterých se word skládá, jsou vždy přenášeny dohromady a nemohou být rozděleny). Konzistenci po wordech použijte, pokud mají přenášená data hodnotu integer.
- ☐ Zásobníková konzistence zajišťuje, že je celý datový zásobník přenášen jako jeden celek a není přerušovaný žádným jiným procesem v CPU. Tato konzistence se používá, pokud mají přenášená data hodnotu double word, obsahují plovoucí desetinnou čárku nebo patří-li celá skupina hodnot k jednomu výpočtu nebo položce.



Obr. A-23 Konzistence dat; po bytech, wordech, zásobníková

Konzistence dat se nastavuje jako součást konfigurace vstupů a výstupů zařízení master. Zvolená konzistence dat se zapisuje do DP slave jako součást jeho inicializace. Obě zařízení, DP master i DP slave, používají zvolenou konzistenci dat na zajištění nepřerušovaného vzájemného přenosu datových hodnot (byty, wordy nebo zásobníky). Různé typy konzistence jsou uvedeny na obrázku A-23.

Upozornění pro uživatelský program

Po úspěšné konfiguraci modulu EM 277 PROFIBUS–DP zařízením DP master vstoupí EM 277 a DP master do režimu výměny dat. V tomto režimu master zapisuje výstupní data do modulu EM 277 PROFIBUS–DP a modul EM 277 odpovídá aktuálními vstupními daty CPU S7-200. EM 277 nepřetržitě aktualizuje svoje vstupy z CPU S7-200, aby mohl poskytovat nejnovější vstupní data zařízení DP masteru. Pak modul přenáší výstupní data do CPU S7-200. Výstupní data ze zařízení master jsou uložena do paměti V (výstupní zásobník) se začátkem na adrese, kterou DP master dodal během inicializace. Vstupní data pro master se čtou z míst paměti V (vstupní zásobník), která následují bezprostředně za výstupními daty.

Výstupní data ze zařízení master musí být přesunuta uživatelským programem v CPU S7-200 z výstupního zásobníku do datových oblastí, kde mají být užívána. Obdobně musí být vstupní data pro master přesunuta z různých datových oblastí do vstupního zásobníku pro přenos do masteru.

Výstupní data z DP masteru jsou uložena do paměti V okamžitě po provedení uživatelského programu v rámci programového cyklu. Současně jsou do masteru kopírována vstupní data z paměti V do EM 277 pro přesun.

Výstupní data z masteru se zapisují do paměti V pouze tehdy, když toto zařízení dodává nová data.

Vstupní data pro master jsou do něj přenášena při další vzájemné datové výměně.

V době tvorby uživatelského programu pro CPU S7-200 CPU musí být známa počáteční adresa datových zásobníků v paměti V a jejich velikost.

Stavové informace

Každému inteligentnímu modulu je přiděleno 50 bytů speciální paměti (SM) na základě jeho fyzického umístění. Modul aktualizuje místa v paměti SM odpovídající vzájemné poloze modulu a CPU (vzhledem k ostatním modulům). Pokud je to první modul, aktualizuje SMB200 až SMB249. Pokud je to druhý modul, aktualizuje SMB250 až SMB299 atd. Viz tabulku A-39.

Poznámka

Způsob přiřazování míst v paměti SM inteligentním modulům se pro verzi 2.2 a pozdější změnil.

Používáte – li verzi CPU starší než 2.2, aby byla zajištěna kompatibilita, musíte umístit všechny inteligentní moduly do pozic těsně vedle CPU před všechny ostatní moduly.

Tabulka A-39 Speciální paměťové byty SMB200 až SMB549

Speciální paměťové byty SMB200 až SMB549						
Inteligentní modul na pozici 0	Inteligentní modul na pozici 1	Inteligentní modul na pozici 2	Inteligentní modul na pozici 3	Inteligentní modul na pozici 4	Inteligentní modul na pozici 5	Inteligentní modul na pozici 6
SMB200 až SMB249	SMB250 až SMB299	SMB300 až SMB349	SMB350 až SMB399	SMB400 až SMB449	SMB450 až SMB499	SMB500 až SMB549

Tato místa v paměti SM ukazují implicitní hodnoty, pokud nebyla navázána DP komunikace se zařízením master. Po tom, co master zapsal do modulu EM 277 PROFIBUS–DP parametry a konfiguraci vstupů a výstupů, ukazují tato místa v paměti SM konfiguraci nastavenou DP masterem. Dříve než použijete informace z míst v paměti SM uvedených v tabulce A-40 nebo data v zásobníku paměti V, musíte zkontrolovat stavový byte protokolu (například SMB224 pro místo 0). Tím se ujistíte, že je EM 277 v danou dobu v režimu výměny informací se zařízením master.

**Tip**

Velikost vstupních/výstupních zásobníků EM 277 PROFIBUS–DP nebo jejich umístění není možné konfigurovat zápisem do paměti SM. Pro provoz v režimu DP může modul EM 277 PROFIBUS–DP nakonfigurovat pouze DP master.

Tabulka A-40 Speciální paměťové byty pro EM 277 PROFIBUS–DP

Inteligentní modul na pozici 0	...	Inteligentní modul na pozici 6	Popis																							
SMB200 až SMB215	...	SMB500 až SMB515	Název modulu (16 ASCII znaků) “EM277 ProfibusDP”																							
SMB216 až SMB219	...	SMB516 až SMB519	Číslo verze firmware (4 ASCII znaky) xxxx																							
SMW220	...	SMW520	Chybový kód 16#0000 Bez chyby 16#0001 Není napájení ze strany uživatele 16#0002 až 16#FFFF Rezervováno																							
SMB222	...	SMB522	Adresa stanice DP slave, jak je nastavena přepínači adres (0 – 99 desítkově)																							
SMB223	...	SMB523	Rezervováno																							
SMB224	...	SMB524	Stavový byte standardního DP protokolu MSB LSB <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>S1</td><td>S0</td></tr></table> <table><tr><td>S1</td><td>S0</td><td>Popis stavového bytu standardního DP</td></tr><tr><td>0</td><td>0</td><td>Komunikace DP nebyla od zapnutí zdroje inicializována</td></tr><tr><td>0</td><td>1</td><td>Zjištěna chyba konfigurace/přiřazení parametrů</td></tr><tr><td>1</td><td>0</td><td>Aktuálně v režimu výměny dat</td></tr><tr><td>1</td><td>1</td><td>Výpadek režimu výměny dat</td></tr></table>	0	0	0	0	0	0	S1	S0	S1	S0	Popis stavového bytu standardního DP	0	0	Komunikace DP nebyla od zapnutí zdroje inicializována	0	1	Zjištěna chyba konfigurace/přiřazení parametrů	1	0	Aktuálně v režimu výměny dat	1	1	Výpadek režimu výměny dat
0	0	0	0	0	0	S1	S0																			
S1	S0	Popis stavového bytu standardního DP																								
0	0	Komunikace DP nebyla od zapnutí zdroje inicializována																								
0	1	Zjištěna chyba konfigurace/přiřazení parametrů																								
1	0	Aktuálně v režimu výměny dat																								
1	1	Výpadek režimu výměny dat																								
SMB225	...	SMB525	Protokol DP Standard – adresa masteru, kterému patří slave (0 až 126)																							
SMW226	...	SMW526	Protokol DP Standard – adresa výstupního zásobníku v paměti V jako posunutí od VB0.																							
SMB228	...	SMB528	Protokol DP Standard – počet bytů výstupních dat																							
SMB229	...	SMB529	Protokol DP Standard – počet bytů vstupních dat																							
SMB230 až SMB249	...	SMB530 až SMB549	Rezervováno – vymazáno při zapnutí																							

Poznámka: Místa v paměti SM jsou aktualizována pokaždé, když modul DP slave přijme informace o konfiguraci/přiřazení parametrů. Tato místa jsou aktualizována dokonce i v případě zjištění chyby konfigurace/přiřazení parametrů. Údaje jsou vymazány při každém zapnutí.

Stavové LED diody EM 277 PROFIBUS–DP

Na předním panelu modulu EM 277 PROFIBUS–DP jsou čtyři stavové kontrolky (LED), které indikují provozní stav DP portu:

- ☐ Po zapnutí CPU S7-200 zůstává LED DX MODE nerozsvícená, dokud nedojde k pokusu o DP komunikaci.
- ☐ Po úspěšné inicializaci DP komunikace (modul EM 277 PROFIBUS–DP vstoupil do režimu výměny dat se zařízením master) začne LED DX MODE svítit zeleně a zůstane rozsvícená, dokud není ukončen režim výměny dat.
- ☐ Dojde-li k přerušení DP komunikace, které způsobí, že modul EM 277 opustí režim výměny dat, LED DX MODE zhasne a rozsvítí se červená kontrolka DP ERROR. Tento stav setrvává, dokud není vypnuta CPU S7-200 nebo není obnovena výměna dat.
- ☐ Pokud je chyba v konfiguraci vstupů a výstupů nebo v informacích o parametrech, které DP master zapisuje do modulu EM 277, LED DP ERROR bliká červeně.
- ☐ Pokud není přivedeno uživatelské napájení 24 V DC, nebude svítit LED POWER.

Tabulka A-41 shrnuje významy stavových LED EM 277.

Tabulka A-41 Stavové LED modulu EM 277 PROFIBUS–DP

LED	NESVÍTÍ	ČERVENÁ	BLIKAJÍCÍ ČERVENÁ	ZELENÁ
CPU FAULT	Modul je v pořádku	Interní porucha modulu	--	--
POWER	Není uživatelské napájení 24 V DC	--	--	Uživatelské napájení 24 V DC v pořádku
DP ERROR	Bez chyby	Opuštěn režim výměny dat	Chyba přiřazení parametrů/konfigurace	--
DX MODE	Není v režimu výměny dat	--	--	V režimu výměny dat

Poznámka: Když je modul EM 277 PROFIBUS–DP používán výhradně jako MPI slave, je rozsvícena pouze zelená LED POWER.

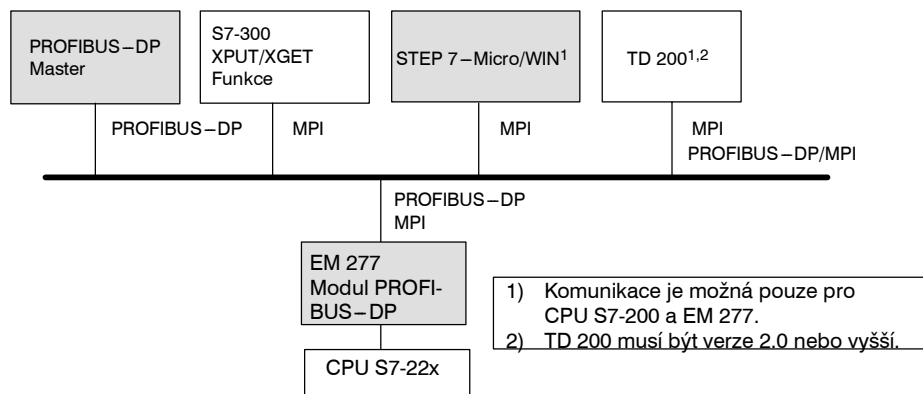
Další vlastnosti konfigurace

Modul EM 277 PROFIBUS–DP je možné použít jako komunikační rozhraní pro ostatní zařízení MPI master, bez ohledu na to, zda se používá jako PROFIBUS–DP slave. Modul může zajistit spojení mezi S7-300/400 a S7-200 pomocí funkcí XGET/XPUT v programu S7-300/400. STEP 7–Micro/WIN a síťová karta (jako například CP5611) používající sadu parametrů MPI nebo PROFIBUS, panel operátora (OP) nebo TD 200 (verze 2.0 nebo novější, objednáací číslo 6ES7 272–0AA20–0YA0) mohou komunikovat s automatem S7-200 prostřednictvím modulu EM 277 PROFIBUS–DP.

Modul EM 277 PROFIBUS–DP podporuje maximálně šest spojení (šest zařízení) kromě zařízení DP master. Jedno spojení je rezervováno pro programovací zařízení (PG) a jedno je rezervováno pro operátorský panel operátora (OP). Ostatní čtyři spojení mohou být použita libovolným zařízením MPI master. Aby mohl modul EM 277 PROFIBUS–DP komunikovat s více zařízeními master, musí všechna tato zařízení pracovat se stejnou přenosovou rychlostí. Jedna možná konfigurace sítě je znázorněna na obrázku A-24.

Pokud je modul EM 277 PROFIBUS–DP použit pro komunikaci MPI, musí MPI master, pro všechny zprávy posílané automatu S7-200, používat adresu stanice modulu, ke kterému je připojen. Zprávy MPI posílané modulu EM 277 PROFIBUS–DP jsou dále předávány automatu S7-200.

Modul EM 277 PROFIBUS–DP je typu slave a nemůže být použit pro komunikaci mezi programovatelnými automaty S7-200, které používají funkce NETR a NETW. Modul EM 277 PROFIBUS–DP také není možné použít pro komunikaci v režimu Freeport.



Obr. A-24 Sít' PROFIBUS-DP/MPI

Databázový soubor zařízení: GSD

Různá zařízení PROFIBUS mají různé vlastnosti. Tyto vlastnosti se liší co do funkčnosti (například počet vstupní/výstupních signálů a diagnostických zpráv) nebo parametrů sběrnice, jako je například přenosová rychlost a monitorování času. Dané parametry jsou pro každý typ zařízení a výrobce jiné a obvykle jsou dokumentovány v technickém manuálu. Aby se dosáhlo jednoduché konfigurace PROFIBUS, jsou vlastnosti konkrétního zařízení specifikovány v elektronické formě nazývané databázový soubor zařízení neboli soubor GSD. Konfigurační nástroje založené na souborech GSD umožňují jednoduchou integraci zařízení od různých výrobců do jedné sítě.

Databázový soubor zařízení obsahuje vyčerpávající popis vlastností zařízení v přesně definovaném formátu. Soubory GSD jsou výrobcem vypracovány pro každý typ zařízení a jsou k dispozici pro uživatele PROFIBUS. Soubor GSD umožňuje konfiguračnímu systému načíst vlastnosti zařízení PROFIBUS a použít tyto informace při konfiguraci sítě.

Nejnovější verze softwaru COM PROFIBUS nebo STEP 7 obsahují konfigurační soubory pro modul EM 277 PROFIBUS-DP. Pokud vaše verze softwaru neobsahuje konfigurační soubor pro EM 277, je nejnovější soubor GSD (SIEM089D.GSD) k dispozici na internetové adrese www.profibus.com.

Pokud používáte zařízení typu master od jiného výrobce než Siemens, najdete potom způsob konfigurace tohoto zařízení master pomocí souboru GSD v dokumentaci dodávané výrobcem.

```

;=====
; GSD File for the EM 277 PROFIBUS-DP with a DPC31
; MLFB : 6ES7 277-0AA2.-0XA0
; DATE : 26-March-2001
;=====
#Profibus_DP
;General parameters
GSD_Revision      = 1
Vendor_Name       = "Siemens"
Model_Name        = "EM 277 PROFIBUS-DP"
Revision          = "V1.02"
Ident_Number      = 0x089D
Protocol_Ident    = 0
Station_Type      = 0
FMS_supp          = 0
Hardware_Release  = "1.00"
Software_Release  = "1.02"
9.6_supp          = 1
19.2_supp         = 1
45.45_supp        = 1
93.75_supp        = 1
187.5_supp        = 1
500_supp          = 1
1.5M_supp         = 1
3M_supp           = 1
6M_supp           = 1
12M_supp          = 1
MaxTsdR_9.6       = 60
MaxTsdR_19.2      = 60
MaxTsdR_45.45     = 250
MaxTsdR_93.75     = 60
MaxTsdR_187.5     = 60
MaxTsdR_500       = 100
MaxTsdR_1.5M      = 150
MaxTsdR_3M        = 250
MaxTsdR_6M        = 450
MaxTsdR_12M       = 800
Redundancy        = 0
Repeater_Ctrl_Sig = 2
24V_Pins          = 2

; Slave-Specification:
OrderNumber="6ES7 277-0AA2.-0XA0"
Periphery="SIMATIC S5"
Slave_Family=10@TdrF@SIMATIC

Freeze_Mode_supp  = 1
Sync_Mode_supp    = 1
Set_Slave_Add_Supp = 0
Auto_Baud_supp    = 1
Min_Slave_Intervall = 1
Fail_Safe         = 0
Max_Diag_Data_Len = 6
Modul_Offset      = 0
Modular_Station   = 1
Max_Module        = 1
Max_Input_len     = 128
Max_Output_len    = 128
Max_Data_len      = 256

; UserPrmData-Definition
ExtUserPrmData=1 "I/O Offset in the V-memory"
Unsigned16 0 0-10239
EndExtUserPrmData
; UserPrmData: Length and Preset:
User_Prm_Data_Len=3
User_Prm_Data= 0,0,0
Max_User_Prm_Data_Len=3
Ext_User_Prm_Data_Const(0)=0x00,0x00,0x00
Ext_User_Prm_Data_Ref(1)=1

```

```

;=====
; Continuation of GSD File
;=====
; Module Definition List
Module = "2 Bytes Out/ 2 Bytes In      -" 0x31
EndModule
Module = "8 Bytes Out/ 8 Bytes In      -" 0x37
EndModule
Module = "32 Bytes Out/ 32 Bytes In    -"
0xC0,0x1F,0x1F
EndModule
Module = "64 Bytes Out/ 64 Bytes In    -"
0xC0,0x3F,0x3F
EndModule
Module = "1 Word Out/ 1 Word In        -" 0x70
EndModule
Module = "2 Word Out/ 2 Word In        -" 0x71
EndModule
Module = "4 Word Out/ 4 Word In        -" 0x73
EndModule
Module = "8 Word Out/ 8 Word In        -" 0x77
EndModule
Module = "16 Word Out/ 16 Word In      -" 0x7F
EndModule
Module = "32 Word Out/ 32 Word In      -"
0xC0,0x5F,0x5F
EndModule
Module = "2 Word Out/ 8 Word In        -"
0xC0,0x41,0x47
EndModule
Module = "4 Word Out/ 16 Word In       -"
0xC0,0x43,0x4F
EndModule
Module = "8 Word Out/ 32 Word In       -"
0xC0,0x47,0x5F
EndModule
Module = "8 Word Out/ 2 Word In        -"
0xC0,0x47,0x41
EndModule
Module = "16 Word Out/ 4 Word In       -"
0xC0,0x4F,0x43
EndModule
Module = "32 Word Out/ 8 Word In       -"
0xC0,0x5F,0x47
EndModule
Module = "4 Byte buffer I/O           -" 0xB3
EndModule
Module = "8 Byte buffer I/O           -" 0xB7
EndModule
Module = "12 Byte buffer I/O          -" 0xBB
EndModule
Module = "16 Byte buffer I/O          -" 0xBF
EndModule

```

Obr. A-25 Výpis souboru GSD pro modul EM 277 PROFIBUS

Vzorový program pro komunikaci DP s CPU

Vzorový program ve formátu STL pro modul PROFIBUS–DP umístěný na pozici 0 a pro CPU, který používá informace o DP portu v paměti SM, je uveden níže. Program určuje umístění zásobníků DP ze SMW226 a velikost zásobníků ze SMB228 a SMB229. Tyto informace jsou použity pro kopírování dat ve výstupním zásobníku do registru obrazu výstupů CPU. Obdobně jsou data v registru obrazu vstupů CPU kopírována do vstupního zásobníku v paměti V.

Poznámka

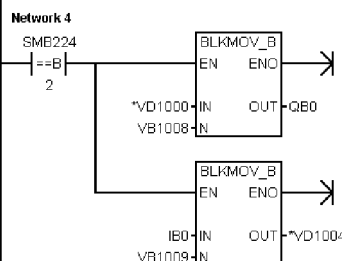
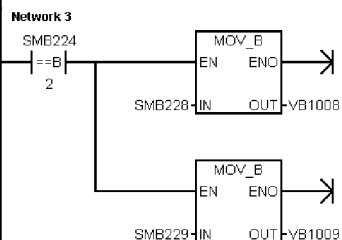
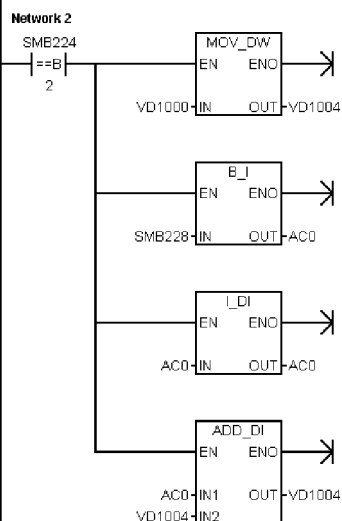
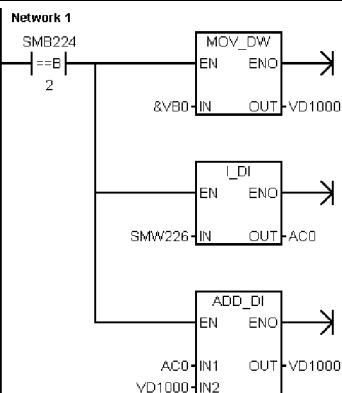
Způsob přiřazování míst v paměti SM inteligentním modulům se pro verzi 2.2 a pozdější změnil.

Pokud používáte verzi CPU starší než 2.2, musíte umístit všechny inteligentní moduly na pozice těsně vedle CPU a před všechny jiné než inteligentní moduly, aby byla zajištěna kompatibilita.

V následujícím vzorovém programu pro DP modul na pozici 0 poskytují konfiguraci zařízení DP slave některé oblasti paměti SM. Program používá následující data:

SMW220	Chybový stav DP modulu
SMB224	Stav DP
SMB225	Adresa masteru
SMW226	Offset výstupů v paměti V
SMB228	Počet bytů výstupních dat
SMB229	Počet bytů vstupních dat
VD1000	Ukazatel výstupních dat
VD1004	Ukazatel vstupních dat

Příklad DP komunikace s CPU



Network 1 //Výpočet ukazatele výstupních dat. Pokud je v režimu
//výměny dat:
//1. Výstupní zásobník má offset od VB0
//2. Převéde offset v paměti V na double integer
//3. Připočte ho k adrese VB0, aby se získal ukazatel
// výstupních dat.

LDB= SMB224, 2
MOVD &VB0, VD1000
ITD SMW226, AC0
+D AC0, VD1000

Network 2 //Výpočet ukazatele vstupních dat. Pokud je v režimu
//výměny dat:
//1. Zkopíruje ukazatel výstupních dat
//2. Získá počet výstupních bytů
//3. Přičte ho k ukazateli výstupních dat, aby získal
// začátek ukazatele vstupních dat.

LDB= SMB224, 2
MOVD VD1000, VD1004
BTI SMB228, AC0
ITD AC0, AC0
+D AC0, VD1004

Network 3 //Nastaví množství dat, která mají být zkopírována.
//Pokud je v režimu výměny dat:
//1. Získá počet výstupních bytů, které mají být
// zkopírovány
//2. Získá počet vstupních bytů, které mají být
// zkopírovány

LDB= SMB224, 2
MOVB SMB228, VB1008
MOVB SMB229, VB1009

Network 4 //Přesune výstupy zařízení master na výstupy CPU.
//Zkopíruje vstupy CPU
//na vstupy zařízení master. Pokud je
//v režimu výměny dat:
//1. Zkopíruje výstupy zařízení master na výstupy CPU
//2. Zkopíruje vstupy CPU na vstupy zařízení master

LDB= SMB224, 2
BMB *VD1000, QB0, VB1008
BMB IB0, *VD1004, VB1009

Specifikace modemového modulu EM 241

Tabulka A-42 Objednací číslo modemového modulu EM 241

Objednací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Odnímatelná svorkovnice
6ES7 241-1AA22-0XA0	Modemový modul EM 241	–	8 ¹	Ne

¹ Osm výstupů Q je použito jako logické řídicí prvky pro ovládání modemu a neřídí přímo žádné vnější signály.

Tabulka A-43 Obecné specifikace modemového modulu EM 241

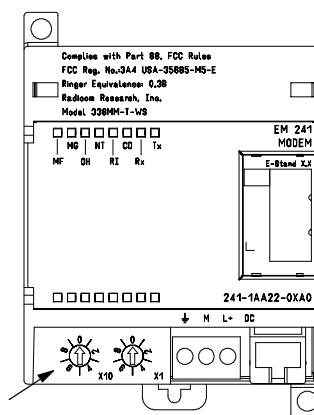
Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Výkonové ztráty	Požadavky V DC +5 V DC	+24 V DC
6ES7 241-1AA22-0XA0	Modemový modul EM 241	71,2 x 80 x 62	190 g	2,1 W	80 mA	70 mA

Tabulka A-44 Specifikace modemového modulu EM 241

Obecné	6ES7 241-1AA22-0XA0
Telefonní připojení	
Napěťové oddělení (telefonní linka od logiky a napájení)	1500 V AC (galvanicky)
Fyzické připojení	RJ11 (6 vývodů, 4 vodiče)
Standardy modemu	Bell 103, Bell 212, V.21, V.22, V.22 bis, V.23c, V.32, V.32 bis, V.34 (implicitní)
Bezpečnostní prvky	Vstupní heslo Zpětné volání
Volba	Pulzní nebo tónová
Protokoly pro přenos zpráv	Numerický TAP (alfanumerický) UCP příkazy 1, 30, 51
Průmyslové protokoly	Modbus PPI
Požadavky na zdroj 24 V DC	
Rozsah napětí	20,4 až 28,8 V DC
Napěťové oddělení (napájení pole od logiky)	500 V AC po dobu 1 minuty

Modemový modul EM 241 nahrazuje funkci externího modemu připojeného ke komunikačnímu portu CPU. Je-li modul EM 241 instalován v systému S7-200, vše, co je zapotřebí pro komunikaci s CPU ze vzdáleného místa, je osobní počítač s externím modemem a STEP 7–Micro/WIN.

Informace o konfiguraci najdete v kapitole 7. Programování a vlastnosti modulu pro pokročilé najdete v kapitole 10.



Přepínač pro číslo země

Obr. A-26 Schéma svorkovnice modemového modulu EM 241

CPU S7-200, které podporují inteligentní moduly

Modemový modul EM 241 je inteligentní rozšiřovací modul určený pro spolupráci s CPU S7-200 uvedenými v tabulku A-45.

Tabulka A-45 Kompatibilita modemového modulu EM 241 s CPU S7-200

CPU	Popis
Model CPU 222 verze 1.10 nebo vyšší	CPU 222 DC/DC/DC a CPU 222 AC/DC/relé
Model CPU 224 verze 1.10 nebo vyšší	CPU 224 DC/DC/DC a CPU 224 AC/DC/relé
Model CPU 226 verze 1.00 nebo vyšší	CPU 226 DC/DC/DC a CPU 226 AC/DC/relé
Model CPU 226XM verze 1.00 nebo vyšší	CPU 226XM DC/DC/DC a CPU 226XM AC/DC/relé

Instalace EM 241

Chcete-li instalovat EM 241, proved'te tyto kroky:

1. Nasad'te EM 241 na DIN lištu a zasuňte plochý kabel rozšiřovací sběrnice.
2. Připojte 24 V DC ze zdroje pro snímače v CPU nebo z vnějšího zdroje a připojte zemnicí svorku k ochrannému uzemnění.
3. Připojte telefonní linku jackem RJ11.
4. Nastavte přepínače pro kód země podle tabulky A-46. Přepínače musíte nastavit před zapnutím napájení CPU, aby byl načten správný kód země.
5. Zapněte napájení CPU. Měla by se rozsvítit zelená LED MG (Module Good – modul v pořádku).

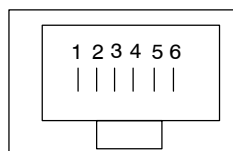
Ted' je modul EM 241 připraven komunikovat.

Tabulka A-46 Kódy zemí podporované EM 241

Kód	Země	Standard Telecom
00	Česká Republika	CTR21
01	Rakousko	CTR21
02	Belgie	CTR21
05	Kanada	IC CS03
08	Dánsko	CTR21
09	Finsko	CTR21
10	Francie	CTR21
11	Německo	CTR21
12	Řecko	CTR21
16	Irsko	CTR21
18	Itálie	CTR21
22	Lucembursko	CTR21
25	Nizozemí	CTR21
27	Norsko	CTR21
30	Portugalsko	CTR21
34	Španělsko	CTR21
35	Švédsko	CTR21
36	Švýcarsko	CTR21
38	Velká Británie	CTR21
39	USA	FCC část 68

Jack RJ11

Obrázek A-27 ukazuje jack RJ11 z blízka. Můžete použít také adaptéry pro jiné standardní telefonní konektory. Více informací najdete v dokumentaci pro přechodový konektor.



Vývod	Popis
3	Vyzvánění
4	Hovor

Je povoleno obrácené zapojení.

Obr. A-27 Pohled na jack RJ11

Varování

Blesk nebo jiná nečekaná vysoká napětí na telefonní lince mohou modemový modul EM 241 poškodit.

Použijte komerčně nabízenou přepětovou ochranu telefonní linky, tedy stejně, jako jsou běžně prodávány na ochranu modemů osobních počítačů. Při výboji může dojít k poškození přepětové ochrany. Vyberte si přepětovou ochranu s indikátorem, který ukazuje, že je ochrana funkční.

Pravidelně kontrolujte přepětovou ochranu, aby bylo jisté, že je trvale modul EM 241 chráněn.

Specifikace polohovacího modulu EM 253

Tabulka A-47 Objednávací číslo polohovacího modulu EM 253

Objednávací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Snímatelný konektor
6ES7 253-1AA22-0XA0	Polohovací modul EM 253	–	8 ¹	Ano

¹ Osm výstupů Q je použito jako logické řídicí prvky modulu a neřídí přímo žádné vnější signály.

Tabulka A-48 Obecné specifikace polohovacího modulu EM 253

Objednávací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Výkonové ztráty	Požadavky V DC +5 V DC +24 V DC
6ES7 253-1AA22-0XA0	Polohovací modul EM 253	71,2 x 80 x 62	0,190 kg	2,5 W	190 mA Viz níže

Tabulka A-49 Specifikace polohovacího modulu EM 253

Obecné	6ES7 253-1AA22-0XA0
Vlastnosti vstupu	
Počet vstupů	5 vstupů
Typ vstupu	Aktivní/pasivní (pasivní IEC typ 1, s výjimkou ZP)
Vstupní napětí	
Maximální trvalé povolené STP, RPS, LMT+, LMT– ZP	30 V DC 30 V DC při 20 mA, maximální
Špička (všechny vstupy)	35 V DC po 0,5 s
Jmenovitá hodnota STP, RPS, LMT+, LMT– ZP	24 V DC při 4 mA, jmenovité 24 V DC při 15 mA, jmenovité
Signál log. 1 (minimální) STP, RPS, LMT+, LMT– ZP	15 V DC při 2,5 mA, minimální 3 V DC při 8,0 mA, minimální
Signál log. 0 (maximální) STP, RPS, LMT+, LMT– ZP	5 V DC při 1 mA, maximální 1 V DC při 1 mA, maximální
Napěťové oddělení (okolí od logiky)	500 V AC po dobu 1 minuty
Optické oddělení (galvanické)	1 bod pro STP, RPS a ZP
Oddělení po skupinách	2 body pro LMT+ a LMT–
Doby zpoždění na vstupu STP, RPS, LMT+, LMT– ZP (započítatelná šířka pulzu)	0,2 ms až 12,8 ms, volitelné uživatelem 2 μs minimálně
Připojení dvou vodičového snímače přiblížení (Bero)	
Povolený svodový proud	1 mA, maximální
Délka kabelu	
Nestíněný STP, RPS, LMT+, LMT– ZP	30 metrů Nedoporučuje se
Stíněný STP, RPS, LMT+, LMT– ZP	100 metrů 10 metrů
Počet současně zapnutých vstupů	
55 °C	5

Tabulka A-49 Specifikace polohovacího modulu EM 253, pokračování

Obecné		6ES7 253–1AA22–0XA0	
Vlastnosti výstupů			
Počet integrovaných výstupů	6 výstupů (4 signály)		
Typ výstupu P0+, P0–, P1+, P1– P0, P1, DIS, CLR	Ovladač RS422/485 Otevřený kolektor		
Výstupní napětí P0, P1, RS–422, rozdílové výstupní napětí Otevřený obvod Diody optoelektronického vazebního členu s 200Ω sériovým odporem 100Ω zatížení 54Ω zatížení P0, P1, DIS, CLR otevřený kolektor doporučené napětí, otevřený obvod povolené napětí, otevřený obvod Pohlcovaný proud Odpor v zapnutém stavu Svodový proud ve vypnutém stavu, 30 V DC Vnitřní zdvihací odpor, výstupní kolektor do T1	3,5 V běžné 2,8 V minimální 1,5 V minimální 1,0 V minimální 5 V DC, k dispozici z modulu 30 V DC ¹ 50 mA, maximální 15 Ω maximální 10 μA maximální 3,3 k Ω ²		
Výstupní proud Počet výstupních skupin Počet sepnutých výstupů (maximální) Svodový proud na výstup P0, P1, DIS, CLR Ochrana proti přetížení	1 6 10 μA maximální Není		
Napěťové oddělení (okolí od logiky) Optické oddělení (galvanické)	500 V AC po dobu 1 minuty		
Zpoždění na výstupu DIS, CLR: Vyp. na zap. / Zap. na vyp.	30 μs, maximální		
Zkreslení pulzu P0, P1, výstupy, ovladače RS–422, 100 Ω vnější zátěž P0, P1 výstupy, otevřený kolektor, 5 V / 470 Ω vnější zátěž	75 ns maximální 300 ns maximální		
Spínací frekvence P0+, P0–, P1+, P1–, P0 a P1	200 kHz		
Délka kabelu Nestíněný Stíněný	Nedoporučuje se 10 metrů		
Napájecí zdroj			
Napájecí napětí L+ Výstup napájení logiky Napájecí proud L+ při zatížení 5 V DC Zatěžovací proud 0 mA (bez zatížení) 200 mA (jmenovitá zátěž)	11 až 30 V DC +5 V DC +/- 10 %, 200 mA maximální		
	Vstup 12 V DC 120 mA 300 mA	Vstup 24 V DC 70 mA 130 mA	
Napěťové oddělení Napájení L+ od logiky Napájení L+ od vstupů Napájení L+ od výstupů	500 V AC po dobu 1 minuty 500 V AC po dobu 1 minuty Nemá		
Obrácená polarita	Vstup L+ a výstup +5V jsou chráněny diodou. Přivedení kladného napětí na svorku M může, vzhledem ke spojení výstupů, mít za následek průtok proudů vedoucích k poškození.		

¹ Provoz výstupů s otevřeným kolektorem nad 5 V DC může zvýšit vysokofrekvenční emise rušení nad povolené mezní hodnoty. Pro váš systém nabo kabeláž mohou být nutná měření emise vysokofrekvenčního rušení.

² Podle typu přijímače pulzů a kabelu, může přídavný vnější odpor zlepšit kvalitu pulzního signálu a odolnost proti rušení.

CPU S7-200, které podporují inteligentní moduly

Polohovací modul EM 253 je inteligentní rozšiřovací modul určený pro spolupráci s CPU S7-200 uvedenými v tabulce A-50.

Tabulka A-50 Kompatibilita polohovacího modulu EM 253 s CPU S7-200

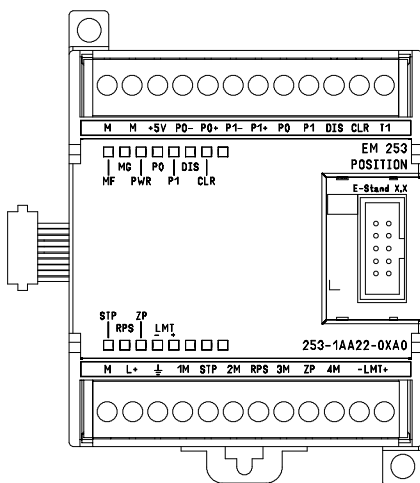
CPU	Popis
Model CPU 222 verze 1.10 nebo vyšší	CPU 222 DC/DC/DC a CPU 222 AC/DC/relé
Model CPU 224 verze 1.10 nebo vyšší	CPU 224 DC/DC/DC a CPU 224 AC/DC/relé
Model CPU 226 verze 1.00 nebo vyšší	CPU 226 DC/DC/DC a CPU 226 AC/DC/relé
Model CPU 226XM verze 1.00 nebo vyšší	CPU 226XM DC/DC/DC a CPU 226XM AC/DC/relé

Stavové LED polohovacího modulu EM 253

Stavové LED diody polohovacího modulu jsou uvedeny v tabulce A-51.

Tabulka A-51 Stavové LED polohovacího modulu

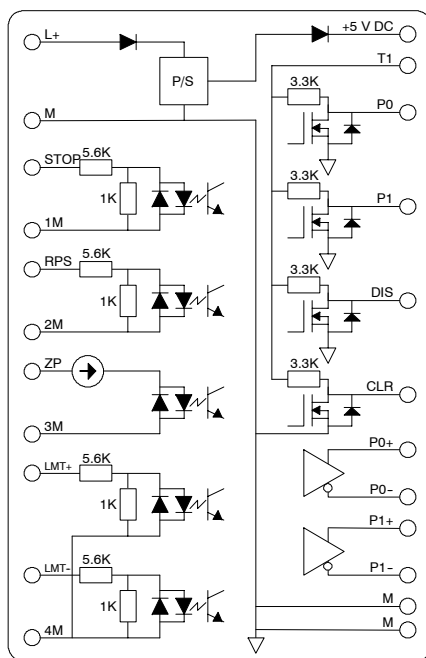
Lokální I/O	Kontrolka (LED)	Barva	Popis funkce
–	MF	Červená	Svítilí, když modul zjistí kritickou chybu
–	MG	Zelená	Svítilí, když modul nemá žádnou poruchu, a bliká rychlostí 1 Hz, když je zjištěna chyba konfigurace
–	PWR	Zelená	Svítilí, když je na svorky modulu L+ a M přiváděno napětí 24 V DC
Vstup	STP	Zelená	Svítilí, když je sepnutý vstup "stop"
Vstup	RPS	Zelená	Svítilí, když je sepnutý vstup přepínače referenčního bodu
Vstup	ZP	Zelená	Svítilí, když je sepnutý vstup nulového impulsu
Vstup	LMT–	Zelená	Svítilí, když je sepnutý vstup záporné mezní hodnoty
Vstup	LMT +	Zelená	Svítilí, když je sepnutý vstup kladné mezní hodnoty
Výstup	P0	Zelená	Svítilí, když výstup P0 vysílá pulzy
Výstup	P1	Zelená	Svítilí, když výstup P1 vysílá pulzy nebo když tento výstup indikuje kladný pohyb
Výstup	DIS	Zelená	Svítilí, když je výstup DIS aktivní
Výstup	CLR	Zelená	Svítilí, když je aktivní výstup pro smazání čítače odchylek



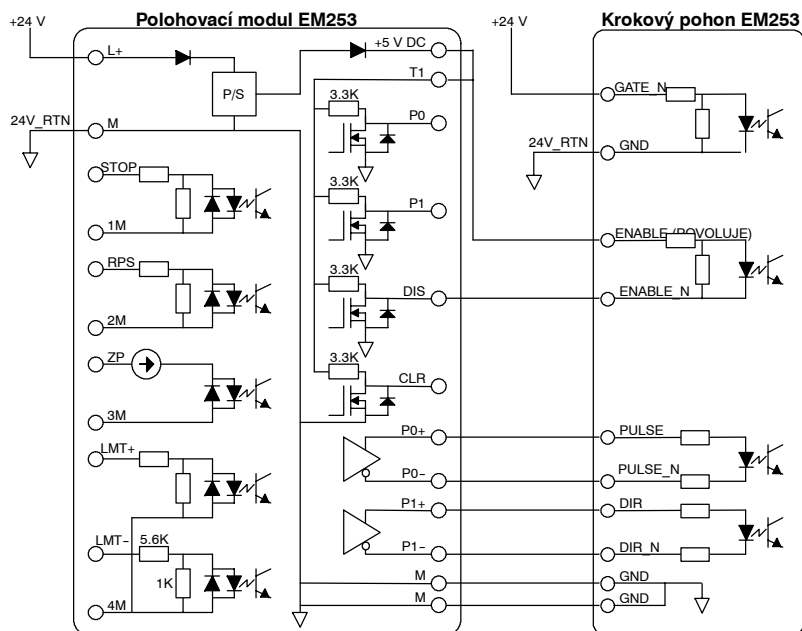
Obr. A-28 Polohovací modul EM 253

Schéma zapojení

Na následujících schematických obrázcích nejsou svorky uváděny v pořadí za sebou. Uspořádání svorek najdete na obrázku A-28.

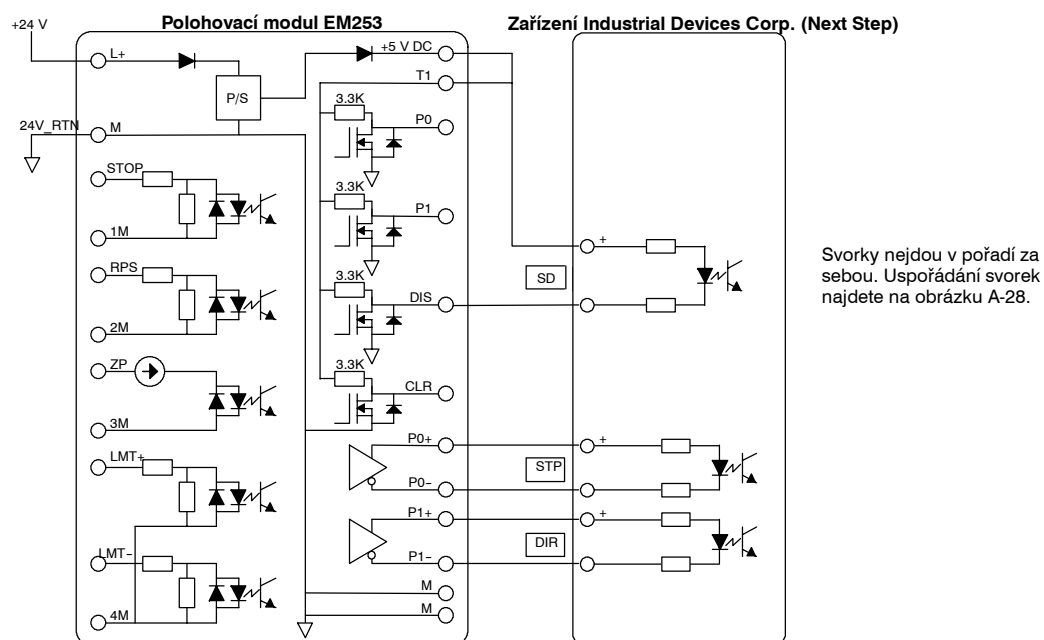


Obr. A-29 Vnitřní schéma pro vstupy a výstupy polohovacího modulu EM 253

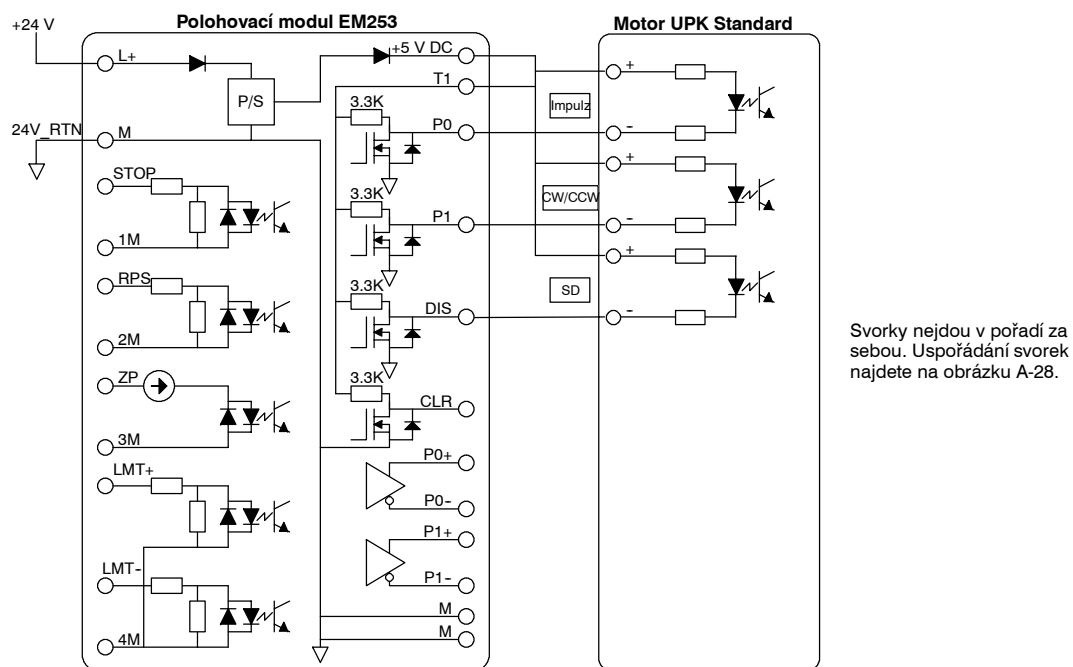


Svorky nejdu v pořadí za sebou. Uspořádání svorek najdete na obrázku A-28.

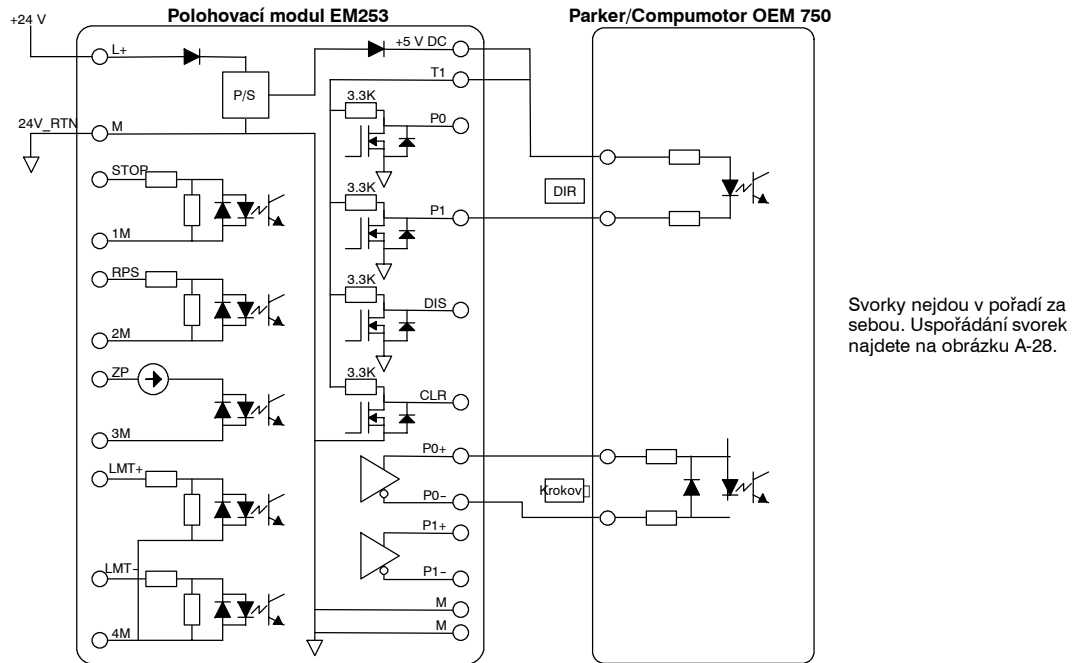
Obr. A-30 Připojení polohovacího modulu EM 253 ke krokovému pohonu SIMATIC FM



Obr. A-31 Připojení polohovacího modulu EM 253 k zařízení Industrial Devices Corp. (Next Step)



Obr. A-32 Připojení polohovacího modulu EM 253 k pohonu UPK Standard



Obr. A-33 Připojení polohovacího modulu EM 253 k pohonu Parker/Compumotor OEM 750

Specifikace modulu AS–Interface (CP 243–2)

Tabulka A-52 Objednací číslo modulu AS-Interface (CP 243–2)

Objednací číslo	Rozšiřovací modul	Vstupy EM	Výstupy EM	Odnímatelná svorkovnice
6GK7 243–2AX01–0XA0	Modul AS–Interface (CP 243–2)	8 digitálních a 8 analogových	8 digitálních a 8 analogových	Ano

Tabulka A-53 Obecné specifikace modulu AS-Interface (CP 243–2)

Objednací číslo	Název a popis modulu	Rozměry (mm) (š x v x h)	Hmotnost	Ztráty	Požadavky V DC + 5 V DC z AS–Interface
6GK7 243–2AX01–0XA0	Modul AS–Interface (CP 243–2)	71 x 80 x 62	cca 250 g	3,7 W	220 mA 100 mA

Tabulka A-54 Specifikace modulu AS-Interface (CP 243–2)

Obecné	6GK7 243–2AX01–0XA0
Perioda	5 ms při 31 zařízeních slave 10 ms při 62 zařízeních AS–i slave, která používají režim rozšířené adresace
Konfigurace	Pomocí tlačítka na předním panelu nebo použijte příkaz celkové konfigurace (viz popis příkazů AS–i v manuálu CP 243–2 AS–i Interface Master)
Podporované profily AS–i master	M1e
Připojení ke kabelu AS–i	Přes svorkovnici S7-200. Povoleno proudové zatížení ze svorky 1 až 3 nebo ze svorky 2 až 4 maximálně 3 A.
Adresový rozsah	Jeden digitální modul s 8 digitálními vstupy a 8 digitálními výstupy a jeden analogový modul s 8 analogovými vstupy a 8 analogovými výstupy

Vlastnosti

Spolu s automatem S7-200 mohou být naráz provozovány až dva moduly AS–Interface, což výrazně zvyšuje počet digitálních a analogových vstupů a výstupů, které máte k dispozici (maximálně 124 digitálních vstupů/124 digitálních výstupů AS–Interface na CP). Čas potřebný pro konfiguraci je redukován díky schopnosti provést konfiguraci stisknutím tlačítka. Stavové LED redukují dobu prostoje v případě chyby tím, že ukazují stav CP a všech připojených zařízení slave, a tím, že monitorují hlavní napětí AS–Interface.

Modul AS–Interface má následující funkce:

- ☐ Podporuje analogové moduly
- ☐ Podporuje všechny funkce master a umožňuje připojení až 62 zařízení AS–Interface slave
- ☐ Kontrolky na předním panelu zobrazují provozní stav a dostupnost připojených zařízení slave.
- ☐ Kontrolky na předním panelu zobrazují chyby (včetně chyby napětí AS–Interface, chyby konfigurace)
- ☐ Dvě svorky umožňují přímé připojení kabelu AS–Interface.
- ☐ Dvě tlačítka dávají informace o stavu zařízení slave, o přepínači pracovního režimu a mohou zajistit převzetí stávající konfigurace jako SET konfiguraci.

Činnost

V obrazu procesu S7-200 zabírá modul AS–Interface digitální vstupní byte (stavový byte), digitální výstupní byte (řídící byte), 8 analogových vstupních a 8 analogových výstupních wordů. Modul AS–Interface využívá prostor dvou logických modulů. Stavový a řídící byte je možné použít pro nastavení režimu modulu AS–Interface pomocí uživatelského programu. V závislosti na svém režimu AS–Interface ukládá buď I/O data zařízení AS–Interface slave, nebo diagnostické hodnoty, nebo povoluje zprávy zařízení master (například změna adresy slave) v oblasti analogových adres S7-200.

Všechna připojená zařízení AS–Interface slave je možné konfigurovat pouhým stisknutím tlačítka. Další konfigurace CP není potřeba.



Varování

Při použití modulu AS–Interface je nutné v CPU zablokovat analogovou filtraci.

Pokud není analogová filtrace v CPU zablokovaná, zničí se digitální data a chyby se nevrací jako bitové hodnoty v analogovém word.

Zkontrolujte, zda je analogová filtrace v CPU zablokovaná.

Funkce

CP 243–2 je zařízení AS–Interface master pro třídu M1e, což znamená, že podporuje všechny specifikované funkce. To umožňuje řídit až 31 digitálních zařízení slave na AS–Interface pomocí dvojnásobného přiřazení adres (A–B). Modul CP 243–2 může být nastaven na dva různé režimy:

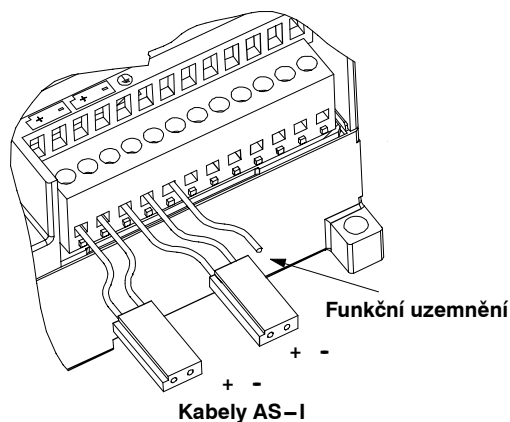
- ☐ Standardní režim: přístup na I/O data zařízení AS–Interface slave
- ☐ Rozšířený režim: volání zařízení master (například zápis parametrů) nebo požadavek na diagnostické hodnoty

Svorky

Modul AS–Interface má následující připojení:

- ☐ Dvě připojení ke kabelu modulu AS–Interface (s vnitřním přemostěním)
- ☐ Jedno připojení pro funkční uzemnění

Svorky jsou umístěny pod krytem předního panelu, jak je znázorněno na obrázku A-34.



Obr. A-34 Připojení kabelu modulu AS–Interface

Varování

Kapacita zatížení kontaktů modulu AS–Interface je maximálně 3 A. Pokud je tato hodnota na kabelu modulu AS–Interface překročena, nesmí být AS–Interface začleněn do smyčky kabelu AS–Interface, ale musí být připojen samostatným kabelem (v tomto případě je využita pouze jedna dvojice svorek modulu AS–Interface). AS–Interface musí být spojen s uzemněním přes zemnicí svorku.



Tip

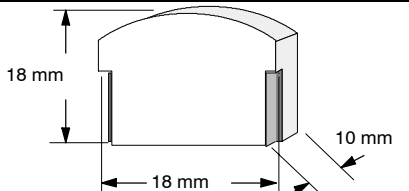
Modul AS–Interface má svorku pro uzemnění. Tato svorka by měla být připojena k vodiči PE s co nejmenším odporem.

Další informace

Více informací o modulu CP 243–2 AS–Interface master najdete v *Manuálu SIMATIC NET CP 243–2 AS-Interface master*.

Volitelné zásuvné moduly

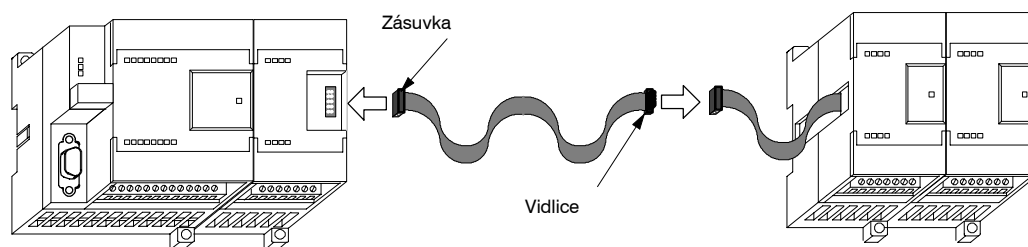
Zásuvný modul	Popis	Objednací číslo
Paměťový modul	Paměťový modul ukládá: program, data a konfiguraci	6ES7 291–8GE20–0XA0
Hodiny reálného času s baterií	Přesnost hodinového modulu: 2 minut/měsíc při 25 °C, 7 minut/měsíc při 0 °C až 55 °C	6ES7 297–1AA20–0XA0
Bateriový modul	Bateriový modul (dobu uchování dat): běžně 200 dnů	6ES7 291–8BA20–0XA0

Obecné vlastnosti		Rozměry
Baterie	3 V, 30 mAhodin, Renata CR 1025	
Velikost	9,9 mm x 2,5 mm	
Typ	Lithiová < 0,6 g	

Paměťový modul uchovává celý program a datový blok pro všechny CPU (CPU 221, CPU 222, CPU 224, CPU 226 a CPU 226XM).

Kabel pro rozšiřovací moduly

Obecné vlastnosti (6ES7 290–6AA20–0XA0)	
Délka kabelu	0,8 m
Hmotnost	25 g
Typ konektoru	10pinový plochý



Obr. A-35 Typická instalace kabelu pro rozšiřovací moduly

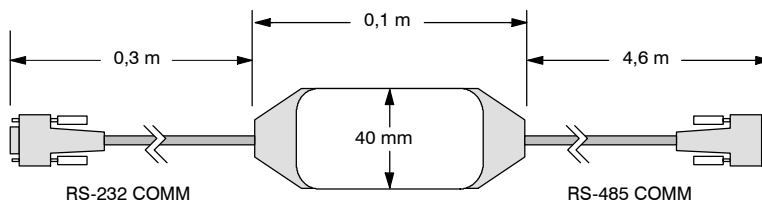


Tip

V řetězci CPU/rozšiřovací moduly je povolen pouze jeden takový kabel.

PC/PPI kabel

PC/PPI kabel (6ES7 901-3BF21-0XA0) Obecná charakteristika	
Napájecí napětí	14,4 až 28,8 V DC
Napájecí proud při jmenovitém napájecím napětí 24 V	max. 50 mA efektivní hodnota
Zpoždění od přijetí hrany start bitu RS-232 po vyslání hrany start bitu RS-485	1,2 μ s. max
Zpoždění od přijetí hrany stop bitu RS-232 po vyslání hrany stop bitu RS-485	1,4 znaku krát maximum (1,4 x 11/ baud) = 1,6 ms při 9600 bodech
Zpoždění šíření	4 μ s max., RS-485 až RS-232, 1,2 μ s max., RS-232 to RS-485
Napětové oddělení (RS-485 až RS-232)	500 V DC
Elektrické vlastnosti RS-485	
Rozsah souhlasného napětí	-7 V až +12 V, 1 sekunda, trvale 3 V efektivní hodnota
Vstupní impedance přijímače	5,4 k Ω min., včetně ukončení
Zakončení/symetrizace	10 k Ω k +5 V na B, PROFIBUS kolík č. 3 10 k Ω k zemi (GND) na A, PROFIBUS kolík č. 8
Práh/citlivost přijímače	+/- 0,2 V, 60 mV typická hystereze
Napětí diferenciálního výstupu vysílače	2 V min. při $R_L = 100 \Omega$, 1,5 V min. při $R_L = 54 \Omega$
Elektrická vlastnosti RS-232	
Vstupní impedance přijímače	3 k Ω min., minimální
Práh/citlivost přijímače	0,8 V min. spodní úroveň, 2,4 V max. vysoká úroveň, 0,5 V typická hystereze
Výstupní napětí vysílače	+/- 5 V min. při $R_L = 3 k\Omega$



SIEMENS		Napětově oddělený PC/PPI kabel		PC	
PPI 1 0		Přepínač DIP č.	123	4	1 = 10 BIT 0 = 11 BIT
		115.2-38.4K	000	5	1 = DTE 0 = DCE
		19.2	001	6	1 = RTS pro XMT 0 = RTS vždy
		9.6K	010		
		2.4K	100		
		1.2K	101		

Obr. A-36 Rozměry kabelu PC/PPI

Tabulka A-55 Nastavení přepínače na PC/PPI kabelu

Rychlost	Přepínače 1,2,3*	Přepínač režimu modemu 4*	Přepínač volby DCE/DTE 5*	Volba RTS pro DTE	Přepínač 6*
115200 – 38400	000	11bitový modem 0	DCE 0	RTS aktivní vždy	0
19200	001	10bitový modem 1	DTE 1	RTS aktivní, když PLC vysílá	1
9600	010				
4800	011				
2400	100				
1200	101				
600	110				

* Přepínač : 1 = nahoru; 0 = dolů

Tabulka A-56 Vývody konektoru RS–485 a RS–232 DTE

Vývody rozhraní RS-485		Vývody rozhraní RS-232 v režimu DCE	
Číslo vývodu	Popis signálu	Číslo vývodu	Popis signálu
1	Uzemnění (logická zem RS-485)	1	Detekce nosné (DCD) (nepoužito)
2	24V zpětný vodič (logická zem RS-485)	2	Přijímaná data (RD) (výstup PC/PPI kabelu)
3	Signál B (Rx/D/TxD+)	3	Vysílaná data (TD) (vstup PC/PPI kabelu)
4	RTS (úroveň TTL)	4	Koncové zařízení připraveno (DTR) (nepoužito)
5	Uzemnění (logická zem RS-485)	5	Uzemnění (logická zem RS-232)
6	Nepřipojeno	6	Pohotovost (DSR) (nepoužito)
7	24V napájení	7	Výzva k vysílání (RTS) (nepoužito)
8	Signál A (Rx/D/TxD–)	8	Připraven k vysílání (CTS) (nepoužito)
9	Volba protokolu	9	Indikátor vyzvánění (RI) (nepoužito)

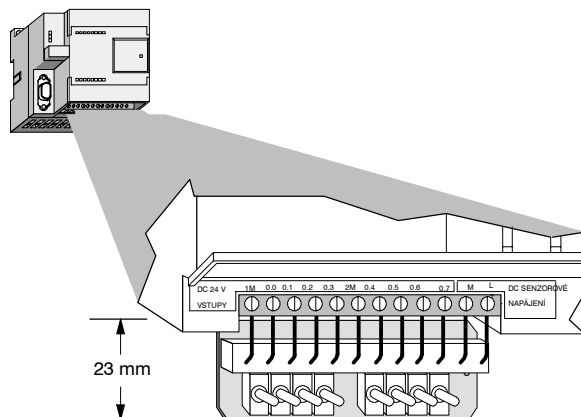
Tabulka A-57 Vývody konektoru RS–485 a RS–232 DTE

Vývody rozhraní RS-485 -		Vývody rozhraní RS-232 v režimu DTE ¹	
Číslo vývodu	Popis signálu	Číslo vývodu	Popis signálu
1	Uzemnění (logická zem RS-485)	1	Detekce nosné (DCD) (nepoužito)
2	24V zpětný vodič (logická zem RS-485)	2	Přijímaná data (RD) (vstup PC/PPI kabelu)
3	Signál B (Rx/D/TxD+)	3	Vysílaná data (TD) (výstup PC/PPI kabelu)
4	RTS (úroveň TTL)	4	Datový terminál připraven (DTR) (nepoužito)
5	Uzemnění (logická zem RS-485)	5	Uzemnění (logická zem RS-232)
6	Nepřipojeno	6	Pohotovost (DSR) (nepoužito)
7	24V napájení	7	Výzva k vysílání (RTS) (výstup z PC/PPI kabelu) (volitelné přepínačem)
8	Signál A (Rx/D/TxD–)	8	Připraven k vysílání (CTS) (nepoužito)
9	Volba protokolu	9	Indikátor vyzvánění (RI) (nepoužito)

¹ Pro modemy je potřeba převod ze zásuvky na vidlici a z 9 vývodů na 25

Simulátory vstupů

Objednací číslo	simulátor 8místný 6ES7 274-1XF00-0XA0	simulátor 14místný 6ES7 274-1XH00-0XA0	simulátor 24místný 6ES7 274-1XK00-0XA0
Velikost (d x š x h)	61 x 36 x 22 mm	91 x 36 x 22 mm	147 x 36 x 25 mm
Hmotnost	0,02 kg	0,03 kg	0,04 kg
Vstupy	8	14	24



Obr. A-37 Instalace simulátoru vstupů



Varování

Tyto simulátory vstupů nejsou schváleny pro použití na nebezpečných místech třídy I DIV 2 nebo třídy I, zóna 2. Přepínače představují eventuální jiskrové nebezpečí.

Simulátory vstupů nepoužívejte na nebezpečných místech třídy I DIV 2 nebo třídy I, zóna 2.

Výpočet energetické bilance



CPU S7-200 má vnitřní napájecí zdroj, který napájí CPU, rozšiřovací moduly a další zařízení uživatele s napětím 24 V DC. Použijte následující informace jako průvodce pro stanovení, jaký proud napájení může CPU S7-200 dodávat pro vaši konfiguraci.

Požadavky na napájení

Každá CPU S7-200 dodává napětí 5 V DC a 24 V DC:

- ☐ Každá CPU má napájení pro snímače 24 V DC, které může dodávat 24 V DC pro lokální vstupy nebo pro cívky relé v rozšiřovacích modulech. Pokud požadavky na napájení 24 V DC překročí energetickou bilanci CPU, můžete přidat vnější zdroj 24 V DC, který bude napájet rozšiřovací moduly. Napájení 24 V DC musíte ke vstupům nebo cívkám relé ručně připojit.
- ☐ CPU také poskytuje 5 V DC napájení pro rozšiřovací moduly, pokud jsou připojeny. Pokud požadavky na 5 V DC napájení rozšiřovacích modulů překročí energetickou bilanci CPU, musíte rozšiřovací moduly odpojit a ponechat jich jen tolik, aby je tato bilance pokryla.

Specifikace v příloze A poskytuje informace o energetické bilanci jednotlivých CPU a požadavcích rozšiřovacích modulů na napájení.



Varování

Připojení externího napájecího zdroje 24 V DC paralelně se stejnosměrným zdrojem pro snímače S7-200 může vést ke konfliktu mezi těmito dvěma zdroji, protože se každý snaží nastavit svou vlastní úroveň výstupního napětí.

Následkem toho může být zkrácena životnost zařízení nebo může dojít k okamžité poruše jednoho nebo obou napájecích zdrojů a k následnému nepředvídatelnému chování řídicího systému. Nepředvídatelné chování by mohlo mít za následek smrt nebo vážné poranění osob a poškození zařízení.

Stejnoseměrný zdroj pro snímače S7-200 a externí zdroj by měly dodávat napětí do různých bodů. Přitom mezi oběma zdroji napětí může existovat jen jedno propojení. Oba zdroje mohou mít navíc propojeny zemnicí vodiče.

Vzor výpočtu požadavků na napájení

Tabulka B-1 uvádí vzor výpočtu požadavků na napájení pro automat S7-200, který obsahuje:

- ☐ CPU S7-200 224 AC/DC/relé
- ☐ 3 EM 223 8 DC vstupů/8 reléových výstupů
- ☐ 1 EM 221 8 DC vstupů

Tato instalace má celkem 46 vstupů a 34 výstupů.

V tomto příkladu CPU S7-200 dává pro rozšiřovací moduly dostatečný proud ze zdroje 5 V DC, ale nedává dostatečný proud ze zdroje pro snímače 24 V DC pro všechny vstupy a cívky relé v rozšiřovacích modulech. Vstupy a výstupy vyžadují 400 mA, CPU S7-200 CPU dává pouze 280 mA. Do této instalace je třeba zapojit další zdroj nejméně 120 mA při 24 V DC, aby bylo možné provozovat všechny 24 V DC vstupy a výstupy.

Tabulka B-1 Výpočty energetické bilance pro vzorovou konfiguraci

Energetická bilance CPU	5 V DC	24 V DC
CPU 224 AC/DC/Relé	660 mA	280 mA

minus

Požadavky systému	5 V DC	24 V DC
CPU 224, 14 vstupů		14 * 4 mA = 56 mA
3 EM 223, potřebné napájení 5 V	3 * 80 mA = 240 mA	
1 EM 221, potřebné napájení 5 V	1 * 30 mA = 30 mA	
3 EM 223, každý s 8 vstupy		3 * 8 * 4 mA = 96 mA
3 EM 223, každý s 8 cívkami relé		3 * 8 * 9 mA = 216 mA
1 EM 221, každý s 8 vstupy		8 * 4 mA = 32 mA
Požadavky celkem	270 mA	400 mA

rovná se

Proudová bilance	5 V DC	24 V DC
Proudová bilance celkem	390 mA	[120 mA]

Výpočet vašich požadavků napájení

Pro stanovení, jaké napětí (nebo proud) může CPU S7-200 pro vaši konfiguraci dodávat, použijte následující tabulku. Energetické bilance vašeho typu CPU a požadavky napájení rozšiřovacích modulů najdete v příloze A.

Energetická bilance	5 V DC	24 V DC

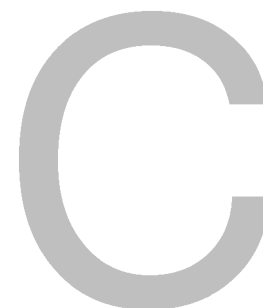
minus

Požadavky systému	5 V DC	24 V DC
Požadavky celkem		

rovná se

Proudová bilance	5 V DC	24 V DC
Proudová bilance celkem		

Chybové kódy



Informace o chybových kódech pomáhají vyřešit problémy s CPU S7-200.

Přehled kapitoly

Kódy a hlášení kritických chyb	404
Problémy programování při běhu programu	405
Porušení pravidel kompilace	406

Kódy a hlášení kritických chyb

Kritické chyby způsobí, že automat S7-200 zastaví provádění uživatelského programu. Kritická chyba může v závislosti na své závažnosti způsobit, že S7-200 nebude schopen provádět některé nebo všechny své funkce. Cílem zvládnutí kritických chyb je přivést automat S7-200 do bezpečného stavu, ve kterém může odpovídat na dotazy o stávajících chybách.

Při zjištění kritické chyby provede S7-200 následující úlohy:

- ☐ Přejde do režimu STOP
- ☐ Rozsvítí LED System Fault (Systémová chyba) a Stop (Zastavení)
- ☐ Vypne výstupy

V tomto stavu S7-200 zůstane, dokud není kritická chyba opravena. Chcete-li si prohlédnout chybové kódy, zvolte příkaz **PLC > Informace** z hlavního menu. Tabulka C-1 uvádí seznam a popis kódů kritických chyb, které lze přečíst z automatu S7-200.

Tabulka C-1 Kódy a hlášení kritických chyb, které uvádí automat S7-200

Chybové kódy	Popis
0000	Bez závažné chyby
0001	Chyba kontrolního součtu v programu
0002	Chyba kontrolního součtu při kompilaci LAD programu
0003	Chyba časové prodlevy obvodu sledování funkce programového cyklu (WATCHDOG)
0004	Interní EEPROM: selhala
0005	Interní EEPROM: Chyba kontrolního součtu v uživatelském programu
0006	Interní EEPROM: Chyba kontrolního součtu v konfiguračních parametrech (SDB0)
0007	Interní EEPROM: Chyba kontrolního součtu ve vnucených datech (force)
0008	Interní EEPROM: Chyba kontrolního součtu v tabulce implicitních výstupů
0009	Interní EEPROM: Chyba kontrolního součtu v uživatelských datech, DB1
000A	Paměťová karta: selhala
000B	Paměťová karta: chyba kontrolního součtu v uživatelském programu.
000C	Paměťová karta: chyba kontrolního součtu v parametrech konfigurace (SDB0)
000D	Paměťová karta: chyba kontrolního součtu ve vnucených datech (force)
000E	Paměťová karta: chyba kontrolního součtu v tabulce implicitních výstupů
000F	Paměťová karta: chyba kontrolního součtu v uživatelských datech, DB1
0010	Interní softwarová chyba
0011 ¹	Chyba nepřímé adresace komparačního kontaktu
0012 ¹	Nepřípustná hodnota s plovoucí desetinnou čárkou komparačního kontaktu
0013	Paměťová karta je prázdná nebo uživatelský program není na tomto S7-200 spustitelný
0014 ¹	Chyba rozsahu komparačního kontaktu

¹ Chyby komparačního kontaktu jsou jediné, které generují kritickou i nekritickou chybu. Důvodem pro generování nekritické chyby je uložení adresy chyby.

Problémy programování při běhu programu

Během provádění uživatelského programu mohou vzniknout nekritické chyby (jako jsou chyby adresace). V takovém případě generuje automat S7-200 chybový kód v době vykonávání programu. Tabulka C-2 uvádí seznam a popis kódů nekritických chyb.

Tabulka C-2 Problémy programování v době vykonávání programu

Chybové kódy	Popis
0000	Bez kritické chyby; bez chyby
0001	HSC blok aktivován před vykonáním HDEF
0002	Vstup je již využit pro HSC; nelze tedy použít jako vstup přerušení
0003	Vstup je již využit jako vstup přerušení nebo pro jiný HSC; nelze tedy použít pro HSC
0004	Pokus o vykonání ENI, DISI, SPA nebo HDEF instrukcí v podprogramu přerušení
0005	Pokus o vykonávání druhého HSC/PLS se stejným číslem jako ještě neukončený předcházející (HSC/PLS v podprogramu přerušení je v konfliktu s HSC/PLS v hlavním programu)
0006	Chyba nepřímé adresace
0007	Chyba dat TODW (Zápis reálného času) nebo TODR (Čtení reálného času)
0008	Překročen maximální počet vnořených podprogramů
0009	Současné provádění instrukcí XMT/RCV na portu 0
000A	Pokus o předefinování HSC vykonáváním další instrukce HDEF pro stejný HSC
000B	Současné provádění instrukcí XMT/RCV na portu 1
000C	Hodinový modul nepřístupný pro TODR, TODW nebo komunikaci
000D	Pokus o předefinování pulzního výstupu v době, kdy je aktivní
000E	Počet segmentů profilu PTO byl nastaven na 0
000F	Nepřípustná číselná hodnota v instrukci komparačního kontaktu
0091	Mimo rozsah (informace o adresách): zkontrolujte rozsah operandů
0092	Chyba při čítání nebo chyba v instrukci čítače (informace o čítání): ověřte maximální hodnotu
0094	Chyba rozsahu pro zápis do pevné paměti s udáním adresy
009A	Pokus o přepnutí do režimu Freeport v době uživatelského přerušení
009B	Nepřípustný index (operace s řetězcem, ve kterém je specifikována jako hodnota výchozího místa 0)

Porušení pravidel kompilace

Když provádíte download programu, automat S7-200 program nejdříve zkompile. Pokud S7-200 zjistí, že program porušuje nějaké pravidlo kompilace (příkladem může být nepřipustná instrukce), přeruší se download a generuje se kód nekritické chyby, porušující pravidla kompilace. Tabulka C-3 uvádí seznam a popis chybových kódů, které jsou generovány, pokud jsou porušena pravidla kompilace.

Tabulka C-3 Porušení pravidel kompilace

Chybové kódy	Chyby kompilace (nekritické)
0080	Program je příliš dlouhý pro kompilaci; zmenšete délku programu
0081	Podtečení zásobníku, rozdělte network na více networků.
0082	Neznámá instrukce; zkontrolujte zápis instrukce.
0083	Chybí MEND nebo instrukce není povolena v hlavním programu: přidejte instrukci MEND nebo odstraňte nesprávnou instrukci.
0084	Rezervováno
0085	Chybí FOR; přidejte instrukci FOR nebo smažte instrukci NEXT
0086	Chybí NEXT, přidejte instrukci NEXT nebo smažte instrukci FOR
0087	Chybí návěští (LBL, INT, SBR); přidejte požadované návěští.
0088	Chybí RET nebo instrukce není povolena v podprogramu: přidejte RET na konec podprogramu nebo odstraňte nesprávnou instrukci.
0089	Chybí RETI nebo instrukce není povolena v podprogramu přerušení: přidejte RETI na konec podprogramu přerušení nebo odstraňte nesprávnou instrukci.
008A	Rezervováno
008B	Nepřípustné JMP do segmentu SCR nebo z něho
008C	Duplikátní návěští (LBL, INT, SBR); přejmenujte jedno z návěští.
008D	Neznámé návěští (LBL, INT, SBR); zajistěte dodržení max. počtu návěští.
0090	Neznámý parametr, zkontrolujte parametry instrukce.
0091	Chyba rozsahu; zkontrolujte rozsah operandů.
0092	Chyba v parametru instrukce čítače, zkontrolujte maximální hodnotu parametru.
0093	Překročena úroveň zanoření FOR/NEXT.
0095	Chybějící instrukce LSCR
0096	Chybějící instrukce SCORE nebo nedovolená instrukce před instrukcí SCORE
0097	Uživatelský program obsahuje nečíslované i číslované instrukce EV/ED
0098	Nepřípustná editace v režimu RUN (pokus o editaci programu s nečíslovanými instrukcemi EV/ED)
0099	Příliš velký počet skrytých programových segmentů (instrukce HIDE)
009B	Nepřípustný index (operace s řetězcem, ve kterém je specifikována jako hodnota výchozího místa 0)
009C	Překročena maximální délka instrukce

Speciální paměťové bity (SM)

Speciální paměťové bity poskytují různé stavové informace a řídicí funkce a zároveň slouží jako prostředek komunikace mezi automatem S7-200 a uživatelským programem. Speciální paměťové bity je možné používat jako bity, byty, word nebo double word.



Přehled kapitoly

SMB0: Stavové bity	408
SMB1: Stavové bity	408
SMB2: Znak přijatý v režimu Freeport	409
SMB3: Chyba parity v režimu Freeport	409
SMB4: Přeplnění fronty	409
SMB5: Stav vstupů a výstupů	410
SMB6: Registr ID CPU	410
SMB7: Rezervován	410
SMB8 až SMB21: ID a chybové registry I/O modulů	411
SMW22 až SMW26: Délka programového cyklu	412
SMB28 a SMB29: Analogové potenciometry	412
SMB30 a SMB130: Řídicí registry pro Freeport	412
SMB31 a SMW32: Řízení zápisu do trvalé paměti (EEPROM)	413
SMB34 a SMB35: Registry časového intervalu pro časovaná přerušení	413
SMB36 až SMB65: Registr HSC0, HSC1 a HSC2	413
SMB66 až SMB85: Registry PTO/PWM	415
SMB86 až SMB94 a SMB186 až SMB194: Řízení příjmu zpráv	416
SMW98: Chyby na rozšiřovací I/O sběrnici	417
SMB130: Řídicí registr pro Freeport (viz SMB30)	417
SMB131 až SMB165: Registr HSC3, HSC4 a HSC5	417
SMB166 až SMB185: Tabulka definice profilu PTO0, PTO1	418
SMB186 až SMB194: Řízení příjmu zpráv (viz SMB86 až SMB94)	418
SMB200 až SMB549: Stav inteligentních modulů	419

SMB0: Stavové bity

Jak je popsáno v tabulce D-1, SMB0 obsahuje osm stavových bitů, které automat S7-200 aktualizuje na konci každého programového cyklu.

Tabulka D-1 Speciální paměťový byte SMB0 (SM0.0 až SM0.7)

SM bity	Popis (pouze ke čtení)
SM0.0	Tento bit má vždy hodnotu 1.
SM0.1	Tento bit je v log. 1 v prvním programovém cyklu. Jedním z použití je volání inicializačního podprogramu.
SM0.2	Tento bit přejde do stavu log. 1 na jeden programový cyklus, pokud byla ztracena remanentní data. Je možné ho použít buď jako chybový bit paměti, nebo jako mechanismus vyvolání speciální spouštěcí sekvence.
SM0.3	Tento bit přejde do stavu log. 1 na jeden programový cyklus, když systém přešel po studeném restartu do režimu RUN. Je možné ho použít pro inicializaci stroje před spuštěním vlastního provozu.
SM0.4	Tento bit generuje hodinový pulz, který je v log. 1 po dobu 30 sekund a v log. 0 po dobu 30 sekund v pracovním cyklu 1 minuta. To umožňuje vytvořit snadno použitelné zpoždění nebo minutový pulz.
SM0.5	Tento bit generuje hodinový pulz, který je v log. 1 po dobu 0,5 sekundy a pak v log. 0 po dobu 0,5 sekundy v pracovním cyklu 1 sekunda. To umožňuje vytvořit snadno použitelné zpoždění nebo sekundový pulz.
SM0.6	Tento bit slouží jako hodiny programového cyklu; v log. 1 je po dobu jednoho programového cyklu a pak v log. 0 během příštího programového cyklu. Je možné ho použít jako vstup čítače programových cyklů.
SM0.7	Tento bit odráží polohu přepínače pracovního režimu (vypnutý znamená polohu TERM, zapnutý polohu RUN). Pokud tento bit použijete pro povolení režimu Freeport ve chvíli, kdy je přepínač v poloze RUN, je možné použít normální komunikaci s programovacím zařízením, pokud je přepínač v poloze TERM.

SMB1: Stavové bity

Jak je uvedeno v tabulce D-2, obsahuje SMB1 indikátory různých potenciálních chyb. Tyto bity jsou nastavovány instrukcemi při jejich provádění.

Tabulka D-2 Speciální paměťový byte SMB1 (SM1.0 až SM1.7)

SM bity	Popis (pouze ke čtení)
SM1.0	Tento bit přejde do log. 1 provedením určitých instrukcí při nulovém výsledku operace.
SM1.1	Tento bit přejde do log. 1 provedením určitých instrukcí, pokud je výsledkem přeplnění, nebo pokud je zjištěna nepřípustná číselná hodnota.
SM1.2	Tento bit přejde do log. 1, když má některá matematická operace záporný výsledek.
SM1.3	Tento bit přejde do log. 1 při pokusu o dělení nulou.
SM1.4	Tento bit přejde do log. 1, když se instrukce přidání do tabulky (Add to Table) pokusí tabulku přeplnit.
SM1.5	Tento bit přejde do log. 1, když se instrukce LIFO nebo FIFO pokusí číst z prázdné tabulky.
SM1.6	Tento bit přejde do log. 1, když dojde k pokusu o převedení jiné než BCD hodnoty na binární.
SM1.7	Tento bit přejde do log. 1, když nemůže být ASCII hodnota převedena na platnou hexadecimální hodnotu.

SMB2: Znak přijatý v režimu Freeport

SMB2 je zásobník pro znak přijatý z Freeportu. Jak je popsáno v tabulce D-3, každý znak přijatý během režimu Freeport je umístěn na tuto adresu kvůli jednoduchému přístupu z programu kontaktního schématu.



Tip

SMB2 a SMB3 jsou sdíleny mezi portem 0 a portem 1. Když se po přijetí znaku na portu 0 provede podprogram přerušení přiřazený k této události (událost 8), obsahuje SMB2 znak přijatý na portu 0 a SMB3 stav parity přijatého znaku. Když se po přijetí znaku na portu 1 provede podprogram přerušení přiřazený k této události (událost 25), obsahuje SMB2 znak přijatý na portu 1 a SMB3 stav parity přijatého znaku.

Tabulka D-3 Speciální paměťový byte SMB2

SM byte	Popis (pouze ke čtení)
SMB2	Tento byte obsahuje každý znak, který je přijat z portu 0 nebo portu 1 během komunikace v režimu Freeport.

SMB3: Chyba parity v režimu Freeport

SMB3 se používá pro režim Freeport a obsahuje bit chyby parity, který je nastaven při detekci chyby parity u přijatého znaku. Jak je ukázáno v tabulce D-4, SM3.0 se zapne, když je zjištěna chyba parity. Tento bit použijte pro detekci chyby zprávy.

Tabulka D-4 Speciální paměťový byte SMB3 (SM3.0 až SM3.7)

SM bity	Popis (pouze ke čtení)
SM3.0	Chyba parity z portu 0 nebo portu 1 (0 = bez chyby; 1 = zjištěna chyba)
SM3.1 až SM3.7	Rezervováno

SMB4: Přeplnění fronty

Jak je popsáno v tabulce D-5, obsahuje SMB4 bity přeplnění fronty přerušení, dále stavový indikátor, který ukazuje, zda jsou přerušení povolena nebo zablokována, a paměťový bit nečinnosti vysílače. Bity přeplnění fronty indikují, že přerušení jsou generována rychleji, než je možné je zpracovat, nebo že přerušení byla zablokována instrukcí globálního zablokování přerušení.

Tabulka D-5 Speciální paměťový byte SMB4 (SM4.0 až SM4.7)

SM bity	Popis (pouze ke čtení)
SM4.0 ¹	Tento bit přejde do log. 1, když došlo k přeplnění fronty komunikačních přerušení.
SM4.1 ¹	Tento bit přejde do log. 1, když došlo k přeplnění fronty vstupních přerušení.
SM4.2 ¹	Tento bit přejde do log. 1, když došlo k přeplnění fronty časovaných přerušení.
SM4.3	Tento bit přejde do log. 1, když je zjištěn problém programování při běhu programu.
SM4.4	Tento bit odráží stav globálního povolení přerušení. Zapne se, když jsou přerušení povolena.
SM4.5	Tento bit přejde do log. 1, když je vysílač nečinný (port 0).
SM4.6	Tento bit přejde do log. 1, když je vysílač nečinný (port 1).
SM4.7	Tento bit přejde do log. 1 při vnucení (force) nějaké hodnoty.

¹ Stavové bity 4.0, 4.1 a 4.2 používejte pouze v podprogramu přerušení. Při vyprázdnění fronty jsou tyto stavové bity vynulovány a řízení se vrací do hlavního programu.

SMB5: Stav vstupů a výstupů

Jak popisuje tabulka D-6, obsahuje SMB5 stavové bity indikující chyby, které byly zjištěny v systému vstupů a výstupů. Tyto bity poskytují přehled zjištěných chyb vstupů a výstupů.

Tabulka D-6 Speciální paměťový byte SMB5 (SM5.0 až SM5.7)

SM bity	Popis (pouze ke čtení)
SM5.0	Tento bit přejde do log. 1, existují-li nějaké chyby vstupů a výstupů.
SM5.1	Tento bit přejde do log. 1, jestliže bylo na I/O sběrnici připojeno příliš mnoho digitálních vstupů/výstupů.
SM5.2	Tento bit přejde do log. 1, jestliže bylo na I/O sběrnici připojeno příliš mnoho analogových vstupů/výstupů.
SM5.3	Tento bit přejde do log. 1, jestliže bylo na I/O sběrnici připojeno příliš mnoho inteligentních modulů.
SM5.4 až SM5.7	Rezervováno.

SMB6: Registr ID CPU

Jak popisuje tabulka D-7, SMB6 je identifikační registr CPU S7-200. SM6.4 až SM6.7 identifikují typ CPU S7-200. SM6.0 až SM6.3 jsou rezervovány pro použití v budoucnosti.

Tabulka D-7 Speciální paměťový byte SMB6

SM bity	Popis (pouze ke čtení)									
Formát	MSB 7 LSB 0 x x x x r r r r Registr ID CPU									
SM6.0 až SM6.3	Rezervováno									
SM6.4 až SM6.7	xxxx = 0000 = CPU 222 0010 = CPU 224 0110 = CPU 221 1001 = CPU 226/CPU 226XM									

SMB7: Rezervováno

SMB7 je rezervován pro použití v budoucnosti.

SMB8 až SMB21: ID a chybové registry I/O modulů

SMB8 až SMB21 jsou organizovány do dvojic bytů pro rozšiřovací moduly na pozici 0 až 6. Jak popisuje tabulka D-8, v každé dvojici je byte se sudým číslem identifikační registr modulu. Tyto byty identifikují typ modulu, typ vstupů a výstupů a jejich počet. Byte s lichým číslem v každé dvojici je chybový registr modulu. Tyto byty indikují jakékoliv chyby zjištěné ve vstupech/výstupech tohoto modulu.

Tabulka D-8 Speciální paměťové byty SMB8 až SMB21

SM byte	Popis (pouze ke čtení)	
Formát	Byte se sudým číslem: Registr ID modulu	Byte s lichým číslem: Chybový registr modulu
	MSB7LSB0 m t t a i i q q m: Přítomnost modulu 0 = Přítomen 1 = Nepřítomen tt: Typ modulu 00 Jiný než inteligentní vstupní/výstupní modul 01 Inteligentní modul 10 Rezervováno 11 Rezervováno a: Typ vstupu/výstupu 0 = Digitální 1 = Analogový ii: Vstupy 00 Žádné vstupy 01 2 AI nebo 8 DI 10 4 AI nebo 16 DI 11 8 AI nebo 32 DI qq: Výstupy 00 Žádné výstupy 01 2 AQ nebo 8 DQ 10 4 AQ nebo 16 DQ 11 8 AQ nebo 32 DQ	MSB7LSB0 c 0 0 b r p f t c: Chyba konfigurace 0 = bez chyby 1 = chyba b: Chyba sběrnice nebo chyba parity r: Chyba překročení rozsahu p: Chyba – nepřítomnost uživ. napájení f: Chyba – spálená pojistka t: Chyba – uvolněná svorkovnice
SMB8 SMB9	Registr ID modulu 0 Chybový registr modulu 0	
SMB10 SMB11	Registr ID modulu 1 Chybový registr modulu 1	
SMB12 SMB13	Registr ID modulu 2 Chybový registr modulu 2	
SMB14 SMB15	Registr ID modulu 3 Chybový registr modulu 3	
SMB16 SMB17	Registr ID modulu 4 Chybový registr modulu 4	
SMB18 SMB19	Registr ID modulu 5 Chybový registr modulu 5	
SMB20 SMB21	Registr ID modulu 6 Chybový registr modulu 6	

SMW22 až SMW26: Délka programového cyklu

Jak je uvedeno v tabulce D-9, SMW22, SMW24 a SMW26 obsahují informace o programovém cyklu: minimální dobu, maximální dobu a dobu posledního programového cyklu v milisekundách.

Tabulka D-9 Speciální paměťová word SMW22 až SMW26

SM word	Popis (pouze ke čtení)
SMW22	Doba posledního programového cyklu v milisekundách
SMW24	Minimální doba programového cyklu, zaznamenaná od začátku režimu RUN v milisekundách
SMW26	Maximální doba programového cyklu, zaznamenaná od začátku režimu RUN v milisekundách

SMB28 a SMB29: Analogové potenciometry

Jak je uvedeno v tabulce D-10, obsahuje SMB28 číselnou hodnotu, která představuje polohu analogového potenciometru 0. SMB29 obsahuje hodnotu, která představuje polohu analogového potenciometru 1.

Tabulka D-10 Speciální paměťové byty SMB28 a SMB29

SM byte	Popis (pouze ke čtení)
SMB28	Tento byte uchovává hodnotu nastavenou analogovým potenciometrem 0. Daná hodnota je aktualizována jedenkrát za programový cyklus v STOP/RUN.
SMB29	Tento byte uchovává hodnotu nastavenou analogovým potenciometrem 1. Daná hodnota je aktualizována jedenkrát za programový cyklus v STOP/RUN.

SMB30 a SMB130: Řídící registry pro Freeport

SMB30 řídí komunikaci v režimu Freeport na portu 0; SMB130 řídí komunikaci na portu 1. Byty SMB30 a SMB130 jsou pro čtení i zápis. Jak je uvedeno v tabulce D-11, tyto byty konfiguruji příslušné porty na komunikaci Freeport a zabezpečují přepnutí mezi režimem Freeport a systémovým protokolem.

Tabulka D-11 Speciální paměťový byte SMB30

Port 0	Port 1	Popis
Formát SMB30	Formát SMB130	Řídící byte režimu Freeport MSB 7 LSB 0 ppdbbbmm
SM30.0 a SM30.1	SM130.0 a SM130.1	mm: Volba protokolu 00 = Protokol Point-to-Point (režim PPI/slave) 01 = protokol Freeport 10 = režim PPI/master 11 = Rezervováno (implicitně režim PPI/slave) Poznámka: Když vyberete kód mm = 10 (PPI master), automat S7-200 se stane v síti zařízením typu master a dovolí vykonání instrukcí NETR a NETW. Bity 2 až 7 jsou v režimech PPI ignorovány.
SM30.2 až SM30.4	SM130.2 až SM130.4	bbb: Přenosová rychlost Freeport 000 = 38.400 baud 001 = 19.200 baud 010 = 9.600 baud 011 = 4.800 baud 100 = 2.400 baud 101 = 1.200 baud 110 = 115.200 baud 111 = 57.600 baud
SM30.5	SM130.5	d: Datové bity na znak 0 = 8 bitů na znak 1 = 7 bitů na znak
SM30.6 a SM30.7	SM130.6 a SM130.7	pp: Volba parity 00 = žádná parita 01 = sudá parita 10 = lichá parita 11 = lichá parita

SMB31 a SMW32: Řízení zápisu do trvalé paměti (EEPROM)

Prostřednictvím vašeho programu můžete uložit hodnotu uloženou v paměti V do trvalé paměti (EEPROM), což provedete tak, že adresu místa, jehož obsah chcete uložit, nahrajete do SMW32. Potom nahrajete do SMB31 příkaz k uložení hodnoty. Jakmile to provedete, neměňte hodnotu v paměti V, dokud automat S7-200 neresetuje SM31.7; to znamená, že operace ukládání byla ukončena.

Na konci každého programového cyklu S7-200 kontroluje, zda byl vydán příkaz k uložení hodnoty do trvalé paměti. Pokud byl příkaz vydán, je daná hodnota uložena do trvalé paměti.

Jak je uvedeno v tabulce D-12, definuje SMB31 velikost dat, která mají být uložena do trvalé paměti, a dává příkaz, který inicializuje operaci uložení. SMW32 uchovává počáteční adresu v paměti V pro data, která mají být uložena do trvalé paměti.

Tabulka D-12 Speciální paměťový byte SMB31 a speciální paměťový word SMW32

SM byte	Popis
Formát	SMB31: Příkaz softwaru MSB 7 0 0 0 0 0 0 1 1 LSB SMW32: adresa v paměti V MSB 15 adresa v paměti V LSB 0
SM31.0 a SM31.1	ss: Velikost dat 00 = byte 10 = word 01 = byte 11 = double word
SM31.7	c: Uložit do EEPROM 0 = Není požadavek na operaci uložení 1 = Uživatelský program požaduje uložení dat Automat S7-200 resetuje tento bit po každé operaci uložení.
SMW32	Adresa v paměti V pro data, která mají být uložena do trvalé paměti, je uložena v SMW32. Tato hodnota je daná jako offset od V0. Při provádění operace uložení je hodnota na této adrese v paměti V uložena na místo odpovídající paměti V v trvalé paměti (EEPROM).

SMB34 a SMB35: Intervaly pro časovaná přerušení

Jak je uvedeno v tabulce D-13, specifikuje SMB34 časový interval pro cyklické přerušení 0 a SMB35 časový interval pro cyklické přerušení 1. Můžete specifikovat časový interval od 1 ms do 255 ms (v přírůstcích po 1 ms). Hodnota časového intervalu je zachycena v S7-200 v okamžiku, kdy je odpovídající událost cyklického přerušení přiřazena k podprogramu přerušení. Chcete-li změnit časový interval, musíte znovu přiřadit událost časovaného přerušení ke stejnému nebo jinému podprogramu přerušení. Událost časovaného přerušení je možné ukončit zrušením přiřazení události k podprogramu přerušení.

Tabulka D-13 Speciální paměťové byty SMB34 a SMB35

SM byte	Popis
SMB34	Tento byte specifikuje interval (v přírůstcích po 1 ms, od 1 ms do 255 ms) pro časové přerušení 0.
SMB35	Tento byte specifikuje interval (v přírůstcích po 1 ms, od 1 ms do 255 ms) pro časové přerušení 1.

SMB36 až SMB65: Registr HSC0, HSC1 a HSC2

Jak popisuje tabulka D-14, SMB36 až SM65 se používají na monitorování a řízení práce vysokorychlostních čítačů HSC0, HSC1 a HSC2.

Tabulka D-14 Speciální paměťové byty SMB36 až SMD62

SM byte	Popis
SM36.0 až SM36.4	Rezervováno
SM36.5	Stavový bit aktuálního směru počítání HSC0: 1 = počítání nahoru
SM36.6	Stavový bit rovnosti požadované a aktuální hodnoty HSC0: 1 = rovnost
SM36.7	Stavový bit příznaku aktuální hodnota větší než požadovaná u HSC0: 1 = větší
SM37.0	Aktivní úroveň říd. bitu pro "Reset": 0 = "Reset" je akt. v log. 1, 1 = "Reset" je akt. v log. 0
SM37.1	Rezervováno
SM37.2	Výběr rychlosti čítání pro kvadrurní čítače: 0 = 4 x rychlost čítání; 1 = 1 x rychlost čítání
SM37.3	Řídící bit pro směr HSC0: 1 = počítání nahoru
SM37.4	Aktualizace směru HSC0: 1 = aktualizovat směr
SM37.5	Aktualizace požadované hodnoty HSC0: 1 = zapsat novou pož. hodnotu do nastavení HSC0
SM37.6	Aktualizace aktuální hodnoty HSC0: 1 = zapsat novou aktuální hodn. do stávající HSC0
SM37.7	Bit povolení HSC0: 1 = povolen
SMD38	Nová aktuální hodnota HSC0
SMD42	Nová požadovaná hodnota HSC0
SM46.0 až SM46.4	Rezervováno
SM46.5	Stavový bit aktuálního směru počítání HSC1: 1 = počítání nahoru
SM46.6	Stavový bit rovnosti požadované a aktuální hodnoty HSC1: 1 = rovnost
SM46.7	Stavový bit příznaku aktuální hodnota větší než požadovaná u HSC1: 1 = větší
SM47.0	Aktivní úroveň říd. bitu pro "Reset" HSC1: 0 = aktivní v úrovni log. 1, 1 = aktivní v úrovni log. 0
SM47.1	Aktivní úroveň říd. bitu pro start HSC1: 0 = aktivní v úrovni log. 1, 1 = aktivní v úrovni log. 0
SM47.2	Volba rychlosti kvadrurního čítače HSC1: 0 = 4 x rychlost, 1 = 1 x rychlost
SM47.3	Řídící bit pro směr HSC1: 1 = počítání nahoru
SM47.4	Aktualizace směru HSC1: 1 = aktualizovat směr
SM47.5	Aktualizace pož. hodnoty HSC1: 1 = zapsat novou požadovanou hodnotu do nastavení HSC1
SM47.6	Aktualizace aktuální hodnoty HSC1: 1 = zapsat novou aktuální hodn. do stávající HSC1
SM47.7	Bit povolení HSC1: 1 = povolen
SMD48	Nová aktuální hodnota HSC1
SMD52	Nová požadovaná hodnota HSC1
SM56.0 až SM56.4	Rezervováno
SM56.5	Stavový bit aktuálního směru počítání HSC2: 1 = počítání nahoru
SM56.6	Stavový bit rovnosti požadované a aktuální hodnoty HSC2: 1 = rovnost
SM56.7	Stavový bit příznaku aktuální hodnota větší než požadovaná u HSC2: 1 = větší
SM57.0	Aktivní úroveň řídícího bitu pro "Reset" HSC2: 0 = akt. v úrovni log. 1, 1 = akt. v úrovni log. 0
SM57.1	Aktivní úroveň řídícího bitu pro start HSC2: 0 = akt. v úrovni log. 1, 1 = aktivní v úrovni log. 0
SM57.2	Volba rychlosti kvadrurního čítače HSC2: 0 = 4 x rychlost, 1 = 1 x rychlost
SM57.3	Řídící bit pro směr HSC2: 1 = počítání nahoru
SM57.4	Aktualizace směru HSC2: 1 = aktualizovat směr
SM57.5	Aktualizace požadované hodnoty HSC2: 1 = zapsat novou pož. hodnotu do nastavení HSC2
SM57.6	Aktualizace aktuální hodnoty HSC2: 1 = zapsat novou aktuální hodn. do stávající HSC2
SM57.7	Bit povolení HSC2: 1 = povolen
SMD58	Nová aktuální hodnota HSC2
SMD62	Nová požadovaná hodnota HSC2

SMB66 až SMB85: Registry PTO/PWM

Jak je uvedeno v tabulce D-15, SMB66 až SMB85 se používají pro monitorování a řízení funkce výstupu série rychlých pulzů a pulzní šířkové modulace. Úplný popis těchto bitů najdete v informacích o instrukcích pro vysokorychlostní pulzní výstup v kapitole 6.

Tabulka D-15 Speciální paměťové byty SMB66 až SMB85

SM byte	Popis
SM66.0 až SM66.3	Rezervováno
SM66.4	Profil PTO0 přerušen: 0 = bez chyby, 1 = přerušen pro chybu ve výpočtu změny
SM66.5	Profil PTO0 přerušen: 0 = nebyl přerušen uživatelským příkazem, 1 = přerušen uživatelským příkazem
SM66.6	Přeplnění PTO0 (smazáno systémem při použití vnějších profilů, jinak musí být smazáno uživatelem): 0 = nepřeplněno, 1 = přeplněno
SM66.7	Bit nečinnosti PTO0: 0 = PTO pracuje, 1 = PTO nečinný
SM67.0	Aktualizace hodnoty doby cyklu PTO0/PWM0: 1 = zápis nové doby cyklu
SM67.1	Aktualizace hodnoty šířky pulzu PWM0: 1 = zápis nové šířky pulzu
SM67.2	Aktualizace hodnoty počtu pulzů PTO0: 1 = zápis nového počtu pulzů
SM67.3	Časová základna PTO0/PWM0: 0 = 1 μ s/pulz, 1 = 1 ms/pulz
SM67.4	Synchronní aktualizace PWM0: 0 = asynchronní aktualizace, 1 = synchronní aktualizace
SM67.5	Práce PTO0: 0 = práce s jedním segmentem (doba cyklu a počet pulzů uloženy v paměti SM), 1 = práce s více segmenty (tabulka profilu uložena v paměti V)
SM67.6	Volba režimu PTO0/PWM0: 0 = PTO, 1 = PWM
SM67.7	Povolovací bit PTO0/PWM0: 1 = povolen
SMW68	Hodnota doby cyklu PTO0/PWM0 (2 až 65.535 jednotek časové základny);
SMW70	Hodnota šířky pulzu PWM0 (0 až 65.535 jednotek časové základny);
SMD72	Hodnota počtu pulzů PTO0 (1 až $2^{32} - 1$);
SM76.0 až SM76.3	Rezervováno
SM76.4	Profil PTO0 přerušen: 0 = bez chyby, 1 = přerušen pro chybu ve výpočtu změny
SM76.5	Profil PTO0 přerušen: 0 = nebyl přerušen uživatelským příkazem, 1 = přerušen uživatelským příkazem
SM76.6	Přeplnění PTO1 (smazáno systémem při použití vnějších profilů, jinak musí být smazáno uživatelem): 0 = nepřeplněno, 1 = přeplněno
SM76.7	Bit nečinnosti PTO1: 0 = PTO pracuje, 1 = PTO nečinný
SM77.0	Aktualizace hodnoty doby cyklu PTO1/PWM1: 1 = zápis nové doby cyklu
SM77.1	Aktualizace hodnoty šířky pulzu PWM1: 1 = zápis nové šířky pulzu
SM77.2	Aktualizace hodnoty počtu pulzů PTO1: 1 = zápis nového počtu pulzů
SM77.3	Časová základna PTO1/PWM1: 0 = 1 μ s/pulz, 1 = 1 ms/pulz
SM77.4	Synchronní aktualizace PWM1: 0 = asynchronní aktualizace, 1 = synchronní aktualizace
SM77.5	Práce PTO1: 0 = práce s jedním segmentem (doba cyklu a počet pulzů uloženy v paměti SM), 1 = práce s více segmenty (tabulka profilu uložena v paměti V)
SM77.6	Volba režimu PTO1/PWM1: 0 = PTO, 1 = PWM
SM77.7	Povolovací bit PTO1/PWM1: 1 = povolen
SMW78	Hodnota doby cyklu PTO1/PWM1 (2 až 65.535 jednotek časové základny);
SMW80	Hodnota šířky pulzu PWM1 (0 až 65.535 jednotek časové základny);
SMD82	Hodnota počtu pulzů PTO1 (1 až $2^{32} - 1$);

SMB86 až SMB94 a SMB186 až SMB194: Řízení příjmu zpráv

Jak je uvedeno v tabulce D-16, SMB86 až SMB94 a SMB186 až SMB194 se používají pro řízení a čtení stavu instrukce pro příjem zpráv.

Tabulka D-16 Speciální paměťové byty SMB86 až SMB94 a SMB186 až SMB194

Port 0	Port 1	Popis								
SMB86	SMB186	Stavový byte pro příjem zprávy MSB7 LSB0 <table><tr><td>n</td><td>r</td><td>e</td><td>0</td><td>0</td><td>t</td><td>c</td><td>p</td></tr></table> n: 1 = Příjem zprávy ukončen uživatelským příkazem pro ukončení r: 1 = Příjem zprávy ukončen: chyba vstupních parametrů nebo chybí startovací či ukončovací podmínka e: 1 = Přijat ukončovací znak t: 1 = Příjem zprávy ukončen: vypršel čas c: 1 = Příjem zprávy ukončen: dosaženo maximálního počtu přenášených znaků p 1 = Příjem zprávy ukončen pro chybu parity	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			
SMB87	SMB187	Řídící byte pro příjem zprávy MSB7 LSB0 <table><tr><td>en</td><td>sc</td><td>ec</td><td>il</td><td>c/m</td><td>tmr</td><td>bk</td><td>0</td></tr></table> en: 0 = Funkce příjmu zpráv je zablokováná. 1 = Funkce příjmu zpráv je povolena. Povolovací bit příjmu zpráv je kontrolován při každém provedení instrukce RCV. sc: 0 = Ignorovat SMB88 nebo SMB188. 1 = Použít hodnotu SMB88 nebo SMB188 pro rozpoznání začátku zprávy. ec: 0 = Ignorovat SMB89 nebo SMB189. 1 = Použít hodnotu SMB89 nebo SMB189 pro rozpoznání konce zprávy. il: 0 = Ignorovat SMW90 nebo SMW190. 1 = Použít hodnotu SMW90 nebo SMW190 pro rozpoznání prázdného úseku. c/m: 0 = Časovač je použit na časování mezi znaky. 1 = Časovač je použit na časování zpráv. tmr: 0 = Ignorovat SMW92 nebo SMW192. 1 = Ukončit příjem, pokud byl překročen čas v SMW92 nebo SMW192. bk: 0 = Ignorovat pauzu (break). 1 = Použít pauzu (break) pro detekci začátku zprávy.	en	sc	ec	il	c/m	tmr	bk	0
en	sc	ec	il	c/m	tmr	bk	0			
SMB88	SMB188	Znak začátku zprávy								
SMB89	SMB189	Znak konce zprávy								
SMW90	SMW190	Doba prázdného úseku uvedená v milisekundách. První znak přijatý po skončení prázdného úseku je startovací znak.								
SMW92	SMW192	Hraniční hodnota pro časovač, který měří čas mezi znaky/zprávami (v milisekundách). Pokud je hodnota překročena, je příjem zpráv ukončen.								
SMB94	SMB194	Maximální počet přijatých znaků (1 až 255 bytů). Poznámka: Tato hodnota musí být nastavena na předpokládanou maximální velikost zásobníku i v případě, že není použito ukončení příjmu zprávy při dosažení maximálního počtu znaků.								

SMW98: Chyby na rozšiřovací I/O sběrnici

Jak popisuje tabulka D-17, poskytuje SMW98 informace o počtu chyb na rozšiřovací I/O sběrnici.

Tabulka D-17 Speciální paměťové byty SMW98

SM byte	Popis
SMW98	Hodnota tohoto paměťového místa je zvýšena při každém zjištění chyby na rozšiřovací I/O sběrnici. Hodnota je vymazána při studeném restartu, může být vymazána uživatelem.

SMB130: Řídicí registr pro Freeport (viz SMB30)

Viz tabulku D-11.

SMB131 až SMB165: Registr HSC3, HSC4 a HSC5

Jak popisuje tabulka D-18, SMB131 až SMB165 se používají pro monitorování a řízení práce vysokorychlostních čítačů HSC3, HSC4 a HSC5.

Tabulka D-18 Speciální paměťové byty SMB131 až SMB165

SM byte	Popis
SMB131 až SMB135	Rezervováno
SM136.0 až SM136.4	Rezervováno
SM136.5	Stavový bit aktuálního směru počítání HSC3: 1 = počítání nahoru
SM136.6	Stavový bit rovnosti požadované a aktuální hodnoty HSC3: 1 = rovnost
SM136.7	Stavový bit příznaku aktuální hodnota větší než požadovaná u HSC3: 1 = větší
SM137.0 až SM137.2	Rezervováno
SM137.3	Řídicí bit pro směr HSC3: 1 = počítání nahoru
SM137.4	Směr aktualizace HSC3: 1 = aktualizovat směr
SM137.5	Aktualizace požadované hodnoty HSC3: 1 = zapsat novou požadovanou hodnotu do HSC3
SM137.6	Aktualizace aktuální hodnoty HSC3: 1 = zapsat novou aktuální hodnotu do stávající hodnoty HSC3
SM137.7	Bit povolení HSC3: 1 = povolen
SMD138	Nová aktuální hodnota HSC3
SMD142	Nová požadovaná hodnota HSC3
SM146.0 až SM146.4	Rezervováno
SM146.5	Stavový bit aktuálního směru počítání HSC4: 1 = počítání nahoru
SM146.6	Stavový bit rovnosti požadované a aktuální hodnoty HSC4: 1 = rovnost
SM146.7	Stavový bit příznaku aktuální hodnota větší než požadovaná u HSC4: 1 = větší
SM147.0	Aktivní úroveň řídicího bitu pro "Reset": 0 = "Reset" je aktivní na vysoké úrovni, 1 = "Reset" je aktivní na nízké úrovni
SM147.1	Rezervováno
SM147.2	Výběr rychlosti čítání pro kvadraturní čítače: 0 = 4 x rychlost čítání, 1 = 1 x rychlost čítání
SM147.3	Řídicí bit pro směr HSC4: 1 = počítání nahoru
SM147.4	Směr aktualizace HSC4: 1 = aktualizovat směr
SM147.5	Aktualizace požadované hodnoty HSC4: 1 = zapsat novou požadovanou hodnotu do HSC4
SM147.6	Aktualizace aktuální hodnoty HSC4: 1 = zapsat novou aktuální hodnotu do stávající hodnoty HSC4
SM147.7	Bit povolení HSC4: 1 = povolen

Tabulka D-18 Speciální paměťové byty SMB131 až SMB165, pokračování

SM byte	Popis
SMD148	Nová aktuální hodnota HSC4
SMD152	Nová požadovaná hodnota HSC4
SM156.0 až SM156.4	Rezervováno
SM156.5	Stavový bit aktuálního směru počítání HSC5: 1 = počítání nahoru
SM156.6	Stavový bit rovnosti požadované a aktuální hodnoty HSC5: 1 = rovnost
SM156.7	Stavový bit příznaku aktuální hodnota větší než požadovaná u HSC5: 1 = větší
SM157.0 až SM157.2	Rezervováno
SM157.3	Řídící bit pro směr HSC5: 1 = počítání nahoru
SM157.4	Směr aktualizace HSC5: 1 = aktualizovat směr
SM157.5	Aktualizace požadované hodnoty HSC5: 1 = zapsat novou požadovanou hodnotu do HSC5
SM157.6	Aktualizace aktuální hodnoty HSC5: 1 = zapsat novou aktuální hodnotu do stávající hodnoty HSC5
SM157.7	Bit povolení HSC5: 1 = povolen
SMD158	Nová aktuální hodnota HSC5
SMD162	Nová požadovaná hodnota HSC5

SMB166 až SMB185: Tabulka definice profilu PTO0, PTO1

Jak popisuje tabulka D-19, SMB166 až SMB185 se používají na zobrazení počtu aktivních kroků profilu a adresy tabulky profilu v paměti V.

Tabulka D-19 Speciální paměťové byty SMB166 až SMB185

SM byte	Popis
SMB166	Aktuální číslo záznamu aktivního kroku profilu pro PTO0
SMB167	Rezervováno
SMD168	Adresa tabulky profilu v paměti V pro PTO0 zadaná jako offset od V0.
SMB170 až SMB175	Rezervováno
SMB176	Aktuální číslo záznamu aktivního kroku profilu pro PTO1
SMB177	Rezervováno
SMD178	Adresa tabulky profilu v paměti V pro PTO1 zadaná jako offset od V0.
SMB180 až SMB185	Rezervováno

SMB186 až SMB194: Řízení příjmu zpráv (viz SMB86 až SMB94)

Viz tabulku D-16.

SMB200 až SMB549: Stav inteligentních modulů

Jak ukazuje tabulka D-20, SMB200 až SMB549 jsou vyhrazeny pro informace poskytované inteligentními rozšiřovacími moduly, například modulem EM 277 PROFIBUS–DP. Informace, jak modul využívá SMB200 až SMB549, najdete v příloze A, uvádějící technická data konkrétního modulu.

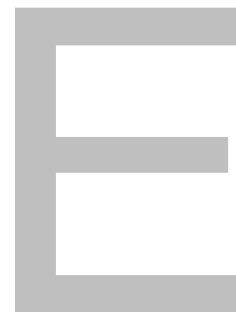
Způsob přiřazování míst v paměti SM inteligentním modulům se pro verzi firmware 2.2 a pozdější změnil.

U CPU S7-200 s verzí firmware starší než 1.2 musíte instalovat inteligentní modul přímo vedle CPU, aby byla zajištěna kompatibilita.

Tabulka D-20 Speciální paměťové byty SMB200 až SMB549

Speciální paměťové byty SMB200 až SMB549							
Inteligentní modul na pozici 0	Inteligentní modul na pozici 1	Inteligentní modul na pozici 2	Inteligentní modul na pozici 3	Inteligentní modul na pozici 4	Inteligentní modul na pozici 5	Inteligentní modul na pozici 6	Popis
SMB200 až SMB215	SMB250 až SMB265	SMB300 až SMB315	SMB350 až SMB365	SMB400 až SMB415	SMB450 až SMB465	SMB500 až SMB515	Název modulu (16 ASCII znaků)
SMB216 až SMB219	SMB266 až SMB269	SMB316 až SMB319	SMB366 až SMB369	SMB416 až SMB419	SMB466 až SMB469	SMB516 až SMB519	Verze firmwaru (4 ASCII znaky)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	Kód chyby
SMB222 až SMB249	SMB272 až SMB299	SMB322 až SMB349	SMB372 až SMB399	SMB422 až SMB449	SMB472 až SMB499	SMB522 až SMB549	Informace, specifické danému modulu

Objednací čísla S7-200



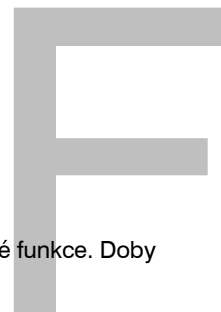
CPU	Objednací číslo
CPU 221 DC/DC/DC 6 vstupů/ 4 výstupy	6ES7 211-0AA22-0XB0
CPU 221 AC/DC/relé 6 vstupů/4 relé	6ES7 211-0BA22-0XB0
CPU 222 DC/DC/DC 8 vstupů/6 výstupů	6ES7 212-1AB22-0XB0
CPU 222 AC/DC/relé 8 vstupů/6 relé	6ES7 212-1BB22-0XB0
CPU 224 DC/DC/DC 14 vstupů/10 výstupů	6ES7 214-1AD22-0XB0
CPU 224 AC/DC/relé 14 vstupů/10 relé	6ES7 214-1BD22-0XB0
CPU 226 DC/DC/DC 24 vstupů/16 výstupů	6ES7 216-2AD22-0XB0
CPU 226 AC/DC/relé 24 vstupů/16 relé	6ES7 216-2BD22-0XB0
CPU 226XM DC/DC/DC 24 vstupů/16 relé	6ES7 216-2AF22-0XB0
CPU 226XM AC/DC/relé 24 vstupů/16 relé	6ES7 216-2BF22-0XB0
Rozšiřovací moduly	Objednací číslo
EM 221 24 V DC digitální 8 vstupů	6ES7 221-1BF22-0XA0
EM 221 digitální 8 AC vstupů (8 x 120/230 V AC)	6ES7 221-1EF22-0XA0
EM 222 24 V DC digitální 8 výstupů	6ES7 222-1BF22-0XA0
EM 222 digitální výstup 8 x relé	6ES7 222-1HF22-0XA0
EM 222 digitální 8 AC výstupů (8 x 120/230 V AC)	6ES7 222-1EF22-0XA0
EM 223 24 V DC digitální kombinace 4 vstupů/4 výstupů	6ES7 223-1BF22-0XA0
EM 223 24 V DC digitální kombinace 4 vstupů/4 reléových výstupů	6ES7 223-1HF22-0XA0
EM 223 24 V DC digitální kombinace 8 vstupů/8 výstupů	6ES7 223-1BH22-0XA0
EM 223 24 V DC digitální kombinace 8 vstupů/8 reléových výstupů	6ES7 223-1PH22-0XA0
EM 223 24 V DC digitální kombinace 16 vstupů/16 výstupů	6ES7 223-1BL22-0XA0
EM 223 24 V DC digitální kombinace 16 vstupů/16 reléových výstupů	6ES7 223-1PL22-0XA0
EM 231 analogový vstup, 4 vstupy	6ES7 231-0HC22-0XA0
EM 231 analogový vstup RTD, 2 vstupy	6ES7 231-7PB22-0XA0
EM 231 analogový vstup z termočlánku, 4 vstupy	6ES7 231-7PD22-0XA0
EM 232 analogový výstup, 2 výstupy	6ES7 232-0HB22-0XA0
EM 235 analogová kombinace 4 vstupy/1 výstup	6ES7 235-0KD22-0XA0
Modemový modul EM 241	6ES7 241-1AA22-0XA0
Polohovací modul EM 253	6ES7 253-1AA22-0XA0
EM 277 PROFIBUS-DP slave	6ES7 277-0AA22-0XA0
CP 243-1 komunikační procesor Ethernet	6GK7 243-1EX00-0XE0
CP 243-2 komunikační procesor AS Interface	6GK7 243-2AX01-0XA0

Zásuvné moduly a kabely	Objednací číslo
MC 291, paměťový modul 32K x 8 EEPROM	6ES7 291 – 8GE20 – 0XA0
CC 292, bateriový modul s integrovanými hodinami reálného času - pro CPU 22x	6ES7 297 – 1AA20 – 0XA0
BC 293, bateriový modul pro CPU 22x	6ES7 291 – 8BA20 – 0XA0
Kabel pro rozšiřovací moduly, 0,8 metru, pro CPU 22x/EM	6ES7 290 – 6AA20 – 0XA0
Kabel, PC/PPI, napětově oddělený, konektor 90 stupňů, přepínač RTS	6ES7 901 – 3BF21 – 0XA0
Programovací software	Objednací číslo
STEP 7 – Micro/WIN 32 (V3.2) plná verze (CD-ROM)	6ES7 810 – 2BC02 – 0YX0
STEP 7 – Micro/WIN 32 (V3.2) upgrade (CD-ROM)	6ES7 810 – 2BC02 – 0YX3
TP – Designer pro TP070, verze 1.0 (CD-ROM)	6ES7 850 – 2BC00 – 0YX0
Add – On pro STEP 7 – Micro/WIN: Knihovna instrukcí V1.1 (CD-ROM)	6ES7 830 – 2BC00 – 0YX0

Komunikační karty	Objednací číslo
CP 5411: Krátká AT ISA	6GK 1 541 – 1AA00
CP 5511: PCMCIA, typ II	6GK 1 551 – 1AA00
CP 5611: Karta PCI (verze 3.0 nebo novější)	6GK 1 561 – 1AA00
Manuály	Objednací číslo
Uživatelský manuál pro textový displej TD 200	6ES7 272 – 0AA20 – 8BA0
Uživatelský manuál pro dotykový panel TP 070 (v angličtině)	6AV6 591 – 1DC01 – 0AB0
Manuál pro komunikaci S7-200 v protokolu Point-to-Point Interface (v angličtině/němčině)	6ES7 298 – 8GA00 – 8XH0
Manuál pro komunikační procesor CP 243–2 SIMATIC NET AS-Interface Master (v angličtině)	6GK7 243 – 2AX00 – 8BA0
Systémový manuál pro programovatelný automat S7-200 (v němčině)	6ES7 298 – 8FA22 – 8AH0
Systémový manuál pro programovatelný automat S7-200 (v angličtině)	6ES7 298 – 8FA22 – 8BH0
Systémový manuál pro programovatelný automat S7-200 (ve francouzštině)	6ES7 298 – 8FA22 – 8CH0
Systémový manuál pro programovatelný automat S7-200 (ve španělštině)	6ES7 298 – 8FA22 – 8DH0
Systémový manuál pro programovatelný automat S7-200 (v italštině)	6ES7 298 – 8FA22 – 8EH0
Kabely, síťové konektory a repeatery	Objednací číslo
MPI kabel	6ES7 901 – 0BF00 – 0AA0
Síťový kabel PROFIBUS	6XVI 830 – 0AH10
Konektor pro síť s konektorem programovacího portu, svislý kabelový výstup	6ES7 972 – 0BB11 – 0XA0
Konektor pro síť (bez konektoru programovacího portu), svislý kabelový výstup	6ES7 972 – 0BA11 – 0XA0
Sběrníkový konektor RS-485 s kabelovým výstupem 35° (bez konektoru programovacího portu)	6ES7 972 – 0BA40 – 0XA0
Sběrníkový konektor RS-485 s kabelovým výstupem 35° (s konektorem programovacího portu)	6ES7 972 – 0BB40 – 0XA0
Svorkovnice CPU 22x/EM, 7 svorek, odnímatelná	6ES7 292 – 1AD20 – 0AA0
Svorkovnice CPU 22x/EM, 12 svorek, odnímatelná	6ES7 292 – 1AE20 – 0AA0
Svorkovnice CPU 22x/EM, 14 svorek, odnímatelná	6ES7 292 – 1AF20 – 0AA0
Svorkovnice CPU 22x/EM, 18 svorek, odnímatelná	6ES7 292 – 1AG20 – 0AA0
Repeater RS-485 IP 20, napětově oddělený	6ES7 972 – 0AA00 – 0XA0

Operátorská rozhraní	Objednací číslo
Textový displej TD 200	6ES7 272-0AA20-0YA0
Operátorské rozhraní OP3	6AV3 503-1DB10T
Operátorské rozhraní OP7	6AV3 607-1JC20-0AX1
Operátorské rozhraní OP17	6AV3 617-1JC20-0AX1
Dotykový panel TP070	6AV6 545-0AA15-2AX0
Dotykový panel TP170A	6AV6 545-0BA15-2AX0
Různé	Objednací číslo
Zarážky DIN lišty	6ES5 728-8MAI1
12místný blok svorek (CPU 221, CPU 222) sada - 10 kusů	6ES7 290-2AA00-0XA0
Sada náhradních dvířek, obsahuje po 4 ks od následujících: kryty svorkovnice pro 7, 12, 14, 18, 2x12, 2x14 svorek; přístupová dvířka CPU, přístupová dvířka EM	6ES7 291-3AX20-0XA0
8místný simulátor vstupů	6ES7 274 1XF00-0XA0
14místný simulátor vstupů	6ES7 274 1XH00-0XA0
24místný simulátor vstupů	6ES7 274 1XK00-0XA0

Doby provádění instrukcí STL



Doby provádění instrukcí jsou velmi důležité, pokud aplikace obsahuje časově kritické funkce. Doby provádění instrukcí jsou uvedeny v tabulce F-3.



Tip

Když použijete doby provádění z tabulky F-3, měli byste vzít v úvahu vliv signálového toku na instrukci, vliv nepřímé adresace a použití určitých paměťových oblastí na dobu provádění. Uvedené faktory mohou přímo ovlivnit doby provádění obsažené v tabulce.

Vliv signálového toku

Tabulka F-3 uvádí dobu nutnou pro provedení instrukce, když je signálový tok přítomen (vrchol zásobníku = 1 nebo ON).

Jestliže signálový tok přítomen není, je doba provádění dané instrukce 3 μ s.

Vliv nepřímé adresace

Tabulka F-3 uvádí dobu potřebnou pro provedení instrukce při použití přímé adresace operandů a konstant.

Jestliže instrukce použijí nepřímo adresované operandy, prodlouží se doba provádění instrukce o 22 μ s na každý nepřímo adresovaný operand použitý v instrukci.

Vliv přístupu k některým paměťovým oblastem

Přístup k některým paměťovým oblastem, jako jsou například AI, AQ, L a akumulátory, vyžaduje delší dobu provádění.

Tabulka F-1 uvádí dodatečný čas, který je třeba připočítat k době provádění instrukce, jestliže jsou tyto paměťové oblasti specifikovány v operandu.

Tabulka F-1 Čas potřebný pro přístup k paměťovým oblastem

Paměťová oblast	Čas přičítaný k době provádění
Analogové vstupy (AI)	
Bez povolení analogové filtrace:	149 μ s
S povolenou analogovou filtrací:	0 μ s
Analogové výstupy (AQ)	73 μ s
Lokální paměť (L)	5.4 μ s
Akumulátory (AC)	4.4 μ s

Vliv použití některých instrukcí CPU 226XM

Provádění některých instrukcí větvení u CPU 226XM vyžaduje prodloužení doby provádění.

Tabulka F-2 uvádí hodnotu, kterou je třeba přičíst k základní době provádění pro každou uvedenou instrukci.

Tabulka F-2 Čas přičítaný pro instrukce CPU 226XM

Instrukce	Čas přičítaný k době provádění
ATCH	1.0 μ s
CALL	4.3 μ s
CSCRE	3.1 μ s
FOR (čas přičítaný k základnímu)	3.1 μ s
(čas přičítaný k násobiteli smyčky)	3.1 μ s
INT	1.7 μ s
JMP	3.1 μ s
RET	2.8 μ s

Tabulka F-3 Doby provádění instrukcí

Instrukce	μs	Instrukce	μs
= S použitím: I SM, T, C, V, S, Q, M L	0.37 1.8 19.2	ATT	70
+D	55	AW < =, =, >=, >, <, <>	45
-D	55	BCDI	66
*D	92	BIR S použitím: Lokálních vstupů Rozšiřovacích vstupů	45 53
/D	376	BIW S použitím: Lokálních výstupů Rozšiřovacích výstupů	46 56
+I	46	BMB Doba = Základní + (délka * LM) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky (LM)	21 51 11
-I	47	BMD Doba = Základní + (délka * LM) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky (LM)	21 51 20
*I	71	BMW Doba = Základní + (délka * LM) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky (LM)	21 51 16
/I	115	BTI	27
=I S použitím: Lokálních výstupů Rozšiřovacích výstupů	29 39	CALL Bez použití parametrů: S použitím parametrů: Doba = Základní + Σ (doba operandu) Základní Doba operandu bit (vstupní, výstupní) byte (vstupní, výstupní) word (vstupní, výstupní) double word (vstupní, výstupní)	15 32 23, 21 21, 14 24, 18 27, 20
+R	110 max. 163	Poznámka: zpracování výstupních operandů probíhá během návratu z podprogramu	
-R	113 max. 166	CFND Max. doba = Základní + N1 * ((LM1 * N2) + LM2) Základní Násobitel délky 1 (LM1) Násobitel délky 2 (LM2) N1 je délka zdrojového řetězce N2 je délka řetězce souboru znaků	79 79 9.2 4.4
*R	100 max. 130	COS	1525 max. 1800
/R	300 max. 360	CRET	13
A S použitím: I SM, T, C, V, S, Q, M L	0.37 1.1 10.8	CRETI	23
AB < =, =, >=, >, <, <>	35	CSCRE	0.9
AD < =, =, >=, >, <, <>	53	CTD Při přechodu na vstupu pro čítání Jinak	48 36
AENO	.6	CTU Při přechodu na vstupu pro čítání Jinak	53 35
AI S použitím: Lokálních vstupů Rozšiřovacích vstupů	27 35	CTUD Při přechodu na vstupu pro čítání Jinak	64 45
ALD	0.37	DECB	30
AN S použitím: I SM, T, C, V, S, Q, M L	0.37 1.1 10.8	DECD	42
ANDB	37	DECO	36
ANDD	55		
ANDW	48		
ANI S použitím: Lokálních vstupů Rozšiřovacích vstupů	27 35		
AR < =, =, >=, >, <, <>	54		
AS=, <> Doba = Základní + (LM * N) Základní Násobitel délky (LM) N je počet porovn. znaků	51 9.2		
ATCH	20		
ATH Doba = Základní + (délka * LM) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky (LM)	41 55 20		

Instrukce	μs	Instrukce	μs
DECW	37	ITS	260
DISI	18	JMP	0.9
DIV	119	LBL	0.37
DTA	540	LD S použitím: I, SM0.0 SM, T, C, V, S, Q, M L	0.37 1.1 10.9
DTI	36	LDB <=, =, >=, >, <, <>	35
DTCH	18	LDD <=, =, >=, >, <, <>	52
DTR	60 max. 70	LDI S použitím: Lokálních vstupů Rozšiřovacích vstupů	26 34
DTS	540	LDN S použitím: I SM, T, C, V, S, Q, M L	0.37 1.1 10.9
ED	15	LDNI S použitím: Lokálních vstupů Rozšiřovacích vstupů	26 34
ENCO	39 max. 43	LDR <=, =, >=, >, <, <>	55
END	0.9	LDS	0.37
ENI	53	LDS=, <> Doba = Základní + (LM * N) Základní Násobitel délky (LM) N je počet porovn. znaků	51 9.2
EU	15	LDW <=, =, >=, >, <, <>	42
EXP	1170 max. 1375	LIFO	70
FIFO Doba = Základní + (délka * LM) Základní Násobitel délky (LM)	70 14	LN	1130 max. 1275
FILL Doba = Základní + (délka * LM) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky (LM)	29 50 7	LPP	0.37
FND <, =, >, <> Doba = Základní + (dél. * LM) Základní Násobitel délky (LM)	85 12	LPS	0.37
FOR Doba = Zákl. + (Počet smyček * LM) Základní Násobitel smyčky (LM)	64 50	LRD	0.37
GPA	31	LSCR	12
HDEF	35	MEND	0.5
HSC	37	MOVB	29
HTA Doba = Základní + (délka * LM) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky (LM)	38 48 11	MOVD	38
IBCD	114	MOVR	38
INCB	29	MOVW	34
INCD	42	MUL	70
INCW	37	NEXT	0
INT Obvykle s 1 přerušením	47	NETR	179
INVB	31	NETW Doba = Základní + (délka * LM) Základní Násobitel délky (LM)	175 8
INVD	42	NOP	0.37
INVW	38	NOT	0.37
ITA	260	O S použitím: I SM, T, C, V, S, Q, M L	0.37 1.1 10.8
ITB	27	OB <=, =, >=, >, <, <>	35
ITD	36	OD <=, =, >=, >, <, <>	53

Instrukce			μs	Instrukce			μs
OI	S použitím:	Lokálních vstupů Rozšiřovacích vstupů	27 35	ROUND			108 max. 183
OLD			0.37	RRB	Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)		42 0.6
ON	S použitím:	I SM, T, C, V, S, Q, M L	0.37 1.1 10.8	RRD	Celkem = Základní + (délka * LM) Základní Násobitel délky (LM))		52 2.5
ONI	S použitím:	Lokálních vstupů Rozšiřovacích vstupů	27 35	RRW	Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)		49 1.7
OR<=, =, >=, >, <, <>			55	RTA	Doba = Základní + (LM * N) Základní (pro první číslici ve výsledku) Násobitel délky (LM) N je počet dalších číslic ve výsledku		1000 240
ORB			37	RTS	Doba = Základní + (LM * N) Základní (pro první číslici ve výsledku) Násobitel délky (LM) N je počet dalších číslic ve výsledku		1000 240
ORD			55	S	Pro délku = 1 a specifikovanou jako konstanta Jinak: Doba = Základní + (délka * LM) Základní Násobitel délky (LM) Pokud je délka uložena jako proměnná, přičtete k základní době		5 27 0.9 29
ORW			48	SBR			0
OS=, < >	Doba + Základní + (LM * N) Základní Násobitel délky (LM) N je počet porovnávaných znaků		51 9.2	SCAT	Doba = Základní + (LM * N) Základní Násobitel délky (LM) N je počet připojených znaků		55 8.8
OW <=, =, >=, >, <, <>			45	SCPY	Doba = Základní + (LM * N) Základní Násobitel délky (LM) N je počet kopírovaných znaků		43 8.8
PID	Základní Dodatečný čas přičítaný pro přepočet integrační a derivační časové konst.		750 1000	SCRE			0.37
PLS:	S použitím:	PWM PTO v režimu s jedním segmentem PTO v režimu s více segmenty	57 67 92	SCRT			17
R	Délka = 1 a specifikována jako konst. S použitím operandu = C, T S použitím všech ostatních operandů Jinak, Čas = Základní + (délka * LM) Základní pro operand = C, T Základní pro všechny ostatní operandy LM pro operand = C, T LM pro všechny ostatní operandy Je-li délka uložena jako proměnná, přičtete k základní době		17, 24 5 19, 19 28 8.6, 16.5 0.9 29	SEG			30
RCV			80	SFND	Max. doba = Základní + (N1 - N2) * ((LM1 * N2) + LM2) Základní Násobitel délky 1 (LM1) Násobitel délky 2 (LM2) N1 je délka zdrojového řetězce N2 je délka vyhledávaného řetězce		79 11.5 17.8
RET			13	SHRB	Celkem = Základní + (délka * LM1) + ((délka / 8) * LM2) Základní (konstantní délka) Základní (proměnná délka) Násobitel délky 1 (LM1) Násobitel délky 2 (LM2)		76 84 1.6 4
RETI			23				
RI	Celkem = Základní + (délka * LM) Základní LM s použitím lokálních výstupů LM s použitím rozšiřovacích výstupů Pokud je délka uložena jako proměnná, přičtete k základní době		18 22 32 30				
RLB	Celkem = Základní + (délka + LM) Základní Násobitel délky (LM)		42 0.6				
RLD	Celkem = Základní + (délka * LM) Základní Násobitel délky (LM))		52 2.5				
RLW	Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)		49 1.7				

Instrukce	μs	Instrukce	μs
SI Celkem = Základní + (délka * LM) Základní LM s použitím lokálního výstupu LM s použitím rozšiřovacího výstupu Pokud je délka uložena jako proměnná, přičtete k základní době	18 22 32 30	STD Doba = Základní + (LM*N) Základní (pro 1. zdrojový znak) Násobitel délky (LM) N je počet dalších zdrojových znaků	84 59
SIN	1525 max. 1800	STI Doba = Základní + (LM*N) Základní (pro 1. zdrojový znak) Násobitel délky (LM) N je počet dalších zdrojových znaků	84 59
SLB Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)	43 0.7	STOP	16
SLD Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)	53 2.6	STR Doba = Základní + (LM*N) Základní (pro 1. zdrojový znak) Násobitel délky (LM) N je počet dalších zdrojových znaků	100 120
SLEN	46	SWAP	32
SLW Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)	51 1.3	TAN	1825 max. 2100
SPA	243	TODR	2400
SQRT	725 max. 830	TODW	1600
SRB Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)	43 0.7	TOF	64
SRD Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)	53 2.6	TON	64
SRW Celkem = Základní + (délka * LM) Základní Násobitel délky (LM)	51 1.3	TONR	56
SSCPY Doba = Základní + (LM * N) Základní Násobitel délky (LM) N je počet kopírovaných znaků	82 8.8	TRUNC	103 max. 178
		WDR	16
		XMT	78
		XORB	37
		XORD	55
		XORW	48

Stručný přehled informací o S7-200



Pro usnadnění vyhledávání informací uvádí tato kapitola stručný přehled následujícího:

- ☐ Speciální paměťové bity
- ☐ Popisy událostí přerušení
- ☐ Stručný přehled rozsahů a vlastností paměti CPU S7-200
- ☐ Vysokorychlostní čítače HSC0, HSC1, HSC2, HSC3, HSC4, HSC5
- ☐ Instrukce pro S7-200

Tabulka G-1 Speciální paměťové bity

Speciální paměťové bity			
SM0.0	Vždy zapnutý	SM1.0	Výsledek operace = 0
SM0.1	První programový cyklus	SM1.1	Přeplnění nebo nepřípustná hodnota
SM0.2	Remanentní data ztracena	SM1.2	Záporný výsledek
SM0.3	Studený start	SM1.3	Dělení nulou
SM0.4	30 s vypnutý / 30 s zapnutý	SM1.4	Plná tabulka
SM0.5	0,5 s vypnutý / 0,5 s zapnutý	SM1.5	Prázdná tabulka
SM0.6	Vypnutý na 1 programový cyklus / zapnutý na 1 programový cyklus	SM1.6	Chyba převodu BCD na binární číslo
SM0.7	Přepínač v poloze RUN	SM1.7	Chyba převodu ASCII na hexadecimální číslo

Tabulka G-2 Události přerušení seřazené podle priority

Číslo události	Popis přerušení	Prioritní skupina	Priorita ve skupině
8	Port 0: Příjem znaku	Komunikace (nejvyšší priorita)	0
9	Port 0: Ukončení přenosu		0
23	Port 0: Ukončení příjmu zprávy		0
24	Port 1: Ukončení příjmu zprávy		1
25	Port 1: Příjem znaku		1
26	Port 1: Ukončení přenosu		1
19	Přerušení od ukončení PTO 0	Diskrétní (střední priorita)	0
20	Přerušení od ukončení PTO 1		1
0	I0.0, Náběžná hrana		2
2	I0.1, Náběžná hrana		3
4	I0.2, Náběžná hrana		4
6	I0.3, Náběžná hrana		5
1	I0.0, Sestupná hrana		6
3	I0.1, Sestupná hrana		7
5	I0.2, Sestupná hrana		8
7	I0.3, Sestupná hrana		9
12	HSC0 CV=PV (aktuální hodnota = nastavená hodnota)		10
27	HSC0 změna směru		11
28	HSC0 externí reset		12
13	HSC1 CV=PV (aktuální hodnota = nastavená hodnota)		13
14	HSC1 změna směru na vstupu		14
15	HSC1 externí reset		15
16	HSC2 CV=PV		16
17	HSC2 změna směru		17
18	HSC2 externí reset		18
32	HSC3 CV=PV (aktuální hodnota = nastavená hodnota)		19
29	HSC4 CV=PV (aktuální hodnota = nastavená hodnota)		20
30	HSC4 změna směru		21
31	HSC4 externí reset		22
33	HSC5 CV=PV (aktuální hodnota = nastavená hodnota)		23
10	Cyklické přerušení 0	Časovaná (nejnižší priorita)	0
11	Cyklické přerušení 1		1
21	Přerušení od časovače T32 CT=PT		2
22	Přerušení od časovače T96 CT=PT		3

Tabulka G-3 Stručný přehled rozsahu a vlastností paměti CPU S7-200

Popis	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Velikost uživatelského programu	2 Kword	2 Kword	4 Kword	4 Kword	8 Kword
Velikost uživatelských dat	1 Kword	1 Kword	2,5 Kword	2,5 Kword	5 Kword
Registr obrazu vstupů	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7
Registr obrazu výstupů	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7
Analogové vstupy (pouze čtení)	--	AIW0 až AIW30	AIW0 až AIW62	AIW0 až AIW62	AIW0 až AIW62
Analogové výstupy (pouze zápis)	--	AQW0 až AQW30	AQW0 až AQW62	AQW0 až AQW62	AQW0 až AQW62
Paměť pro proměnné (V)	VB0 až VB2047	VB0 až VB2047	VB0 až VB5119	VB0 až VB5119	VB0 až VB10239
Lokální paměť (L) ¹	LB0 až LB63	LB0 až LB63	LB0 až LB63	LB0 až LB63	LB0 až LB63
Bitová paměť (M)	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7
Speciální paměť (SM) Pouze čtení	SM0.0 až SM179.7 SM0.0 až SM29.7	SM0.0 až SM299.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7
Časovače	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)
Remanentní zpožděné zapnutí	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
1 ms	T1 až T4, a	T1 až T4, a	T1 až T4, a	T1 až T4, a	T1 až T4, a
10 ms	T65 až T68	T65 až T68	T65 až T68	T65 až T68	T65 až T68
100 ms	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95
Zpožděné zapnutí/vypnutí	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
1 ms	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100
10 ms	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255
100 ms					
Čítače	C0 až C255	C0 až C255	C0 až C255	C0 až C255	C0 až C255
Vysokorychlostní čítač	HC0, HC3, HC4 a HC5	HC0, HC3, HC4 a HC5	HC0 až HC5	HC0 až HC5	HC0 až HC5
Sekvenční řídicí relé (S)	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7
Registry akumulátoru	AC0 až AC3	AC0 až AC3	AC0 až AC3	AC0 až AC3	AC0 až AC3
Skoky/návěsti	0 až 255	0 až 255	0 až 255	0 až 255	0 až 255
Volání/podprogramy	0 až 63	0 až 63	0 až 63	0 až 63	0 až 127
Podprogramy přerušení	0 až 127	0 až 127	0 až 127	0 až 127	0 až 127
Náběžné/sestupné hrany	256	256	256	256	256
Smyčky PID	0 až 7	0 až 7	0 až 7	0 až 7	0 až 7
Porty	Port 0	Port 0	Port 0	Port 0, Port 1	Port 0, Port 1

¹ LB60 až LB63 jsou rezervovány pro STEP 7–Micro/WIN, verze 3.0 nebo pozdější.

Tabulka G-4 Vysokorychlostní čítače HSC0, HSC3, HSC4 a HSC5

Režim	HSC0			HSC3	HSC4			HSC5
	I0.0	I0.1	I0.2	I0.1	I0.3	I0.4	I0.5	I0.4
0	Pulzy			Pulzy	Pulzy			Pulzy
1	Pulzy		Nulování		Pulzy		Nulování	
2								
3	Pulzy	Směr			Pulzy	Směr		
4	Pulzy	Směr	Nulování		Pulzy	Směr	Nulování	
5								
6	Pulzy nahoru	Pulzy dolů			Pulzy nahoru	Pulzy dolů		
7	Pulzy nahoru	Pulzy dolů	Nulování		Pulzy nahoru	Pulzy dolů	Nulování	
8								
9	Fáze A	Fáze B			Fáze A	Fáze B		
10	Fáze A	Fáze B	Nulování		Fáze A	Fáze B	Nulování	
11								

Tabulka G-5 Vysokorychlostní čítače HSC1 a HSC2

Režim	HSC1				HSC2			
	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
0	Pulzy				Pulzy			
1	Pulzy		Nulování		Pulzy		Nulování	
2	Pulzy		Nulování	Start	Pulzy		Nulování	Start
3	Pulzy	Směr			Pulzy	Směr		
4	Pulzy	Směr	Nulování		Pulzy	Směr	Nulování	
5	Pulzy	Směr	Nulování	Start	Pulzy	Směr	Nulování	Start
6	Pulzy nahoru	Pulzy dolů			Pulzy nahoru	Pulzy dolů		
7	Pulzy nahoru	Pulzy dolů	Nulování		Pulzy nahoru	Pulzy dolů	Nulování	
8	Pulzy nahoru	Pulzy dolů	Nulování	Start	Pulzy nahoru	Pulzy dolů	Nulování	Start
9	Fáze A	Fáze B			Fáze A	Fáze B		
10	Fáze A	Fáze B	Nulování		Fáze A	Fáze B	Nulování	
11	Fáze A	Fáze B	Nulování	Start	Fáze A	Fáze B	Nulování	Start

Booleovské instrukce			Instrukce matematiky, inkrementace a dekrementace		
LD	bit	Uloží hodnotu na vrchol zásobníku	+I	IN1, OUT	Sečtení integer, double integer nebo reálného čísla
LDI	bit	Uloží okamžitou hodnotu	+D	IN1, OUT	IN1 + OUT = OUT
LDN	bit	Uloží negovanou hodnotu	–I	IN1, OUT	Odečtení integer, double integer nebo reálného čísla
LDNI	bit	Uloží negovanou okamžitou hodnotu	–D	IN1, OUT	OUT – IN1 = OUT
A	bit	AND (logický součin)	–R	IN1, OUT	Vynásobení integer (16*16 – >32)
AI	bit	AND s okamžitou hodnotou	*I	IN1, OUT	Vynásobení integer, double integer nebo reálného čísla
AN	bit	AND s negovanou hodnotou	*D	IN1, OUT	IN1 * OUT = OUT
ANI	bit	AND s negovanou okamžitou hodnotou	*R	IN1, IN2	Dělení integer (16/16 – >32)
O	bit	OR (logický součet)	DIV	IN1, OUT	Dělení integer, double integer nebo reálného čísla
OI	bit	OR s okamžitou hodnotou	/I	IN1, OUT	OUT / IN1 = OUT
ON	bit	OR s negovanou hodnotou	/D,	IN1, OUT	Druhá odmocnina
ONI	bit	OR s negovanou okamžitou hodnotou	/R	IN1, OUT	Přirozený logaritmus
LDBx	IN1, IN2	Uloží výsledek porovnání bytů IN1 (x: <, <=, =, >=, >, <>) IN2	EXP	IN, OUT	Přirozený exponenciál
ABx	IN1, IN2	AND výsledek porovnání bytů IN1 (x: <, <=, =, >=, >, <>) IN2	SIN	IN, OUT	Sinus
OBx	IN1, IN2	OR výsledek porovnání bytů IN1 (x: <, <=, =, >=, >, <>) IN2	COS	IN, OUT	Kosinus
LDWx	IN1, IN2	Uloží výsledek porovnání wordů IN1 (x: <, <=, =, >=, >, <>) IN2	TAN	IN, OUT	Tangens
AWx	IN1, IN2	AND výsledek porovnání wordů IN1 (x: <, <=, =, >=, >, <>) IN2	INCB	OUT	Inkrementace bytu, word nebo double word
OWx	IN1, IN2	OR výsledek porovnání wordů IN1 (x: <, <=, =, >=, >, <>) IN2	INCW	OUT	Dekrementace bytu, word nebo double word
LDDx	IN1, IN2	Uloží výsledek porovnání double word IN1 (x: <, <=, =, >=, >, <>) IN2	INCD	OUT	Smyčka PID
ADx	IN1, IN2	AND výsledek porovnání double word IN1 (x: <, <=, =, >=, >, <>) IN2	DECB	OUT	
ODx	IN1, IN2	OR výsledek porovnání double word IN1 (x: <, <=, =, >=, >, <>) IN2	DECW	OUT	
LDRx	IN1, IN2	Uloží výsledek porovnání reálných čísel IN1 (x: <, <=, =, >=, >, <>) IN2	DECD	OUT	
ARx	IN1, IN2	AND výsledek porovnání reálných čísel IN1 (x: <, <=, =, >=, >, <>) IN2	PID	TBL, LOOP	
ORx	IN1, IN2	OR výsledek porovnání reálných čísel IN1 (x: <, <=, =, >=, >, <>) IN2	Instrukce časovače a čítače		
NOT		Negace zásobníku	TON	Txxx, PT	Časovač zpožděného zapnutí
EU		Detekce náběžné hrany	TOF	Txxx, PT	Časovač zpožděného vypnutí
ED		Detekce sestupné hrany	TONR	Txxx, PT	Remanentní časovač zpožděného zapnutí
=	Bit	Přiřazení hodnoty	CTU	Cxxx, PV	Čítání nahoru
=I	Bit	Okamžité přiřazení hodnoty	CTD	Cxxx, PV	Čítání dolů
S	Bit, N	Set (nastavení bitu)	CTUD	Cxxx, PV	Čítání nahoru/dolů
R	Bit, N	Reset (nulování bitu)	Instrukce hodin reálného času		
SI	Bit, N	Okamžité nastavení bitu	TODR	T	Přečtení času z hodin
RI	Bit, N	Okamžité nulování bitu	TODW	T	Zápis času do hodin
LDSx	IN1, IN2	Uloží výsledek porovnání řetězce IN1 (x: =, <>) IN2	Instrukce řízení programu		
ASx	IN1, IN2	AND výsledek porovnání řetězce IN1 (x: =, <>) IN2	END		Podmíněné ukončení programu
OSx	IN1, IN2	OR výsledek porovnání řetězce IN1 (x: =, <>) IN2	STOP		Přechod do režimu STOP
ALD		Logický součin s vrcholem zásobníku	WDR		Reset watchdogu (300 ms)
OLD		Logický součet s vrcholem zásobníku	JMP	N	Skok na definované návěští
LPS		Duplikace horní hodnoty zásobníku	LBL	N	Definuje návěští pro skok
LRD		Zkopírování druhé hodnoty zásobníku	CALL	N [N1,...]	Volání podprogramu [N1, ... až do 16 volitelných parametrů]
LPP		Vybrání hodnoty na vrcholu zásobníku	CRET		Podmíněný návrat z podprogramu
LDS	N	Duplikace bitu n zásobníku	FOR	INDX,INIT,FINAL	Smyčka For/Next
AENO		Logický součin s hodnotou ENO	NEXT		
			LSCR	N	
			SCRT	N	
			CSCRE		Uložení, přechod, podmíněné ukončení a ukončení sekvenčního řídicího relé
			SCRE		

Instrukce přesunu, posunu a rotace		
MOVB	IN, OUT	Přesun hodnot typu byte, word, double word, reálné číslo
MOVW	IN, OUT	
MOVD	IN, OUT	
MOVR	IN, OUT	
BIR	IN, OUT	Přesun bytu s okamžitým čtením
BIW	IN, OUT	
BMB	IN, OUT, N	Přesun bloku hodnot typu byte, word, double word
BMW	IN, OUT, N	
BMD	IN, OUT, N	
SWAP	IN	
SHRB	DATA, S_BIT, N	Záměna vyššího a nižšího bytu
SRB	OUT, N	Posun bitů posuvného registru
SRW	OUT, N	
SRD	OUT, N	
SLB	OUT, N	
SLW	OUT, N	Posun bytu, word, double word doleva
SLD	OUT, N	
RRB	OUT, N	
RRW	OUT, N	
RRD	OUT, N	Rotace bytu, word, double word doprava
RLB	OUT, N	
RLW	OUT, N	
RLD	OUT, N	
Logické instrukce		
ANDB	IN1, OUT	Logický součin (AND) hodnot typu byte, word a double word
ANDW	IN1, OUT	
ANDD	IN1, OUT	
ORB	IN1, OUT	Logický součet (OR) hodnot typu byte, word a double word
ORW	IN1, OUT	
ORD	IN1, OUT	
XORB	IN1, OUT	
XORW	IN1, OUT	Logická nonekvivalence (XOR) hodnot typu byte, word a double word
XORD	IN1, OUT	
INVB	OUT	Inverze byte, word a double word (jednotkový doplněk)
INWV	OUT	
INVD	OUT	
Instrukce pro řetězce		
SLEN	IN, OUT	Délka řetězce
SCAT	IN, OUT	Spojování řetězců
SCPY	IN, OUT	Kopírování řetězce
SSCPY	IN, INDX, N, OUT	Kopírování části řetězce
CFND	IN1, IN2, OUT	Nalezení prvního znaku uvnitř řetězce
SFND	IN1, IN2, OUT	Nalezení části řetězce

Instrukce tabulky, vyhledávání a konverze		
ATT	DATA, TBL	Přidání dat do tabulky
LIFO	TBL, DATA	Výběr dat z tabulky
FIFO	TBL, DATA	
FND=	TBL, PTN, INDX	Nalezení datové hodnoty, která odpovídá srovnání (v tabulce)
FND<>	TBL, PTN, INDX	
FND<	TBL, PTN, INDX	
FND>	TBL, PTN, INDX	
FILL	IN, OUT, N	Vyplnění místa v paměti
BCDI	OUT	Konverze BCD na integer
IBCD	OUT	Konverze integer na BCD
BTI	IN, OUT	Konverze byte na integer
ITB	IN, OUT	Konverze integer na byte
ITD	IN, OUT	Konverze integer na double integer
DTI	IN, OUT	Konverze double integer na integer
DTR	IN, OUT	Konverze double word na reálné číslo
TRUNC	IN, OUT	Konverze real čísla na double integer
ROUND	IN, OUT	Konverze real čísla na double integer
ATH	IN, OUT, LEN	Konverze ASCII na HEX hodnotu
HTA	IN, OUT, LEN	Konverze HEX hodnoty na ASCII
ITA	IN, OUT, FMT	Konverze integer na ASCII
DTA	IN, OUT, FM	Konverze double integer na ASCII
RTA	IN, OUT, FM	Konverze reálného čísla na ASCII
DECO	IN, OUT	Dekódování
ENCO	IN, OUT	Kódování
SEG	IN, OUT	Generuje hodnotu pro sedmisegmentový displej
ITS	IN, FMT, OUT	Konverze integer na řetězec
DTS	IN, FMT, OUT	Konverze double integer na řetězec
RTS	IN, FMT, OUT	Konverze reálného čísla na řetězec
STI	STR, INDX, OUT	Konverze části řetězce na integer
STD	STR, INDX, OUT	Konverze části řetězce na double integer
STR	STR, INDX, OUT	Konverze části řetězce na reálné číslo
Instrukce přerušení		
CRETI		Podmíněný návrat z přerušení
ENI		Povolení přerušení
DISI		Zákaz přerušení
ATCH	INT, EVNT	Přiřazení podprogramu přerušení události
DTCH	EVNT	Zrušení přiřazení přerušení
Instrukce komunikace		
XMT	TBL, PORT	Přenos v režimu Freeport
RCV	TBL, PORT	Příjem zprávy v režimu Freeport
NETR	TBL, PORT	Čtení ze sítě
NETW	TBL, PORT	Zápis do sítě
GPA	ADDR, PORT	Načtení adresy portu
SPA	ADDR, PORT	Nastavení adresy portu
Vysokorychlostní instrukce		
HDEF	HSC, MODE	Definuje režim vysokorychlostního čítače
HSC	N	Aktivace vysokorychlostního čítače
PLS	Q	Pulzní výstup

Rejstřík

Symboly

&, 32
*, 32

A

ACCEL_TIME (Doba zrychlení), polohovací modul EM 253, 249

Adresa

Modbus, 331
nastavení vzdálené, 210
přiřazení, 58
síťová, 208
symbolická, 58
vysokorychlostní čítače, 119

Adresa nejvyšší stanice (HSA), 228

Adresování

akumulátory, 27
analogové vstupy, 29
analogové výstupy, 29
bitová paměť, 25
bity speciální paměti (SM), 28
byte:bit, 24
lokální paměť, 28
lokální vstupy/výstupy, 31
nepřímé (ukazatel), 32
paměť časovače, 26
paměť čítače, 26
paměť pro proměnné, 25
paměť S7–200, 24
paměť sekvenčních řídicích relé (SCR), 29
paměťové oblasti, 25–28
přímé, 24
rozšiřitelné vstupy/výstupy, 31
registr obrazu vstupů, 25
registr obrazu výstupů, 25
vysokorychlostní čítače, 27

AENO, instrukce, 70

Aktivní úroveň vstupu, polohovací modul EM 253, 247

Aktualizace, vlny PWM, 127

Aktuální hodnota

nastavení pro HSC, 118
změna v HSC, 123

Akumulátory, 27

AI. Viz Analogový vstup (AI)

American Bureau of Shipping (ABS), 338

Analogové moduly, 3

EM 231 analogový vstup, 354
EM 231 RTD, 361
EM 231 termočlánek, 361
EM 232 analogové výstupy, 358
EM 235 analogové vstupy/výstupy, 355

Analogové potenciometry

potenciometry, 45
SMB28 a SMB29, 412

Analogový I/O modul EM 235

blokové schéma vstupů, 357
blokové schéma výstupů, 358
formát vstupního datového word, 356
formát výstupního datového word, 357
instalace, 358
kalibrace, 353
konfigurace, 355
přesnost a opakovatelnost, 359
specifikace, 360

Analogový vstup (AI)

adresování, 29
filtrace, 42

Analogový vstupní modul EM 231

blokové schéma vstupů, 356
formát vstupního datového word, 356
instalace, 358
kalibrace, 353
konfigurace, 354
přesnost a opakovatelnost, 359
specifikace, 360

Analogový výstup (AQ), adresování, 29

Analogový výstupní modul EM 232

blokové schéma výstupů, 358
formát výstupního datového word, 357

AND, instrukce, 163

příklad, 164

AQ. Viz Analogový výstup (AQ)

ASCII na Hex, instrukce, 96

příklad, 99

Asynchronní aktualizace, instrukce PWM, 127

Automatický režim, smyčka PID, 151

B

Bateriový zásuvný modul, 34, 395

objednací číslo, 422

BCD na Integer, instrukce, 93

Běh, programu, 11

Bezpečnostní obvody, navrhování, 48

Blokové schéma, modem EM 241, 385

Blokové schéma vstupů

EM 231, 356
EM 235, 357

Blokové schéma výstupů

EM 232, 358
EM 235, 358

Booleovské instrukce

cívky, 68
kontakty, 66
logický zásobník, 70
RS a SR klopný obvod, 72

Bureau Veritas (BV), 338

Byte na Integer, instrukce, 93

Bytová konzistence, PROFIBUS, 378
Bytový adresní formát, 24

C

Control panel pro EM 253, 274–276

CP 243–2, připojení, 394

CP5411, 422

CP5511, 422

CP5611, 422

CP karty

objednací čísla, 422

volba, 220

CPU

digitální vstup/výstup, 2

hodiny reálného času, 2

ID registr (SMB6), 410

komunikační porty, 2

ochrana heslem, 44

ošetření chyb, 56

paměť, 2

požadavky na napájení, 15, 399

programový cyklus, 22

prováděcí rychlost, 2

rozměry, 2

rozšiřovací moduly, 2

simulátory vstupu, 398

velikost šroubů pro instalaci, 16

zálohování, 2

CPU, podporující inteligentní moduly, 303

CPU 224, vzorový program DP, 383

CPU moduly

autodiagnostika, 23

demontáž, 17

instalace, 16

objednací čísla, 340, 421

rozsahy paměti, 64, 433, 461

schémata zapojení, 343–345

specifikace, 340

specifikace napájení, 341

specifikace vstupů, 341

specifikace výstupů, 342

vlastnosti, 64, 461

Cyklická přerušení, registry časového intervalu (SMB34, SMB35), 413

Č

Čas, nastavení, 73

Čas cyklu (funkce PTO), 125

Časovač měřící čas mezi znaky, 84

Časovač zpráv, 84

Časovač zpoždění náběžné hrany, instrukce (TON), 196, 201

příklad, 198

Časovač zpoždění sestupné hrany, instrukce (TOF), 196, 201

příklad, 199

Časovače zpoždění náběžné hrany (TON), 196, 201

Časovače zpoždění sestupné hrany (TOF), 196, 201

Časy vstupního filtru, polohovací modul EM 253, 247

Čísla, reprezentace, 24, 30

Číslo, reprezentace, 29

Čítač dolů, instrukce, 106

příklad, 108

Čítač nahoru, instrukce, 106

Čítač nahoru/dolů, instrukce, 106

příklad, 108

Čítače, vysokorychlostní, 46

Čtení dat ze vstupů, 22, 23

Čtení hodin reálného času, instrukce, 73

Čtení ze sítě, instrukce, 74, 75

chybové kódy, 75

příklad, 77

D

Data

příjem, 80, 85

uložení a obnovení, 34

vysílání, 80

Databázový soubor zařízení (GSD), EM 277

PROFIBUS – DP, 381–382

Datový blok, 50

Datum, nastavení, 73

DECEL_TIME (Doba zpomalení), polohovací modul EM 253, 249

Definice vysokorychlostního čítače, instrukce (HDEF), 111

Dekódovat, instrukce, 105

příklad, 105

Dekrementace, instrukce, 144

příklad, 144

Dělení, instrukce, 140

Dělení Integer se zbytkem, instrukce (DIV), 142

příklad, 142

Délka řetězce, instrukce, 184

Demontáž

modulu CPU, 17

paměťového modulu, 36

rozšiřovacího modulu, 17

svorkovnice, 17

Derivační člen, algoritmus PID, 148

Det Norske Veritas (DNV), 338

Diagnostické informace, polohovací modul EM 253, 275

Diagnostika, autodiagnostika CPU, 23

Dialogové okno “Informace PLC”, 56

Digitální rozšiřovací modul, adresace, 31

Digitální vstupy

čtení, 39

filtrace, 41

zachycení pulzu, 41

Digitální výstupy, konfigurace stavů, 40

DIP přepínače

RTD, 368–369

termočlánek, 364

Diskrétní moduly, 3

Doba oběhu tokenu, 228
 porovnání, 230
 Doba zaoblení, polohovací modul EM 253, 250
 Doby programového cyklu: SMW22 až SMW26, 412
 Doby provádění, instrukce STL, 425
 Double Integer na ASCII, instrukce, 97
 Double Integer na Integer, instrukce, 93
 Double Integer na Real, instrukce, 93
 Download, programu, 11, 34
 Dvoubodová (peer-to-peer) komunikace, 213–214

E

Editace, v režimu RUN, 236
 Editor datového bloku
 přiřazení adres, 58
 přiřazení počátečních hodnot, 58
 Editor FBD
 konvence, 54
 popis, 52
 vlastnosti, 52
 Editor LAD
 konvence, 54
 popis, 52
 vlastnosti, 52
 Editor STL
 popis, 51
 vlastnosti, 51
 Editory
 funkční bloky, FBD, 52
 kontaktní schémata, LAD, 52
 výpis příkazů, STL, 51
 EEPROM, 34
 chybové kódy, 404
 uložení bitové paměti (M), 35
 uložení proměnné paměti (V), 38, 413
 Elektrická životnost, 338
 Elektromagnetická
 kompatibilita, S7–200, 339
 normy pro emise, 339
 normy pro odolnost, 339
 EN, 55, 63
 ENO, 55, 63

F

Faktor obnovení mezery (GUF), 228
 FIFO, instrukce, 190
 příklad, 190
 Filtrace
 analogové vstupy, 42
 digitální vstupy, 41
 Formát telefonního čísla pro zprávu, modemový modul EM 24, 308
 Formát textové zprávy, modemový modul EM 241, 309
 Formát vstupního datového word
 EM 231, 356
 EM 235, 356

Formát výstupního datového word
 EM 232, 357
 EM 235, 357
 Formát zprávy s přenosem dat CPU, modemový modul EM 241, 310
 Fronta časovaných přerušení, 159
 Fronta přerušení od vstupů/výstupů, 159
 Fronty, podprogramy přerušení, 159
 Funkce paměti
 instrukce posunu, 179
 instrukce posunutí bitů posuvného registru, 181
 instrukce přesunu, 165
 instrukce přesunu bloku dat, 167
 instrukce rotace, 179
 instrukce výměny, 183
 Funkce, Modbus, 332
 Funkce PTO/PWM, registry (WMB66 až SMB85), 415
 Funkce zachycování pulzů, 42
 Funkční bloky Viz Editor FBD
 Fyzické vstupy, odezva polohovacího modulu EM 253, 248

G

Germanisher Lloyd (GL), 338

H

Hardware, odstraňování poruch, 241
 Hex na ASCII, instrukce, 96
 Hledání, instrukce, 193
 Hledání poruch
 hardware S7–200, 241
 chybové kódy, 404
 kritické chyby, 57
 návod, 241
 nekritické chyby, 56
 Hledání řetězce, instrukce, 187
 Hledání znaku, instrukce, 187
 Hodiny
 zásuvný modul, 395
 stavové bity, 408
 Hodiny, instrukce
 čtení hodin reálného času, 73
 nastavení hodin reálného času, 73
 Hodiny reálného času (TOD), 73
 Hodiny TOD, 73
 Hodnoty
 data S7–200, 24, 32
 přímá adresace, 24
 stavová tabulka, 59
 Hodnoty reálných čísel, 24, 29
 Hodnoty s plovoucí desetinnou čárkou, 29, 149
 Hodnoty tabulky profilu, generátory PTO/PWM, 130
 HSC, 46

CH

Chybové kódy, 404

- chyby polohovacího modulu EM 253, 277
- instrukce nonekvivalence (XOR), 163
- instrukce pro polohovací modul EM 253, 276
- instrukce USS protokolu, 323
- kritické chyby, 404
- polohovací modul EM 253, 276
- porušení pravidel kompilace, 406
- problémy programování při běhu programu, 405
- příklad, 164

Chybové kódy instrukcí, polohovací modul EM 253, 276

Chybové kódy modulu, polohovací modul EM 253, 277

Chybové kódy provádění, instrukce protokolu USS, 323

Chyby

- instrukce čtení a zápis v síti, 75
- instrukce modemového modulu EM 241, 301
- kompilace programu, 56
- konfigurace modemového modulu EM 241, 293
- kritické, 57
- nekritické, 56
- ošetření, 56
- prohlížení, 56
- provádění programu, 57
- provádění protokolu Modbus slave, 335
- SMB1, chyby provádění, 408
- smyčka PID, 151
- v době běhu programu, 57
- vstupy a výstupy (I/O), 57

Chyby kompilace, 56

Chyby parity

- SMB3, 409
- SMB30 a SMB130, 85

Chyby provádění, 57

Chyby v době běhu programu, 57

I

Identifikační a chybový registr I/O modulu, 411

IEC instrukce časovače, 201

- příklad, 201

IEC instrukce čítače

- čítač dolů, 109
- čítač nahoru, 109
- čítač nahoru/dolů, 109
- příklad, 109

Inicializace

- Protokol Modbus, 330
- PTO pro režim s jedním segmentem, 134
- PTO pro režim s více segmenty, 135
- vysokorychlostní čítače, 120
- výstup PWM, 132

Inkrementace, instrukce, 144

- příklad, 144

Instalace

- elektrické rušení, 14
- EM 231, 358
- EM 235, 358
- kabel pro rozšiřovací moduly, 395

modul CPU, 16

- napájení, 15S7–200, 15
- nároky na prostor, 14
- návody, 14
- paměťový zásuvný modul, 36
- požadavky na montáž, 16
- rozšiřovací modul, 16
- STEP 7–Micro/WIN, 4
- vysokonapěťová zařízení, 14
- zařízení vytvářející teplo, 14

Instrukce

- AENO, 70
- AND, 163
- AND Load, 70
- ASCII na Hex, 96
- BCD na Integer, 93
- bitové logiky, 66
- časovače zpoždění náběž. hrany s pam. (TONR), 196
- čítače dolů, 106
- čítače nahoru, 106
- čítače nahoru/dolů, 106
- čtení hodin reálného času, 73
- čtení zásobníku, 70
- čtení ze sítě, 74
- definice vysokorychlostního čítače (HDEF), 111
- dekódování, 105
- dekrementace, 144
- dělení, 140
- dělení integer se zbytkem (DIV), 142
- délky řetězce, 184
- double integer na ASCII, 97
- double integer na Integer, 93
- double Integer na Real, 93
- druhá odmocnina, 143
- FIFO, 190
- for, 170
- HEX na ASCII, 96
- hodin reálného času, 73
- inkrementace, 144
- Integer na ASCII, 96
- Integer na BCD, 93
- Integer na Byte, 93
- Integer na Double Integer, 93
- Inverze Byte, 162
- Inverze Double Word, 162
- Inverze Word, 162
- konverze bytu na integer, 93
- kopírování části řetězce, 186
- konverze dílčího řetězce na double integer, 100, 103
- konverze dílčího řetězce na integer, 100, 103
- konverze dílčího řetězce na reálné číslo, 100, 103
- konverze double integer na řetězec, 100, 103
- konverze integer na řetězec, 100, 103
- konverze reálného čísla na řetězec, 100, 103
- kopírování řetězce, 184
- kosinus, 143
- LIFO, 190
- MBUS_INIT, 333
- MBUS_SLAVE, 335
- modemového modulu EM 241, 298

- MODx_CTRL, 299
- MODx_MSG, 300
- MODx_XFR, 299
- náběžná hrana, 66
- načtení adresy portu, 88
- načtení sekvenčního řídicího relé, 173
- načtení zásobníku, 70
- nalezení části řetězce, 187
- nalezení znaku uvnitř řetězce, 187
- naplnění paměti (Memory Fill), 192
- násobení, 140
- násobení integer s 32bitovým výsledkem (MUL), 142
- nastavení, 68
- nastavení adresy portu, 88
- nastavení hodin reálného času, 73
- návěští, 172
- návratu z podprogramu, 203
- návratu z přerušení, 155
- negativního přechodu, 66
- next, 170
- NOT, 66
- odčítání, 140
- odpojení přerušení, 155
- okamžitého čtení a přesunu bytu, 166
- okamžitého nastavení, 68
- OR, 163
- OR načtení, 70
- podmíněného návratu z podprogramu, 203
- podmíněného návratu z přerušení, 155
- podmíněného konce sekvenčního řídicího relé, 173
- podmíněného ukončení programu, 168
- polohovacího modulu EM 253, 257
- pop, 70
- porovnání, 55
- posunu bytu vlevo, 179
- posunu bytu vpravo, 179
- posunu double word vlevo, 179
- posunu double word vpravo, 179
- posunu word vlevo, 179
- posunu word vpravo, 179
- posunutí bitů posuvného registru, 181
- POSx_CFG, 268
- POSx_CLR, 267
- POSx_CTRL, 258
- POSx_DIS, 266
- POSx_GOTO, 260
- POSx_LDOFF, 263
- POSx_LDPOS, 264
- POSx_MAN, 259
- POSx_RSEEK, 262
- POSx_RUN, 261
- POSx_SRATE, 265
- povolení přerušení, 155
- protokolu Modbus slave, 332
- provádění, 23
- přemístit blok Byte, 167
- přemístit blok Double Word, 167
- přemístit blok Word, 167
- přerušení, 155–161
- přesunu bytu, 165
- přesunu double word, 165
- přesunu reálného čísla, 165
- přesunu word, 165
- přidání do tabulky, 189
- příjmu, 79
- přímého kontaktu, 66
- přímého vynulování, 68
- přímého výstupu, 68
- přirozeného exponenciálu, 143
- přirozeného logaritmu, 143
- přiřazení přerušení, 155
- pulzní šířkové modulace (PWM), 125
- pulzního časovače (TP), 201
- pulzního výstupu (PLS), 125
- logické Push, 70
- Real na ASCII, 98
- rotace bytu doleva, 179
- rotace bytu doprava, 179
- rotace double word doleva, 179
- rotace double word doprava, 179
- rotace word doleva, 179
- rotace word doprava, 179
- RS klopného obvodu, 72
- řízení smyčky (PID), 145
- sčítání (Add), 140
- segment, 95
- sekvenční řídicí relé – konec, 173
- sekvenční řídicí relé – přechod, 173
- sestupného čítače, 109
- sinus, 143
- skok na návěští, 172
- smyčky PID, 145
- spojování řetězců, 184
- SR klopný obvod, 72
- standardního kontaktu, 66
- stop, 168
- tvorba pohybu, 286
- ukončení, 168XOR, 163
- volání podprogramu, 203
- vynulování, 68
- vysokorychlostního čítače (HSC), 111
- vysokorychlostního pulzního výstupu (PLS), 125
- výstupu, 68
- výstupu série pulzů (PTO), 125
- zakázání přerušení, 155
- zakódování, 105
- zaokrouhlení, 94
- zápis do sítě, 74
- žádná operace, 68
- Instrukce, stručný přehled, 435
- Instrukce, vložení, 9
- Instrukce bitové logiky
 - instrukce cívky, 68
 - instrukce kontaktů, 66
 - instrukce logického zásobníku, 70
 - RS a SR klopný obvod, 72
 - RS klopný obvod, 72
- Instrukce cívky
 - nastavení, 68
 - přímé nastavení, 68

- přímé vynulování, 68
- přímý výstup, 68
- vynulování, 68
- výstup, 68
- žádná operace, 68
- Instrukce časovače
 - IEC
 - časovač zpoždění náběžné hrany (TON), 201
 - časovač zpoždění sestupné hrany (TOF), 201
 - pulzní časovač (TP), 201
 - přerušení, 158
 - SIMATIC
 - časovač zpoždění náběžné hrany (TON), 196
 - časovač zpoždění náběžné hrany s pamětí (TONR), 196
 - časovač zpoždění sestupné hrany (TOF), 196
- Instrukce čítače
 - definice vysokorychlostního čítače (HDEF), 111
 - IEC
 - čítač dolů, 109
 - čítač nahoru, 109
 - čítač nahoru/dolů, 109
 - SIMATIC
 - čítač dolů, 106
 - čítač nahoru, 106
 - čítač nahoru/dolů, 106
 - vysokorychlostní čítač (HSC), 111
- Instrukce čítače dolů, 109
- Instrukce hodin reálného času, 73
- Instrukce "invertovat Byte", 162
- Instrukce "invertovat Double Word", 162
- Instrukce "invertovat Word", 162
- Instrukce inverze, příklad, 162
- Instrukce komunikace
 - čtení ze sítě, 74
 - načtení adresy portu (Get Port Address), 88
 - nastavení adresy portu, 88
 - příjem, 79
 - vysílání, 79
 - zápis do sítě, 74
- Instrukce kontaktů, 66
- příklad, 67
- Instrukce logického zásobníku
 - AENO, 70
 - AND Load, 70
 - bez výstupů, 55
 - čtení, 70
 - instrukce (pokračování) výměny bytů, 183
 - načtení zásobníku, 70
 - OR, 70
 - ořezání, 94
 - pop, 70
 - prohledávání tabulky, 193
 - příklad, 71
 - push, 70
 - tabulky, 190–195
 - tangens, 143
 - USS protokolu, 314
 - vysílání, 79
 - vzestupného čítače, 109
 - vzestupného/sestupného čítače, 109
 - Watchdog Reset, 168
- Instrukce logických operací
 - AND, OR, XOR, 163
 - inverze, 162
- Instrukce matematiky
 - dekrementace, 144
 - dělení, 140
 - dělení integer se zbytkem (DIV), 142
 - inkrementace, 144
 - násobení, 140
 - násobení integer s 32bitovým výsledkem (MUL), 142
 - odčítání, 140
 - sčítání (Add), 140
- Instrukce matematiky celých čísel, příklad, 141
- Instrukce matematiky reálných čísel, příklad, 141
- Instrukce MODx_CTRL, modemový modul EM 241, 299
- Instrukce MODx_XFR, modemový modul EM 241, 299
- Instrukce MODx_MSG, modemový modul EM 241, 300
- Instrukce numerických funkcí
 - druhá odmocnina, 143
 - kosinus, 143
 - přírozený exponenciál, 143
 - přírozený logaritmus, 143
 - sinus, 143
 - tangens, 143
- Instrukce podprogramu
 - podmíněný návrat z podprogramu, 203
 - příklad, 205
 - volání podprogramu, 203
- Instrukce porovnání , 55
 - porovnání bytu, 89
 - porovnání double word, 89
 - porovnání integer, 89
 - porovnání reálných čísel, 89
 - porovnání řetězců, 91
 - příklad, 89
- Instrukce posunu
 - příklad, 180
 - typy, 179
- Instrukce pro řetězce
 - délky řetězce, 184
 - kopírování části řetězce, 186
 - kopírování řetězce, 184
 - nalezení části řetězce, 187
 - nalezení znaku uvnitř řetězce, 187
 - spojování řetězců, 184
- Instrukce pro konverzi
 - ASCII, 96
 - ASCII na Hex, 96
 - dekódování, 105
 - Hex na ASCII, 96
 - ořezání, 94
 - příklad, 94
 - řetězec, 100, 103
 - segment, 95

- standardní, 92
 - zakódování, 105
 - zaokrouhlení, 94
 - Instrukce pro standardní konverzi, 92
 - Instrukce přechodu, 66
 - Instrukce přerušení
 - odpojení přerušení, 155
 - podmíněný návrat z přerušení, 155
 - povolení přerušení, 155
 - příklad, 161
 - připojení přerušení, 155
 - zakázání přerušení, 155
 - Instrukce přímých kontaktů, 66
 - Instrukce rotace, 179
 - příklad, 180
 - typy, 179
 - Instrukce řízení programu
 - instrukce skoku, 172
 - podmíněné ukončení, 168
 - Reset Watchdog, 168
 - sekvenční řídicí relé (SCR), 173
 - smyčka For–Next, 170
 - stop, 168
 - základní řízení programu, 168
 - Instrukce sekvenčního řídicího relé
 - divergence řízení, 176
 - konec sekvenčního řídicího relé, 173
 - konvergence řízení, 176
 - načtení sekvenčního řídicího relé, 173
 - omezení, 173
 - podmíněný konec sekvenčního řídicího relé, 173
 - přechod sekvenčního řídicího relé, 173
 - příklad, 173
 - Instrukce smyčky
 - automatický režim, 151
 - derivační člen, 148
 - chyby, 151
 - integrační člen, 147
 - kontrola poruchových signálů, 151
 - konverze vstupů smyčky, 149
 - konverze výstupů smyčky na integer s upraveným měřítkem, 150
 - manuální režim, 151
 - normalizace vstupů smyčky, 149
 - PID, 145
 - proměnné, 150
 - proporcionální člen, 147
 - průvodce, 145
 - příklad, 153
 - příklad programu, 152
 - přímá, 150
 - reverzní, 150
 - režimy, 151
 - rozsahy, 150
 - řízení smyčky, 148
 - tabulka smyčky, 152
 - typy řízení smyčky, 148
 - vysvětlení, 146
 - Instrukce smyčky For–Next
 - for, 170
 - next, 170
 - příklad, 171
 - Instrukce standardní kontakt, 66
 - Instrukce STL doby provádění, 425
 - stručný přehled, 435
 - Instrukce tabulky
 - FIFO, 190
 - LIFO, 190
 - naplnění paměti, 192
 - prohledávání tabulky, 193
 - přidávání do tabulky (Add to Table), 189
 - registr obrazu vstupů, 23
 - Instrukce USS protokolu
 - chybové kódy provádění, 323
 - návod k užívání, 314
 - vzorový program, 322
 - USS4_DRV_CTRL, 316
 - USS4_INIT, 315
 - USS4_RPM_x and USS4_WPM_x, 319, 320
 - Instrukce vysokorychlostního čítače (HSC), 111
 - příklad, 124
 - Instrukce vzestupného čítače, 109
 - Instrukce vzestupného/sestupného čítače, 109
 - Instrukční soubor IEC 1131–3, 53
 - Instrukční soubory IEC 1131–3, 53
 - příklad, 99
 - SIMATIC, 53
 - volba, 53
 - Instrukční strom, 9, 51
 - Integer na ASCII, instrukce, 96
 - Integer na BCD, instrukce, 93
 - Integer na Byte, instrukce, 93
 - Integer na Double Integer, instrukce, 93
 - Integer s uprav. měřítkem, konverze výstupů smyčky, 150
 - Integrační člen, algoritmus PID, 147
 - Inteligentní moduly, 3
 - CPU podporující, 303
 - stav (SMB200 až SMB549), 419
- ## J
- JOG_INCREMENT, polohovací modul EM 253, 249
 - JOG_SPEED, polohovací modul EM 253, 249
- ## K
- Kabel pro rozšiřovací moduly, 395
 - Kabel pro rozšiřovací moduly, instalace, 395
 - Kabeláž, 18, 19
 - Kabely
 - objednací čísla, 422
 - PC/PPI, 396
 - rozšiřovací sběrnice, 395
 - síťové, 218, 219
 - symetrizace, 220
 - ukončení, 220
 - Kalibrace
 - EM 231, 353
 - EM 235, 353

vstupů, 353
 Knihovna protokolu Modbus, 329
 Knihovna USS protokolu, řízení pohonů MicroMaster, 311
 Knihovny instrukcí, 60
 Knihovny instrukcí STEP 7 – Micro/WIN 32, 422
 Kódy zemí, podporované EM 241, 288
 Kompatibilita
 EM 231 RTD, 362
 EM 231 termočlánekový, 362
 EM 277 PROFIBUS – DP, 374
 Komunikace
 konflikty, 231
 modem, 226–227
 podporované protokoly, 211
 Protokol Modbus slave, 330
 přenosová rychlost, 226
 s pohony MicroMaster, 312S7 – 200, 7
 Komunikace PPI, změna na režim Freeport, 80
 Komunikace s pohonem, výpočet potřebné doby, 313
 Komunikace v režimu Freeport, 409
 SMB30 a SMB130, 412
 Komunikační fronta, 159
 Komunikační karty, objednáací čísla, 422
 Komunikační port
 přerušení, 158
 režim Freeport, 222
 rozmístění vývodů konektoru, 219
 Komunikační procesor CP 243 – 2
 činnost, 394
 funkce, 394
 objednáací číslo, 393
 připojení, 394
 vlastnosti, 393
 Komunikační procesory, CP 243 – 2, 393
 Komunikační procesory. Viz CP karty
 Komunikační protokol
 definovaný uživatelem, 222
 dvoubodové spojení (point-to-point) (PPI), 211, 230
 mnohabodové spojení (multipoint) (MPI), 212, 230
 PROFIBUS, 212, 230
 volba, 211
 Konec, instrukce, 168
 Konec sekvenčního řídicího relé, instrukce, 173
 Konektory, objednáací čísla, 422
 Konfigurace
 EM 231, 354
 EM 231 RTD, 368
 EM 231 termočlánekový, 363
 EM 235, 355
 EM 277 PROFIBUS – DP, 376–377
 funkce PTO/PWM, 128
 hesla, 44
 polohovací modul EM 253, 275
 remanentní oblasti paměti, 41
 sítě, 218
 stavy výstupů, 40
 tabulka symbolů pro Modbus, 331
 zobrazení stavu FBD a LAD, 238
 zobrazení stavu STL, 239

Konfigurace vstupů, editace implicitních hodnot v polohovacím modulu EM 253, 247
 Konfigurace výstupů, editace implicitní v polohovacím modulu, 247
 Konflikty, vyhnutí v síti, 231
 Konstanty, 30
 Kontaktní schéma Viz Editor LAD
 Kontrola poruchových signálů, smyčka PID, 151
 Konvence
 programování S7 – 200, 55
 programové editory, 54
 Konvergence řízení
 instrukce sekvenčního řídicího relé, 176
 příklad, 177
 Konverze
 vstupů smyčky, 149
 výstupů smyčky na integer s uprav. měřítkem, 150
 Konverze ASCII, instrukce, 96
 Konverze Double Integer na řetězec, instrukce, 100, 103
 Konverze Integer na řetězec, instrukce, 100, 103
 Konverze Real na řetězec, instrukce, 100, 103
 Konverze Substring na Double Integer, instrukce, 100, 103
 Konverze Substring na Integer, instrukce, 100, 103
 Konverze Substring na Real, instrukce, 100, 103
 Konzistence dat, PROFIBUS, 378
 Konzistence wordů, PROFIBUS, 378
 Kopírování programu do paměťového zásuvného modulu, 36
 Kopírování řetězce, instrukce, 184
 Kopírování řetězců, instrukce, 186
 Kosinus, instrukce, 143
 Kritické chyby, 57
 prohlížení, 404
 Kroky profilu, polohovací modul EM 253, 253

L

Ladění
 editace v režimu RUN, 236
 více programových cyklů, 240
 vlastnosti, 236
 vnucení hodnot, 240
 Ladění softwaru, 235
 LIFO, instrukce, 190
 příklad, 191
 Lišta
 montáž, 16
 rozměry, 16
 Lloyds Register of Shipping (LRS), 338
 Logická spojení
 MPI, 212
 PPI, 211
 Logické Read, instrukce, 70
 Logický Pop, instrukce, 70
 Logický Push, instrukce, 70
 Logika, řídicí, 22
 Lokální vstup/výstup, 31

M

- Manuální režim, smyčka PID, 151
 - Manuály, objednáací čísla, 422
 - MAX_SPEED, polohovací modul EM 253, 248
 - Maximální počet přijatých znaků, 85
 - MBUS_INIT, instrukce, 333
 - MBUS_SLAVE, instrukce, 335
 - Montáž na DIN lištu, 16
 - Montáž na panel, 16
 - Mapování adresy pro S7–200, Modbus, 331
 - Mezinárodní normy, 338
 - Micro/WIN. Viz STEP 7–Micro/WIN
 - Mikro–PLC, navrhování, 48
 - Místa speciální paměti, polohovací modul EM 253, 281
 - Modem, s PC/PPI kabelem, 226
 - Modemový modul, 385
 - formát textové zprávy, 309
 - formát zprávy s přesunem dat CPU, 310
 - chyby z instrukcí, 301
 - instrukce, 298
 - konektor RJ11, 288
 - MODx_CTRL, instrukce, 299
 - MODx_MSG, instrukce, 300
 - MODx_XFR, instrukce, 299
 - ochrana heslem, 291
 - průvodce modemem, 294
 - přenos numerických zpráv, 290
 - přenos textových zpráv, 290
 - přenos zpráv pro pager, 290
 - přesuny dat, 291
 - příklad, 303
 - rozhraní mezinárodní telefonní linky, 288
 - SMS, posílání zpráv, 290
 - stavové LED, 294
 - tabulka konfigurace, 293
 - telefonní číslo pro přenos zpráv, 308
 - textové zprávy, 290
 - vlastnosti, 288
 - zabezpečení zpětným voláním, 292
 - Modemový modul EM 241
 - adresy Modbus, 290
 - blokové schéma, 385
 - formát zprávy s přesunem dat CPU, 310
 - chyby z instrukcí, 301
 - instalace, 386
 - instrukce, 298
 - jack RJ11, 288
 - místa speciální paměti, 304
 - MOD_XFR, instrukce, 299
 - MODx_CTRL, instrukce, 299
 - MODx_MSG, instrukce, 300
 - ochrana heslem, 291
 - podporované země, 288
 - protokol Modbus RTU, 289
 - průvodce modemem, 294
 - přenos numerických stránek, 290
 - přenos stránek, 290
 - přenos textových stránek, 290
 - přesuny dat, 291
 - příklad, 303
 - rozhraní mezinárodní telefonní linky, 288
 - rozhraní STEP 7–Micro/WIN, 289
 - služba krátkých textových zpráv (SMS), 290
 - specifikace, 385
 - stavové kontrolky, 294
 - tabulka konfigurace, 293, 306
 - telefonní číslo pro přenos zpráv, 308
 - vlastnosti, 288
 - zabezpečení zpětným voláním, 292
 - Modul EM 277 PROFIBUS–DP
 - další vlastnosti, 380
 - jako DP slave, 375
 - kompatibilita s CPU, 374
 - konfigurační soubor, 381–382
 - konfigurování, 376–377
 - protokol DP, 375
 - přepínače adres, 374
 - režim výměny dat, 378
 - speciální paměťové byty, 379
 - specifikace, 373
 - stavové kontrolky, 374
 - stavové LED, 380
 - v síti PROFIBUS, 375
 - volby konfigurace, 377
 - Moduly AS–interface, objednáací čísla manuálů, 422
 - Moduly CP242, CP243, objednáací čísla manuálů, 422
 - Momentové charakteristiky, typické pro motory, 248
 - Montáž
 - DIN lišta, 16
 - nároky na prostor, 16
 - panel, 16
 - rozměry, 16
 - Motory, momentová charakteristika, typická, 248
 - MPI kabel, 4
 - MPI, protokol, 212, 230
- ## N
- Náběžná hrana, instrukce, 66
 - Načtení AND, instrukce, 70
 - Načtení sekvenčního řídicího relé, instrukce, 173
 - Načtení zásobníku, instrukce, 70
 - Najít v tabulce, instrukce, 193
 - příklad, 195
 - Napájení, 6, 15
 - Napěťové oddělení
 - návod pro kabeláž, 18
 - sítě, 218
 - Naplnění paměti, instrukce, 192
 - příklad, 192
 - Násobení, instrukce, 140
 - Násobení Integer na Double Integer (MUL), instrukce, 142
 - příklad, 142
 - Nastavení
 - aktuálních a požadovaných hodnot pro HSC, 118
 - času, 73
 - data, 73
 - přenosové rychlosti, 208

řídícího bytu (HSC), 118
 vzdálené adresy pro S7–200, 210
 Nastavení adresy portu, instrukce, 88
 Nastavení DIP přepínače, PC/PPI kabel, 6, 223
 Nastavení komunikace, STEP 7–Micro/WIN, 7
 Nastavování
 pohonu MM3, 325
 pohonu MM4, 328
 Nástrojové lišty, 51
 Návěští, instrukce, 172
 Navigační lišta, 51
 Návod
 instalace, 14
 instrukce pro polohovací modul EM 253, 257
 kabeláž, 18
 konfigurace sítě, 218
 napěťové oddělení, 18
 navrhování systému mikro–PLC, 48
 ochranné obvody, 20
 prostředí s vysokou vibrací, 17
 podprogramy, 50
 podprogramy přerušení, 50, 157
 uzemnění a kabeláž, 19
 uzemnění a zapojení, 18
 vertikální montáž, 17
 změny ukazatelů pro nepřímé adresování, 32
 Návrat z podprogramu, instrukce, 203
 příklad, 205
 Návrat z přerušení, instrukce, 155
 Návrh
 bezpečnostních obvodů, 48
 Mikro–PLC, 48
 Navrhování systému, micro–PLC, 48
 Negativní polarita, polohovací modul EM 253, 247
 Nekritické chyby, 56
 Nepodmíněné vstupy, 55
 Nepřímá adresace, 32
 & a *, 32
 změny ukazatelů, 32
 Next, instrukce, 170
 příklad, 171
 Nippon Kaiji Kyokai (NK), 338
 Normalizace, vstupů smyčky, 149
 Normy, státní a mezinárodní, 338
 NOT, instrukce, 66
 Nulový modem, 226

O

Objednací čísla, 421
 analogový rozšiřovací modul, 351, 361
 CP 243–2, 393
 CPU, moduly, 340
 digitální rozšiřovací modul, 346
 kabel PC/PPI, 396
 simulátory vstupu, 398
 Oblast bitové paměti (M), 25
 uložení při ztrátě napájení, 35

Oblast lokální paměti (L), 28
 Oblast paměti časovače (T), 26
 Oblast paměti čítače (C), 26
 Oblast paměti sekvenčních řídicích relé (S), 29
 Oblast proměnné paměti (V), 25
 Oblast speciální paměti (SM), 28
 analogové potenciometry, 45
 Obnovení
 dat z EEPROM, 35
 programu z paměťového modulu, 37
 Obnovení, ztraceného hesla, 45
 Odčítání, instrukce, 140
 Odmocnina, instrukce, 143
 Odpojit přerušení, instrukce, 155
 Ochrana heslem, modemový modul EM 241, 291
 Ochranná dioda, 20
 Ochranné obvody, 20
 Ochranný obvod, 338
 Okamžitý Reset, instrukce, 68
 Okamžitý Set, instrukce, 68
 Okamžitý výstup, instrukce, 68
 Omezení přístupu, 44
 OP3, OP7, OP17, objednáací čísla, 423
 Operátorská rozhraní, objednáací číslo, 423
 Operátorské stanice, specifikování, 48
 Opětná montáž, svorkovnice, 17
 Oprávnění, 338
 Optimalizace, výkonu sítě, 228
 OR, instrukce, 163
 příklad, 164
 OR Load, instrukce, 70
 Oříznout, instrukce, 94
 Otevření, programového editoru, 9

P

Obnovení z EEPROM, 35
 přiřazení adres, 58
 uložení do EEPROM, 38
 Paměť
 CPU, 2
 obnovení z EEPROM, 35
 přístup, 24
 remanentní, 41
 Paměť C, 26
 Paměť I, 25
 Paměť L, 28
 Paměť M, 25
 Paměť Q, 25
 Paměť S, 29
 Paměť SM, 28
 funkce PTO/PWM, 128
 protokol Modbus slave, 330
 Paměť T, 26
 Paměť V, 25
 Paměťové oblasti
 mazání, 45
 rozsahy operandů, 65

- Paměťový bit ztráty remanentních dat SM0.2, 35
- Paměťový zásuvný modul, 34, 36, 395
 - chybové kódy, 404
 - instalace, 36
 - kopírování programu, 36
 - objednáací číslo, 422
 - obnovení programu, 37
- Parametry
 - typy pro podprogramy, 204
 - v podprogramech, 203
- Parametry manuálního posunu
 - polohovací modul EM 253, 249
 - provoz s manuálním posunem, 249
- Parametry vyhledávání, polohovací modul EM 253, 250
- PC/PPI, kabel, 4, 6
 - objednáací číslo, 396RS – 232, 223
 - přepínač volby přenosové rychlosti, 226, 397
 - režim Freeport, 223
 - s modemem, 226
 - specifikace, 396
 - volba, 220
- Počáteční hodnoty, přiřazení, 58
- Podmíněné přechody, příklad, 178
- Podmíněné vstupy, 55
- Podmínění návrat z přerušení, instrukce, 155
- Podmíněný návrat z podprogramu, instrukce, 203
- Podmínkový konec, instrukce, 168
 - příklad, 169
- Podmínkový konec sekvenčního řídicího relé, instrukce, 173
- Podmínky prostředí
 - doprava a skladování, 339
 - provozní, 339
- Podprogram časovaného přerušení, příklad, 161
- Podprogramy
 - návody, 50
 - parametr signálového toku, 204
 - příklad, 49
 - s parametry, 203
 - typy dat, 204
 - typy parametrů, 204
 - vnoření, 203
 - volání z přerušovacích podprogramů, 158
- Podprogramy přerušení, 23, 39
 - fronty, 159
 - komunikační port, 158
 - náběžná/sestupná hrana, 158
 - návody, 50, 157
 - od času, 158
 - priorita, 159
 - příklad, 49
 - sdílení dat s hlavním programem, 157
 - systémová podpora, 157
 - typy podporované S7–200, 158
 - volání podprogramů z, 158
 - vstupy a výstupy (I/O), 158
 - vysvětlení, 157
- Pohon MicroMasterkomunikace, 312
 - čtení a psaní, 319, 320
 - připojení, 324
 - řízení, 311
- Pohon MM3
 - nastavení, 325
 - připojení, 324
- Pohon MM4
 - nastavení, 328
 - připojení, 327
- Pohony. Viz Pohony MicroMaster
- Pohybové příkazy, polohovací modul EM 253, 284
- Pohybový profil, def. pro polohovací modul EM 253, 252
- Polohovací modul
 - ACCEL_TIME, 249
 - aktivní úroveň vstupu, 247
 - Control panel pro EM 253, 274–276
 - časy vstupního filtru, 247
 - DECEL_TIME, 249
 - diagnostické informace, 275
 - doba zaoblení, 250
 - eliminace vůle, 256
 - hledání referenčního bodu (RS seek), 250
 - chybové kódy, 276
 - chybové kódy instrukcí, 276
 - chybové kódy modulu, 277
 - instrukce, 257
 - kladná polarita, 247
 - konfigurace, 246, 275
 - konfigurace vstupů a výstupů, 247
 - kroky profilu, 253
 - max. rychlost a spouštěcí/zastavovací rychlost, 248
 - MAX_SPEED, 248
 - návod pro instrukce, 257
 - parametry pomalého posunu, 249
 - pohybové příkazy, 284
 - pohybový profil, definování, 252
 - POSx_CFG, 268
 - POSx_CLR, 267
 - POSx_CTRL, 258
 - POSx_DIS, 266
 - POSx_GOTO, 260
 - POSx_LD OFF, 263
 - POSx_LD POS, 264
 - POSx_MAN, 259
 - POSx_RSEEK, 262
 - POSx_RUN, 261
 - POSx_SRATE, 265
 - potenciometry, analogové nastavení, 45
 - pracovní režim profilu, 252
 - profily, 252
 - programování, 245
 - průvodce polohováním, 246
 - referenční bod (RP), 250
 - režimy vyhledání RP, 253–257
 - sekvence vyhledání RP, 251
 - speciální paměť, 281
 - SS_SPEED, 248
 - tabulka konfigurace profilů, 278
 - tvorba instrukcí, 286

- typ měření, výběr, 246
- vlastnosti, 244
- vstupy a výstupy, 244
- vzorový program řízení modulu, 286
- záporná polarita, 247
- zobrazení a řízení práce, 274
- Polohovací modul EM 253
 - ACCEL_TIME, 249
 - aktivní úrovně vstupu, 247
 - Control panel pro EM 253, 274–276
 - časy vstupního filtru, 247
 - DECEL_TIME, 249
 - diagnostické informace, 275
 - doba zaoblení, 250
 - eliminace vůle, 256
 - hledání referenčního bodu (RS seek), 250
 - chybové kódy, 276
 - chybové kódy instrukcí, 276
 - chybové kódy modulu, 277
 - instrukce, 257
 - kladná polarita, 247
 - konfigurace, 246, 275
 - kroky profilu, 253
 - max. rychlost a spouštěcí/zastavovací rychlost, 248
 - návod pro instrukce, 257
 - odezva na fyzické vstupy, 248
 - parametry pomalého posunu, 249
 - pohybové příkazy, 284
 - pohybový profil, definování, 252
 - POSx_CFG, 268
 - POSx_CLR, 267
 - POSx_CTRL, 258
 - POSx_DIS, 266
 - POSx_GOTO, 260
 - POSx_LDOFF, 263
 - POSx_LDPOS, 264
 - POSx_MAN, 259
 - POSx_RSEEK, 262
 - POSx_RUN, 261
 - POSx_SRATE, 265
 - pracovní režim profilu, 252
 - profily, 252
 - programování, 245
 - Průvodce polohováním, 246
 - příkazový byte, 283
 - příklady, 269–273
 - pulzní a směrové výstupy, 247
 - referenční bod (RP), 250
 - režimy hledání RP, 253–257
 - schéma zapojení, 391–392
 - speciální paměť, 281
 - SS_SPEED, 248
 - tabulka konfigurace profilů, 278
 - tvorba instrukcí, 286
 - typ měření, výběr, 246
 - vlastnosti, 244
 - vstupy a výstupy, 244
 - vyhledání RP, 251
 - vzorový program řízení modulu, 286
 - záporná polarita, 247
 - zobrazení a řízení práce, 274
- Porovnání Byte, instrukce, 89
- Porovnání Double Word, instrukce, 89
- Porovnání Integer, instrukce, 89
- Porovnání Real, instrukce, 89
- Porovnání String, instrukce, 91
- Porovnání, doby oběhu tokenu, 230
- Porušení pravidel kompilace, 406
- POSx_CFG, 268
- POSx_CLR, 267
- POSx_CTRL, 258
- POSx_DIS, 266
- POSx_GOTO, 260
- POSx_LDOFF, 263
- POSx_LDPOS, 264
- POSx_MAN, 259
- POSx_RSEEK, 262
- POSx_RUN, 261
- POSx_SRATE, 265
- Pořadí, přerušovacích událostí, 160
- Posunout Byte vlevo, instrukce, 179
- Posunout Byte vpravo, instrukce, 179
- Posunout Double Word vlevo, instrukce, 179
- Posunout Double Word vpravo, instrukce, 179
- Posunout Register Bit, instrukce, 181
 - příklad, 182
- Posunout Word vpravo, instrukce, 179
- Potlačení šumu, vstupní filtr, 41
- Povolení výstupu (ENO), 55
- Povolení vstupu (EN), 55
- Povolení, vysokorychlostních čítačů, 123
- Povolit přerušování, ENI, instrukce, 155
- Pozitivní polarita, polohovací modul EM 253, 247
- Požadavky komunikace, zpracování, 23
- Požadavky na napájení, 15
 - CPU, 399
 - rozšiřovací modul, 399
 - tabulka pro výpočet, 401
 - výpočet, 399, 401
 - vzor, 400
- Požadavky na zařízení, 3
- Požadovaná hodnota
 - nastavení pro HSC, 118
 - změna v HSC, 123
- Pracovní režim profilu, polohovací modul EM 253, 252
- Pracovní režimy, CPU
 - protokol Freeport, 222
 - stavové bity, 408
 - změna, 37
- Priorita
 - podprogramů přerušování, 159
 - přerušovacích událostí, 160
- PROFIBUS–DP
 - konzistence dat, 378
 - modul (EM 277), 375
 - standardní komunikace, 374
 - vzorový program, 383
- Profil popisu pohybu, polohovací modul EM 253, 252

- Program
 - analogové vstupy, 22
 - běh, 11
 - download, 11, 34
 - editace v režimu RUN, 236
 - funkce ladění, 236
 - chyby kompilace, 56
 - chyby provádění, 57
 - kopírování do paměťového modulu, 36
 - obnovení z paměťového modulu, 37
 - podprogramy, 50
 - příklad PID, 152
 - sdílení dat s přerušovacími podprogramy, 157
 - sledování, 11
 - stav sledování, 238
 - stavová tabulka, 59
 - strukturování, 49
 - tvorba, 8
 - uložení, 11, 34–38
 - upload, 34
 - určený počet programových cyklů, 240
 - vytvoření pomocí STEP 7 – Micro/WIN, 51
 - základní prvky, 49
- Program, vzorový, řízení polohovacího profilu, 286
- Programovací kabel, 4
- Programovací software, objednáací čísla, 422
- Programování
 - polohovacího modulu EM 253, 245
 - vysokorychlostních čítačů, 112
- Programování v době provádění programu, chybové kódy, 405
- Programové editory, 51
 - funkční bloky (FBD), 51
 - kontaktní schéma (LAD), 51
 - konvence, 54
 - otevření, 9
 - volba, 51
 - výpis instrukcí (STL), 51
- Programový cyklus, 22
 - časovače 198
 - určený počet, 240
- Prohlížení, chyby, 56
- Proměnné
 - monitorování pomocí stavové tabulky, 239
 - sledování, 59
 - smyčky PID, 150
 - symbolická adresace, 58
- Proporcionální člen, algoritmus PID, 147
- Prostor, nároky 14
- Protokol definovaný uživatelem, režim Freeport, 222
- Protokol, komunikační
 - definovaný uživatelem, 222
 - volba, 211
- Protokol Modbus RTU, 332
 - doba provádění, 330
 - funkce podporované modemovým modulem, 289
 - chybové kódy provádění, 335
 - inicializace, 330
 - instrukce, 332
 - konfigurace tabulky symbolů, 331
 - mapování adres, 290
 - mapování adres pro S7–200, 331
 - MBUS_INIT, 333
 - MBUS_SLAVE, 335
 - Modemový modul EM 241, 289
 - podporované funkce, 332
 - použité prostředky, 330
 - protokol Modbus slaveadresy, 331
 - příklad programování, 336
 - speciální paměť, 330
 - tabulka CRC, 330
- Protokol Modbus slave, požadavky, 330
- Protokol PPI, 211, 230
 - síť s jedním zařízením typu master, 213
 - síť s více zařízeními typu master, 213
 - složitá síť, 213
- Protokol PROFIBUS, 212, 230
- Protokoly, PROFIBUS–DP, 374
- Provádění řídicí logiky, 22
 - instrukcí, 23
- Průvodce
 - modemem, 294
 - PID, 145
 - polohováním, 246
- Průvodce modemem, modemový modul EM 241, 294
- Průvodce polohováním, 246
 - PTO/PWM, 125
- Pulzní a směrové výstupy, polohovací modul EM 253, 247
- Pulzní časovač, instrukce (TP), 201
- Pulzní šířková modulace, instrukce (PWM), 46, 125
 - čas cyklu, 127
 - hodnoty tabulky profilu, 130
 - inicializace, 132
 - konfigurace pamětí SM, 128
 - metody aktualizace, 127
 - průvodce polohováním, 125
 - příklad, 132
 - řízení krokového motoru, 130
 - vysvětlení, 127
 - změna šířky pulzu, 132
- Pulzní výstup, instrukce (PLS), 125
- Pulzní výstup, instrukce (PTO), 46, 125
 - čas cyklu, 126
 - hodnoty tabulky profilu, 130
 - inicializace pro režim s jedním segmentem, 134
 - inicializace pro režim s více segmenty, 135
 - konfigurace pamětí SM, 128
 - průvodce polohováním, 125
 - příklady, 136, 138
 - řetězení jednoho segmentu, 126
 - řetězení více segmentů, 126
 - řízení krokového motoru, 130
 - vysvětlení, 126
 - změna času cyklu, 134
 - změna času cyklu a počtu pulzů, 135
 - změna počtu pulzů, 135
- Pulzní výstupy
 - instrukce pulzní šířkové modulace (PWM), 125
 - instrukce pulzního výstupu (PLS), 125

- instrukce výstupu série pulzů (PTO), 125
- vysokorychlostní, 46
- Přehození Bytů, instrukce, 183
- příklad, 183
- Přechod sekvenčního řídicího relé, instrukce, 173
- Přenos, instrukce, 79
- příklad, 86
- režim Freeport, 79
- vysílání dat, 80
- Přenos numerických zpráv, modemový modul EM 241, 290
- Přenos textových zpráv na pager, modemový modul EM 241, 290
- Přenos zpráv, modemový modul, 290
- Přenosová rychlost, síť, 218
- Přenosové rychlosti
- nastavení, 208
- optimální, 228
- síťová, 218
- volby přepínače, PC/PPI kabel, 6, 210, 221, 223, 226, 397
- Přenosy dat, modemový modul EM 241, 291
- Přepětová ochrana, 20
- Přepínač režimu, 37
- Přepínače adres, EM 277 PROFIBUS–DP, 374
- Přeplnění fronty (SMB4), 409
- Přerušení
- přiřazení HSC, 119
- vysokorychlostní čítače, 113
- Přerušení od času, 158
- Přerušení od vstupů/výstupů, 158
- Přerušovací události
- priorita, 160
- stručný přehled, 432
- typy, 156
- Přesunout, instrukce, příklad, 183
- Přesunout blok, instrukce, příklad, 167
- Přesunout blok Byte, instrukce, 167
- Přesunout blok Double Word, instrukce, 167
- Přesunout blok Word, instrukce, 167
- Přesunout Byte, instrukce, 165
- Přesunout Byte okamžitě pro čtení, instrukce, 166
- Přesunout Byte okamžitě pro zápis, instrukce, 166
- Přesunout Double Word, instrukce, 165
- Přesunout Real, instrukce, 165
- Přesunout Word, instrukce, 165
- Přidat do tabulky, instrukce, 189
- příklad, 189
- Příjem, instrukce, 79
- časovač měřící čas mezi znaky, 84
- časovač zpráv, 84
- chyby parity, 85
- maximální počet přijatých znaků, 85
- podmínky ukončení, 82
- podmínky začátku zprávy, 82
- příjem dat, 80
- příklad, 86
- režim Freeport, 79
- rozpoznání pauzy (break), 83
- rozpoznání prázdného úseku, 82
- rozpoznání startovacího znaku, 82
- rozpoznání ukončovacího znaku, 84
- SMB86 až SMB94, SMB186 až SMB194, 416
- ukončení uživatelem, 85
- Příjem dat, 85
- Příkazový byte, polohovací modul EM 253, 283
- Příklady
- časovače SIMATIC, 198, 199, 200
- čítače SIMATIC, 108
- IEC časovače, 201
- IEC instrukce čítače, 109
- instrukce "AND", 164
- instrukce "ASCII to Hex", 99
- instrukce "Real na ASCII", 99
- instrukce "Remanentní zpožděné zapnutí", 200
- instrukce "Stop", 169
- instrukce "Zpožděné vypnutí", 199
- instrukce "Zpožděné zapnutí", 198
- instrukce čtení a zápisu v síti, 77
- instrukce dekódování, 105
- instrukce dekrementace, 144
- instrukce DIV, 142
- instrukce FIFO, 190
- instrukce inkrementace, 144
- instrukce Integer na ASCII, 99
- instrukce inverze, 162
- instrukce kontaktů, 67
- instrukce LIFO, 191
- instrukce logického zásobníku, 71
- instrukce matematiky celých čísel, 141
- instrukce matematiky reálných čísel, 141
- instrukce MUL, 142
- instrukce naplnění paměti, 192
- instrukce nastavení, 68
- instrukce návratu z podprogramu, 205
- instrukce OR, 164
- instrukce podmíněného ukončení programu, 169
- instrukce podprogramu, 205
- instrukce porovnání, 89
- instrukce posunu, 180
- instrukce posunutí bitů posuvného registru, 182
- instrukce pro standardní konverzi, 94
- instrukce prohledávání tabulky (Table Find), 195
- instrukce přerušení, 161
- instrukce přesunu, 183
- instrukce přesunu bloku dat, 167
- instrukce přidávání do tabulky, 189
- instrukce příjmu, 86
- instrukce Reset watchdog, 169
- instrukce rotace, 180
- instrukce Segment, 95
- instrukce sekvenčního řídicího relé, 173
- instrukce sestupného čítače, 108
- instrukce skoku na návěští, 172
- instrukce smyčky For–Next, 171
- instrukce smyčky PID, 153
- instrukce výměny, 183
- instrukce vynulování, 68
- instrukce vysílání, 86
- instrukce vysokorychlostního čítače, 124

- instrukce vzestupného/sestupného čítače, 108
 - instrukce XOR, 164
 - instrukce zakódování, 105
 - konvergence toků řízení, 177
 - modemový modul EM 241, 303
 - podmíněné přechody, 178
 - podprogram, 49
 - podprogram časovaného přerušení, 161
 - podprogramy přerušení, 49
 - polohovací modul, 269–273
 - program PID, 152
 - program protokolu USS, 322
 - protokol Modbus slave, programování, 336
 - PTO v režimu s jedním segmentem, 136
 - pulzní šířková modulace (PWM), 132
 - režim Freeport, 222
 - režim PTO s více segmenty, 138
 - režimy vysokorychlostních čítačů, 114
 - rozdělení toků řízení, 176
 - síť s oběhem tokenu, 229
 - volání podprogramu, 204
 - výpočet požadavků na napájení, 399
 - Přímé smyčky, 150
 - Připojení
 - kabelu PC/PPI, 6k S7–200, 7
 - modemu k S7–200, 226
 - pohonu MM3, 324
 - pohonu MM4, 327
 - síťových zařízení, 230
 - připojit přerušení, instrukce, 155
 - Přirozený exponent, instrukce, 143
 - Přirozený logaritmus, instrukce, 143
 - Přiřazení adres, 58
 - počátečních hodnot, 58
 - přerušení HSC, 119
 - síťových adres, 208
 - Přístup k word, 24
- ## R
- Real na ASCII, instrukce, 98
 - příklad, 99
 - Referenční bod (RP), 250
 - Hledání referenčního bodu (RP Seek)
 - RP_APPR_DIR, 250
 - RP_FAST, 250
 - RP_SEEK_DIR, 250
 - RP_SLOW, 250
 - směr vyhledávání, 250
 - Sekvence "vyhledání RP", režimy, 251
 - RP_OFFSET, 251
 - Registr HSC0, HSC1, HSC2 (SMB36 až SMB65), 413
 - Registr HSC3, HSC4, HSC5 (SMB131 až SMB165), 417
 - Registr obrazu vstupů (I), 25
 - Registr obrazu vstupů a výstupů, 39
 - Registr obrazu výstupů, 22
 - Registr obrazu výstupů (Q), 25
 - Relé, 20
 - životnost, 338
 - Remanentní bitová paměť, 35
 - Remanentní paměť, 41
 - Remanentní zpožděné zapnutí, instrukce (TONR), 196
 - příklad, 200
 - Repeatery
 - objednací čísla, 422
 - síť, 218
 - Reset, instrukce, 68
 - příklad, 68
 - Restartování, po kritické chybě, 57
 - Reverzní smyčky, 150
 - Režim Freeport
 - definování, 158
 - chyba parity (SMB3), 409
 - instrukce vysílání a příjmu, 79
 - povolení, 79
 - protokoly definované uživatelem, 222
 - přechod z PPI, 80
 - příjem znaku (SMB2), 409
 - příklad, 222
 - řízení znakovým přerušením, 85
 - standard RS–232, 223
 - Režim RUN, 11, 37
 - editace programu, 236
 - Režim s jedním segmentem
 - inicializace PTO, 134
 - změna času cyklu a počtu impulzů PTO, 135
 - změna času cyklu PTO, 134
 - změna počtu impulzů PTO, 135
 - Režim s více segmenty, inicializace PTO, 135
 - Režim STOP, 11, 37
 - Režim "vyhledání RP", volby, 253–257
 - polohovací modul EM 253, 253–257
 - Režim výměny dat, EM 277 PROFIBUS–DP, 378
 - Režimy
 - smyčky PID, 151
 - vysokorychlostních čítačů, 113
 - RJ11 konektor, modemový modul EM 241, 288, 386
 - Rotace Byte vlevo, instrukce, 179
 - Rotace Byte vpravo, instrukce, 179
 - Rotace Double Word vlevo, instrukce, 179
 - Rotace Double Word vpravo, instrukce, 179
 - Rotace Word vlevo, instrukce, 179
 - Rotace Word vpravo, instrukce, 179
 - Rozlišení časovače, 197, 198
 - Rozměry, CPU, 2
 - Rozmístění vývodů, komunikační port, 219
 - Rozpoznání pauzy (break), 83
 - Rozpoznání prázdného úseku, 82
 - Rozpoznání startovacího znaku, 82
 - Rozpoznání ukončovacího znaku, 84
 - Rozsah bytů a celých čísel, 24
 - Rozsahy, smyčka PID, 150
 - Rozsahy operandů, 65
 - Rozsahy paměti
 - moduly CPU, 64, 461
 - stručný přehled, 433
 - Rozsahy teploty
 - EM 231 RTD, 371–372
 - EM 231 termočlánkový, 366–367

Rozšiřovací moduly, 3
 adresování vstupů a výstupů, 31
 demontáž, 17
 ID a chybový registr, 411
 instalace, 16
 objednáací čísla, 421
 požadavky na napájení, 15, 399
 rozměry, 16

Rozšiřovací moduly (EM)
 analogové
 objednáací čísla, 351, 361
 schémata zapojení, 352
 specifikace, 351
 specifikace vstupů, 351
 specifikace výstupů, 352
 digitální, objednáací čísla, 346
 diskrétní
 obecná specifikace, 346
 schémata zapojení, 349–350
 specifikace vstupů, 346
 specifikace výstupů, 347

Rozšiřovací vstupy/výstupy, 31
 chyby sběrnice (SMW98), 417
 RS klopný obvod, instrukce, 72
 RTD modul (EM 231), 368

RTD modul EM 231
 kompatibilita s CPU, 362
 konfigurace, 368
 rozsahy teploty a přesnost, 371–372
 specifikace, 361
 stavové LED, 370
 svorkovnice, 362
 volba DIP přepínačů, 368–369

Ř

Řídicí byte, nastavení pro HSC, 118
 Řídicí logika, 22
 Řídicí registry pro Freeport (SMB30 a SMB130), 412
 Řízení
 složitě komunikace, 231chyby, 56
 Řízení krokového motoru, generátory PTO/PWM, 130
 Řízení přerušení od příjmu znaku, 85
 Řízení smyčky(PID) instrukce, 145–156
 chyby, 151
 konverze vstupů, 149
 konverze výstupů, 150
 příklad programu, 152–154
 přímá/reverzní, 150
 režimy, 151
 rozsahy/proměnné, 150
 upravení integrační sumy, 150
 volba typu, 148

S

S7–200
 adresování, 24
 akumulátory, 27
 analogové vstupy (AI), 29
 analogové výstupy (AQ), 29
 CPU moduly, 2
 čtení a zápis dat, 22
 download, 34
 elektromagnetická kompatibilita, 339
 funkce zachycování impulzů, 42
 hardware, odstraňování poruch, 241
 chybové kódy, 404
 instalace, 15
 jako zařízení typu slave, 208, 375
 komponenty systému, 2
 komunikace, 208
 konstantní hodnoty, 30
 konvence programování, 55
 modem, 226
 napájení, 6
 návod pro kabeláž, 19
 návod pro montáž, 14
 návod pro uzemnění, 19
 oblast bitové paměti (M), 25
 oblast lokální paměti (L), 28
 oblast paměti časovače (T), 26
 oblast paměti čítače (C), 26
 oblast paměti sekvenčních řídicích relé (S), 29
 oblast proměnné paměti (V), 25
 oblast speciální paměti (SM), 28
 odezva na kritickou chybu, 404
 odstraňování poruch, 241
 ochrana heslem, 44
 ošetření chyb, 56
 paměť, 24
 paměť C, 26
 paměť I, 25
 paměť L, 28
 paměť M, 25
 paměť Q, 25
 paměť S, 29
 paměť SM, 28
 paměť T, 26
 paměť V, 25
 paměťový modul, 36
 podmínky prostředí, 339
 podporované přerušovací podprogramy, 158
 podprogramy přerušení, 157
 programový cyklus, 22, 39
 provádění řídicí logiky, 22
 přenosové rychlosti, 208, 209–211
 připojení STEP 7–Micro/WIN, 7
 přístup k datům, 24
 registr obrazu vstupů a výstupů, 39
 registr obrazu vstupů (I), 25
 registr obrazu výstupů (Q), 25
 remanentní paměť, 41
 režim RUN, 11, 37

- režim STOP, 11, 37
- rozměry, 2
- rozsahy paměti, 64, 461
- rozšiřovací moduly, 3
- RS–232, 223
- schémata zapojení, 343–345
- síťová adresa, 209–211
- speciální možnosti, 39
- stavová tabulka, 239
- technická specifikace, 339
- ukládání dat, 34
- upload, 34
- vysokorychlostní čítače, 27
- vzory konfigurace sítě, 213, 214, 215
- zpracování komunikace, 40
- S7–200 systémový manuál, objednáčí čísla, 422
- S7–300, vzory konfigurace sítě, 214
- S7–400, vzory konfigurace sítě, 214
- Schémat konfigurace, 49
- Schémat, tvorba schémat konfigurace, 49
- Schémat zapojení
 - analogové rozšiřovací moduly, 352
 - digitální rozšiřovací moduly, 349–350
 - moduly CPU, 343–345
 - polohovací modul EM 253, 391–392
 - vstupy a výstupy CPU, 343
- Schválení, 338
- Sečíst, instrukce, 140
- Sedmisegmentový displej, instrukce, 95
 - příklad, 95
- Sestupná hrana, instrukce, 66
- Set, instrukce, 68
 - příklad, 68
- Set Real–Time Clock, instrukce, 73
- Signálový tok, parametr podprogramu, 204
- SIMATIC instrukce časovače, 196
 - příklad, 198, 199, 200
- SIMATIC instrukce čítače
 - příklady, 108
 - sestupný, 106
 - vzestupný, 106
 - vzestupný/sestupný, 106
- SIMATIC instrukční soubory, 53
- Simulátory, objednáčí čísla, 423
- Simulátory, vstupu, 398
- Sinus, instrukce, 143
- Síť
 - adresa nejvyšší stanice (HSA), 228
 - adresy, 208
 - adresy zařízení, 211
 - CP karta, 220
 - dooba oběhu tokenu, 228
 - faktor obnovení mezery (GUF), 228
 - kabel, 218, 219
 - kabel PC/PCI, 220
 - komponenty, 218–221
 - komunikační port, 219
 - modem, 226
 - MPI, méně než 187,5 kbaud, 214
 - MPI, více než 187,5 kbaud, 214
 - napěťové oddělení, 218
 - nastavení komunikace, 208–219
 - optimalizace výkonu, 228
 - pokyny pro konfigurování, 218
 - PPI s jedním zařízením typu master, 213
 - PPI s více zařízením typu master, 213
 - PROFIBUS–DP, 215
 - přenosová rychlost, 218
 - repeatery, 218
 - složitá, 231
 - složitá PPI, 213
 - symetrizace kabelu, 220
 - ukončení kabelu, 220
 - výpočet vzdáleností, 218
 - vytvoření, 218
 - vzory konfigurace, 213, 214, 215
 - zařízení HMI, 221
 - zařízení typu master, 208
 - zařízení typu slave, 208
- Síť PPI s jedním zařízením typu master, 213
- Síť PPI s více zařízením typu master, 213
- Síť MPI
 - méně než 187,5 kbaud, 214
 - více než 187,5 kbaud, 214
- Síť PROFIBUS–DP
 - repeatery, 218
 - rozmístění vývodů, 219
 - S7–315–2 a EM 277, 215
 - specifikace kabelu, 218
 - STEP 7–Micro/WIN a HMI, 215
- Síť s oběhem tokenu, příklad, 229
- Skok na návěští, instrukce, 172
 - příklad, 172
- Skokové instrukce
 - návěští, 172
 - skok na návěští, 172
- Sledování, 11
 - procesních proměnných, 59
 - proměnné se stavovou tabulkou, 239
 - stavu programu, 238
- Složitá síť PPI, 213
- Služba krátkých textových zpráv, modemový modul EM 241, 290
- SMB0: stavové bity, 408
- SMB1: stavové bity, 408
- SMB2: znak přijatý v režimu Freeport, 409
- SMB3: chyba parity v režimu Freeport, 409
- SMB4: přeplnění fronty, 409
- SMB5: stav vstupů a výstupů, 410
- SMB6: ID registr CPU, 410
- SMB7: rezervován, 410
- SMB8 až SMB21: identifikační a chybové registry
 - vstup/výstupního modulu, 411
- SMB28, SMB29 analogové nastavení, 45, 412
- SMB30 a SMB130: řídicí registry pro Freeport, 412
- SMB31 a SMW32: řízení zápisu do EEPROM, 413
- SMB34 a SMB35: registry časovaného přerušení, 413
- SMB36 až SMB65: registr HSC0, HSC1, HSC2, 413
- SMB66 až SMB85: registry PTO/PWM, 415

- SMB86 až SMB94, SMB186 až SMB194: řízení příjmu zpráv, 416
- SMB130: řídicí registry pro Freeport, 412
- SMB131 až SMB165: registr HSC3, HSC4, HSC5, 417
- SMB166 až SMB185: tabulka definice profilu PTO0, PTO1, 418
- SMB186 až SMB194: řízení příjmu zpráv, 416
- SMB200 až SMB549: stav inteligentních modulů, 419
- Směr, změna v HSC, 123
- Směrnice pro instalaci stejnosměrného napětí, 19
- Směrnice pro instalaci střídavého napětí, 19
- SMS, modemový modul, 290
- SMW22 až SMW26: doby programového cyklu, 412
- SMW98: chyby na rozšiřovací vstup/výstupní sběrnici, 417
- Soubor GSD, EM 277 PROFIBUS–DP, 381–382
- Speciální paměť
 - modemový modul EM 241, 304
 - protokol Modbus slave, 330
- Speciální paměťové bity, 408–416
 - stručný přehled, 431
- Speciální paměťové byty, EM 277 PROFIBUS–DP, 379
- Specifikace
 - analogového rozšiřovacího modulu, 351
 - diskrétního rozšiřovacího modulu, 346
 - modulu CPU, 340
 - EM 231 RTD, 361
 - EM 231 termočlánek, 361
 - EM 277 PROFIBUS–DP, 373
 - kabelu PC/PPI, 396
 - mikro–PLC, 48
 - modemu EM 241
- Specifikace napájení, moduly CPU, 341
- Specifikace vstupů
 - analogový rozšiřovací modul, 351
 - digitální rozšiřovací modul, 346, 348
 - filtrace, analogová, 42
 - filtrace, digitální, 41
 - kalibrace, 353
 - modul CPU, 343
 - podmíněné/nepodmíněné, 55
 - polohovací modul EM 253, 244
 - start a nulování (HSC), 116
 - vstupy, 22, 23
 - vysokorychlostní čítače, 113
- Specifikace výstupů
 - analogový rozšiřovací modul, 352
 - CPU moduly, 342
 - digitální rozšiřovací modul, 347
- Spojit řetězce, instrukce, 184
- Spuštění, STEP 7–Micro/WIN, 7
- SR klopný obvod, instrukce, 72
- SS_SPEED, polohovací modul EM 253, 248
- Standard RS–232
 - kabel PC/PPI, 223
 - režim Freeport, 223
- Standard RS–485, 219
- Standardní DIN lišta, 15
- Start, vysokorychlostní čítač, 116
- Státní normy, 338
- Stav
 - konec programového cyklu (scanu), 238
 - provádění, 238
 - sledování programu, 238
 - zobrazení v LAD a FBD, 238
 - zobrazení v STL, 239
- Stav vstupů a výstupů (SMB5), 410
- Stavová tabulka, 59
 - sledování hodnot, 239
 - vnučení hodnot, 240
- Stavové LED
 - EM 277 PROFIBUS–DP, 374, 380
 - EM 231 RTD, 370
 - EM 231 termočlánek, 365
 - modemový modul EM 241, 294
 - polohovacího modulu EM 253, 389
- Stavový byte, vysokorychlostní čítač, 119
- Stejnosměrné a reléové výstupy, 20
- STEP 7–Micro/WIN
 - CP karty, 221
 - instalace, 4
 - instrukční soubory
 - IEC 1131–3, 53
 - SIMATIC, 53
 - volba, 53
 - jako zařízení typu master, 208
 - kabel PC/PPI, 221
 - nastavení komunikace, 7
 - nástroje pro ladění, 235
 - navázání komunikace se S7–200, 7
 - objednací čísla, 422
 - otevření, 51
 - požadavky na počítač, 3
 - požadavky na zařízení, 3
 - programovací balík, 3
 - programové editory, 51
 - přenosové rychlosti, 209–211
 - rozhraní pro EM 241, 289
 - síťová adresa, 209–211
 - spuštění, 7
 - tvorba programů, 51
 - vzory konfigurace sítě, 213–216
- Stop, instrukce, 168
 - příklad, 169
- Stručný přehled informací, 431
- Strukturování, programu, 49
- Střídavé a reléové výstupy, 20
- Svorkovnice
 - CPU 221 AC/DC/Relé, 343
 - CPU 221 DC/DC/DC, 343
 - CPU 222 AC/DC/Relé, 344
 - CPU 222 DC/DC/DC, 344
 - CPU 224 AC/DC/Relé, 344
 - CPU 224 DC/DC/DC, 344
 - CPU 226 AC/DC/Relé, 345
 - CPU 226 DC/DC/DC, 345

- CPU 226XM AC/DC/Relé, 345
 - CPU 226XM DC/DC/DC, 345
 - demontáž, 17
 - EM 223 4 x 24 V DC vstup/4 x 24 V DC výstup, 349
 - EM 231 AI 4 x 12 bitů, 352
 - EM 223 DI 4/DO 4 x DC 24 V/relé, 349
 - EM 221 DI 8 x AC, 349
 - EM 221 DI 8 x 24 V DC, 349
 - EM 223 DI 8/DO 8 x 24 V DC, 350
 - EM 223 DI 8/DO 8 x 24 V DC/relé, 350
 - EM 223 DI 16/DO 16 x 24 V DC relé, 350
 - EM 223 DI 16/DO 16 x DC 24 V, 350
 - EM 222 DO 8 x 24 V DC, 349
 - EM 222 DO 8 x relé, 349
 - EM 231 RTD, 362
 - EM 231 termočlánek, 362
 - EM 232 AQ 2 x 12 bitů, 352
 - EM 235 AI 4/AQ 1 x 12 bitů, 352
 - opakovaná montáž, 17
 - Symbolická adresace, 58
 - Symbolické názvy, tvorba seznamu, 49
 - Symetrizace
 - síťový kabel, 220
 - smyčka PID, 145, 147
 - Synchronní aktualizace, instrukce PWM, 127
 - Systémový blok, 50
 - Systémová podpora, pro podprogramy přerušení, 157
- T**
- Tabulka CRC, protokol Modbus slave, 330
 - Tabulka definování profilu PTO0, PTO1 (SMB166 až SMB185), 418
 - Tabulka globálních proměnných, 58
 - Tabulka konfigurace, modemový modul EM 241, 293, 306
 - Tabulka konfigurace profilů, polohovací modul EM 253, 278
 - Tabulka křížových odkazů, 236
 - Tabulka lokální proměnné, 51, 59
 - Tabulka profilů, polohovací modul EM 253, 278
 - Tabulka smyčky, 152
 - Tabulka symbolů, 58
 - adresace, 58
 - konfigurace pro Modbus, 331
 - Tangens, instrukce, 143
 - Telefonní linka, mezinár. modemový modul EM 241, 288
 - Teleservis, 289
 - Termočlánekový modul EM 231
 - kompatibilita s CPU, 362
 - konfigurace, 363
 - rozsahy teploty a přesnost, 366–367
 - specifikace, 361
 - stavové kontrolky, 365
 - stavové LED, 365
 - svorkovnice, 362
 - volba DIP přepínačů, 364
 - základní informace, 363
 - Textový displej TD 200
 - objednací číslo, 423
 - objednací číslo manuálu, 422
 - Touch panel TP070, 4
 - objednací čísla, 423
 - objednací číslo manuálu, 422
 - TP–Designer pro TP070, verze 1.0, 422
 - Tranzistor, ochrana, 20
 - Trvalé uložení programu, 38
 - Typ měření, polohovací modul EM 253, 246
 - Typy dat, parametry podprogramu, 204
- U**
- Uchování paměti, 34–37
 - bateriový modul (volitelný), 34
 - EEPROM, 34–36
 - rozsahy, 41
 - zapnutí napájení, 35–37
 - Ukazatele, nepřímá adresace, 32
 - Ukládání programových dat S7–200, 34
 - Ukončení, síťového kabelu, 220
 - Ukončení uživatelem, 85
 - Uložení
 - bitové paměti (M) do EEPROM, 35
 - hodnoty do EEPROM, 413
 - programových dat S7–200, 34
 - programu, 11
 - proměnné paměti (V) do EEPROM, 38
 - Umístění prac. pásma, polohovací modul EM 253, 256
 - Upload, programu, 34
 - USS protokol, požadavky, 312
 - USS4_DRV_CTRL, instrukce, 316
 - USS4_INIT, instrukce, 315
 - USS4_RPM_x, instrukce, 319, 320
 - USS4_WPM_x, instrukce, 319, 320
 - Uzemnění, 18, 19
 - Uživatelské knihovny, 60
- V**
- Vlastnosti
 - Modemový modul EM 241, 288
 - Moduly CPU, 64, 461
 - Polohovací modul EM 253, 244
 - Vložené proměnné v textových a SMS zprávách, 291
 - Vložení, instrukce, 9
 - Vnoření, podprogramy, 203
 - Vnucení hodnot (force), 239, 240
 - Volání podprogramu, instrukce, 203
 - příklad, 204
 - Volba komunikačního protokolu, 211
 - CP karty, 220
 - DIP přepínače termočlánek, 364
 - instrukčních souborů, 53
 - kabelu PC/PPI, 220
 - pracovního režimu S7–200, 37
 - programového editoru, 51
 - RTD DIP přepínače, 368–369

Vstupní heslo
 funkce CPU, 44
 konfigurace, 44
 mazání, 45
 obnovení ztraceného, 45
 omezení přístupu, 45
Vstupy a výstupy, adresace, 31
Vstupy a výstupy, čtení a zapisování, 39
Vstupy a výstupy, chyby, 57
Vstupy smyčky
 konverze, 149
 normalizace, 149
Vůle
 kompenzace, 251
 nastavení pracovního pásma, 256
Výkon, optimalizace výkonu sítě, 228
Vynulování, vysokorychlostního čítače, 116
Výpis instrukcí Viz Editor STL
Vyplnění, instrukce, 192
Výpočet
 doby oběhu tokenu, 228
 požadavků na napájení, 399–401
Vysokonapěťová izolační zkouška, 339
Vysokorychlostní čítače, 46
 adresace, 119
 časové diagramy, 114–116
 definování režimů a vstupů, 113
 funkce s "Reset" a "Start", 116
 inicializační sekvence, 120
 paměťová oblast, adresace, 27
 programování, 112
 přerušování, 113
 přiřazení přerušování, 119
 naprogramování aktivního stavu, 117
 nastavení aktuálních a požadovaných hodnot, 118
 nastavení řídicího bytu, 118
 režimy, 114, 434
 řídící byte, 112
 SMB36–SMB65, 413
 stavový byte, 119
 vysvětlení, 112
 zablokování, 123
 změna aktuální hodnoty, 123
 změna nastavené hodnoty, 123
 změna směru, 123
Vysokorychlostní pulzní výstup
 činnost, 125
 SMB36–SMB65, 415
 změna šířky pulzu, 132
Vysokorychlostní pulzní výstupy, 46
Výstup, instrukce, 68
Výstupy, 22
 digitálního rozšiřovacího modulu, 348
 instrukce bez, 55
 konfigurování stavů, 40
 modulu CPU, 343
 polohovacího modulu EM 253, 244
Výstupy a relé, 20
Výstupy smyčky, konverze na integer s upraveným měřítkem, 150

Vytvoření
 programu, 8
 programu s Micro/WIN, 51
 protokolů definovaných uživatelem, 222
 seznamu symbolických názvů, 49
 schémat konfigurace, 49
Vytvoření sítě, 218
Vývody konektoru, rozmístění v komunikačním portu, 219
Vzdálená adresa, nastavení pro S7–200, 210
Vzorový program, 8

W

Watchdog Reset, instrukce, 168
 příklad, 169

Z

Zabezpečení, hesla, 44
Zabezpečení zpětným voláním, modemový modul EM 241, 292
Zabránění konfliktům v síti, 231
Zadání
 nové aktuální hodnoty HSC, 123
 nové nastavené hodnoty HSC, 123
Zachycení impulzu, 41
Zakázat přerušování, instrukce, 155
Zakódovat, instrukce, 105
 příklad, 105
Záložky, 236
Zaokrouhlit, instrukce, 94
Zápis do sítě, instrukce, 74, 75
 chybové kódy, 75
 příklad, 77
Zapnutí napájení, uchování paměti, 35–37
Zařízení HMI, 221
Zařízení typu master, 208
Zařízení typu slave, 208
 EM 277 PROFIBUS–DP, 375
 S7–200, 375
Zásobník, PROFIBUS, 378
Zásuvné moduly
 objednací čísla, 422
 paměťový, 395
Zjistit adresu portu, instrukce, 88
Změna
 aktuální hodnoty HSC, 123
 času cyklu PTO, 134
 času cyklu a počtu impulzů PTO, 135
 nastavené hodnoty HSC, 123
 počtu impulzů PTO, 135
 směru vysokorychlostního čítače, 123
 šířky impulzu, 132
Změny, ukazatelů, 32
Zobrazení programových prvků, 51
 stavu programu, 238
Zobrazovací panely
 dotykový panel TP070, 4
 textový displej TD 200, 4

Zpětné volání, modemový modul EM 241, 292
Zpracování, požadavků komunikace, 23
Zpracování komunikace, 40
Zprávy, síť s oběhem tokenu, 229
Zřetězení, impulzy PTO, 126
Ztráta hesla, 45

Ztráta napájení, remanentní paměť, 35, 41
Zvýšení hodnoty ukazatelů, 32

Ž

Žadná operace, instrukce, 68

Pro

Siemens s.r.o.

Technická 15

616 00 Brno

Česká Republika

Od

Jméno: _____

Pracovní zařazení: _____

Název společnosti: _____

Ulice: _____

Město a stát: _____

Země: _____

Telefon: _____

Zaškrtněte prosím průmyslové odvětví, ve kterém působíte:

☐ Automobilový průmysl

☐ Chemický průmysl

☐ Elektrické stroje

☐ Potraviny

☐ Měření a regulace

☐ Jiné - než elektrické stroje

☐ Petrochemický průmysl

☐ Farmaceutický průmysl

☐ Plasty

☐ Celulóza a papír

☐ Textil

☐ Doprava

☐ Jiné _____



Formulář pro připomínky

Vaše poznámky a doporučení nám pomohou zlepšit kvalitu a užitečnost našich publikací. Využijte prosím první příležitosti, vyplňte tento dotazník a odešlete ho zpět společnosti Siemens.

Ohodnoťte prosím každou následující otázku podle svého názoru známkou od 1 (výborně) až do 5 (nedostatečně).

1. Splňuje obsah vaše požadavky?	☐
2. Našli jste snadno potřebné informace?	☐
3. Je text snadno pochopitelný?	☐
4. Odpovídá úroveň technických podrobností vašim požadavkům?	☐
5. Ohodnoťte prosím kvalitu grafiky a tabulek.	☐

Další poznámky:

Rozsahy paměti a vybavení S7-200

Popis	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Velikost uživatelského programu	2 Kword	2 Kword	4 Kword	4 Kword	8 Kword
Velikost uživatelských dat	1 Kword	1 Kword	2,5 Kword	2,5 Kword	5 Kword
Registr obrazu vstupů	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7	I0.0 až I15.7
Registr obrazu výstupů	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7	Q0.0 až Q15.7
Analogové vstupy (pouze čtení)	– –	AIW0 až AIW30	AIW0 až AIW62	AIW0 až AIW62	AIW0 až AIW62
Analogové výstupy (pouze zápis)	– –	AQW0 až AQW30	AQW0 až AQW62	AQW0 až AQW62	AQW0 až AQW62
Paměť pro proměnné (V)	VB0 až VB2047	VB0 až VB2047	VB0 až VB5119	VB0 až VB5119	VB0 až VB10239
Lokální paměť (L) ¹	LB0 až LB63	LB0 až LB63	LB0 až LB63	LB0 až LB63	LB0 až LB63
Bitová paměť (M)	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7	M0.0 až M31.7
Speciální paměť (SM) Pouze čtení	SM0.0 až SM179.7 SM0.0 až SM29.7	SM0.0 až SM299.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7	SM0.0 až SM549.7 SM0.0 až SM29.7
Časovače	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)	256 (T0 až T255)
Remanentní zpožděné zapnutí					
1 ms	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
10 ms	T1 až T4, a T65 až T68	T1 až T4, a T65 až T68	T1 až T4, a T65 až T68	T1 až T4, a T65 až T68	T1 až T4, a T65 až T68
100 ms	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95	T5 až T31, a T69 až T95
Zpožděné zapnutí/vypnutí					
1 ms	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
10 ms	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100	T33 až T36, a T97 až T100
100 ms	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255	T37 až T63, a T101 až T255
Čítače	C0 až C255	C0 až C255	C0 až C255	C0 až C255	C0 až C255
Vysokorychlostní čítač	HC0, HC3, HC4 a HC5	HC0, HC3, HC4 a HC5	HC0 až HC5	HC0 až HC5	HC0 až HC5
Sekvenční řídicí relé (S)	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7	S0.0 až S31.7
Registry akumulátoru	AC0 až AC3	AC0 až AC3	AC0 až AC3	AC0 až AC3	AC0 až AC3
Skoky/návěští	0 až 255	0 až 255	0 až 255	0 až 255	0 až 255
Volání/podprogramy	0 až 63	0 až 63	0 až 63	0 až 63	0 až 127
Podprogramy přerušení	0 až 127	0 až 127	0 až 127	0 až 127	0 až 127
Náběžné/sestupné hrany	256	256	256	256	256
Smyčky PID	0 až 7	0 až 7	0 až 7	0 až 7	0 až 7
Port	Port 0	Port 0	Port 0	Port 0, Port 1	Port 0, Port 1

¹ LB60 až LB63 jsou rezervovány pro STEP 7 – Micro/WIN, verze 3.0 nebo pozdější.

STL	Strana
=	68
+D	140
-D	140
*D	140
/D	140
+I	140
-I	140
=I	68
*I	140
/I	140
+R	140
-R	140
*R	140
/R	140
A	66
AB < =	89
AB =	89
AB >	89
AB<	89
AB > =	89
AB <>	89
AD <	89
AD < =	89
AD =	89
AD >	89
AD > =	89
AD <>	89
AENO	70
AI	66
ALD	70
AN	66
ANDB	163
ANDD	163
ANDW	163
ANI	66
AR=	89
AR <	89
AR<=	89
AR >	89
AR>=	89
AR <>	89
AS=	91
AS<>	91
ATCH	155
ATH	96
ATT	189
AW <	89
AW < =	89

STL	Strana
AW=	89
AW >	89
AW > =	89
AW <>	89
BCDI	92
BIR	166
BIW	166
BMB	167
BMD	167
BMW	167
BTI	92
CALL	203
CFND	187
COS	143
CRET	203
CRETI	155
CSCRE	173
CTD	106
CTU	106
CTUD	106
DECB	144
DECD	144
DECO	105
DECW	144
DISI	155
DIV	142
DTA	96
DTCH	155
DTI	92
DTR	92
DTS	100
ED	66
ENCO	105
END	168
ENI	155
EU	66
EXP	143
FIFO	190
FILL	192
FND <	193
FND <>	193
FND =	193
FND >	193
FOR	170
GPA	88
HDEF	111
HSC	111
HTA	96

STL	Strana
IBCD	92
INCB	144
INCD	144
INCW	144
INVB	162
INVD	162
INVW	162
ITA	96
ITB	92
ITD	92
ITS	100
JMP	172
LBL	172
LD	66
LDB <=	89
LDB =	89
LDB >=	89
LDB >	89
LDB <	89
LDB <>	89
LDD >=	89
LDD <	89
LDD <=	89
LDD =	89
LDD >	89
LDD <>	89
LDI	66
LDN	66
LDNI	66
LDR=	89
LDR <	89
LDR<=	89
LDR >	89
LDR>=	89
LDR <>	89
LDS	70
LDS=	91
LDS<>	91
LDW <=	89
LDW <	89
LDW =	89
LDW >	89
LDW >=	89
LDW <>	89
LIFO	190
LN	143
LPP	70
LPS	70

STL	Strana
LRD	70
LSCR	173
MOVB	165
MOVD	165
MOVR	165
MOVW	165
MUL	142
NEXT	170
NETR	74
NETW	74
NOT	66
O	66
OB =	89
OB > =	89
OB >	89
OB <	89
OB < =	89
OB <>	89
OD <	89
OD < =	89
OD =	89
OD >	89
OD > =	89
OD <>	89
OI	66
OLD	70
ON	66
ONI	66
OR=	89
OR <	89
OR<=	89
OR >	89
OR > =	89
OR <>	89
ORB	163
ORD	163
ORW	163
OS=	91
OS<>	91
OW <	89
OW < =	89
OW =	89
OW >	89
OW > =	89
OW <>	89
PID	145
PLS	125
R	68

STL	Strana
RCV	79
RI	68
RLB	179
RLD	179
RLW	179
ROUND	92
RRB	179
RRD	179
RRW	179
RTA	96
RTS	100
S	68
SCAT	184
SCPY	184
SCRE	173
SCRT	173
SEG	92
SFND	187
SHRB	181
SI	68
SIN	143
SLB	179
SLD	179
SLEN	184
SLW	179
SPA	88
SQRT	143
SRB	179
SRD	179
SRW	179
SSCPY	186
STD	100
STI	100
STOP	168
STR	100
SWAP	183
TAN	143
TODR	73
TODW	73
TOF	196
TON	196
TONR	196
TRUNC	92
WDR	168
XMT	79
XORB	163
XORD	163
XORW	163