

SIMATIC

S7-1200 Easy Book

マニュアル

はじめに

パワフルで使いやすい
S7-1200

1

カンタン 操作 STEP 7

2

ゲッティング・スタート

3

カンタン PLC コンセプト

4

カンタン デバイスコンフィ
グレーション

5

カンタン プログラミング

6

カンタン デバイス間通信

7

カンタン PID

8

カンタン Web サーバ接続

9

カンタン モーションコント
ロール

10

カンタン オンライン操作

11

カンタン IO-Link

12

技術仕様

A

V3.0 CPU を V4.1 CPU に
交換

B

法律上の注意

警告事項

本書には、ユーザーの安全を確保し製品の損傷を防止する上で守るべき注意事項が記載されています。ユーザーの安全に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損傷に関する注意事項には表示されません。注意事項は、危険度によって以下の等級に分類されています。

危険

回避しなければ、直接的な死亡または重傷に至る危険な状態を示します。

警告

回避しなければ、死亡または重傷に至るおそれのある危険な状況を示します。

注意

回避しなければ、軽度または中度の人身傷害を引き起こすおそれがある危険な状況を示します(安全警告サイン付き)。

注意

回避しなければ、物的損傷を引き起こすおそれのある危険な状況を示します(安全警告サインなし)。

複数の危険レベルに相当する場合は、通常、最も危険度の高い(番号の低い)事項が表示されることになっています。安全警告サイン付きの人身傷害に関する注意事項があれば、物的損傷に関する警告が付加されます。

有資格者

装置/システムのセットアップおよび使用にあたっては必ず本書を参照してください。機器のインストールおよび操作は**有資格者**のみが行うものとします。有資格者とは、法的な安全規制/規格に準拠してアースの取り付け、電気回路、設備およびシステムの設定に携わることを承認されている技術者のことをいいます。

シーメンス製品の適切な使用

以下の事項に注意してください。

警告

シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限りです。シーメンス製品を正しく安全にご使用いただくには、適切な運搬、保管、取り付け、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場所は、許容された範囲を必ず守ってください。付属の技術説明書に記述されている指示を順守してください。

商標

本書において®で識別されるすべての名称は、Siemens AG の登録商標です。本書に記載するその他の称号は商標であり、第三者が自己の目的において使用した場合、所有者の権利を侵害することになります。

免責事項

本書のハードウェアおよびソフトウェアに関する記述と、実際の製品内容との一致については検証済みです。しかしながら、本書の記述が実際の製品内容と異なる可能性もあり、完全な一致が保証されているわけではありません。記述内容については定期的に検証し、訂正が必要な場合は次の版で更新いたします。

本マニュアルは、英語版を原本として参照のみを目的として作成されるものであり、当社は、当該翻訳の不十分や正確性に関して責任を負わないものとします。

英語マニュアル : *SIMATIC S7-1200 Easy Book*

(<http://support.industry.siemens.com/cs/ww/en/view/39710145>)

はじめに

S7-1200 の世界によろこそ。SIMATIC S7-1200 コンパクトコントローラは、ロジック、HMI、およびネットワーク接続のためのシンプルな機能と高度な機能の両方を必要とする小型のオートメーションシステム用のモジュール型の場所を取らないコントローラです。コンパクトな設計、低コスト、パワフルな機能により、S7-1200 は小型アプリケーションを制御するための完璧なソリューションになっています。

「Totally Integrated Automation」(TIA)への SIMATIC の取り組みの一環として、S7-1200 製品ファミリーおよび TIA ポータルプログラミングソフトウェアは、お客様のオートメーションニーズの解決に必要な柔軟性を提供します。

S7-1200 は極めて難度の高い作業を簡単にこなすお手伝いをします!

「コンパクトな」コントローラクラス用に設計された SIMATIC S7-1200 コントローラソリューションは、SIMATIC S7-1200 コントローラと SIMATIC HMI ベーシックパネルで構成されていて、その両方を TIA ポータルエンジニアリングソフトウェアでプログラミングすることができます。同じエンジニアリングソフトウェアで両方のデバイスをプログラミングすることで、開発コストが大幅に削減されます。TIA ポータルには、S7-1200 のプログラミング用の STEP 7 とベーシックパネルプロジェクトの設計用の WinCC が含まれています。



S7-1200 コンパクトコントローラには以下が含まれています。

- 内蔵 PROFINET
- モーションコントロール対応の高速 I/O、スペース要件と追加 I/O を最小限に抑えるオンボードアナログ入力、パルストレインおよびパルス幅アプリケーション用の 4 つのパルスジェネレータ(72 ページ)、および最大 6 つの高速カウンタ (131 ページ)
- CPU モジュールに内蔵されたオンボード I/O 点により、6~14 の入力点と 4~10 の出力点が提供されます。



DC、リレー、またはアナログ I/O 用のシグナルモジュールにより I/O 点の数が増大し、革新的なシグナルボードを CPU の前面にスナップ式で取り付けて追加 I/O を提供できます(20 ページ)。

SIMATIC HMI ベーシックパネル (22 ページ) は特に S7-1200 専用設計されています。

この Easy Book では、S7-1200 PLC の概要を説明します。以降のページで、各デバイスのさまざまな特長と機能の概要を述べます。

追加情報については、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。UL および FM 認定、CE ラベル、C-Tick、およびその他の規格については、技術仕様 (363 ページ) を参照してください。

このマニュアルでは以下の製品について説明します。

- STEP 7 V13 SP1 Basic および Professional
- S7-1200 CPU ファームウェアリリース V4.1

文書と情報

S7-1200 および STEP 7 には、必要な技術情報を見つけるためのさまざまな文書とその他のリソースがあります。

- 『S7-1200 プログラマブルコントローラシステムマニュアル』には、S7-1200 製品ファミリー全体の操作、プログラミング、仕様に関する固有の情報が記載されています。システムマニュアルに加えて、『S7-1200 Easy Book』で S7-1200 ファミリーの機能の一般的な概要について述べます。

システムマニュアルと Easy Book はどちらも、電子(PDF)マニュアルとして提供されます。電子マニュアルはカスタマーサポートウェブサイトからダウンロードすることができます。また、それぞれの S7-1200 CPU に同梱されているマニュアルディスクにも収容されています。

- オンライン STEP 7 情報システムから、SIMATIC CPU のプログラミングパッケージの基本操作と機能、および SIMATIC PU の基本操作について説明する概念情報と固有の説明に即時にアクセスすることができます。
- My Documentation Manager は、システムマニュアル、Easy Book、STEP 7 情報システムを含む SIMATIC マニュアルセットの電子(PDF)版にアクセスすることができます。My Documentation Manager を使用して、各種の文書のトピックをドラッグ&ドロップし、独自のカスタムマニュアルを作成できます。

カスタマーサポートエントリポータル(<http://support.automation.siemens.com>)の mySupport に、My Documentation Manager へのリンクがあります。

- カスタマーサポートウェブサイトでは、ポッドキャスト、FAQ、その他の S7-1200 および STEP 7 の役に立つ文書も提供されています。ポッドキャストは、特定の機能またはシナリオに焦点を絞った短い教育ビデオのプレゼンテーションを使用して、STEP 7 が提供する相互作用、利便性、効率をデモンストレーションします。以下のウェブサイトでは、ポッドキャストのコレクションにアクセスしてください。
 - STEP 7 Basic Web ページ(<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-basic/Pages/Default.aspx>)
 - STEP 7 Professional Web ページ(<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-professional/Pages/Default.aspx>)
- サービス&サポート技術フォーラムで製品についての討議をフォローしたり、参加することもできます (<https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid=34612486>)。このフォーラムでは、各製品のエキスパートと相互にやり取りすることもできます。

- S7-1200 のフォーラム
(<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=258&Language=en&onlyInternet=False>)
- STEP 7 Basic のフォーラム
(<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=265&Language=en&onlyInternet=False>)

サービスとサポート

さまざまな文書のほかに、シーメンスはインターネットおよびカスタマーサポートウェブサイトで技術上のさまざまな情報を提供しています(<http://www.siemens.com/tiaportal>)。

技術的なご質問、トレーニングに関するお問い合わせ、S7 製品のご注文については、シーメンス正規販売店または弊社各支店にご相談ください。営業担当者は、専門的な技術訓練を受け、ご使用の個別のシーメンス製品についての知識だけでなく、担当先での運用、処理、業務についての詳細な知識を身に付けています。よって、どのような問題に対しても、迅速に効果的な答えを用意することができます。

セキュリティ情報

シーメンスは、当社製品およびソリューションに対して、プラント、ソリューション、機械またはネットワークの安全な運転をサポートする産業セキュリティファンクションを提供します。これらの製品は、産業セキュリティコンセプト全体にとって重要な構成要素となります。この点を踏まえて、シーメンスの製品は日々発展を続けています。そのため、当社製品に関する最新情報を常に確認することを強くお勧めします。

シーメンス製品およびソリューションの安全な稼働を確実にするために、適切な予防処置(たとえば、セルプロテクションコンセプト)を行うことや、最先端の総合的な産業セキュリティコンセプトに各構成要素を組み入れることも必要です。使用されている可能性があるサードパーティ製品についても同様に考慮する必要があります。産業セキュリティの詳細は、インターネットで参照できます(<http://www.siemens.com/industrialsecurity>)。

常に弊社製品の最新情報を入手するには、製品情報のニュースレターにご登録ください。詳細はインターネットを参照してください(<http://support.automation.siemens.com>)。

目次

	はじめに	4
1	パワフルで使いやすい S7-1200	17
1.1	S7-1200 PLC について	17
1.2	CPU の拡張機能	20
1.3	S7-1200 モジュール	21
1.4	HMI ベーシックパネル	22
1.5	取り付け寸法とクリアランス要件	22
1.6	新機能	27
2	カンタン 操作 STEP 7	31
2.1	ユーザープログラムに命令を簡単挿入	32
2.2	ツールバーからよく使用する命令に簡単アクセス	32
2.3	LAD および FBD 命令に入力または出力を簡単追加	33
2.4	拡張可能な命令	33
2.5	CPU の動作モードを簡単変更	34
2.6	STEP 7 の外観と構成を簡単変更	34
2.7	プロジェクトライブラリおよびグローバルライブラリに簡単アクセス	35
2.8	命令のバージョンを簡単選択	35
2.9	エディタ間で簡単ドラッグ&ドロップ	36
2.10	DB の呼び出しタイプの変更	37
2.11	ネットワークからデバイスを一時的に切断	38
2.12	コンフィグレーションを失わずに簡単にバーチャルでモジュールを「アンプラグ」	39
3	ゲッティング・スタート	41
3.1	プロジェクトの作成	41
3.2	CPU の I/O タグの作成	42
3.3	ユーザープログラムで簡単ネットワーク作成	44
3.4	タグテーブルの PLC タグを使用して命令をアドレス指定	46
3.5	「空ボックス」命令の追加	47
3.6	複雑な数学計算式用の CALCULATE 命令の使用	48
3.7	HMI デバイスをプロジェクトに追加	50
3.8	CPU と HMI デバイス間のネットワーク接続作成	51
3.9	タグの共有のための HMI 接続作成	51

3.10	HMI 画面の作成.....	52
3.11	HMI エLEMENT の PLC タグの選択	53
4	カンタン PLC コンセプト	55
4.1	スキャンサイクルごとのタスクの実行.....	55
4.2	CPU の動作モード	57
4.3	ユーザープログラムの実行.....	58
4.3.1	RUN モードでスキャンサイクル処理	58
4.3.2	OB を使用したユーザープログラムの構造化	59
4.3.3	イベント実行の優先度およびキュー	60
4.4	メモリ領域、アドレス指定、データタイプ	63
4.4.1	S7-1200 がサポートするデータタイプ.....	64
4.4.2	メモリ領域をアドレス指定する.....	66
4.4.3	タグ付きデータタイプの「スライス」アクセス	69
4.4.4	AT オーバーレイでタグにアクセス	70
4.5	パルス出力	72
5	カンタン デバイスコンフィグレーション	75
5.1	接続された CPU のコンフィグレーションをアップロード.....	76
5.2	コンフィグレーションに CPU を追加	78
5.3	デバイスの変更	79
5.4	コンフィグレーションにモジュールを追加	80
5.5	コンフィグレーション制御.....	81
5.6	CPU およびモジュールの動作設定 CPU およびモジュールの動作設定	82
5.6.1	システムメモリおよびクロックメモリ.....	84
5.7	CPU の IP アドレスの設定.....	87
5.8	CPU またはコードブロックへのアクセス保護が簡単.....	89
5.8.1	ノウハウプロテクト.....	91
5.8.2	コピー保護	92
6	カンタン プログラミング.....	95
6.1	ユーザープログラムの簡単設計.....	95
6.1.1	OB を使用したユーザープログラムの構成.....	97
6.1.2	FB および FC によりモジュール型タスクの簡単プログラミング.....	99
6.1.3	データブロックを使用したプログラムデータの簡単保存.....	100
6.1.4	コードブロックの新規作成.....	101
6.1.5	再利用できるプログラムブロックの作成	102
6.1.6	別のコードブロックからコードブロックの呼び出し	103
6.2	使いやすいプログラミング言語.....	103
6.2.1	ラダーロジック(LAD)	104
6.2.2	ファンクションブロックダイアグラム(FBD)	105
6.2.3	SCL の概要.....	105
6.2.4	SCL プログラミングエディタ	106
6.3	パワフルな命令により簡単プログラミング	107
6.3.1	ユーザーが期待する基本命令の提供	107

6.3.2	Comparator 命令と Move 命令	110
6.3.3	変換操作	111
6.3.4	Calculate 命令を使用した簡単数値計算	113
6.3.5	タイマの動作	115
6.3.6	カウンタの動作	120
6.3.7	パルス幅振幅(PWM)	123
6.4	データログの簡単作成	124
6.5	ユーザープログラムの簡単モニタとテスト	126
6.5.1	ウォッチテーブルとフォーステーブル	126
6.5.2	クロスリファレンスの使用状況表示	127
6.5.3	呼び出し構造体を使用した呼び出し階層検査	128
6.5.4	診断命令でハードウェアモニタ	129
6.5.4.1	CPU の LED 状態の読み取り	129
6.5.4.2	デバイスの診断ステータスを読み取るための命令	130
6.6	高速カウンタ(HSC)	131
6.6.1	高速カウンタの動作	133
6.6.2	HSC のコンフィグレーション	139
7	カンタン デバイス間通信	141
7.1	ネットワーク接続の作成	142
7.2	通信オプション	143
7.3	V4.1 非同期通信接続	145
7.4	PROFINET 命令と PROFIBUS 命令	148
7.5	PROFINET	149
7.5.1	オープンユーザーコミュニケーション	149
7.5.1.1	アドホックモード	150
7.5.1.2	オープンユーザーコミュニケーション命令用の接続 ID	150
7.5.1.3	PROFINET 接続のパラメータ	154
7.5.2	ローカル/パートナー接続パスの構成	156
7.6	PROFIBUS	159
7.6.1	PROFIBUS CM の通信サービス	160
7.6.2	PROFIBUS CM ユーザーマニュアルへの参照	161
7.6.3	CM 1243-5 (DP マスタ)モジュールと DP スレーブの追加	162
7.6.4	CM 1243-5 モジュールと DP スレーブへ PROFIBUS アドレスの割り当て	163
7.7	AS-i	165
7.7.1	AS-i マスタ CM 1243-2 と AS-i スレーブの追加	166
7.7.2	AS-i アドレスを AS-i スレーブに割り当て	167
7.8	S7 通信	170
7.8.1	GET 命令と PUT 命令	170
7.8.2	S7 コネクションの作成	171
7.8.3	GET/PUT 接続パラメータの割り当て	172
7.9	GPRS	173
7.9.1	GSM ネットワークの接続	173
7.9.2	CP 1242-7 の用途	175
7.9.3	CP-1242-7 のその他のプロパティ	176
7.9.4	構成と電氣的接続	177
7.9.5	その他の情報	177

7.9.6	付属品	178
7.9.7	GSM アンテナマニュアルの参照	179
7.9.8	遠隔制御の構成例	179
7.10	PtP、USS、および Modbus 通信プロトコル	184
7.10.1	ポイントツーポイント通信	184
7.10.2	シリアル通信インターフェースの使用	186
7.10.3	PtP 命令	187
7.10.4	USS 命令	188
7.10.5	Modbus 命令	190
8	カンタン PID	193
8.1	PID 命令とテクノロジーオブジェクトの挿入	195
8.2	PID_Compact 命令	197
8.3	PID_Compact 命令の ErrorBit パラメータ	201
8.4	PID_3Step 命令	203
8.5	PID_3Step 命令の ErrorBit パラメータ	210
8.6	PID_Temp 命令	212
8.6.1	概要	212
8.6.2	PID_Temp コントローラの動作	216
8.6.3	カスケードコントローラ	218
8.7	PID_Temp 命令の ErrorBit パラメータ	222
8.8	PID_Compact および PID_3Step コントローラの構成	224
8.9	PID_Temp コントローラの構成	227
8.10	PID_Compact および PID_3Step コントローラのコミッショニング	241
8.11	PID_Temp コントローラのコミッショニング	243
9	カンタン Web サーバ接続	255
9.1	標準 Web ページの簡単使用	256
9.2	Web サーバーの使用に影響する制約事項	258
9.3	ユーザー定義 Web ページの簡単作成	259
9.3.1	カスタム「ユーザー定義」Web ページの簡単作成	259
9.3.2	ユーザー定義 Web ページに固有の制約事項	261
9.3.3	ユーザー定義 Web ページの設定	261
9.3.4	WWW 命令の使用	262
10	カンタン モーションコントロール	265
10.1	位相調整	270
10.2	パルスジェネレータの構成	272
10.3	開ループモーションコントロール	273
10.3.1	軸の構成	273
10.3.2	コミッショニング	277
10.4	閉ループモーションコントロール	282
10.4.1	軸の構成	282
10.4.2	コミッショニング	289

10.5	TO_CommandTable_PTO の構成	295
10.6	S7-1200 のモーションコントロールの動作	298
10.6.1	モーションコントロールに使用する CPU 出力	298
10.6.2	モーションコントロール用のハードウェアおよびソフトウェアリミットスイッチ	300
10.6.3	原点復帰	304
10.6.3.1	軸の原点復帰	304
10.6.3.2	原点復帰パラメータの設定	305
10.6.3.3	アクティブ原点復帰のシーケンス	308
10.7	モーションコントロール命令	309
10.7.1	MC 命令の概要	309
10.7.2	MC_Power (軸の有効化/無効化)命令	310
10.7.3	MC_Reset (軸エラーリセット)命令	313
10.7.4	MC_Home (軸の原点復帰)命令	314
10.7.5	MC_Halt (軸の一時停止)命令	317
10.7.6	MC_MoveAbsolute (絶対値位置決め)命令	319
10.7.7	MC_MoveRelative (相対値位置決め)命令	321
10.7.8	MC_MoveVelocity (速度制御)命令	323
10.7.9	MC_MoveJog (ジョグモード)命令	326
10.7.10	MC_CommandTable (複数の軸コマンドを移動シーケンスとして実行)命令	328
10.7.11	MC_ChangeDynamic (軸の動的設定の変更)命令	330
10.7.12	MC_WriteParam (テクノロジーオブジェクトのパラメータを書き込み)命令	332
10.7.13	MC_ReadParam (テクノロジーオブジェクトのパラメータを読み取り)命令	334
11	カンタン オンライン操作	337
11.1	オンラインで CPU に接続	337
11.2	オンラインの CPU との相互作用	338
11.3	オンラインで CPU の値をモニタ	339
11.4	ユーザープログラムのステータスの表示が簡単	340
11.5	ウォッチテーブルを使用して CPU をモニタリング	340
11.6	フォーステーブルの使用	342
11.7	DB のオンライン値を取得して開始値のリセット	345
11.8	プロジェクトエレメントのアップロード	346
11.9	CPU のオフラインとオンライン比較	347
11.10	診断イベントの表示	348
11.11	IP アドレスと時刻の設定	348
11.12	工場出荷時設定にリセット	349
11.13	ファームウェアの更新	350
11.14	オンライン CPU に IP アドレスのダウンロード	351
11.15	「未指定 CPU」を使用したハードウェアコンフィグレーションのアップロード	340
11.16	RUN モードでダウンロード	353
11.16.1	RUN モードでプログラムの変更	354
11.17	トリガ条件を使用した CPU データのトレースとレコーディング	355

12	カンタン IO-Link.....	357
12.1	IO-Link テクノロジーの概要	357
12.2	IO-Link システムのコンポーネント	357
12.3	電源投入後	357
12.4	IO-Link プロトコル.....	358
12.5	フィールドバスでの構成	358
12.6	IO-Link と STEP 7 プログラム	358
12.7	SM 1278 4xIO-Link マスタ	359
A	技術仕様	363
A.1	一般技術仕様.....	363
A.2	CPU モジュール.....	373
A.3	デジタル I/O モジュール	377
A.3.1	SB 1221、SB 1222、SB 1223 デジタル入出力(DI、DQ、DI/DQ)	377
A.3.2	SM 1221 デジタル入力(DI)	380
A.3.3	SM 1222 デジタル出力(DQ)	382
A.3.4	SM 1223 V DC デジタル入出力(DI/DQ)	383
A.3.5	SM 1223 120/230V AC 入力/リレー出力	385
A.4	デジタル入力および出力の仕様.....	386
A.4.1	24V DC デジタル入力(DI)	386
A.4.2	120/230V AC デジタル AC 入力	388
A.4.3	デジタル出力(DQ).....	389
A.5	アナログ I/O モジュール	392
A.5.1	SB 1231 および SB 1232 アナログ入力(AI)および出力(AQ)	392
A.5.2	SM 1231 アナログ入力(AI)	393
A.5.3	SM 1232 アナログ出力(AQ).....	393
A.5.4	SM 1234 アナログ入出力(AI/AQ).....	394
A.5.5	SM 1231 (AI)、SM 1232 (AQ)、SM 1234 (AI/AQ)の配線図	394
A.6	BB 1297 バッテリボード	396
A.7	アナログ I/O の仕様	397
A.7.1	アナログ入力(CPU、SM、SB)の仕様	397
A.7.2	電圧および電流の入力(AI)測定範囲	398
A.7.3	アナログ入力(AI)のステップ応答	400
A.7.4	アナログ入力のサンプリング時間と更新時間	400
A.7.5	アナログ出力の仕様.....	401
A.7.6	電圧および電流の出力(AQ)測定範囲	402
A.8	RTD および熱電対モジュール	404
A.8.1	SB 1231 RTD および SB 1231 TC 仕様.....	405
A.8.2	SM 1231 RTD 仕様	407
A.8.3	SM 1231 TC 仕様.....	409
A.8.4	RTD および TC (SM および SB)のアナログ入力仕様.....	410
A.8.5	熱電対タイプ	412
A.8.6	熱電対フィルタの選択と更新時間	413
A.8.7	RTD センサタイプ選択表	413
A.8.8	RTD フィルタの選択と更新時間.....	415

A.9	通信インターフェース	416
A.9.1	PROFIBUS マスタ/スレーブ	416
A.9.1.1	CM 1242-5 PROFIBUS DP スレーブ	416
A.9.1.2	CM 1242-5 の D-sub ソケットのピンアウト	417
A.9.1.3	CM 1243-5 PROFIBUS DP マスタ	418
A.9.1.4	PROFIBUS マスタ(CM 1243-5)には CPU からの 24V DC 電源が必要	419
A.9.1.5	CM 1243-5 の D-sub ソケットのピンアウト	420
A.9.2	GPRS CP	421
A.9.2.1	CP 1242-7 GPRS	421
A.9.2.2	GSM/GPRS アンテナ ANT794-4MR	423
A.9.2.3	平面アンテナ ANT794-3M	424
A.9.3	TeleService (TS)	424
A.9.4	RS485、RS232、RS422 通信	425
A.9.4.1	CB 1241 RS485 仕様	425
A.9.4.2	CM 1241 RS422/485 仕様	427
A.9.4.3	CM 1241 RS232 仕様	428
A.10	テクノロジーモジュール	430
A.10.1	SM 1278 4xIO-Link マスタ SM	430
A.10.1.1	SM 1278 4xIO-Link マスタ シグナルモジュール仕様	430
A.10.1.2	SM 1278 4xIO-Link マスタ SM 配線図	432
A.11	コンパニオン製品	433
A.11.1	PM 1207 電源モジュール	433
A.11.2	CSM 1277 コンパクトスイッチモジュール	433
A.11.3	CM CANopen モジュール	434
B	V3.0 CPU を V4.1 CPU に交換	435
B.1	V3.0 CPU を V4.1 CPU に交換	435
	索引	441

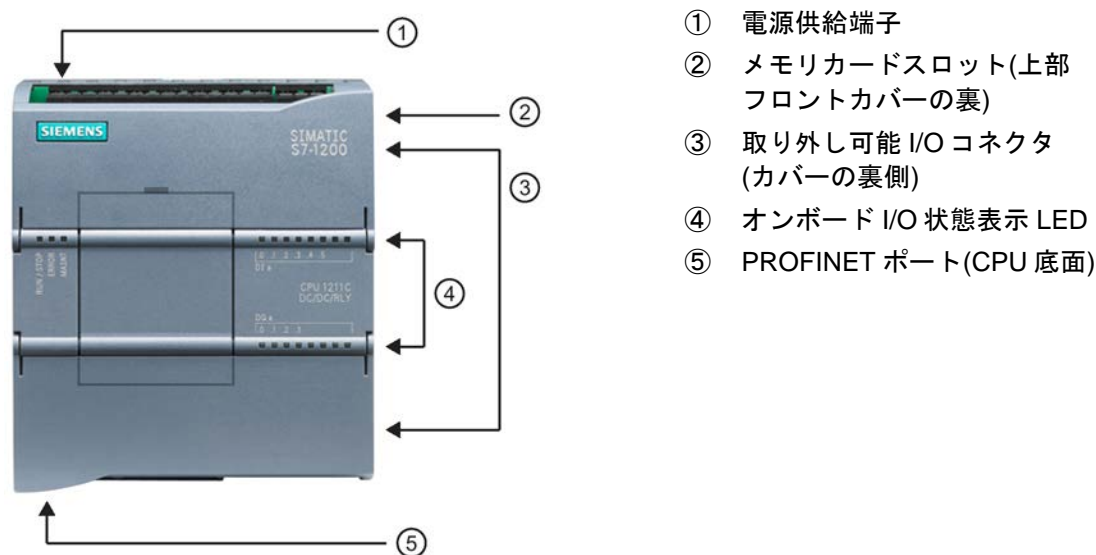
パワフルで使いやすい S7-1200

1.1 S7-1200 PLC について

S7-1200 コントローラには、オートメーションをサポートするさまざまなデバイスの制御に必要な柔軟性と能力が備わっています。コンパクトなデザイン、柔軟な構成、幅広い命令セットの組み合わせを備えた S7-1200 は、さまざまなアプリケーションの制御に最適です。

CPU は、マイクロプロセッサ、内蔵電源、入力および出力回路、内蔵 PROFINET、高速モーションコントロール I/O、オンボードアナログ入力をコンパクトなハウジングにまとめたパワフルなコントローラになっています。プログラムをダウンロードすると、アプリケーションで使用されているデバイスの監視および制御に必要なロジックが CPU に保存されます。CPU によって入力の監視が行われ、ユーザープログラムに基づいて出力されます。ビット演算、カウント、タイミング、複雑な数学演算、インテリジェント機能を備えた機器との通信を行うことができます。

CPU には PROFINET ポートが用意されていて、PROFINET 通信を行うことができます。追加モジュールを使用して、PROFIBUS、GPRS、RS485、RS232、IEC、DNP3、および WDC ネットワーク経由で通信を行うことができます。



CPU および制御プログラムへのアクセスを保護する複数のセキュリティ機能が備わっています。

- 「パスワード保護」機能(89 ページ)を備え、CPU の各ファンクションにアクセスするためのアクセス権限を設定することができます。
- 「ノウハウ保護」機能(91 ページ)を使用して、特定のブロック内のプログラムを非表示にすることができます。
- 「コピー保護」機能(92 ページ)を使用して、プログラムを特定のメモリカードまたは CPU にバインドすることができます。

1.1 S7-1200 PLC について

表 1-1 CPU モジュールの比較

特徴		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C	
外形寸法(mm)		90 x 100 x 75			110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
ユーザーメモリ	ワーク	50 KB	75 KB	100 KB	125 KB	150 KB	
	ロード	1MB		4 MB			
	保持	10 KB					
オンボード I/O	デジタル	6 入力 / 4 出力	8 入力 / 6 出力	14 入力 / 10 出力			
	アナログ	2 入力			2 入力 / 2 出力		
プロセスイメージ サイズ	入力(I)	1024 バイト					
	出力(Q)	1024 バイト					
ビットメモリ(M)		4096 バイト		8192 バイト			
シグナルモジュール(SM)の拡張		なし	2	8			
シグナルボード(SB)、バッテリー ボード(BB)、または通信ボード (CB)		1					
通信モジュール(CM) (左側での拡張)		3					
高速カウンタ	合計	任意の内蔵または SB 入力を使用するために、最大で 6 を構成					
	1 MHz	-				1b.2～1b.5	
	100/ ¹ 80 kHz	1a.0～1a.5					
	30/ ¹ 20 kHz	--	1a.6～1a.7	1a.6～1a.5		1a.6～1a.1	
	200 kHz ³						
パルス出力 ²	合計	任意の内蔵または SB 入力を使用するために、最大で 4 を構成					
	1 MHz	--				Qa.0～Qa.3	
	100 kHz	Qa.0 to Qa.3				Qa.4～Qb.1	
	20 kHz	--	Qa.4～Qa.5	Qa.4～Qb.		--	
メモ리카ード		SIMATIC メモ리카ード(オプション)					
リアルタイムクロック保持時間		40°C で通常 20 日/最小 12 日(メンテナンスフリーの大容量キャパシタ)					
PROFINET イーサネット通信ポート		1			2		
実数演算実行速度		2.3 μs / 命令					
ビット演算実行速度		0.08 μs / 命令					

¹ HSC が直交位相モード用に構成されている場合、より低い速度が適用されます。

² リレー出力付きの CPU モデルの場合、パルス出力を使用するためにデジタル信号(SB)をインストールする必要があります。

³ SB 1221 DI x 24V DC 200 kHz および SB 1221 DI 4 x 5V DC 200 kHz では、最大 200 kHz が使用可能です。

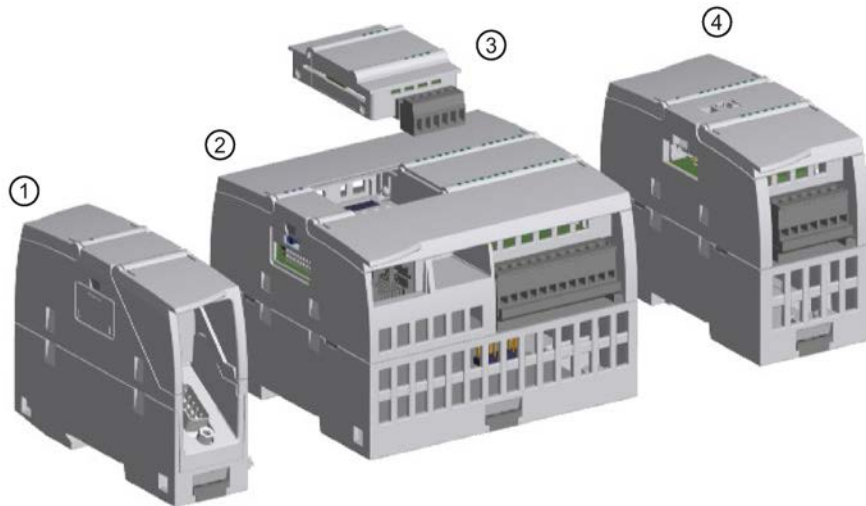
各 CPU モデルには、アプリケーションに対する効果的なソリューションの作成に役立つ機能および能力が備えられています。それぞれの CPU の詳細については、技術仕様(363 ページ)を参照してください。

表 1-2 S7-1200 がサポートするブロック、タイマ、カウンタ

エレメント		説明
ブロック	タイプ	OB、FB、FC、DB
	サイズ	50 K バイト (CPU 1211C) 75 K バイト (CPU 1212C) 100 K バイト (CPU 1214C) 125 K バイト (CPU 1215C) 150 K バイト (CPU 1217C)
	数量	合計で最大 1024 ブロック (OB + FB + FC + DB)
	ネストレベル	プログラムサイクルまたはスタートアップ OB からは 16 レベル; 割り込みイベント OB からは 6 レベル
	モニタリング	2 つのコードブロックのステータスを同時にモニタできます。
OB	プログラムサイクル	複数
	STARTUP	複数
	遅延割り込み	4 (イベントあたり 1)
	周期割り込み	4 (イベントあたり 1)
	ハードウェア割り込み	50 (イベントあたり 1)
	タイムエラー割り込み	1
	診断エラー割り込み	1
	モジュールの取り出しまたはプラグ	1
	ラックまたはステーション障害	1
	時刻	複数
	ステータス	1
	更新	1
	プロファイル	1
タイマ	タイプ	IEC
	数量	メモリサイズによってのみ制限されます
	保存	DB 内の構造体、タイマあたり 16 バイト
カウンタ	タイプ	IEC
	数量	メモリサイズによってのみ制限されます
	保存	DB 内の構造体、サイズはカウントタイプに依存 <ul style="list-style-type: none"> • SInt、USInt: 3 バイト • Int、UInt: 6 バイト • DInt、UDInt: 12 バイト

1.2 CPU の拡張機能

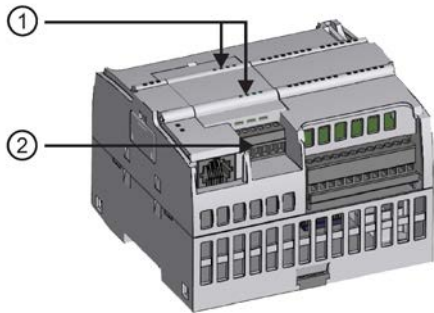
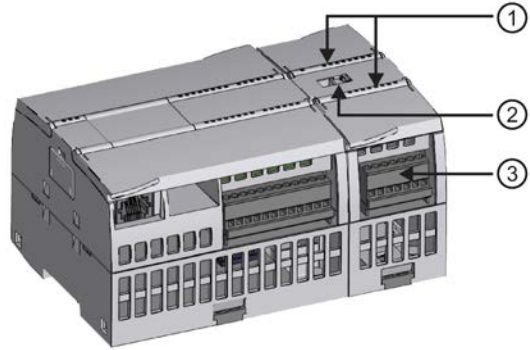
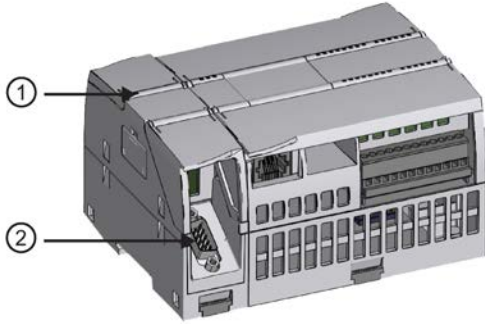
S7-1200 ファミリーには、追加 I/O またはその他の通信プロトコルで CPU の機能を拡張するためのさまざまなシグナルモジュールおよびプラグインボードが用意されています。それぞれのモジュールの詳細については、技術仕様(363 ページ)を参照してください。



- ① 通信モジュール(CM)または通信プロセッサ(CP)
- ② CPU (CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C、CPU 1217C)
- ③ シグナルボード(SB) (デジタル SB、アナログ SB)、通信ボード(CB)、またはバッテリーボード(BB) CPU (CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C、CPU 1217C)
- ④ シグナルモジュール(SM) (デジタル SM、アナログ SM、熱電対 SM、RTD SM、テクノロジー SM)

1.3 S7-1200 モジュール

表 1-3 S7-1200 拡張モジュール

モジュールのタイプ	説明
<p>CPU は 1 つの拡張プラグインボードを取り付けることができます。</p> <ul style="list-style-type: none"> シグナルボード(SB)は CPU に追加 I/O を提供します。SB は CPU の前面に接続します。 通信ボード(CB)を使用して、CPU にもう 1 つ通信ポートを追加することができます。 バッテリーボード(BB)を使用して、リアルタイムクロックを長期間バックアップすることができます。 	 <p>① SB 上のステータス LED</p> <p>② 取り外し可能ユーザー配線コネクタ</p>
<p>シグナルモジュール(SM)は CPU に追加機能を追加します。SM は CPU の右側に接続します。</p> <ul style="list-style-type: none"> デジタル I/O アナログ I/O RTD および熱電対 SM 1278 IO-Link マスタ 	 <p>① ステータス LED</p> <p>② バスコネクタスライドタブ</p> <p>③ 取り外し可能ユーザー配線コネクタ</p>
<p>通信モジュール(CM)および通信プロセッサ(CP)は、PROFIBUS または RS232/RS485 接続用(PtP、Modbus、または USS の場合)、または AS-i マスタ用などの通信オプションを CPU に追加します。</p> <p>CP は、GPRS、IEC、DNP3、WDC ネットワーク経由の CPU への接続などの他のタイプの通信機能を提供します。</p> <ul style="list-style-type: none"> CPU は最大 3 つの CM または CP をサポートします。 各 CM または CP は、CPU (または他の CM または CP)の左側に接続します。 	 <p>① ステータス LED</p> <p>② 通信コネクタ</p>

1.4 HMI ベーシックパネル

SIMATIC HMI ベーシックパネルには、オペレータが基本的な制御および監視に使用するタッチスクリーンデバイスが備わっています。すべてのパネルは保護等級 IP65 に準拠し、CE、UL、cULus、NEMA 4x の認定を取得しています。

使用可能な HMI ベーシックパネルを以下に示します。

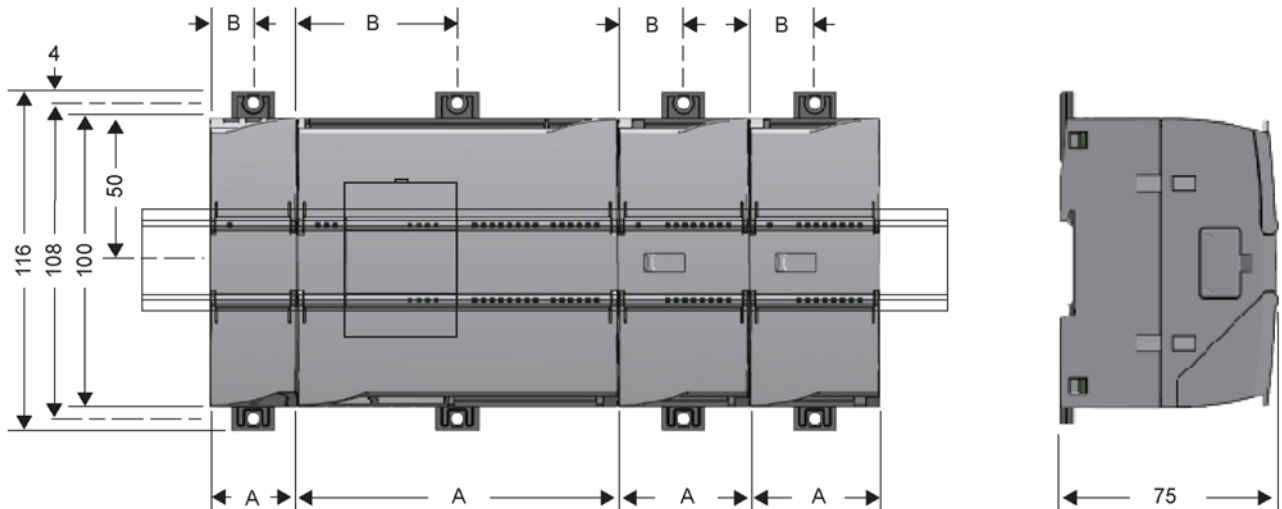
- KTP400 Basic: 4 インチタッチスクリーン、設定可能な 4 つのキー付き、解像度 480 x 272 および 800 タグ
- KTP700 Basic: 7 インチタッチスクリーン、設定可能な 8 つのキー付き、解像度 800 x 480 および 800 タグ
- KTP700 Basic DP: 7 インチタッチスクリーン、設定可能な 8 つのキー付き、解像度 800 x 480 および 800 タグ
- KTP900 Basic: 9 インチタッチスクリーン、設定可能な 8 つのキー付き、解像度 800 x 480 および 800 タグ
- KTP1200 Basic: 12 インチタッチスクリーン、設定可能な 10 のキー付き、解像度 800 x 480 および 800 タグ
- KTP 1200 Basic DP: 12 インチタッチスクリーン、設定可能な 10 のキー付き、解像度 800 x 400 および 800 タグ

1.5 取り付け寸法とクリアランス要件

S7-1200 PLC は簡単に設置できるように設計されています。パネルに取り付けても標準 DIN レールに取り付けても、コンパクトなサイズによりスペースを有効に使用できます。

設置のための特定の要件とガイドラインについては、『S7-1200 プログラマブルコントローラシステム マニュアル』を参照してください。

CPU 1211C、CPU 1212C、CPU 1214C
(寸法 単位: mm)



CPU 1215C、CPU 1217C

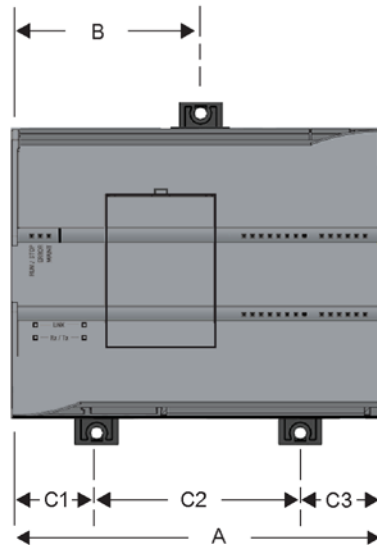


表 1-4 取り付け寸法(mm)

S7-1200 機器		幅 A (mm)	幅 B (mm)	幅 C (mm)
CPU	CPU 1211C および CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (上面)	底面: C1: 32.5 C2: 65 C3: 32.5

1.5 取り付け寸法とクリアランス要件

S7-1200 機器		幅 A (mm)	幅 B (mm)	幅 C (mm)
	CPU 1217C	150	75	底面: C1: 37.5 C2: 75 C3: 37.5
シグナル モジュール	デジタル 8 および 16 点 アナログ 2、4、および 8 点 熱電対 4 および 8 点 RTD 4 点 SM 1278 IO Link-マスタ	45	22.5	--
	デジタル DQ 8 x リレー(切り替え)	70	35	--
	アナログ 16 点 RTD 8 点	70	35	--
通信インター フェース	CM 1241 RS232、および CM 1241 RS422/485 CM 1243-5 PROFIBUS マスタおよび CM 1242-5 PROFIBUS スレーブ CM 1242-2 AS-i マスタ CP 1242-7 GPRS V2 CP 1243-7 LTE-EU CP 1243-1 DNP3 CP 1243-1 IEC CP 1243-1 CP1243-1 PCC CP 1243-8 ST7 RF120C	30	15	--
	TS (TeleService)アダプタ IE Advanced ¹ TS (TeleService)アダプタ IE Basic ¹ TS アダプタ TS モジュール	30 30	15 15	-- --

¹ TS (TeleService)アダプタ IE Advanced または IE Basic を取り付ける前に、まず TS アダプタと TS モジュールを接続する必要があります。合計の幅(「幅 A」)は 60 mm です。

それぞれの CPU、SM、CM、CP は DIN レールへの取り付けにもパネルへの取り付けにも対応します。レールに取り付ける場合は、DIN レールクリップでモジュールを固定します。このクリップを引き出して突き出した位置にすると、ユニットをパネルに直付けするためのネジ取り付け位置になります。デバイス上の DIN クリップの取り付け穴の内寸は 4.3 mm です。

自然空冷による冷却のために、ユニットの上下に 25 mm の空間を確保する必要があります。

S7-1200 機器は簡単に設置できるように設計されています。S7-1200 は、パネルまたは標準レールに取り付けることができます。さらに S7-1200 は、縦にも横にも取り付けすることができます。S7-1200 は小型のため、スペースを効率良く使用することができます。

S7-1200 フェールセーフ CPU は、PROFIBUS または PROFINET リモートフェールセーフ I/O をサポートしていません。


電気機器規格では、SIMATIC S7-1200 システムはオープン機器として分類されています。S7-1200 は、ハウジング、キャビネット、電気制御室内に格納する必要があります。ハウジング、キャビネット、電気制御室内への格納は有資格者が行う必要があります。

S7-1200 は湿気のない環境に設置してください。SELV/PELV 回路は湿気のない場所では、感電に対する保護策になると思われます。

適用される電気規約および建築規約に従った特定のロケーションカテゴリで、オープン機器に認定された機械的および環境的保護対策が実現されるよう設置を行ってください。

埃、湿気、大気汚染による通電汚染により、PLC に動作的小および電氣的な障害が発生するおそれがあります。

通電汚染が存在すると思われる領域に PLC を配置する場合、適切な保護等級の筐体で PLC を保護する必要があります。IP54 は汚染環境での電気機器の筐体に通常使用される等級の 1 つであり、お客様のアプリケーションに適切であると思われます。

<p> 警告</p> <p>S7-1200 を正しく設置しない場合、電氣的障害や予測しない機械の動作が生じるおそれがあります。</p> <p>電氣的障害や予測しない機械の動作によって、死傷事故などの重大な人的傷害や物的損害が発生するおそれがあります。</p> <p>機器の安全な動作を保証するために、適切な動作環境の設置とメンテナンスのためのすべての指示に従ってください。</p>

S7-1200 機器を高温、高電圧、電氣的ノイズから避けて設置する

システムを構成する各装置を設置する際の一般的なルールとして、高電圧および大きな電気ノイズを発生する装置は、S7-1200 のような低電圧の論理装置から必ず離すようにします。

パネル内に S7-1200 を配置するとき、熱が発生するデバイスを考慮して、キャビネット内のなるべく低温の領域に、電子機器を配置するようにします。高温の環境にさらさないようにすると、電子機器の寿命が延びます。

パネル内の配線の引き回しについても考慮します。AC 電源配線やスイッチング周波数の高い DC 配線と同じトレイに、低電圧信号の配線や通信ケーブルを収容することは避けるようにします。

冷却および配線に必要な空間を確保する

S7-1200 の機器は自然空冷方式です。適切な冷却が行われるように、この機器の上下に 25 mm 以上の空間を確保する必要があります。また、このモジュールの前面と筐体の内側との空間を 25 mm 以上確保する必要があります。

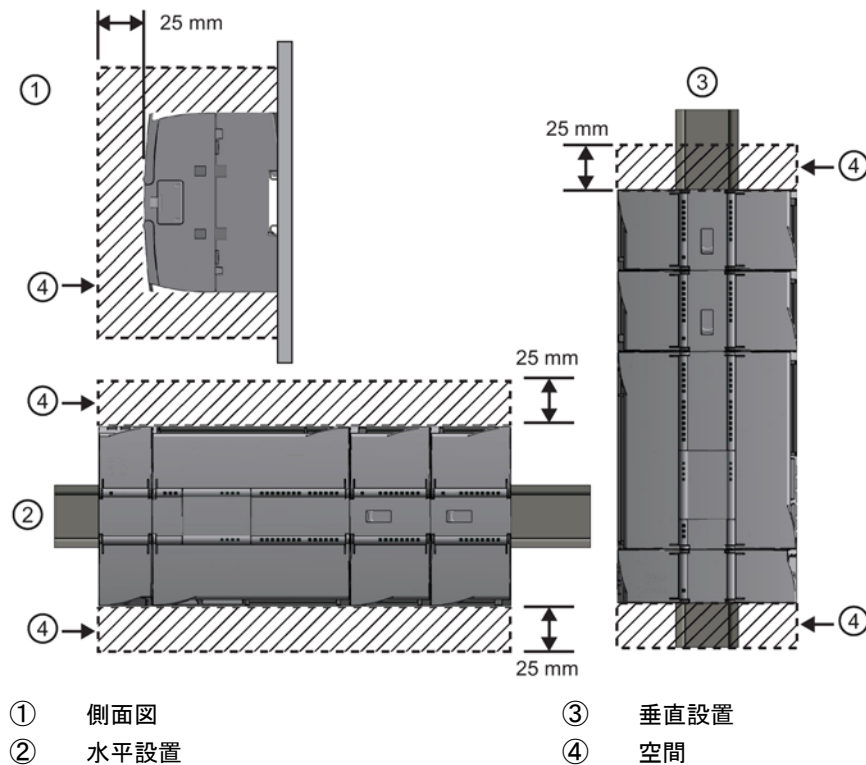
▲ 注意

垂直設置の場合、最大許容周囲温度は 10°C 下がります。

上下に並べて取付ける S7-1200 システムは、下の図に示すような向きにしてください。

S7-1200 システムが正しく取り付けられていることを確認してください。

S7-1200 システムのレイアウトを計画するとき、配線の引き回しおよび通信ケーブルの接続に必要な空間を十分に確保します。



**警告**

S7-1200 および電源が供給されている関連機器の取り付けおよび取り外しによって、感電または予測できない装置の動作の原因になることがあります。

取り付けおよび取り外し作業中は、S7-1200 および関連機器のすべての電源をオフにしないと、感電または予測できない装置の動作によって、死傷事故などの重大な人的傷害や物的損害が発生する可能性があります。

安全上の注意事項を遵守し、S7-1200 CPU または関連機器の取り付けおよび取り外しを行う前に、S7-1200 の電源を必ずオフにしてください。

S7-1200 デバイスの交換または取り付けを行うときは、適正なモジュールまたは等価のデバイスを必ず使用してください。

**警告**

S7-1200 モジュールを正しく取り付けないと、S7-1200 のプログラムに予測できない動作が発生することがあります。

交換した S7-1200 機器のモデル、方向、順序が元のものとは異なると、予測できない装置の動作によって、死傷事故などの重大な人的傷害や物的損害が発生する可能性があります。

S7-1200 機器を交換する場合は、元のモデルと同じモデルを使用し、正しい向きと正しい位置に取り付けてください。

1.6 新機能

このリリースでは以下の機能が新しくなっています。

- S7-1200 フェールセーフ CPU とシグナルモジュール(SM)のハードウェアおよびファームウェアと、ソフトウェア(ES)でダウンロードした安全プログラムを組み合わせ使用し、安全機能を実装できるようになりました。詳細は、『S7-1200 機能安全マニュアル』(<http://support.automation.siemens.com/WW/view/en/104547552>)を参照してください。
- ファームウェアバージョン V4.0 以降による S7-1200 CPU のシミュレーション。
S7 PLCSIM V13 SP1 により、実際のハードウェアを使用する必要なしに、シミュレートされた PLC で PLC プログラムをテストすることができます。S7 PLCSIM は、TIA ポータルで STEP 7 とともに動作する個別にインストールされたアプリケーションです。
PLC とすべての関連するモジュールを STEP 7 で構成し、アプリケーションロジックをプログラミングし、ハードウェアコンフィグレーションとプログラムを S7 PLCSIM にダウンロードできます。これで、S7 PLCSIM のツールを使用して、プログラムのシミュレートとテストが行えます。完全な文書は、S7-PLCSIM のオンラインヘルプを参照してください。フェールセーフ CPU はシミュレーションできません。
- 構成制御(オプション処理) (81 ページ): 操作中に実際は使用しないモジュールを含めた最大の機械構成に合わせてハードウェアを構成することができます。この柔軟なモジュールの構成と指定は、今回の STEP 7 および S7-1200 のリリースの新機能です。このように指定したモジュールは、存在していなくてもエラー状態を発生させません。

1.6 新機能

- Web サーバー (255 ページ)が、S7-1200 CPU の IP アドレス経由だけでなく、ローカル ラック内の選択された(通信プロセッサ)モジュールの IP アドレス経由のアクセスもサポートするようになりました。
- モーション機能の強化
 - アナログおよび PROFIdrive 接続
 - モジュールおよび制御ループ拡張パラメータ
- 高速カウンタ(HSC)(131 ページ)を使用した周期測定
- SCL コンパイラの性能の改良
- パスワードによるプログラムブロックへコピー保護 (92 ページ) 用シリアルバインド
- 共有デバイスのサポートを含めた PROFINET 機能の強化
- 新しいプログラミング命令
 - EQ_Type、NE_Type、EQ_ElemType、NE_ElemType
 - IS_NULL、NOT_NULL
 - IS_ARRAY
 - Deserialize、Serialize
 - VariantGet、VariantPut、CountOfElements
 - Variant_to_DB_Any、DB_Any_To_Variant
 - GET_IM_DATA
 - RUNTIME
 - GEO2LOG、IO2MOD
 - ReadLittle、WriteLittle、ReadBig、WriteBig (SCL のみ)
 - T_RESET、T_DIAG、TMAIL_C
 - PID_Temp
 - 新しい Modbus 命令(190 ページ)
 - 新しいポイントツーポイント(PtP)命令(187 ページ)
 - 新しい USS 命令(188 ページ)

S7-1200 の新しいモジュール

新しいモジュールは S7-1200 CPU の機能が拡張され、お客様のオートメーションニーズに答えた柔軟性が実現されます。

- 産業用リモートコントロール通信モジュール:この CP は通信モジュールとして S7-1200 V4.1 CPU で使用できます。
- フェールセーフ CPU および I/O: S7-1200 V4.1 以降のリリースとともに使用できる 4 つのフェールセーフ CPU と 3 つのフェールセーフシグナルモジュール(SM)があります。

- CPU 1214FC DC/DC/DC (6ES7 214-1AF40-0XB0)
- CPU 1214FC DC/DC/RLY (6ES7 214-1HF40-0XB0)
- CPU 1215FC DC/DC/DC (6ES7 215-1AF40-0XB0)
- CPU 1215FC DC/DC/RLY (6ES7 215-1HF40-0XB0)
- SM 1226 F-DI 16 x 24V DC (6ES7 226-6BA32-0XB0)
- SM 1226 F-DQ 4 x 24V DC (6ES7 226-6DA32-0XB0)
- SM 1226 F-DQ 2 x リレー (6ES7 226-6RA32-0XB0)

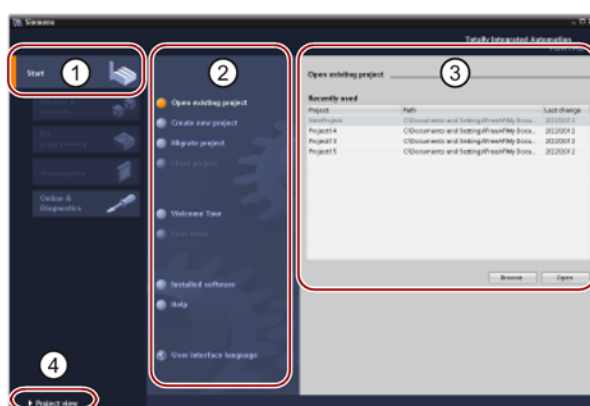
S7-1200 標準シグナルモジュール(SM)、通信モジュール(CM)、シグナルボード(SB)を同じシステムでフェールセーフ SM とともに使用し、機能安全定格を必要としないアプリケーション制御ファンクションを完成させることができます。フェールセーフ SM との併用がサポートされている標準 SM の商品番号は(6ES7 --- ---32 0XB0)以降です。

V3.0 CPU を V4.1 CPU に交換

S7-1200 V3.0 CPU を S7-1200 V4.1 CPU に交換する場合、記載されているバージョンの相違点(435 ページ)と必要なユーザーの操作に注意してください。

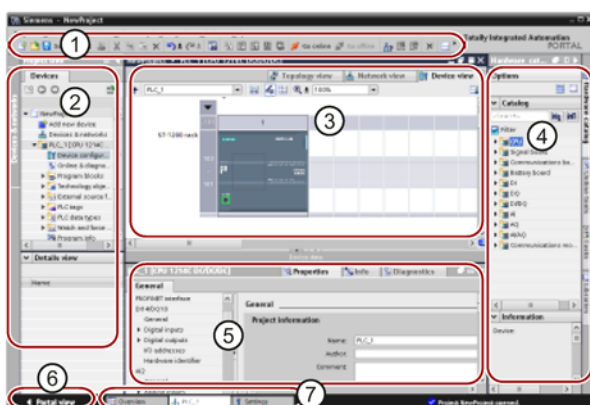
カンタン操作 STEP7

STEP 7は、コントローラロジックの開発、HMI ビジュアルライゼーションの設定、ネットワーク通信のセットアップのためのユーザーフレンドリーな環境を提供します。生産性の向上を助けるために、STEP 7は2種類のビューを用意しています:各ツールがワークフローに基づいて分類されているポータルビュー、またはプロジェクト内の要素で構成されているプロジェクトビューを使用することができます。効率良く作業できる表示方式を選択することができます。ポータルビューとプロジェクトビューをワンクリックで切り替えることができます。



ポータルビュー

- ① さまざまなタスク用のポータル
- ② 選択されたポータルのタスク
- ③ 選択された操作の選択パネル
- ④ プロジェクトビューの変更



プロジェクトビュー

- ① メニューとツールバー
- ② プロジェクトナビゲータ
- ③ ワークエリア
- ④ タスクカード
- ⑤ インспекタウィンドウ
- ⑥ ポータルビューの変更
- ⑦ エディタバー

すべてのコンポーネントが1つの画面に表示されるため、プロジェクトのどの要素にも簡単にアクセスすることができます。たとえば、インспекタウィンドウにはワークエリアで選択したオブジェクトのプロパティと情報が表示されます。異なったオブジェクトを選択すると、インспекタウィンドウには設定可能なプロパティが表示されます。インспекタウィンドウには、診断情報およびその他のメッセージを表示するためのタブが含まれています。

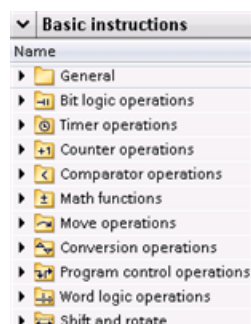
開いているすべてのエディタを表示することで、エディタバーはユーザーがより迅速で効率的に作業できるようにします。開いているエディタ間を切り替えるには、各エディタをクリックするだけです。2つのエディタを同時に配置することもできます(左右にでも上下にでも配置可能)。この機能を使用して、エディタ間でドラッグ&ドロップが行えます。

2.1 ユーザープログラムに命令を簡単挿入

STEP 7には、プログラムの命令の入ったタスクカードが備わっています。命令はファンクションに従ってグループ化されています。



プログラムを作成するには、タスクカードからネットワークに命令をドラッグします。



2.2 ツールバーからよく使用する命令に簡単アクセス

STEP 7には、よく使用する命令にすぐにアクセスするための「お気に入り」ツールバーがあります。命令のアイコンをクリックするだけで、ネットワークに挿入することができます！



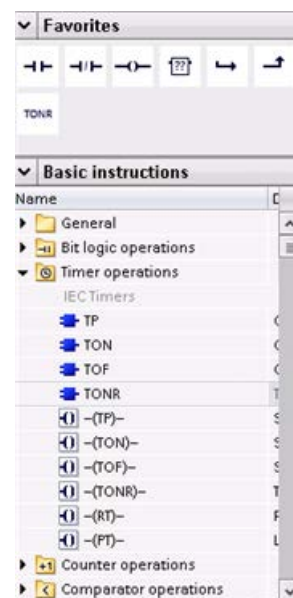
(命令ツリーの「お気に入り」の場合は、アイコンをダブルクリックします。)



新しい命令を追加して、「お気に入り」を簡単にカスタマイズできます。

命令を「お気に入り」にドラッグ&ドロップするだけです。

これで、クリックするだけで命令を実行できます！



2.3 LAD および FBD 命令に入力または出力を簡単追加

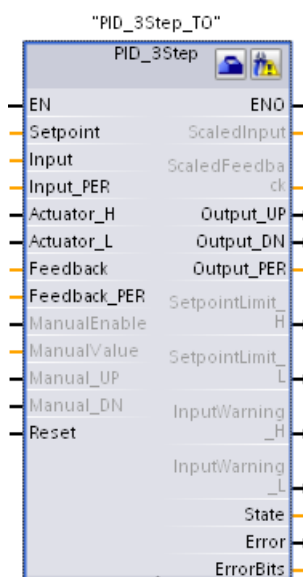
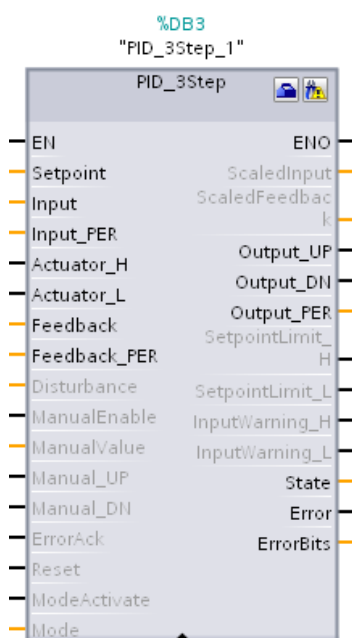


追加の入力または出力の作成を可能にする命令もあります。

- 入力または出力を追加するには、[作成]アイコンをクリックするか、既存の IN または OUT パラメータの入カスタブを右クリックし、[入力の挿入]コマンドを選択します。
- 入力または出力を削除するには、既存の IN または OUT パラメータのスタブを右クリックし(元の 2 つの入力以外に追加した入力が存在する場合)、[削除]コマンドを選択します。

2.4 拡張可能な命令

より複雑な命令のなかには、拡張可能で主要な入力および出力のみを表示するものもあります。すべての入力および出力を表示するには、命令の一番下にある矢印をクリックします。



2.5 CPU の動作モードを簡単変更

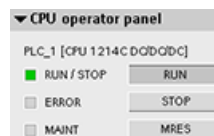
CPU には動作モード(STOP または RUN)を変更するための物理的なスイッチがありません。

CPU の動作モードを変更するには、[CPU のスタート]および[CPU のストップ] ツールバーボタンを使用します。



デバイスコンフィグレーションで CPU を構成定義するときに、CPU のプロパティのスタートアップ動作(82 ページ)を設定します。

「オンラインおよび診断」ポータルにも、オンライン CPU の動作モードを変更するためのオペレータパネルがあります。CPU オペレータパネルを使用するには、CPU へのオンライン接続が必要です。[オンラインツール]タスクカードには、オンライン CPU の動作モードを表示するオペレータパネルが表示されます。オペレータパネルで、オンライン CPU の動作モードも変更できます。



動作モード(STOP または RUN)を変更するには、オペレータパネルのボタンを使用します。オペレータパネルには、メモリのリセットのための MRES ボタンもあります。

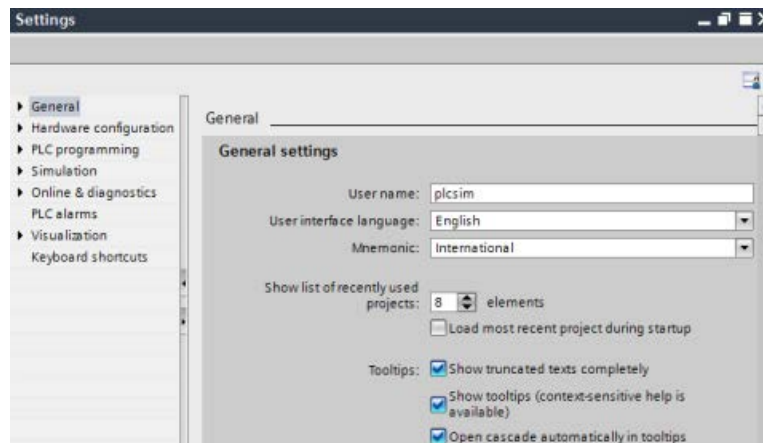
RUN/STOP インジケータの色は、CPU の現在の動作モードを示します。黄色は STOP モード、緑色は RUN モードを示します。

STEP 7 のデバイスコンフィグレーションから、CPU の電源投入時のデフォルトの動作モードを設定することもできます。

2.6 STEP 7 の外観と構成を簡単変更

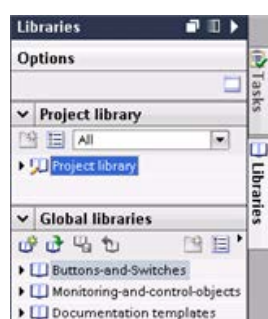
インターフェースの外観、言語、作業の保存用フォルダなどのさまざまな設定を選択することができます。

上記の設定を変更するには、[オプション]メニューから[設定]コマンドを選択します。



2.7 プロジェクトライブラリおよびグローバルライブラリに簡単アクセス

グローバルライブラリおよびプロジェクトライブラリにより、保存されたオブジェクトを1つのプロジェクト全体で、または複数のプロジェクト間で再使用することができます。たとえば、異なったプロジェクトで使用するためのブロックテンプレートを作成し、オートメーションタスクの特定の要件に合わせて調整することができます。FC、FB、DB、デバイスコンフィグレーション、データタイプ、ウォッチテーブル、プロセス画面、フェースプレートなどのさまざまなオブジェクトをライブラリに保存できます。HMI デバイスのコンポーネントをプロジェクトに保存することもできます。

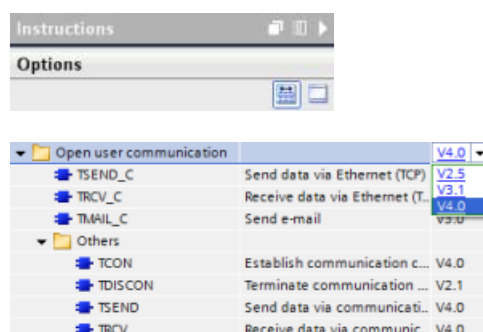


各プロジェクトには、プロジェクト内で複数回使用するオブジェクトを保存するためのプロジェクトライブラリがあります。このプロジェクトライブラリはプロジェクトの一部です。プロジェクトを開いたり閉じたりすると、プロジェクトライブラリも開かれたり閉じられたりし、プロジェクトを保存すると、プロジェクトライブラリ内のすべての変更が保存されます。

独自のグローバルライブラリを作成して、他のプロジェクトで使用したいオブジェクトを保存することができます。新しいグローバルライブラリを作成するときに、このライブラリをコンピュータまたはネットワーク上の場所に保存します。

2.8 命令のバージョンを簡単選択

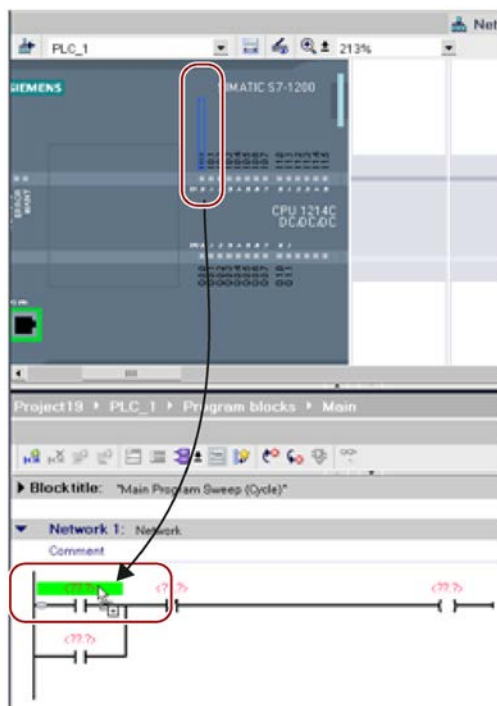
特定の命令 (Modbus、PID、モーションなど)は、開発サイクルおよびリリースサイクルにより、これらの命令の複数のリリース済みバージョンが作成されます。古いプロジェクトとの互換性と移行性を保証しやすくするために、STEP 7 ではユーザープログラムに挿入する命令のバージョンを選択することができます。



命令ツリータスクカードのアイコンをクリックして、命令ツリーのヘッダーと列を有効にします。

命令のバージョンを変更するには、ドロップダウンリストから該当するバージョンを選択します。

2.9 エディタ間で簡単ドラッグ&ドロップ



タスクを迅速かつ簡単に行いやすくするために、STEP 7ではエディタから別のエディタにエレメントをドラッグ&ドロップすることができます。たとえば、CPUからの入力をユーザープログラムの命令のアドレスにドラッグすることができます。

CPUの入力または出力を選択するには、最低でも 200%のズームインが必要です。

タグ名は PLC タグテーブルだけでなく、CPU にも表示されます。

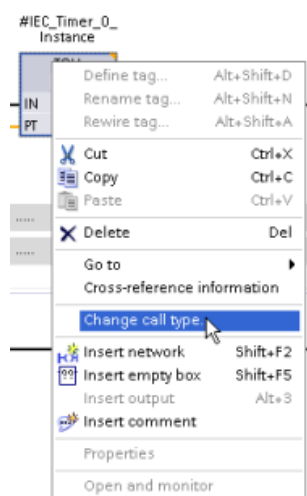
2つのエディタを一度に表示するには、[エディタスペースの分割]メニューコマンドまたはツールバーのボタンを使用します。



開いているエディタを切り替えるには、エディタバーのアイコンをクリックします。



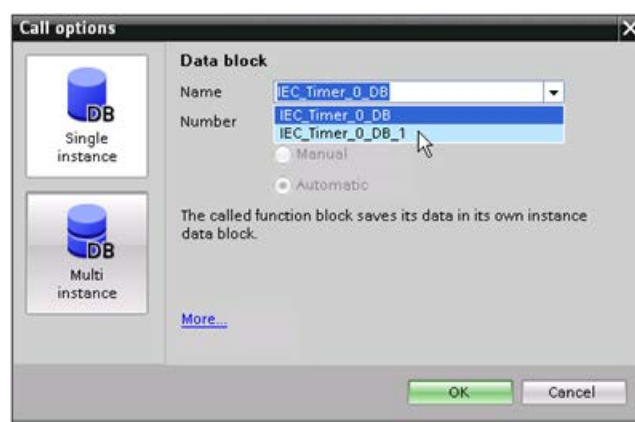
2.10 DB の呼び出しタイプの変更



STEP 7 では、FB 内の命令または FB に対する DB の関連を簡単に作成または変更できます。

- 異なった DB 間の関連を切り替えることができます。
- シングルインスタンス DB とマルチインスタンス DB 間の関連を切り替えることができます。
- インスタンス DB を作成することができます(インスタンス DB が失われているか使用できない場合)。

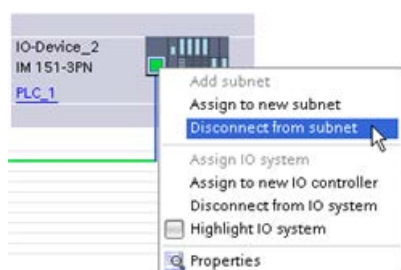
プログラミングエディタで命令または FB を右クリックするか、[オプション]メニューから[ブロック呼び出し]コマンドを選択して、[呼び出しタイプの変更]コマンドにアクセスできます。



[呼び出しオプション]ダイアログで、シングルインスタンス DB またはマルチインスタンス DB を選択できます。使用可能な DB のドロップダウンリストから特定の DB を選択することもできます。

2.11 ネットワークからデバイスを一時的に切断

個々のネットワークデバイスをサブネットから切断することができます。デバイスの構成はプロジェクトから削除されないため、デバイスへの接続を簡単に復元できます。



ネットワークデバイスのインターフェースポートを右クリックし、コンテキストメニューから[サブネットからの切断]コマンドを選択します。

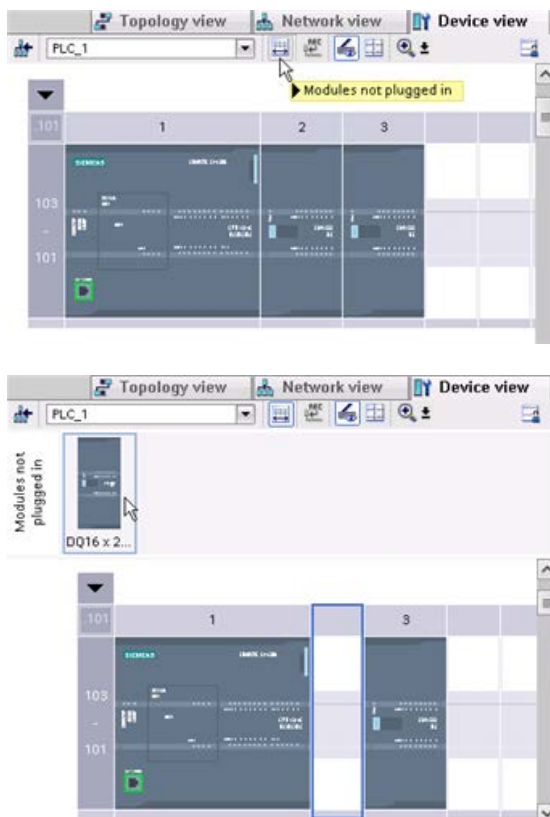
STEP 7はネットワーク接続を再構成しますが、切断されたデバイスをプロジェクトから削除しません。ネットワーク接続は削除されますが、インターフェースアドレスは変更されません。



新しいネットワーク接続をダウンロードする場合は、CPU を STOP モードに設定する必要があります。

デバイスを再接続するには、デバイスのポートへの新しいネットワーク接続を作成するだけです。

2.12 コンフィグレーションを失わずに簡単にバーチャルでモジュールを「アンプラグ」



STEP 7 は「アンプラグ」モジュール用の記憶領域を提供します。ラックからモジュールをドラッグして、そのモジュールの構成を保存できます。アンプラグモジュールはプロジェクトとともに保存され、将来、パラメータを再設定する必要なしにモジュールを再挿入することができます。

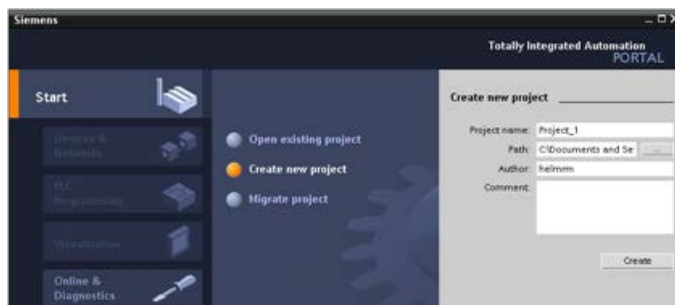
この機能の用途の 1 つは、一時的なメンテナンスです。交換用モジュールが届くのを待っていて、別のモジュールを短期間の代替品として一時的に使用する状況を考えてみてください。構成済みのモジュールをラックから「アンプラグされたモジュール」にドラッグし、一時モジュールを挿入します。

2.12 コンフィグレーションを失わずに簡単にバーチャルでモジュールを「アンプラグ」

ゲッティング・スタート

3.1 プロジェクトの作成

STEP 7 の操作は簡単です! どれほど簡単にプロジェクトの作成を開始できるかを見てください。



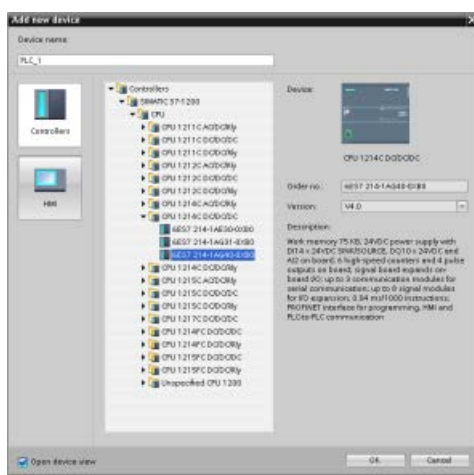
「スタート」ポータルで、[新規プロジェクトの作成]タスクをクリックします。

プロジェクト名を入力し、[作成]ボタンをクリックします。



プロジェクトを作成したら、「デバイス&ネットワーク」ポータルを選択します。

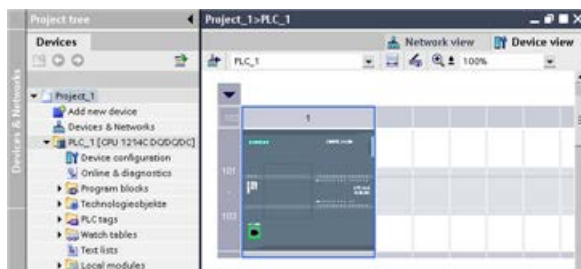
[新しいデバイスの追加]タスクをクリックします。



プロジェクトに追加する CPU を選択します。

1. [新しいデバイスの追加]ダイアログで、[コントローラ]ボタンをクリックします。
2. リストから CPU を選択します。
3. 選択された CPU をプロジェクトに追加するには、[追加]ボタンをクリックします。

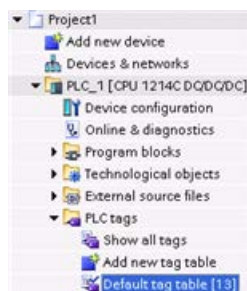
[デバイスビューを開く]オプションが選択されていることに注意してください。このオプションが選択された状態で[追加]をクリックすると、プロジェクトビューの「デバイスコンフィグレーション」が開きます。



デバイスビューに追加した CPU が表示されます。

3.2 CPU の I/O タグの作成

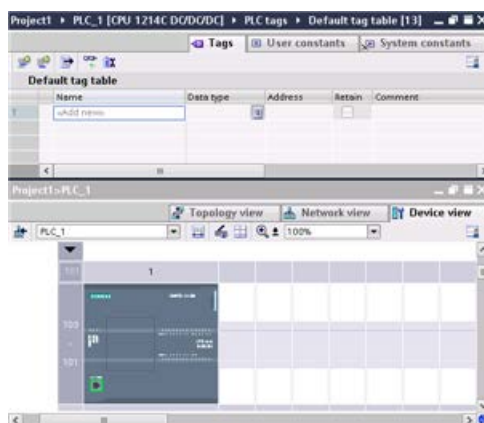
「PLC タグ」は I/O およびアドレスのシンボル名です。PLC タグを作成すると、STEP 7 はそのタグをタグテーブルに保存します。プロジェクト内のすべてのエディタ(プログラミングエディタ、デバイスエディタ、ビジュアルイゼーションエディタ、ウォッチテーブルエディタなど)がタグテーブルにアクセスできます。



デバイスエディタを開いた状態で、タグテーブルを開きます。
エディタバーに開いたエディタが表示されます。



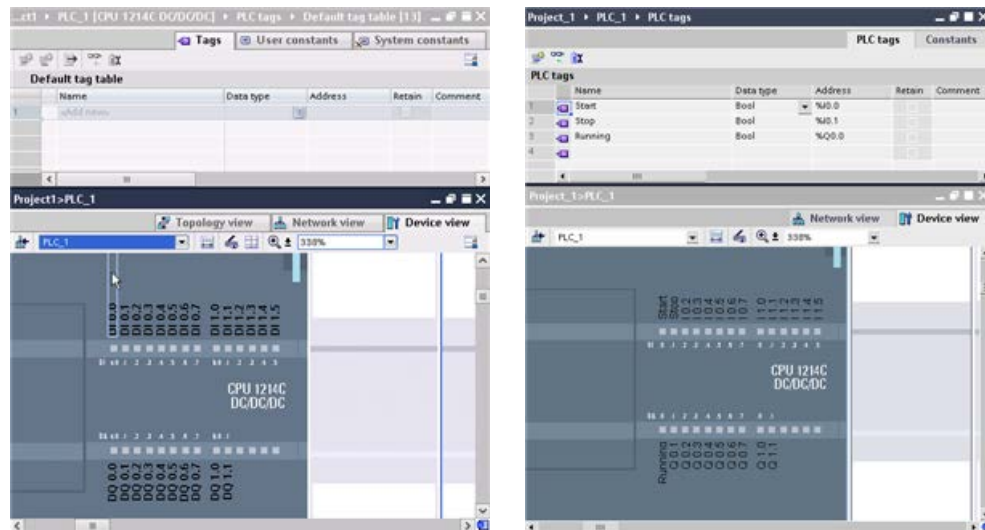
ツールバーで、[エディタスペースを上下に分割]ボタンをクリックします。



STEP 7 はタグテーブルとデバイスエディタの両方を一緒に表示します。

CPU の I/O 点を読み取って選択ができるよう、デバイスコンフィグレーションを 200%以上にズームします。入力と出力を CPU からタグテーブルにドラッグします。

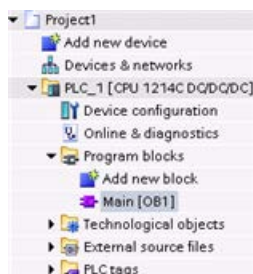
1. I0.0 を選択し、タグテーブルの最初の列にドラッグします。
2. タグ名を「I0.0」から「Start」に変更します。
3. I0.1 をタグテーブルにドラッグし、名前を「Stop」に変更します。
4. Q0.0 (CPU の一番下)をタグテーブルにドラッグし、名前を「Running」に変更します。



タグを PLC タグテーブルに入力すると、タグがユーザープログラムで使用可能になります。

3.3 ユーザープログラムで簡単ネットワーク作成

プログラムコードはCPUが順番に実行する命令で構成されています。この例では、ラダーロジック(LAD)を使用してプログラムコードを作成します。LAD プログラムは、はしごの段に類似しているネットワークのシーケンスです。

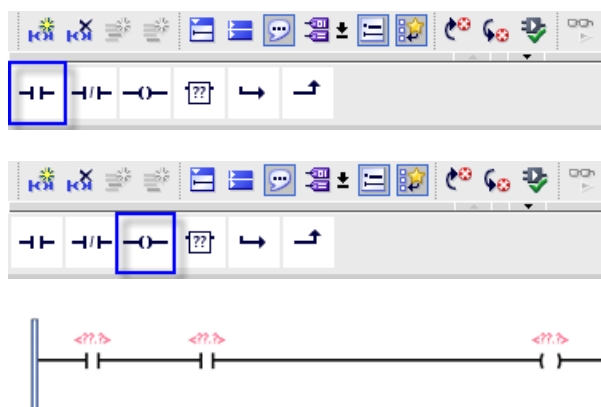


プログラミングエディタを開くには、以下の手順に従います。

1. プロジェクトツリーで「プログラムブロック」フォルダを展開し、「Main [OB1]」ブロックを表示します。
2. 「Main [OB1]」ブロックをダブルクリックします。

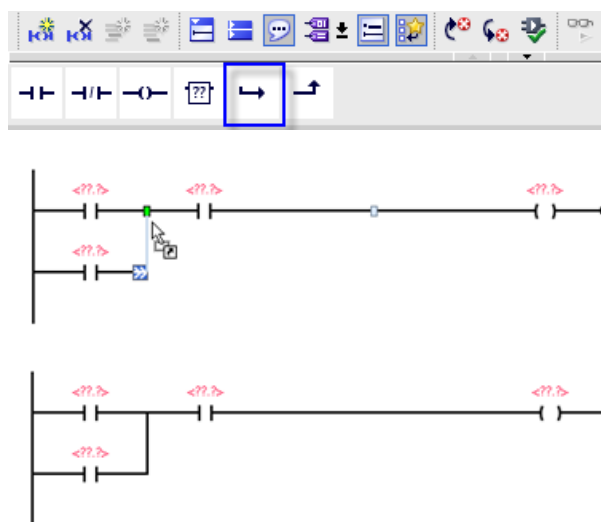
プログラミングエディタがプログラムブロック(OB1)を開きます。

[お気に入り]のボタンを使用して、接点とコイルをネットワークに挿入します。



1. [お気に入り]の[ノーマルオープン]ボタンをクリックして、接点をネットワークに追加します。
2. この例では、2 番目の接点を追加します。
3. [コイルの出力]ボタンをクリックして、コイルを挿入します。

[お気に入り]には、分岐を作成するためのボタンもあります。

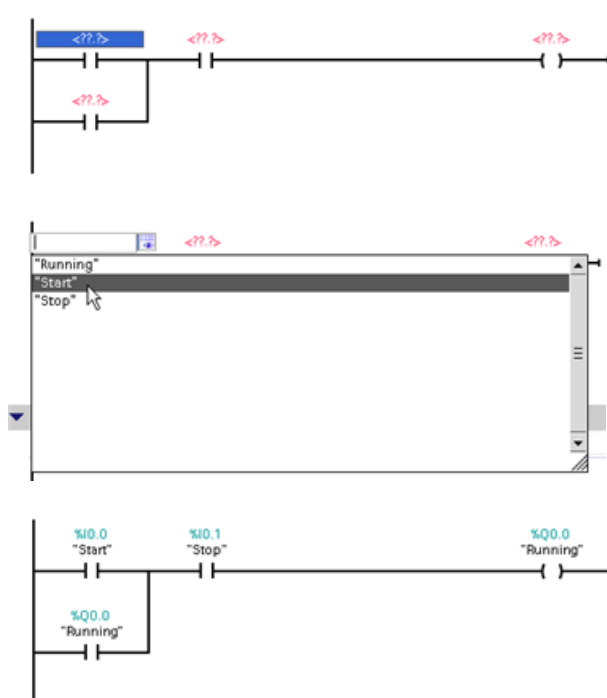


1. 分岐の左側のレールを選択するには、左側のレールを選択します。
2. [分岐を開く]アイコンをクリックして、分岐をネットワークのレールに追加します。
3. さらにもう 1 つのノーマルオープンを開いている分岐に挿入します。
4. 両矢印を、最初の段の 2 つの接点間の接続点(段上の緑色の四角形)にドラッグします。

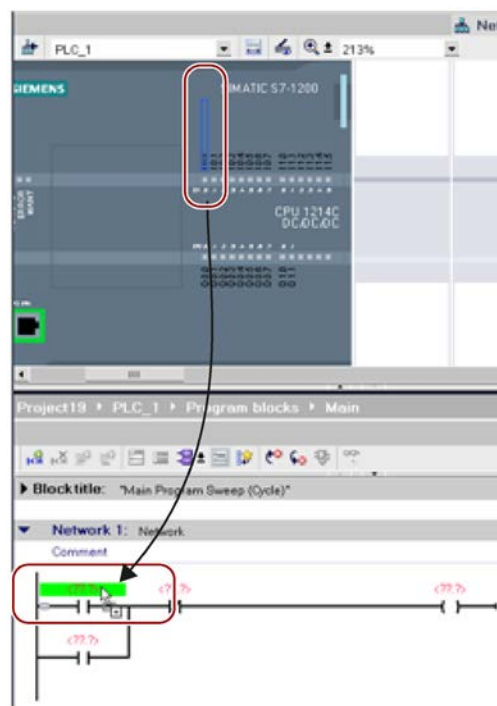
プロジェクトを保存するには、ツールバーの[プロジェクトの保存]ボタンをクリックします。保存する前にラダーの編集を終了する必要はありません。これからタグ名とこれらの命令を関連付けます。

3.4 タグテーブルの PLC タグを使用して命令をアドレス指定

タグテーブルを使用して、接点とコイルのアドレス用の PLC タグを迅速に入力することができます。



1. 1 番目のノーマルオープンの上にあるデフォルトアドレス<??.>をダブルクリックします。
2. アドレスの右側にあるセレクトアイコンをクリックして、タグテーブルのタグを開きます。
3. ドロップダウンリストから、1 番目の接点に[Start]を選択します。
4. 2 番目の接点に対して上記の手順を繰り返し、タグ[Stop]を選択します。
5. コイルとラッチ接点に、タグ[Running]を選択します。



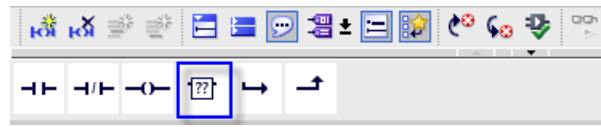
I/O アドレスを CPU から直接ドラッグすることもできます。プロジェクトビューのワークエリアを分割するだけです(36 ページ)。

I/O 点を選択するために、CPU を 200%以上にズームする必要があります。

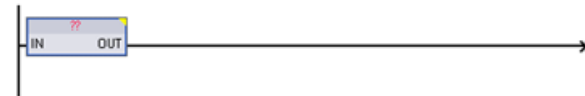
「デバイスコンフィグレーション」の CPU 上の I/O をプログラミングエディタの LAD 命令までドラッグし、命令のアドレスだけでなく、PLC タグテーブルのエントリも作成することができます。

3.5 「空ボックス」 命令の追加

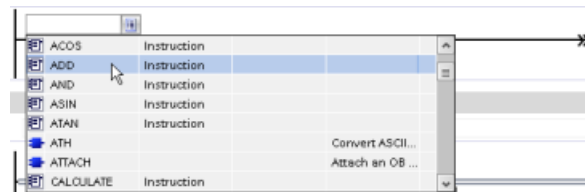
プログラミングエディタは、汎用「空ボックス」命令を特長としています。このボックス命令を挿入した後で、ADD 命令などの命令タイプをドロップダウンリストから選択します。



[お気に入り]ツールバーの汎用「空ボックス」命令をクリックします。

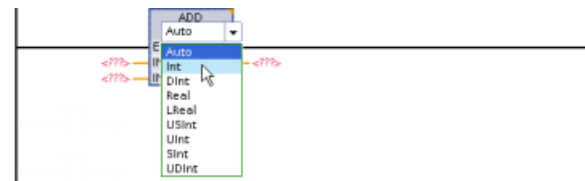


汎用「空ボックス」命令はさまざまな命令に対応しています。この例では、ADD 命令を作成します。



1. ボックス命令の黄色の隅をクリックして、命令のドロップダウンリストを表示します。

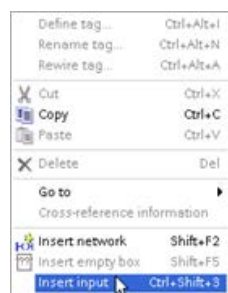
2. リストを下にスクロールして、ADD 命令を選択します。



3. 「?」の横の黄色の隅をクリックして、入力および出力のデータタイプを選択します。



これで、ADD 命令で使用する値のタグ(またはメモリアドレス)を入力できます。



特定の命令に対して追加入力を作成することもできます。

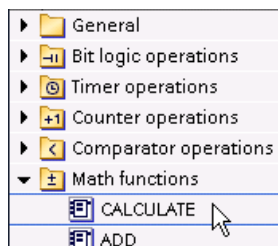
1. ボックス内部の入力のどれかをクリックします。
2. 右クリックしてコンテキストメニューを表示し、[入力の挿入]コマンドを選択します。



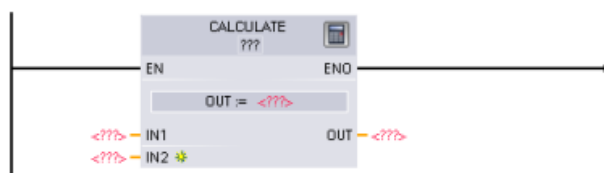
これで、ADD 命令は 3 つの入力を使用できます。

3.6 複雑な数学計算式用の CALCULATE 命令の使用

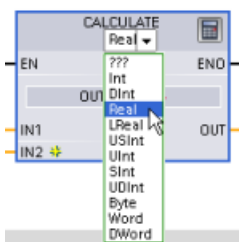
Calculate 命令(113 ページ)を使用して、複数の入力パラメータで動作して定義した計算式に従って結果を生成する演算ファンクションを作成できます。



Basic 命令ツリーで、四則演算ファンクションフォルダを展開します。Calculate 命令をダブルクリックして、命令をユーザープログラムに挿入します。



構成されていない Calculate 命令には、2つの入力パラメータと1つの出力パラメータがあります。

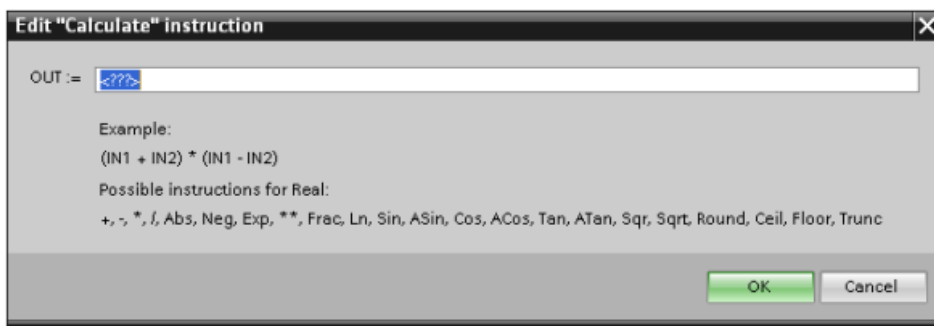


「???」をクリックして、入力および出力パラメータのデータタイプを選択します。(入力および出力パラメータは、すべて同じデータタイプでなければなりません。)

この例では、「Real」データタイプを選択します。



[計算命令の編集]アイコンをクリックして、計算式を入力します。



この例では、アナログ値のスケーリングのために以下の計算式を入力します。(「In」および「Out」指定は Calculate 命令のパラメータに対応しています。)

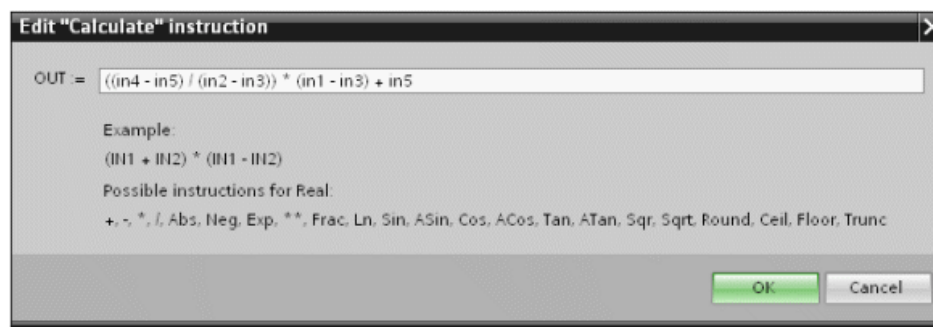
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

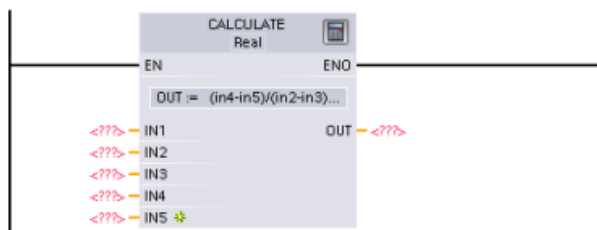
ここで、	Out _{value}	(Out)	スケーリングされた出力値
	In _{value}	(in1)	アナログ入力値
	In _{high}	(in2)	スケーリングされた入力値の上限
	In _{low}	(in3)	スケーリングされた入力値の下限
	Out _{high}	(in4)	スケーリングされた出力値の上限
	Out _{low}	(in5)	スケーリングされた出力値の下限

[計算命令の編集]ボックスに、計算式をパラメータ名とともに入力します。

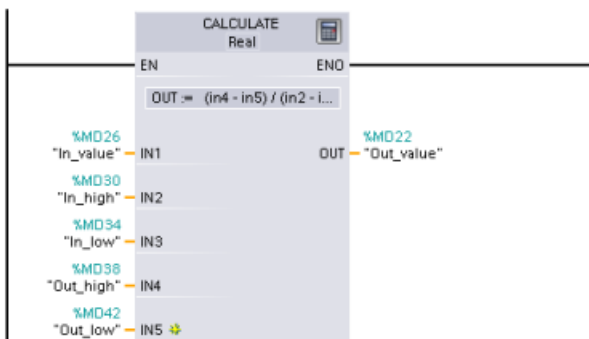
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



[OK]をクリックすると、Calculate 命令が命令に必要な入力を作成します。



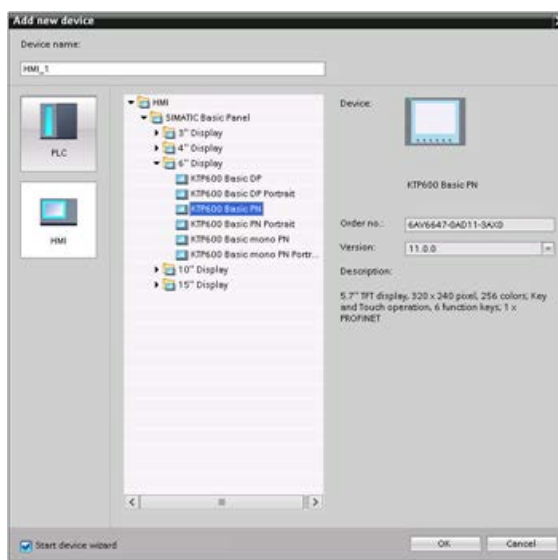
パラメータに対応する値のタグ名を入力します。



3.7 HMI デバイスをプロジェクトに追加



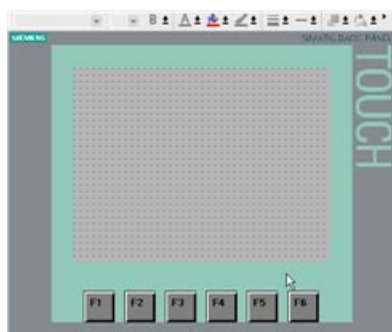
HMI デバイスを簡単にプロジェクトに追加できます!



1. [新しいデバイスの追加]アイコンをダブルクリックします。
2. [新しいデバイスの追加]ダイアログで、[HMI]ボタンをクリックします。
3. リストから特定の HMI デバイスを選択します。

HMI ウィザードの実行を選択して、HMI デバイスの画面を簡単に構成することができます。

4. [OK]をクリックして HMI デバイスをプロジェクトに追加します。

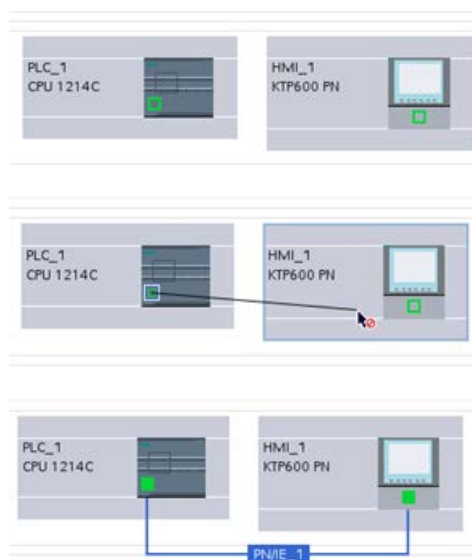


TIA ポータルが HMI デバイスをプロジェクトに追加します。

TIA ポータルは、HMI デバイスのすべての画面と構造を簡単に構成するための HMI ウィザードを提供します。

HMI ウィザードを実行しない場合、TIA ポータルは簡単なデフォルトの HMI 画面を作成します。後から追加の画面またはオブジェクトを追加することができます。

3.8 CPU と HMI デバイス間のネットワーク接続作成

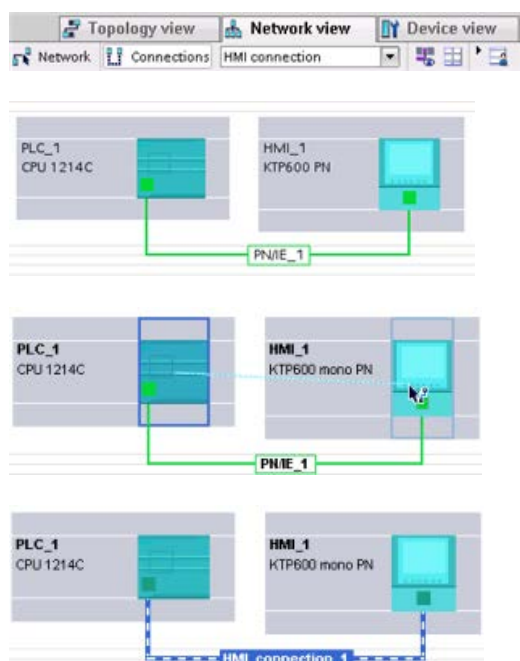


ネットワークを簡単に作成できます!

- [デバイスとネットワーク]にジャンプ後、ネットワークビューを選択してCPUとHMIデバイスを表示します。
- PROFINET ネットワークを作成するには、デバイスの緑色のボックス(イーサネットポート)からもう一方のデバイスの緑色のボックスに線をドラッグします。

2つのデバイスのネットワーク接続が作成されます。

3.9 タグの共有のための HMI 接続作成



2つのデバイス間にHMI接続を作成することで、2つのデバイス間のタグを簡単に共有できます。

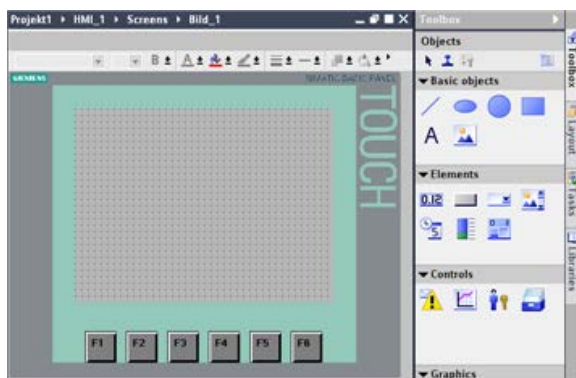
- ネットワーク接続を選択して[接続]ボタンをクリックし、ドロップダウンリストから[HMI 接続]を選択します。
- HMI 接続により2つのデバイスが青色に変わります。
- CPU デバイスを選択し、HMI デバイスまで線をドラッグします。
- HMI 接続により、PLC タグのリストを選択してHMI タグを構成することができます。

これ以外のオプションを使用して HMI 接続を作成することもできます。

- PLC タグテーブル、プログラミングエディタ、またはデバイスコンフィグレーションエディタから PLC タグを HMI 画面エディタまでドラッグすると、自動的に HMI 接続が作成されます。
- PLC の参照用の HMI ウィザードを使用すると、自動的に HMI 接続が作成されます。

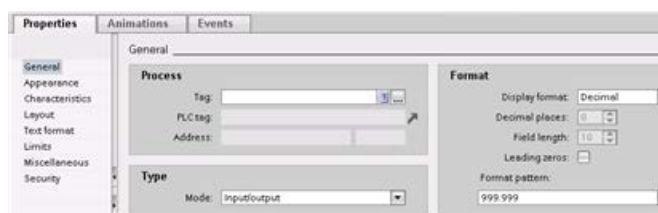
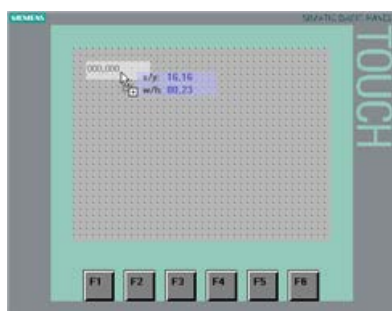
3.10 HMI 画面の作成

HMI ウィザードを使用しなくても、HMI 画面を簡単に構成できます。



STEP 7 には、基本形状、インタラクティブエレメント、さらには標準グラフィックスを挿入するための標準のライブラリセットがあります。

エレメントを追加するには、エレメントを画面にドラッグ＆ドロップするだけです。エレメントのプロパティを使用して(インスペクタウィンドウで)、エレメントの外観と動作を設定します。



プロジェクトツリーまたはプログラミングエディタから PLC タグを HMI 画面にドラッグ＆ドロップして、画面上でエレメントを作成することもできます。PLC タグが画面上でエレメントになります。次に、プロパティを使用してこのエレメントのパラメータを変更します。

3.11 HMI ELEMENTの PLC タグの選択

画面上でELEMENTを作成したら、ELEMENTのプロパティを使用して PLC タグをELEMENTに割り当てます。タグフィールドの横にあるセクタボタンをクリックして、CPUの PLC タグを表示します。



プロジェクトツリーから HMI 画面に PLC タグをドラッグ&ドロップすることもできます。プロジェクトツリーの[詳細]ビューに PLC タグを表示し、タグを HMI 画面にドラッグします。

カンタン PLC コンセプト

4.1 スキャンサイクルごとのタスクの実行

各スキャンサイクルには、出力の書き込み、入力の読み取り、ユーザープログラム命令の実行、およびシステムメンテナンスまたはバックグラウンド処理の実行が含まれます。

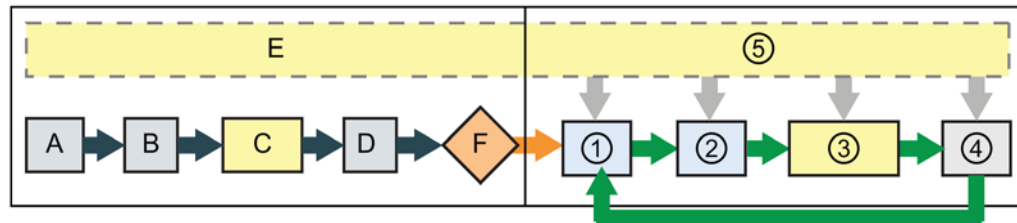


このサイクルは、スキャンサイクルまたは単にスキャンと呼ばれます。デフォルトの条件では、すべてのデジタルおよびアナログの I/O 点は、プロセスイメージと呼ばれる内部メモリ領域を使用して、スキャンサイクルに同期して更新されます。プロセスイメージには CPU、シグナルボード、シグナルモジュールの物理的な入力および出力のスナップショットが含まれます。

- CPU は、ユーザープログラムを実行する直前に物理的な入力から入力を読み取り、その値をプロセスイメージの入力領域に格納します。これによって、ユーザー命令実行中の値の一貫性が保証されます。
- CPU は、ユーザー命令のロジックを実行し、実際の物理的な出力に書き込むのではなく、プロセスイメージの出力領域の出力値を更新します。
- ユーザープログラムの実行後、CPU は、プロセスイメージ出力領域から結果として得られた出力を物理的な出力に書き込みます。

4.1 スキャンサイクルごとのタスクの実行

この処理によって、所定のサイクルのユーザー命令の実行中のロジックの一貫性を保持するとともに、物理的な出力点の変動を防止し、プロセスイメージ出力領域の状態が頻繁に変化するのを防いでいます。



STARTUP

- A I (イメージ)メモリ領域をクリアする
- B Q 出力(イメージ)メモリ領域を設定に従ってゼロ、最後の値、代替値のいずれかで初期化し、PB、PN、よび AS-i 出力をゼロにする
- C 非保持 M メモリおよびデータブロックを初期値に初期化し、設定された周期割り込みと時刻イベントを有効にする
スタートアップ OB を実行する
- D 物理的な入力の状態を I メモリにコピーする
- E 割り込みイベントが発生した場合は、RUN モードで処理するために、キューに格納する
- F 物理的な出力への Q メモリの書き込みを可能にする

RUN

- ① Q メモリを物理的な出力に書き込む
- ② 物理的な入力の状態を I メモリにコピーする
- ③ プログラムサイクル OB を実行する
- ④ 自己診断を実行する
- ⑤ スキャンサイクルの各部で割り込みおよび通信を処理する

モジュールのデフォルトの動作を、上記の I/O の自動更新から削除することで変更できます。また、命令が実行されたときに、即時にデジタルおよびアナログ I/O 値の読み取りと書き込みをモジュールに対して行うことができます。物理的な入力の即時読み取りは、プロセスイメージの入力領域を更新しません。物理的な出力への即時書き込みは、プロセスイメージの出力領域と物理的な出力の出力点の両方を更新します。

4.2 CPU の動作モード

CPU には、3つのオペレーティングモードがあります。「STOP モード」、「STARTUP モード」、「RUN モード」の3種類の動作モードがあります。CPU の前面のステータス LED によって、現在の動作モードが表示されます。

- STOP モードでは、CPU はプログラムを実行していないため、プロジェクトをダウンロードすることができます。RUN/STOP LED は単色の黄色です。
- STARTUP モードでは、CPU はすべてのスタートアップロジックを実行します(存在する場合)。CPU は STARTUP モード中は割り込みイベントを処理しません。RUN/STOP LED は緑色と黄色に交互に点滅します。
- RUN モードでは、スキャンサイクルが反復して実行されます。割り込みイベントが発生し、CPU はプログラムサイクルのどの時点でも割り込みイベントの処理が可能です。RUN モードではプロジェクトの一部をダウンロードすることができます。RUN/STOP LED は単色の緑色です。

CPU はウォームリスタートメソッドによる RUN モードへの移行をサポートしています。ウォームリスタートはメモリリセットを含みませんが、STEP 7 からメモリリセットをコマンドすることができます。メモリリセットは、すべてのワークメモリをクリアし、保持および保持でないメモリ領域をクリアして、ロードメモリをワークメモリにコピーし、出力を設定された「CPU STOP への応答」に設定します。メモリリセットでは、診断バッファおよび IP アドレスの値はクリアされません。ウォームリスタートは保持システムでないすべてのシステムおよびユーザーデータを初期化します。

リスタートメソッドによる CPU の「電源オン後のスタートアップ」設定を STEP 7 を使用して設定することができます。この構成項目は、この CPU の[デバイスコンフィギュレーション]の下[スタートアップ]の下に表示されます。電源を投入すると、CPU は電源投入診断とシステムの初期化を実行します。システムの初期化中に、CPU はすべての非保持ビットメモリを削除し、すべての非保持 DB の内容を初期値にリセットします。次に、CPU は所定の電源投入モードに移行します。エラーによって CPU が RUN モードに移行できないことがあります。CPU は次の電源投入モードをサポートしています。STOP モード、「ウォームリスタート後に RUN モードに移行」、「ウォームリスタート後に以前のモードに移行」

注意

ウォームリスタートモードの設定

CPU は、交換可能なシグナルモジュールの故障などの修理可能な故障、または電源ラインの外乱や異常な電源投入イベントなどの一時的な障害が理由で STOP モードに移行することがあります。

CPU が「電源オフ前はウォームリスタートモード」に設定されている場合、故障が修理されたり取り除かれても、RUN に移行する新しいコマンドを STEP 7 から受信しない限り、RUN モードには戻りません。新しいコマンドがない場合、STOP モードは電源オフ前のモードとして保持されます。

STEP 7 接続とは無関係に動作するよう意図された CPU は、通常、故障状態が取り除かれた後でパワーサイクルによって RUN モードに戻れるよう、「ウォームリスタート - RUN」に設定されます。



CPU には動作モードを変更するための物理的なスイッチがありません。CPU の動作モードを変更するために、STEP 7 には以下のツールがあります。

- STEP 7 ツールバーの[STOP]および[RUN]ボタン
- オンラインツールの CPU オペレータパネル

プログラム内に STP 命令を記述して、CPU の動作モードを STOP モードに変更することもできます。このようにすれば、プログラムロジックに基づいて、プログラムの実行を停止することができます。Web サーバー(256 ページ)にも動作モードを変更するためのページがあります。

4.3 ユーザープログラムの実行

プログラムを効率良く構造化することのできる、次のタイプのプログラムブロックが CPU によってサポートされています。

- オーガニゼーションブロック(OB)はプログラムの構造を定義します。動作および開始イベントが定義済みの OB も存在しますが、独自の開始イベント(60 ページ)を定義した OB を作成することもできます。
- ファンクション(FC)およびファンクションブロック(FB)には、特定のタスクまたはパラメータの組み合わせに対応したプログラムコードが含まれています。それぞれの FC または FB には、呼び出し元のブロックとデータを共有するための入力と出力のパラメータのセットが提供されています。また、FB は、関連のデータブロック(インスタンス DB)を使用して実行間の値の状態を保持し、それをプログラム内の他のブロックが使用することができます。
- データブロック(DB)には、プログラムブロックが使用することのできるデータが格納されます。

ユーザープログラム、データ、コンフィギュレーションのサイズは、CPU 内のロードメモリおよびワークメモリの使用可能な容量(17 ページ)によって制限されます。個々の OB、FC、FB、DB ブロックの数には特に制限はありません。ただし、ブロックの合計数は 1024 に制限されます。

4.3.1 RUNモードでスキャンサイクル処理

各スキャンサイクルでは、CPU は出力の書き込み、入力の読み取り、ユーザープログラムの実行、通信モジュールの更新、ユーザー割り込みイベントおよび通信要求への応答を実行します。通信要求は、スキャン中を通して定期的に処理されます。

これらの操作(ユーザー割り込みイベントを除く)は、定期的に順次に行われます。有効なユーザー割り込みイベントのサービスは、発生した順序で優先度に基づいて行われます。割り込みイベントでは、該当する場合、CPU は対応するプロセスイメージパーティション(PIP)を使用して入力を読み取り、OB を実行し、出力を書き込みます。

システムによって、最大サイクルタイムと呼ばれる期間内にスキャンサイクルを完了することが保証されています。そうでない場合は、タイムエラーイベントが生成されます。

- スキャンサイクルでは、プロセスイメージからデジタルおよびアナログの各出力の現在値が読み取られ、自動 I/O 更新を行うように構成されている(デフォルトの構成)CPU、SB、SM の物理的な出力にその値が書き込まれます。命令によって物理的な出力へのアクセスが行われたときは、出力プロセスイメージと物理的な出力の両方の更新が行われます。
- 自動 I/O 更新を行うように構成されている(デフォルトの構成)CPU、SB、SM の物理的な出力の現在の値を読み取り、それをプロセスイメージに書き込むことによって、スキャンサイクルが続行されます。命令によって物理的な入力へのアクセスが行われたときは、その物理的な入力の値のアクセスは行われますが、入力プロセスイメージの更新は行われません。
- 入力の読み取りが終了すると、ユーザープログラムの最初の命令から最後の命令までが実行されます。これには、すべてのプログラムサイクル OB と、それに関連するすべての FC および FB が含まれます。プログラムサイクル OB は、OB 番号の小さなものから順に実行されます。

スキャンサイクルを通じて定期的に発生する通信処理によって、ユーザープログラムの実行が中断されることがあります。

自己診断には、システムおよび I/O モジュールの状態の定期チェックが含まれています。

割り込みイベントは、スキャンサイクルのどこでも発生する可能性があります。イベントが発生すると、CPU はスキャンサイクルを中断して、そのイベントを処理するように構成されている OB を呼び出します。OB によるイベントの処理が終了すると、CPU は、ユーザープログラムの実行を割り込み発生時点から再開します。

4.3.2 OBを使用したユーザープログラムの構造化

OB は、ユーザープログラムの実行を制御します。CPU 内の所定のイベントによってオーガニゼーションブロックの実行が開始されます。OB の相互呼び出しおよび FC および FB からの呼び出しを行うことはできません。OB を開始できるのは診断割り込みや時間間隔などの開始イベントだけです。CPU は、OB の優先度に基づいて OB の処理を行います。優先度の高い OB は優先度の低い OB よりも先に実行されます。優先度が最も低いのは 1 (メインプログラムサイクル用)で、優先度が最も高いのは 26 です。

4.3.3 イベント実行の優先度およびキュー

CPU の処理はイベントによって制御されます。イベントは割り込み OB の実行をトリガします。ブロックの作成時、デバイスの構成時、または ATTACH または DETACH 命令によって、イベントに対応する割り込み OB を指定します。プログラムサイクルイベントや周期イベントのように定期的に発生するイベントもあります。また、スタートアップイベントや遅延イベントのように 1 回しか発生しないイベントもあります。入力点でのエッジイベントや高速カウンタイベントなど、ハードウェアによってトリガされたときに発生するイベントもあります。診断エラーおよびタイムエラーイベントなどのイベントは、エラーが生じた場合にのみ発生します。イベント割り込み OB の処理順序を決定するために、イベントの優先度とキューが使用されます。

CPU は優先度の順番にイベントを処理します。1 が最も優先度が低く、26 が最も優先度が高くなります。V4.0 以前の S7-1200 CPU では、各タイプの OB が固定の優先度クラス(1~26)に属していました。V4.0 以降では、設定する OB ごとに優先度クラスを割り当てることができます。OB のプロパティの属性で、優先度番号を設定します。

割り込み可能および割り込み不可能な実行モード

OB (59 ページ)は、トリガされるイベントの優先度順に実行されます。V4.0 以降では、OB の実行を割り込み可能または割り込み不可能として設定することができます。プログラムサイクル OB は常に割り込み可能ですが、その他のすべての OB は割り込み可能または割り込み不可能のどちらにでも設定できます。

割り込み可能モードを設定していて、OB が実行中で OB の実行が完了する前に優先度の高いイベントが発生した場合、実行中の OB は優先度の高いイベント OB の実行を可能にするために割り込まれます。優先度の高いイベントが実行され、それが完了すると、割り込まれた OB の実行が継続されます。割り込み可能な OB の実行中に複数のイベントが発生した場合、CPU は優先度順にこれらのイベントを処理します。

割り込み可能モードを設定していない場合、OB はトリガされると、実行中にほかにどのようなイベントがトリガされても最後まで実行されます。

割り込みイベントがサイクリック OB と遅延 OB をトリガする以下の 2 つのケースを考えてみてください。どちらのケースでも、遅延 OB (OB201)にはプロセスイメージパーティションの割り当てがなく、優先度 4 で実行されます。サイクリック OB (OB200)にはプロセスイメージパーティションの割り当て PIP1 があり、優先度 2 で実行されます。下の図に、割り込み不可能実行モードと割り込み可能実行モード間の実行の相違点を示します。

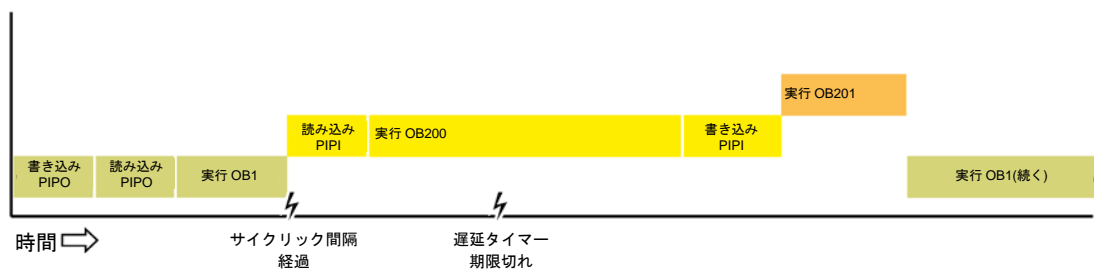


図 4-1 ケース 1: 割り込み不可能な OB の実行

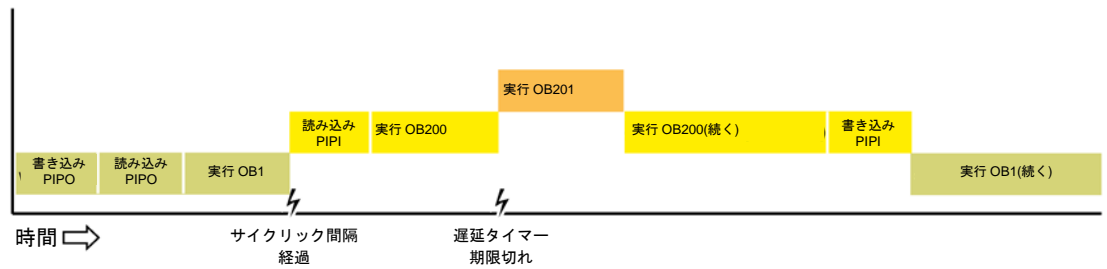


図 4-2 ケース 2: 割り込み可能な OB の実行

注記

OB 実行モードを割り込み不可能に設定している場合、タイムエラー OB はプログラムサイクル OB 以外の OB に割り込むことはできません。S7-1200 CPU の V4.0 以前では、タイムエラー OB はすべての実行中の OB に割り込むことができました。V4.0 以降では、タイムエラー OB (またはその他のすべての優先度の高い OB) がプログラムサイクル OB 以外の実行中の OB に割り込めるようにするには、OB の実行を割り込み可能に設定する必要があります。

イベント実行の優先度およびキューを理解する

CPU は、キュー内に保留するイベントの数を、イベントのタイプによって異なるキューを使用して制限します。所定のイベントタイプの保留イベントの数が限度に達すると、その次のイベントは失われます。キューのオーバーフローに対応するために、タイムエラー割り込み OB を使用することができます。

各 CPU イベントには優先度が割り付けられています。一般に、CPU はイベントの実行を優先度順に(優先度の高いものから先に)行います。イベントの優先度が同じ場合は、先に発生したものから順にサービスを行います。

表 4-1 OB イベント

イベント	許容数量	デフォルトの OB 優先度
プログラムサイクル	1 つのプログラムサイクルイベント 複数の OB が可	1 ⁴
スタートアップ	1 つのスタートアップイベント ¹ 複数の OB が可	1 ⁴
遅延	最大 4 つのタイムイベント イベントあたり 1 つの OB	3
周期割り込み	最大 4 つのイベント イベントあたり 1 つの OB	8
ハードウェア割り込み	最大 50 のハードウェア割り込みイベント ² イベントあたり 1 つの OB、ただし、複数のイベントに同じ OB を使用することができます	18
		18
タイムエラー	1 イベント(設定されている場合のみ) ³	22 または 26 ⁴
診断エラー	1 イベント(設定されている場合のみ)	5

4.3 ユーザープログラムの実行

イベント	許容数量	デフォルトの OB 優先度
モジュールの取り出し またはプラグ	1 イベント	6
ラックまたはステーション障害	1 イベント	6
時刻	最大 2 つのイベント	2
ステータス	1 イベント	4
更新	1 イベント	4
プロファイル	1 イベント	4

- ¹ スタートアップイベントとプログラムサイクルイベントは同時には発生しません。スタートアップイベントはプログラムイベントが開始される前に実行を完了するからです。
- ² DETACH 命令と ATTACH 命令を使用する場合、50 以上のハードウェア割り込みイベント OB を持つことができます。
- ³ スキャンサイクルが最大スキャンサイクルタイムを超過しても CPU が RUN のままになるよう設定したり、RE_TRIGR 命令を使用してサイクルタイムをリセットすることができます。ただし、スキャンサイクルタイムが 2 回目に最大スキャンサイクルタイムを超過すると、CPU は STOP モードに移行します。
- ⁴ 新しい V4.0 または V4.1 CPU の優先度は 22 です。V3.0 CPU を V4.0 または V4.1 CPU に交換した場合、優先度は 26 です: これは V3.0 に有効だった優先度です。いずれの場合も、優先度フィールドは編集可能で、優先度を 22~26 の範囲で任意の値に設定できます。

詳細は、項目「V3.0 CPU を V4.1 CPU に交換(435 ページ)」を参照してください。

また、CPU は関連する OB を持たないイベントも認識します。以下の表に、これらのイベントと対応する CPU の動作を示します。

表 4-2 その他のイベント

イベント	説明	CPU の動作
I/O アクセスエラー	ダイレクト I/O 読み取り/書き込みエラー	CPU は最初の発生を診断バッファにログし、RUN モードのままとなります。
最大サイクルタイムエラー	CPU が設定されたサイクルタイムを 2 回超過	CPU はエラーを診断バッファにログし、STOP モードに移行します。
周辺アクセスエラー	プロセスイメージ更新時の I/O エラー	CPU は最初の発生を診断バッファにログし、RUN モードのままとなります。
プログラミングエラー	プログラム実行エラー	エラーのあるブロックがエラー処理を行う場合、エラーのある構造体を更新します。そうでない場合、CPU はエラーを診断バッファにログし、RUN モードのままとなります。

割り込み待ち時間

割り込みイベント待ち時間(イベントが発生したとの CPU の通知から、そのイベントのサービスを行う OB 内の最初の命令の実行を CPU が開始するまでの時間)は、プログラムサイクル OB が、この割り込みイベントの発生時に唯一のイベントである場合、175 μsec です。

4.4 メモリ領域、アドレス指定、データタイプ

CPU には、ユーザープログラム、データ、および構成を保存するための、次のメモリ領域が提供されています。

- ロードメモリは、ユーザープログラム、データ、および構成を保存するための不揮発性メモリ領域です。プロジェクトを CPU にダウンロードすると、まずこのロードメモリ領域に保存されます。この領域は、メモリカード(存在する場合)または CPU のどちらかに配置されています。この不揮発性メモリ領域の内容は、電源が切断された場合にも保持されます。メモリカードを取り付けて、データログに使用できるロードメモリの量を増やすことができます。
- ワークメモリは、ユーザープログラムの実行中に、ユーザープロジェクトのいくつかの要素を格納しておくための揮発性メモリ領域です。CPU は、プロジェクトの必要な要素をロードメモリからワークメモリにコピーします。このワークメモリは揮発性メモリ領域であるので、電源が切断されると情報が失われ、電源が復旧すると CPU によって復元されます。
- 保持型メモリは、ワークメモリの値の一部を格納しておくための保持型メモリ領域です。保持型メモリ領域は、電源が切断された時でも、選択したユーザーメモリの情報を保持するために使用されます。電源が切断されると、CPU は電源投入時にこの保持値を復元します。



オプションの SIMATIC メモリカードは、ユーザープログラムを保存するための代替メモリとして使用したり、プログラムの転送に使用できます。メモリカードを使用する場合、CPU はプログラムを CPU のメモリではなく、メモリカードから実行します。

メモリカードが書き込み禁止になっていないことを確認します。保護スイッチをスライドして「ロック」を解除します。

オプションの SIMATIC メモリカードを、プログラムカード、転送カードとして、または、データログファイルの収集、ファームウェア更新の実行に使用してください。

- 転送カードを使用して、プロジェクトを STEP 7 を使用せずに複数の CPU にコピーすることができます。転送カードは、保存されたプロジェクトをカードから CPU のメモリにコピーします。プログラムを CPU にコピーした後は、転送カードを取り出してください。
- プログラムカードは CPU ロードメモリに取って代わります。すべての CPU ファンクションはプログラムカードで制御されます。プログラムカードを挿入すると、CPU の内部ロードメモリのすべて(ユーザープログラムと強制 I/O 値)が消去されます。CPU はユーザープログラムをプログラムカードから実行します。
- プログラムカードは、データログファイル(124 ページ)の収集にも使用できます。プログラムカードは CPU の内部メモリよりも多くのメモリを提供します。CPU の Web サーバーファンクション(255 ページ)を使用して、データログファイルをコンピュータにダウンロードできます。
- メモリカードは、ファームウェア更新の実行にも使用できます。説明については、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。

注記

プログラムカードを CPU に挿入したままにしておく**必要があります**。プログラムカードを抜き取ると、CPU は STOP モードに移行します。

4.4.1 S7-1200 がサポートするデータタイプ

データタイプは、データの要素と、そのデータの解釈方法の両方を指定するために使用します。命令パラメータのそれぞれは少なくとも 1 つのデータタイプをサポートしています。複数のデータタイプをサポートしているパラメータもあります。命令のパラメータフィールドの上にカーソルを移動すると、そのパラメータがサポートしているデータタイプが表示されます。

表 4-3 S7-1200 がサポートするデータタイプ

データタイプ	説明
ビットおよびビットシーケンスデータタイプ	<ul style="list-style-type: none"> Bool は Boolean またはビット値です。 Byte は 8 ビットバイト値です。 Word は 16 ビット値です。 DWord は 32 ビットダブルワード値です。
整数データタイプ	<ul style="list-style-type: none"> USInt (符号なし 8 ビット整数)および SInt (符号付き 8 ビット整数)は、符号付きまたは符号なしの「短い」整数(メモリの 8 ビットまたは 1 バイト)です。 UInt (符号なし 16 ビット整数)および Int (符号付き 16 ビット整数)は、符号付きまたは符号なしの整数(メモリの 16 ビットまたは 1 ワード)です。 UDInt (符号なし 32 ビット整数)および DInt (符号付き 32 ビット整数)は、符号付きまたは符号なしの倍精度整数(メモリの 32 ビットまたは 1 ワード)です。
実数データタイプ	<ul style="list-style-type: none"> Real は 32 ビット実数または浮動小数点値です。 LReal は 64 ビット実数または浮動小数点値です。
日付と時刻データタイプ	<ul style="list-style-type: none"> Date は、1990 年 1 月 1 日からの日数を含む 16 ビット日付値(UInt に類似)です。最大日付値は 65378 (16#FF62)で、2168 年 12 月 31 日に対応しています。使用可能なすべての日付値は有効です。 DTL (date and time long)データタイプは、日付と時刻の情報が定義済みの 12 バイトの構造体です。 <ul style="list-style-type: none"> 年(UInt): 1970~2554 月(USInt): 1~12 日(USInt): 1~31 週日(USInt): 1 (日曜日)~7 (土曜日) 時間(USInt): 0~23 分(USInt): 0~59 秒(USInt): 0~59 ナノ秒(UDInt): 0~999999999 Time は、ミリ秒数(0~24 日、20 時間、31 分、23 秒、647 ミリ秒)を保存する 32 ビット IEC 時間値(Dint に類似)です。使用可能なすべての時間値は有効です。時間値は計算に使用され、負の時間も可能です。 TOD (time of day)は、午前零時からのミリ秒数(0~86399999)を含む 32 ビットの時刻値(Dint に類似)です。
文字および文字列データタイプ	<ul style="list-style-type: none"> Char は 8 ビットの単一文字です。 String は最大 254 文字の変長可変長の文字列です。

データタイプ	説明
配列および構造体データタイプ	<ul style="list-style-type: none"> 配列は同じデータタイプの複数のエレメントを含みます。配列は、OB、FC、FB、DB 用のブロックインターフェースエディタで作成することができます。PLC タグエディタでは配列を作成するとはできません。 Struct は異なるデータタイプで構成されるデータの構造を定義します。Struct データタイプは、関連するプロセスデータのグループを 1 つのデータ単位として処理するのに使用します。Struct データタイプの名前と内部データ構造を、データブロックエディタまたはブロックインターフェースエディタで宣言します。 <p>配列と構造体をより大きな 1 つの構造体にまとめることもできます。構造体は最大で 8 レベルの深さにネストできます。たとえば、複数の配列を含む複数の構造体でできた構造体を作成することができます。</p>
PLC データタイプ	<p>PLC データタイプは、ユーザーがプログラムで複数回使用できるカスタムデータ構造体を定義するユーザー定義のデータ構造体です。PLC データタイプを作成すると、新しい PLC データタイプが DB エディタとコードブロックインターフェースエディタのデータタイプセレクトドロップダウンリストに表示されます。</p> <p>PLC データタイプは、コードブロックインターフェースまたはデータブロックで直接、データタイプとして使用できます。</p> <p>また、PLC データタイプは、同じデータ構造体を使用する複数のグローバルデータブロックを作成するためのテンプレートとして使用できます。</p>
ポインタデータタイプ	<ul style="list-style-type: none"> ポインタは、タグのアドレスを間接的に参照します。ポインタはメモリ内で 6 バイト(48 ビット)を占有し、変数に対する以下の情報を含めることができます: DB 番号(データが DB に保存されていない場合は 0)、CPU のメモリ領域、メモリアドレス。 Any はデータ領域の始まりを間接的に参照し、その長さを識別します。Any ポインタはメモリ内で 10 バイトを使用し、以下の情報を含めることができます: データエレメントのデータタイプ、データエレメントの数、メモリ領域または DB 番号、データの「Byte.Bit」開始アドレス。 Variant は異なったデータタイプまたはパラメータのタグを間接的に参照します。Variant ポインタは構造体と個々の構造コンポーネントを認識します。Variant はメモリ内でスペースを占有しません。

データタイプとしては使用できませんが、次の BCD (2 進数 10 進数) 数字フォーマットも変換命令でサポートされています。

- BCD16 は 16 ビット値(-999~999)です。
- BCD32 は 32 ビット値(-9999999~9999999)です。

4.4.2 メモリ領域をアドレス指定する

STEP 7 では、簡単にシンボリックプログラミングを行うことができます。データのアドレスのシンボル名、すなわち「タグ」を、メモリアドレスおよび I/O 点に関連した PLC タグ、またはコードブロックで使用されるローカル変数のいずれかとして作成します。ユーザープログラムでこのタグを使用するには、命令パラメータにタグ名を入力するだけです。CPU によるメモリ領域の構造化とアドレス指定を詳しく知るために、以下のパラグラフに PLC タグが参照する「絶対」アドレス指定について説明します。CPU には、ユーザープログラム実行中のデータを保存するためのオプションがいくつか用意されています。

- グローバルメモリ: CPU には、入力(I)、出力(Q)、ビットメモリ(M)など、さまざまな専用メモリ領域が用意されています。このメモリには、すべてのコードブロックが制限なしにアクセスできます。
- データブロック(DB): プログラムで使用するデータを保存するために、ユーザープログラムに DB を作成できます。保存されたデータは、関連付けられているプログラムが終了しても残ります。「グローバル」DB にはプログラム内で使用するすべてのデータを保存することができ、インスタンス DB には対応する FB のデータだけを保存し、FB 用のパラメータで構造化します。
- テンポラリメモリ: プログラムブロックが呼び出されると、CPU のオペレーティングシステムは、ブロックの実行中に使用するテンポラリメモリまたはローカルメモリ(L)を割り付けます。このプログラムブロックの実行が終了すると、CPU は、他のプログラムブロックの実行用にローカルメモリを割り付け直します。

それぞれのメモリロケーションには、固有のアドレスが存在します。ユーザープログラムでは、これらのアドレスを使用して、メモリロケーション内の情報にアクセスします。

入力(I)または出力(Q)メモリ領域への参照(I0.3 または Q1.7 など)で、プロセスイメージにアクセスします。物理的な入力または物理的な出力に即時アクセスするには、参照先に「:P」を付加します(I0.3:P、Q1.7:P、Stop:P など)。

強制は、固定値を物理的な入力(Ix.y:P)または物理的な出力(Qx.y:P)のみに適用します。入力または出力を強制するには、PLC タグまたはアドレスに「:P」を付加します。詳細は、「CPU の変数の強制」(342 ページ)を参照してください。

表 4-4 メモリ領域

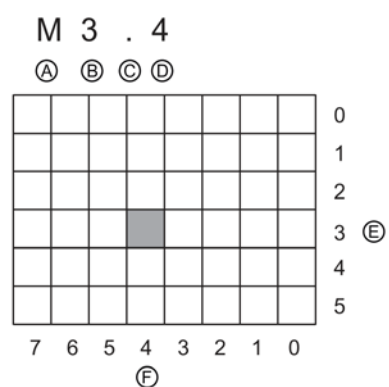
メモリ領域	説明	強制	保持
I プロセスイメージ入力 I_:P ¹ (物理的な入力)	スキャンサイクル開始時に物理的な入力からコピーされる	なし	なし
	CPU、SB、SM 上の物理的な入力点の即時読み取り	あり	なし
Q プロセスイメージ出力 Q_:P ¹ (物理的な出力)	スキャンサイクル開始時に物理的な出力からコピーされる	なし	なし
	CPU、SB、SM 上の物理的な出力点への即時書き込み	あり	なし
M ビットメモリ	制御およびデータメモリ	なし	あり (オプション)
L テンポラリメモリ	ブロック用のテンポラリデータ、1つのブロック専用	なし	なし
DB データブロック	データメモリおよび FB のパラメータ用メモリ	なし	あり (オプション)

¹ 物理的な入力および物理的な出力に即時アクセス(または強制)するには、アドレスまたはタグに「:P」を付加します(I0.3:P、Q1.7:P、「Stop:P」など)。

それぞれのメモリロケーションには、固有のアドレスが存在します。ユーザープログラムでは、これらのアドレスを使用して、メモリロケーション内の情報にアクセスします。絶対アドレスは以下のエレメントで構成されます。

- メモリ領域(I、Q、M など)
- アクセス先のデータのサイズ(バイトの場合は「B」、ワードの場合は「W」 など)
- データのアドレス(バイト 3 またはワード 3 など)

ブール値のアドレスのビットにアクセスする場合は、サイズのニモニックを入力しません。データのメモリ領域、バイト位置、およびビット位置のみを入力します(I0.0、Q0.1、M3.4 など)。



メモリ領域の絶対アドレス

- A メモリ領域 ID
- B バイトアドレス バイト 3
- C セパレータ (byte.bit)
- D バイト内のビットの位置(8 バイトのビット 4)
- E メモリ領域のバイト
- F 選択したバイトのビット

上の例では、メモリ領域とバイトアドレス(M = ビットメモリ領域、3 = バイト 3)は、ビットアドレス(ビット 4)と区別するために最後にピリオド(「.」)が付けられています。

CPU および I/O モジュールの I/O を設定する



Device overview					
Module	Slot	I address	Q address	Type	Order
	103				
	102				
RS485_1	101			CM 1241 (RS485)	6ES7
▼ PLC_1	1			CPU 1214C DCDCI	6ES7
DI14/DO10	1.1	0...1	0...1	DI14/DO10	
AI2	1.2	64...67		AI2	
AO1 x 12bit	1.3		80...81	AO1 signal board	6ES7
HSC_1	1.16	1000...		High speed counts	
HSC_2	1.17			High speed counts	
HSC_3	1.18			High speed counts	
HSC_4	1.19			High speed counts	
HSC_5	1.20			High speed counts	
HSC_6	1.21			High speed counts	
Pulse_1	1.32			Pulse generator (P)	
Pulse_2	1.33			Pulse generator (P)	
► PROFINET L X1				PROFINET interface	
DI8 x 24VDC	2	8		SM 1221 DI8 x 24	6ES7

デバイスコンフィグレーションに CPU および I/O モジュールを追加すると、STEP 7 は I アドレスと Q アドレスを自動的に割り付けます。デフォルトのアドレス指定を変更するには、デバイスコンフィグレーションのアドレスフィールドを選択して新しい番号を入力します。

- STEP 7 はデジタル入力および出力を、モジュールがすべての点を使用するかどうかにかかわらず、8 点(1 バイト)のグループに割り当てます。
- STEP 7 はアナログ入力および出力を 2 つのグループに割り当て、各アナログ点が 2 バイト(16 ビット)を占有します。

上の図は、2 つの SM と 1 つの SB を備えた CPU 1214C の例を示しています。この例では、DI8 モジュールのアドレスを 8 ではなく 2 に変更できました。誤ったサイズや他のアドレスと競合するアドレス範囲を入力しないようにツールが支援してくれます。

4.4.3 タグ付きデータタイプの「スライス」アクセス

PLC タグおよびデータブロックタグは、サイズに従ってビット、バイト、またはワードレベルでアクセスできます。このようなデータスライスにアクセスするための構文は以下のとおりです。

- "<PLC タグ名>".xn (ビットアクセス)
- "<PLC タグ名>".bn (バイトアクセス)
- "<PLC タグ名>".wn (ワードアクセス)
- "<データブロック名>".<タグ名>.xn (ビットアクセス)
- "<データブロック名>".<タグ名>.bn (バイトアクセス)
- "<データブロック名>".<タグ名>.wn (ワードアクセス)

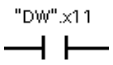
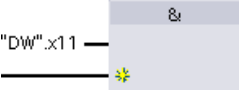
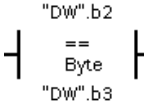


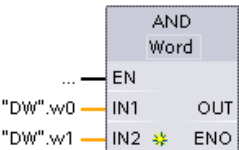
ダブルワードサイズのタグは、ビット 0~31、バイト 0~3、ワード 0~1 でアクセスできます。ワードサイズのタグは、ビット 0~15、バイト 0~1、ワード 0 でアクセスできます。バイトサイズのタグは、ビット 0~7、バイト 0 でアクセスできます。ビットスライス、バイトスライス、およびワードスライスは、ビット、バイト、ワードがオペランドとして予想される場所ならどこでも使用可能です。

																								バイト							
																ワード															
ダブルワード																															
x31	x30	x29	x28	x27	x26	x25	x24	x23	x22	x21	x20	x19	x18	x17	x16	x15	x14	x13	x12	x11	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	x0
b3								b2								b1								b0							
w1																w0															

注記

スライスアクセスできる有効なデータタイプは、Byte、Char、Conn_Any、Date、DInt、DWord、Event_Any、Event_Att、Hw_Any、Hw_Device、HW_Interface、Hw_Io、Hw_Pwm、Hw_SubModule、Int、OB_Any、OB_Att、OB_Cyclic、OB_Delay、OB_WHINT、OB_PCYCLE、OB_STARTUP、OB_TIMEERROR、OB_Tod、Port、Rtm、SInt、Time、Time_Of_Day、UDInt、UInt、USInt、Word です。Real タイプの PLC タグはスライスアクセスできますが、Real タイプのデータブロックはアクセスできません。

例
 PLC タグテーブルで、「DW」は DWORD タイプとして宣言されたタグです。この例では、ビット、バイト、およびワードのスライスアクセスを示しています。

	LAD	FBD	SCL
ビットアクセス			<pre>IF "DW".x11 THEN ... END_IF;</pre>
バイトアクセス			<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
ワードアクセス			<pre>out:= "DW".w0 AND "DW".w1;</pre>

4.4.4 ATオーバーレイでタグにアクセス

AT タグのオーバーレイでは、異なったデータタイプのオーバーレイされた宣言で、標準アクセスブロックの宣言済みタグにアクセスできます。たとえば、Byte、Word、DWord データタイプのタグの個々のビットを Bool の配列でアドレス指定できます。

宣言

パラメータをオーバーレイするには、オーバーレイするパラメータのすぐ後ろにある他のパラメータを宣言し、データタイプ「AT」を選択します。エディタがオーバーレイを作成し、データタイプ struct またはオーバーレイに使用する配列を選択できます。

例
 この例は、標準アクセス FB の入力パラメータを示しています。バイトタグ B1 が Boolean の配列でオーバーレイされています。


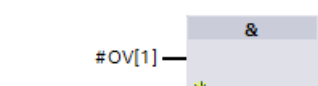
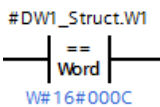
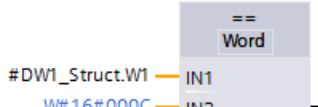
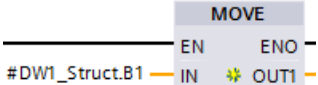
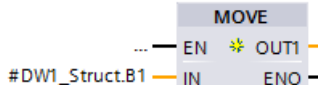
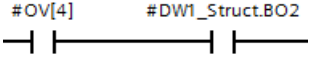
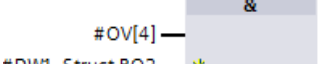
		B1	Byte	0.0
		OV	AT "B1"	Array[0..7] of Bool
		OV[0]	Bool	0.0
		OV[1]	Bool	0.1
		OV[2]	Bool	0.2
		OV[3]	Bool	0.3
		OV[4]	Bool	0.4
		OV[5]	Bool	0.5
		OV[6]	Bool	0.6
		OV[7]	Bool	0.7

もう 1 つの例では、Struct でオーバーレイされている DWord タグを示します。Word、Byte、および 2 つの Boolean が含まれます：

DW1	DWord	2.0
DW1_Struct AT "DW1"	Struct	2.0
W1	Word	0.0
B1	Byte	2.0
BO1	Bool	3.0
BO2	Bool	3.1

ブロックインターフェースの[オフセット]列に、元のタグを基準にしたオーバーレイされたデータタイプの位置を示します。

プログラムロジックで直接、オーバーレイタイプをアドレス指定することができます：

LAD	FBD	SCL
		<pre>IF #OV[1] THEN ... END_IF;</pre>
		<pre>IF #DW1_Struct.W1 = W#16#000C THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.B1;</pre>
		<pre>IF #OV[4] AND #DW1_Struct.BO2 THEN ... END_IF;</pre>

ルール

- タグのオーバーレイは、標準(最適化されていない)アクセスの FB および FC ブロックでのみ可能です。
- すべてのブロックタイプおよびすべての宣言セクションのパラメータをオーバーレイできます。
- オーバーレイされたパラメータは、他のすべてのブロックパラメータと同じように使用できます。
- VARIANT タイプのパラメータはオーバーレイできません。
- オーバーレイパラメータのサイズは、オーバーレイされるパラメータのサイズに等しいかそれより小さくなければなりません。
- オーバーレイされる変数のすぐ後ろにある、その変数をオーバーレイする変数を宣言し、最初のデータタイプの選択としてキーワード「AT」を選択する必要があります。

4.5 パルス出力

CPU またはシグナルボード(SB)を、高速パルス出力ファンクション(パルス幅振幅(PWM)またはパルストレイン出力(PTO)のいずれかとして)制御用の 4 つのパルスジェネレータとなるよう構成することができます。基本モーション命令は PTO 出力を使用します。各パルスジェネレータを PWM または PTO のいずれかに割り付けることはできますが、同時に両方に割り付けることはできません。



パルス出力は、ユーザープログラムで他の命令によって使用することはできません。CPU または SB の出力をパルスジェネレータとして構成する場合、対応する出力アドレスは Q メモリから削除され、ユーザープログラムで他の目的で使用することはできません。ユーザープログラムがパルスジェネレータとして使用されている出力に値を書き込んだとしても、その CPU はこの値を物理出力に書き込みません。

注記

最大パルス周波数を超えないでください。

パルス出力ジェネレータの最大パルス周波数は CPU 1217C の場合は 1 MHz、CPU 1211C、1212C、1214C、1215C の場合は 100 kHz; 20 kHz (標準 SB の場合); または 200 kHz (高速 SB の場合)です。

4 つのパルスジェネレータにはデフォルトの I/O 割り当てがあります; ただし、CPU または SB の他のどのようなデジタル出力としても構成することができます。CPU 上のパルスジェネレータをリモート I/O に割り当てることはできません。

基本モーション命令を設定する場合、STEP 7 は、最大速度またはこのハードウェア制限を超える周波数で軸を設定しても、警告を出さないことに注意してください。これにより、アプリケーションで問題が発生する可能性があります。したがって、常にハードウェアの最大パルス周波数を超えないようにしてください。

オンボード CPU 出力、またはオプションのシグナルボード出力を使用できます。出力点番号を下の表に示します(デフォルト出力コンフィグレーションを想定)。デフォルトの出力点を変更した場合、出力点の番号は割り付けたものになります。PWM には 1 つの出力だけが必要ですが、PTO はチャンネル当たりオプションで 2 つの出力を使用することができます。出力がパルスファンクションに必要な場合、他の用途に利用できます。

4つのパルスジェネレータにはデフォルトの I/O 割り当てがあります; ただし、CPU または SB の他のどのようなデジタル出力としても構成することができます。CPU 上のパルスジェネレータを SM またはリモート I/O に割り当てることはできません。

表 4-5 パルスジェネレータのデフォルトの出力割り当て

説明	パルス	方向
PTO1		
内蔵 I/O	Q0.0	Q0.1
SB I/O	Q4.0	Q4.1
PWM1		
内蔵出力	Q0.0	-
SB 出力	Q4.0	-
PTO2		
内蔵 I/O	Q0.2	Q0.3
SB I/O	Q4.2	Q4.3
PWM2		
内蔵出力	Q0.2	-
SB 出力	Q4.2	-
PTO3		
内蔵 I/O	Q0.4 ¹	Q0.5 ¹
SB I/O	Q4.0	Q4.1
PWM3		
内蔵出力	Q0.4 ¹	-
SB 出力	Q4.1	-
PTO4		
内蔵 I/O	Q0.6 ²	Q0.7 ²
SB I/O	Q4.2	Q4.3
PWM4		
内蔵出力	Q0.6 ²	-
SB 出力	Q4.3	-

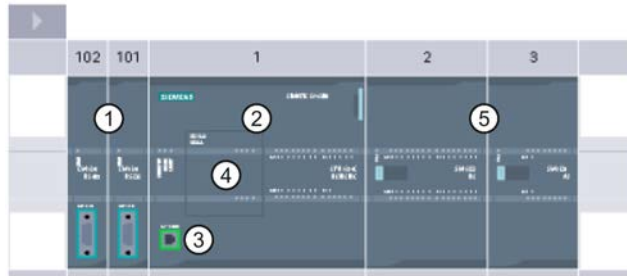
¹ CPU 1211C には出力 Q0.4、Q0.5、Q0.6、または Q0.7 はありません。したがって、これらの出力は CPU 1211C では使用できません。

² CPU 1212C には出力 Q0.6、または Q0.7 がありません。したがって、これらの出力は CPU 1212C では使用できません。

³ この表は CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C、CPU 1217C PTO/PWM ファンクションに適用されます。

カンタン デバイスコンフィグレーション

プロジェクトに CPU および付加モジュールを追加して、PLC 用のデバイスコンフィグレーションを作成します。



- ① 通信モジュール(CM)または通信プロセッサ(CP):最大 3 モジュール、スロット 101、102、103 に挿入
- ② CPU: スロット 1
- ③ CPU のイーサネットポート
- ④ シグナルボード(SB)、通信ボード(CB)、またはバッテリーボード(BB):最大 1 枚、CPU に挿入
- ⑤ デジタルまたはアナログ I/O 用シグナルモジュール(SM): 最大 8 モジュール、スロット 2～9 に挿入
(CPU 1214C、CPU 1215C、CPU 12178C の場合は、8 モジュール、CPU 1212C の場合は 2 モジュール、CPU 1211C の場合は使用不可)

デバイスコンフィグレーションを作成するにはプロジェクトにデバイスを追加します。

- ポータルビューで、[デバイス&ネットワーク]を選択して[新しいデバイスの追加]をクリックします。
- プロジェクトビューで、プロジェクト名の下の[新しいデバイスの追加]をダブルクリックします。



5.1 接続された CPU のコンフィグレーションをアップロード

STEP 7 では、接続されている CPU のハードウェアコンフィグレーションを 2 つの方法でアップロードできます。

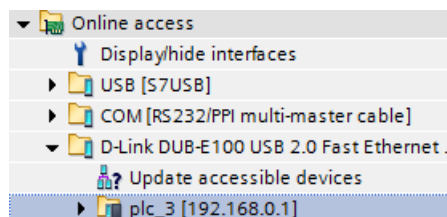
- 接続されているデバイスを新しいステーションとしてアップロードする
- 未指定の CPU を構成し接続されている CPU のハードウェアコンフィグレーションを検出する

ただし、1 番目の方法では、CPU のハードウェアコンフィグレーションとソフトウェアの両方がアップロードされます。

デバイスを新しいステーションとしてアップロードする

接続されているデバイスを新しいステーションとしてアップロードするには、次の手順を実行します。

1. プロジェクトツリーの「オンラインアクセス」ノードから通信インターフェースを展開します。
2. [アクセス可能なデバイスの更新]をダブルクリックします。
3. 検出されたデバイスから PLC を選択します。



4. STEP 7 をオンラインにし、のオンラインメニューから、[デバイスを新しいステーションとしてアップロード(ハードウェアおよびソフトウェア)]メニューコマンドを選択します。

STEP 7 がハードウェアコンフィグレーションとプログラムブロックの両方をアップロードします。

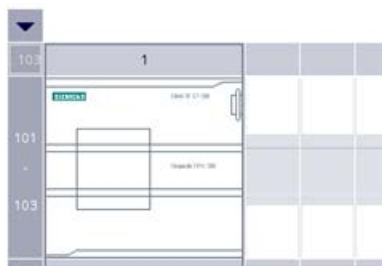
未指定の CPU のハードウェアコンフィグレーションを検出する



CPUに接続している場合は、すべてのモジュールを含め、その CPU の構成をプロジェクトにアップロードすることができます。新しいプロジェクトを作成し、特定の CPU ではなく[未指定の CPU]を選択します。([ファーストステップ]から[PLC プログラムの書き込み]を選択することによって、デバイスコンフィグレーションの全体をスキップすることもできます。この場合、STEP 7によって未指定の CPU が自動的に作成されます。)

プログラミングエディタで、[オンライン]メニューから[ハードウェア検出]を選択します。

デバイスコンフィグレーションエディタで、接続デバイスの構成検出用のオプションを選択します。

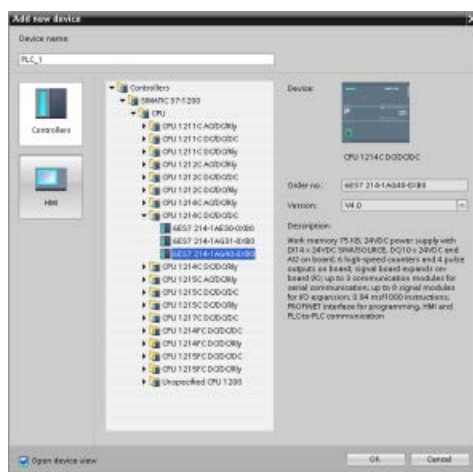


The device is not specified.
→ Please use the [hardware catalog](#) to specify the CPU.
→ or [detect](#) the configuration of the connected device.

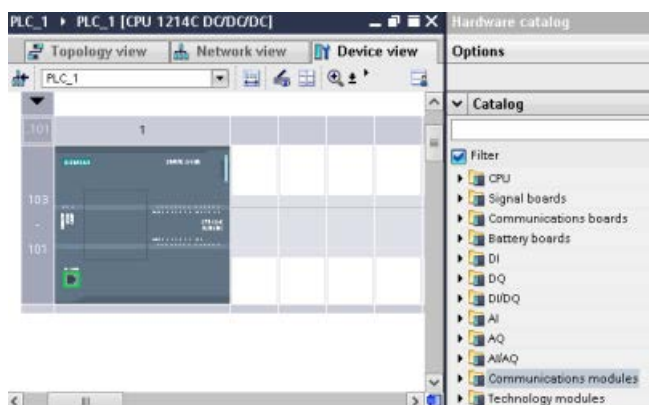
オンラインダイアログで CPU を選択して[検出]ボタンをクリックすると、STEP 7は、その CPU から、すべてのモジュール(SM、SB、CM)を含めてハードウェアコンフィグレーションをアップロードします。CPU とモジュールのパラメータ(82 ページ)を構成することができます。



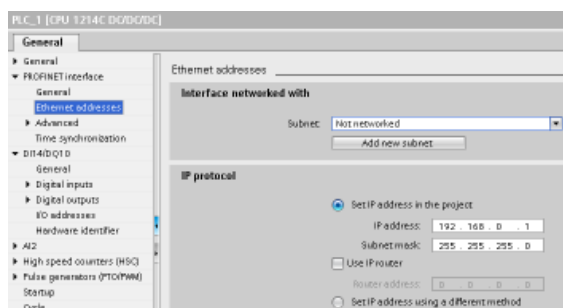
5.2 コンフィグレーションに CPU を追加



プロジェクトに CPU を挿入することによりデバイスコンフィグレーションを作成します。[新しいデバイスの追加]ダイアログで CPU を選択し、[OK]をクリックしてプロジェクトに CPU を追加します。



デバイスビューに CPU とラックが表示されます。



デバイスビューで CPU を選択すると、インスペクタウィンドウにその CPU のプロパティが表示されます。このプロパティを使用して、CPU の動作パラメータを構成します(82 ページ)。

注記

CPU の IP アドレスは未設定です。デバイスコンフィグレーションで、CPU に IP アドレスを手動で割り付ける必要があります。CPU がネットワーク上のルータに接続されている場合は、そのルータ用の IP アドレスも入力する必要があります。

5.3 デバイスの変更

コンフィグレーション済みの CPU またはモジュールのデバイスタイプを変更できます。デバイスコンフィグレーションで、デバイスを右クリックしてコンテキストメニューから[デバイスの変更]を選択します。ダイアログで、交換したい CPU またはモジュールに移動して選択します。[デバイスの変更]ダイアログに、2つのデバイス間の互換性情報が表示されます。

注記

デバイスの交換:V3.0 CPU を V4.1 CPU に交換

STEP 7 V13 で STEP 7 V12 プロジェクトを開いて、V3.0 CPU を V4.1 CPU と交換することができます。V3.0 以前のバージョンの CPU は交換できません。V3.0 CPU を V4.1 CPU と交換する場合は、2つのバージョン間の機能および動作の相違点(435 ページ)と、行うべき対処法を考慮してください。

V3.0 以前の CPU バージョン用のプロジェクトがある場合は、まず CPU を V3.0 にアップグレードしてから V4.1 にアップグレードしてください。

5.4 コンフィグレーションにモジュールを追加

CPUにモジュールを追加するには、ハードウェアカタログを使用します。

- シグナルモジュール(SM) は、デジタルまたはアナログ I/O 点を追加するために使用します。シグナルモジュールは CPU の右側に接続します。
- シグナルボード(SB)は、CPU に少数の I/O 点を追加するために使用します。SB は CPU の前面に装着します。
- バッテリボード 1297 (BB)は、リアルタイムクロックの長期のバックアップを行うために使用します。BB は CPU の前面に装着します。
- 通信ボード(CB)は通信ポート(RS485 など)を追加するために使用します。CB は CPU の前面に装着します。
- 通信モジュール(CM)と通信プロセッサ(CP)は、PROFIBUS または GPRS 用の通信ポートを追加するために使用します。通信モジュールは CPU の左側に接続します。

デバイスコンフィグレーションにモジュールを挿入するには、ハードウェアカタログでモジュールを選択し、そのモジュールをダブルクリックするか、強調表示されているスロットにドラッグします。モジュールを機能させるには、デバイスコンフィグレーションにモジュールを追加し、CPU にハードウェアコンフィグレーションをダウンロードする必要があります。

表 5-1 モジュールをデバイスコンフィグレーションに追加する

モジュール	モジュールを選択する	モジュールを挿入する	結果
SM			
SB、BB、または CB			
CM または CP			

「コンフィグレーション制御」機能(81 ページ)を使用して、シグナルモジュールとシグナルボードを、特定のアプリケーションでは実際のハードウェアに対応していないことがあります。共通のユーザープログラム、CPU モデル、(おそらく)コンフィグレーション済みのモジュールの一部をさまざまなアプリケーションで共有することができるハードウェアコンフィグレーションを作成できます。

5.5 コンフィグレーション制御

コンフィグレーション制御は、さまざまなオートメーションソリューション(機械)タイプを作成するときに役に立つソリューションです。

STEP 7 および S7-1200 でのコンフィグレーション制御で、標準機械の最大構成をコンフィグレーションし、このコンフィグレーションのサブセットを使用するバージョン(オプション)を操作することができます。『STEP 7 での PROFINET』マニュアル (<http://support.automation.siemens.com/WW/view/en/49948856>)では、このようなタイプのプロジェクトを「スタンダードマシンプロジェクト」と呼んでいます。

STEP 7 のデバイスコンフィグレーションとユーザープログラムを、インストール済みの異なった PLC 構成にロードすることができます。ほんの少し簡単な調整を行うだけで、STEP 7 プロジェクトを実際のハードウェア構成に対応させることができます。

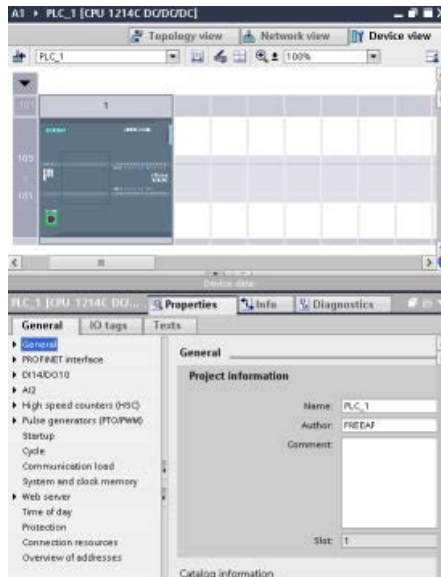
スタートアッププログラムブロックでプログラミングする制御データレコードは、コンフィグレーションと比較した実際の設置で抜けているモジュールや、コンフィグレーションと比較して異なったスロットに配置されているモジュールを CPU に通知します。コンフィグレーション制御はモジュールのパラメータ割り当てには影響を及ぼしません。

コンフィグレーション制御では、STEP 7 の最大デバイスコンフィグレーションから実際の構成を使用する限り、設置を柔軟に変更することができます。

コンフィグレーション制御の説明と例については、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。

5.6 CPU およびモジュールの動作設定

モジュールの動作パラメータを設定するには、デバイスビューでモジュールを選択し、インスペクタウィンドウの[プロパティ]タブを使用します。



以下の CPU のプロパティを設定できます。

- CPU の PROFINET IP アドレスと時刻の同期
- 電源オフからオンへの移行後の CPU のスタートアップ動作
- ローカル(オンボード)デジタルおよびアナログ I/O、高速カウンタ(HSC)、パルスジェネレータ
- システムクロック(時刻、タイムゾーン、サマータイム)
- CPU の読み取り/書き込み保護およびアクセス用パスワード
- 最大サイクルタイムまたは最小固定サイクルタイム、および通信負荷
- Web サーバーのプロパティ

CPU の STOP から RUN への動作を設定する

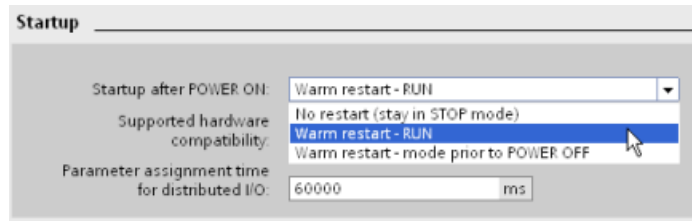
動作モードが STOP モードから RUN モードに変化すると、CPU はプロセスイメージ入力をクリアし、プロセスイメージ出力を初期化して、スタートアップ OB を処理します。(したがって、スタートアップ OB 内の命令が使用するプロセスイメージ入力の値は、現在の物理的な入力値ではなく 0 になります。)スタートアップ時に物理的な入力の現在の状態を読み取るには、即時読み取りを実行する必要があります。スタートアップ OB および関連の FC および FB が次に実行されます。複数のスタートアップ OB が存在する場合は、OB 番号の最も小さいものから、OB 番号の順序で、それぞれが実行されます。

また、スタートアップ処理では、CPU は次のタスクも実行します。

- スタートアップ中は、割り込みはキューに格納され、処理されません。
- スタートアップ中は、サイクルタイムの監視は行われません。
- スタートアップ時に、HSC (高速カウンタ)、PWM (パルス幅変調)、PtP (ポイントツーポイント通信)モジュールへの設定変更を行うことができます。
- 実際の HSC、PWM、PtP 通信モジュールの操作は、実行時にのみ発生します。

CPU は、スタートアップ OB の実行が終了すると RUN モードに移行し、連続スキャンサイクル内の制御タスクを処理します。

電源投入サイクルの後の CPU のスタートアップを設定するには CPU のプロパティを使用します。



- STOP モード
- RUN モード
- 前のモード(電源投入サイクルの前)

CPU は RUN モードに入る前にウォームリスタートを実行します。ウォームリスタートでは、すべての非保持型メモリがデフォルト値にリセットされますが、CPU は保持型メモリに格納されている現在の値を保持します。

注記

CPU は、ダウンロード後には常にリスタートを実行します。

プロジェクトの要素(プログラムブロック、データブロック、ハードウェアコンフィグレーションなど)を 1 つでもダウンロードすると、CPU は次回の RUN モードへの移行時に必ずリスタートを実行します。リスタートでは、入力のクリア、出力の初期化、非保持型メモリの初期化に加え、保持型メモリ領域の初期化も行われます。

ダウンロードに続けて実行されるリスタートの後は、以降のすべての STOP から RUN への移行時にウォームリスタートが実行されます（保持型メモリの初期化は行われません）。

5.6.1 システムメモリおよびクロックメモリ

「システムメモリ」および「クロックメモリ」用のバイトを有効設定にするには CPU のプロパティを使用します。プログラム内ロジックで、これらのファンクションの個別のビットをタグ名で参照することができます。

- システムメモリとして M メモリに 1 バイトを割り付けることができます。このシステムメモリのバイトの次の 4 ビットをユーザープログラムから以下のタグ名で参照することができます。
 - 最初の周期: (タグ名「FirstScan」) ビットは、スタートアップ OB の完了後の最初のスキンの間は 1 にセットされます。(最初のスキンの実行後に「First scan」ビットは 0 にセットされます)。
 - 変更された診断ステータス: (タグ名:「DiagStatusUpdate」) は、CPU が診断イベントをログに記録してから 1 スキンの間、1 にセットされます。プログラムサイクル OB の最初の実行が終了するまで CPU は「DiagStatusUpdate」ビットをセットしないため、ユーザープログラムで診断変更がスタートアップ OB の実行時に発生したのか、プログラムサイクル OB の最初の実行で発生したのかを検出することができません。
 - 常時 1 (高): (タグ名「AlwaysTRUE」) ビットは常に 1 にセットされます。
 - 常時 0 (低): (タグ名「AlwaysFALSE」) ビットは常に 0 にセットされます。
- クロックメモリとして M メモリに 1 バイトを割り付けることができます。クロックメモリとしてのバイトの各ビットは、方形波パルスを作成します。クロックメモリのバイトは、0.5 Hz (低速) ~ 10 Hz (高速) の 8 種類の周波数を供給します。これらのビットを制御ビットとして使用し、特にエッジ命令と組み合わせて、ユーザープログラム内の操作を周期的にトリガすることができます。

これらのバイトは、CPU の STOP モードから STARTUP モードへの移行時に初期化されません。クロックメモリの各ビットは、STARTUP モードと RUN モードの期間を通して同期して変化します。



注意

システムメモリまたはクロックメモリのビットの上書きに伴うリスク

システムメモリまたはクロックメモリのビットを上書きすると、これらのファンクションのデータが破損し、ユーザープログラムの誤動作が発生し、人的傷害や物的損害の原因になることがあります。

システムメモリもクロックメモリも M メモリ上の予約されている領域ではないため、命令または通信によってこれらのロケーションへの書き込みが発生し、データが破壊される可能性があります。

これらのファンクションの正しい動作を保証するために、これらのロケーションへのデータの書き込みを防止する必要があります。また、プロセスまたは機器には緊急停止回路を必ず実装してください。

システムメモリは、特定のイベントの間はオン(値 = 1)になるビットでバイトを構成します。

System memory bits

☒ Enable the use of system memory byte

Address of system memory byte (MBx):

First cycle:

Diagnostics status changed:

Always 1 (high):

Always 0 (low):

表 5-2 システムメモリ

7	6	5	4	3	2	1	0
予約済み 値 0				常時オフ 値 0	常時オン 値 1	診断ステータスインジ ケータ <ul style="list-style-type: none"> 1: 変更 0: 変更なし 	最初のスキャンインジ ケータ <ul style="list-style-type: none"> 1: スタートアップ後の 最初のスキャン 0: 最初のスキャンなし

クロックメモリは、一定の周期で個別のビットがオンとオフを繰り返す 1 バイトを構成します。それぞれのクロックビットは、M メモリの対応するビット上に方形波パルスを生成します。これらのビットを制御ビットとして使用し、特にエッジ命令と組み合わせて、ユーザーコード内の操作を周期的にトリガすることができます。

Clock memory bits

☒ Enable the use of clock memory byte

Address of clock memory byte (MBx):

10 Hz clock:

5 Hz clock:

2.5 Hz clock:

2 Hz clock:

1.25 Hz clock:

1 Hz clock:

0.625 Hz clock:

0.5 Hz clock:

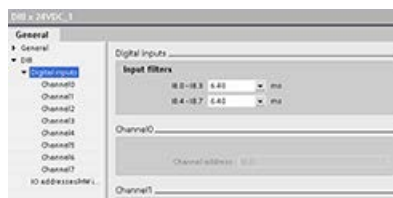
表 5-3 クロックメモリ

ビット番号	7	6	5	4	3	2	1	0
タグ名								
周期	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
周波数(Hz)	0.5	0.625	1	1.25	2	2.5	5	10

クロックメモリは CPU サイクルと非同期に実行されるため、クロックメモリのステータスは長いサイクルの間に複数回変わる可能性があります。

I/O および通信モジュールの動作を設定する

シグナルモジュール(SM)、シグナルボード(SB)、または通信モジュール(CM)の動作パラメータを設定するには、デバイスビューでモジュールを選択してインスペクタウィンドウの[プロパティ]タブを使用します。



シグナルモジュール(SM)とシグナルボード(SB)

- デジタル I/O: それぞれの入力を、エッジ検出および「パルスキャッチ」(瞬時の高または低パルス後オンまたはオフの状態を保持する)などに対して設定します。出力を、RUN モードから STOP モードへの移行時にフリーズまたは代替値を使用するために設定します。
- アナログ I/O: それぞれの入力(電圧/電流、範囲および平滑化)に対してパラメータを設定し、また、アンダーフローまたはオーバーフロー診断を有効にします。それぞれのアナログ出力に対してパラメータを設定し、短絡(電圧出力の場合)またはオーバーフロー値などの診断を有効にします。
- I/O アドレス: モジュールの入力および出力のセットに対する開始アドレスを設定します。



通信モジュール(CM)と通信ボード(CB)

- ポートの設定: ボーレート、パリティ、データビット、ストップビット、待ち時間などの通信パラメータを設定します。
- メッセージの伝送と受信: データの送信と受信に関連するオプションを設定します(メッセージ開始およびメッセージ終了パラメータなど)。

ユーザープログラムでこの設定パラメータを変更することもできます。

5.7 CPU の IP アドレスの設定

CPU には事前に設定された IP アドレスがないため、手動で IP アドレスを割り当てる必要があります。CPU のプロパティを設定するときに、PROFINET インターフェースの IP アドレスとその他のパラメータを設定します。

- PROFINET ネットワークの各デバイスには、製造メーカーによって識別用の一意のメディアアクセス制御アドレス(MAC アドレス)が割り付けられています。各デバイスには IP アドレスも割り付ける必要があります。
- サブネットは、接続されているネットワークデバイスの論理的な分類です。マスク(サブネットマスクまたはネットワークマスクとも呼ばれる)は、サブネットの境界を定義します。異なるサブネット間の接続のみがルータを経由します。ルーターは LAN 間のリンクで、IP アドレスを使用してデータパケットの送受信を行います。

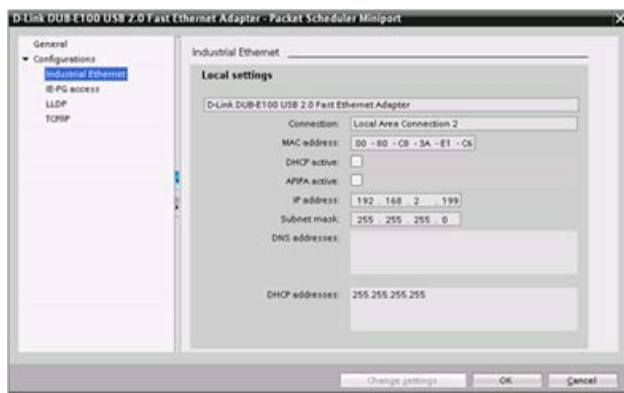
CPU に IP アドレスをダウンロードする前に、CPU の IP アドレスがプログラミングデバイスの IP アドレスと互換性があることを確認する必要があります。

STEP 7 を使用して、プログラミングデバイスの IP アドレスを決定できます。

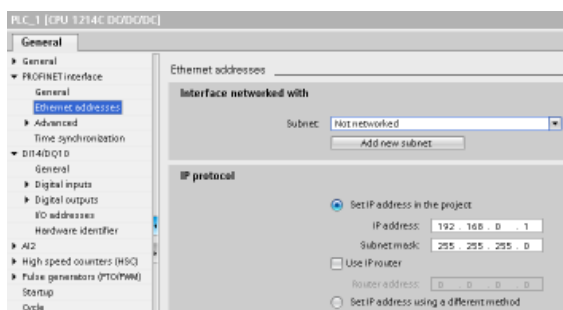
1. プロジェクトツリーの「オンラインアクセス」フォルダを展開して、ネットワークを表示します。
2. CPU に接続するネットワークを選択します。
3. 特定のネットワークを右クリックして、コンテキストメニューを表示します。
4. [プロパティ]コマンドを選択します。

注記

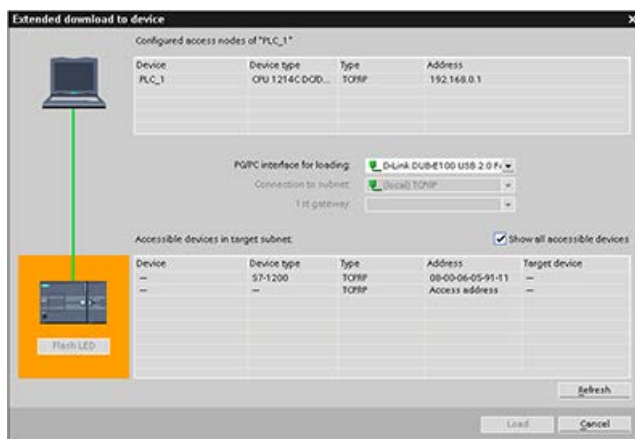
CPU の IP アドレスは、プログラミングデバイスの IP アドレスおよびサブネットマスクと互換性を持つ必要があります。CPU に最適な IP アドレスおよびサブネットマスクについては、ネットワーク担当者にご相談ください。



[プロパティ]ウィンドウにプログラミングデバイスの設定が表示されます。



CPU の IP アドレスとサブネットマスクを決定したら、CPU およびルーター(該当する場合)の IP アドレスを入力します。詳細については、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。



コンフィグレーションを完了した後、CPU にプロジェクトをダウンロードします。

CPU およびルーター(該当する場合)の IP アドレスは、プロジェクトをダウンロードすると設定されます。

5.8 CPU またはコードブロックへのアクセス保護が簡単

特定のファンクションへのアクセスを制限するために、CPU には 4 レベルのセキュリティが備わっています。CPU に対するセキュリティレベルおよびパスワードを設定して、パスワードを使用せずにアクセスできるファンクションとメモリ領域を制限します。

各レベルに、パスワードを使用せずにアクセスできるファンクションがあります。CPU のデフォルトの条件は無制限で、パスワード保護のない状態です。CPU へのアクセスを制限するには、CPU のプロパティを設定して、パスワードを入力します。

ネットワークを通じてのパスワードの入力は、CPU のパスワード保護情報を漏洩しません。パスワード保護は、通信ファンクションを含め、ユーザープログラムの実行には適用されません。正しいパスワードを入力することによって、すべてのファンクションに当該レベルでアクセスすることができます。

PLC 同士の通信(プログラムブロック内の通信命令を使用)は、CPU 内のセキュリティレベルによる制限を受けません。

表 5-4 CPU のセキュリティレベル

セキュリティレベル	アクセス制限
フルアクセス(保護なし)	パスワード保護を行わずに、すべてのファンクションにアクセスすることができます。
読み取りアクセス	パスワードを使用せずに、HMI と PLC 間通信の情報へのアクセスが許可されます。 CPU の修正(CPU への書き込み)および CPU のモード(RUN/STOP)の変更にはパスワードが必要です。
HMI アクセス	パスワードを使用せずに、HMI と PLC 間通信の情報へのアクセスが許可されます。 CPU 内のデータの読み取り、CPU の修正(CPU への書き込み)および CPU のモード(RUN/STOP)の変更にはパスワードが必要です。
アクセスなし(完全保護)	パスワード保護なしでのアクセスを許可しません。 HMI アクセス、CPU のデータの読み取り、および CPU の修正(CPU への書き込み)にはパスワードが必要です。

パスワードは大文字と小文字を区別します。保護レベルおよびパスワードを設定するには、次の手順を実行します。

1. [デバイスコンフィグレーション]で CPU を選択します。
2. インспекタウィンドウの[プロパティ]タブを選択します。
3. [保護]プロパティを選択し、保護レベルを選択して、パスワードを入力します。

The screenshot shows a 'Protection' dialog box with the following content:

Protection

Select the access level for the PLC.

Access level	Access			Access permission	
	HMI	Read	Write	Password	Confirmation
<input type="radio"/> Full access (no protection)	✓	✓	✓	*****	*****
<input type="radio"/> Read access	✓	✓		*****	*****
<input checked="" type="radio"/> HMI access	✓				
<input type="radio"/> No access (complete protection)					

この設定を CPU にダウンロードすると、HMI アクセス権が取得され、パスワードなしで HMI ファンクションにアクセスできます。データを読み取るには、「読み取りアクセス」に設定されたパスワードまたは「フルアクセス(保護なし)」に設定されたパスワードを入力する必要があります。データを書き込むには、「フルアクセス(保護なし)」に設定されたパスワードを入力する必要があります。



警告

保護された CPU への未許可のアクセス

CPU のフルアクセス権限を持つユーザーは、PLC 変数の読み取りおよび書き込み権限があります。CPU のアクセスレベルには関係なく、Web サーバーユーザーは PLC 変数の読み取りおよび書き込み権限を持つことができます。CPU に未許可でアクセス、または PLC 変数を無効な値に変更すると、プロセスオペレーションが混乱し、死傷事故などの重大な人的傷害や物的損害が発生するおそれがあります。

許可を得たユーザーは動作モードの変更、PLC データの書き込み、ファームウェアの更新を行うことができます。シーメンスは以下のセキュリティ慣行を守ることをお奨めします。

- パスワードで保護された CPU アクセスレベルとパワフルなパスワードを持つ Web サーバーユーザー ID (256 ページ)。パワフルなパスワードは文字、数字、特殊文字を組合わせた最低 10 文字の長さで、辞書にある言葉や個人情報から取得できる名前や識別子でないものです。パスワードは秘密とし、頻繁に変更してください。
- Web サーバーへのアクセスは、HTTPS プロトコルを使用したものだけを可能にします。
- Web サーバーの「Everybody」ユーザーのデフォルトの最低限の権限を拡張しないでください。
- Web ページユーザーは PLC 変数を無効な値に変更できるため、プログラムロジックの変数のエラーチェックと範囲チェックを行ってください。

接続メカニズム

PUT/GET 命令でリモート接続パートナーにアクセスするには、許可を得ている必要があります。

デフォルトでは、[PUT/GET 通信でのアクセスを許可]オプションは無効になっています。この場合、CPU データの読み取りおよび書き込みアクセスは、ローカル CPU と通信パートナーの両方に対する設定またはプログラミングが必要な通信接続でのみ可能です。たとえば、BSEND/BRCV 命令によるアクセスが可能です。

したがって、たとえば、ローカル CPU が単なるサーバーにすぎない接続(つまり、ローカル CPU に通信パートナーとの通信の設定/プログラミングが存在しない)は、CPU の動作中は行えません。

- 通信モジュールを介した PUT/GET、FETCH/WRITE、または FTP アクセス
- 他の S7 CPU からの PUT/GET アクセス
- PUT/GET 通信を介した HMI アクセス

クライアント側からの CPU データへのアクセスを許可する場合、つまり、CPU の通信サービスを制約したくない場合は、次の手順を実行します。

1. 保護アクセスレベルを「アクセスなし(完全保護)」以外の任意のレベルに設定します。
2. [リモートパートナーからの PUT/GET 通信によるアクセスを許可]チェックボックスを選択します。



この設定を CPU にダウンロードすると、CPU はリモートパートナーからの PUT/GET 通信を許可します。

5.8.1 ノウハウプロテクト

ノウハウプロテクトによって、作成したプログラム内の 1 つ以上のプログラムブロック(OB、FB、FC、または DB)への不正なアクセスを防ぐことができます。プログラムブロックへのアクセスを制限するにはパスワードを作成します。パスワード保護によって、そのプログラムブロックへの不正な読み取りや修正を防止することができます。パスワードを入力しないと、読み取ることのできる情報が、そのプログラムブロックに関する下記の情報に限定されます。

- ブロックタイトル、ブロックコメント、ブロックのプロパティ
- 転送パラメータ(IN、OUT、IN_OUT、Return)
- プログラムの呼び出し構造
- クロスリファレンスのグローバルタグ(使用ポイントに関する情報を除く)、ローカルタグは不可

ブロックに対して「ノウハウ」保護を設定すると、パスワードを入力しないと、そのブロック内のプログラムにアクセスできなくなります。

コードブロックの[プロパティ]タスクカードを使用して、当該ブロックのノウハウプロテクトを設定します。コードブロックを開いた後で、プロパティから[保護]を選択します。



1. コードブロックのプロパティで、[保護] ボタンをクリックして[ノウハウプロテクト]ダイアログを表示します。
2. [定義]ボタンをクリックしてパスワードを入力します。



パスワードを入力して確認した後、[OK]をクリックします。



5.8.2 コピー保護

もう1つのセキュリティ機能を使用して、プログラムブロックを特定のメモリカードまたはCPUで使用するためにバインドすることができます。この機能は特に、知的財産を保護するのに便利です。プログラムブロックを特定のデバイスにバインドすると、プログラムブロックまたはコードブロックの使用を特定のメモリカードまたはCPUでのみに制限することができます。この機能により、プログラムブロックまたはコードブロックを電子的に(インターネット経由や電子メールなどで)、またはメモリカートリッジの送付で配布することができます。コピー保護はOB(97 ページ)、FB(99 ページ)、およびFC(99 ページ)に使用可能です。S7-1200 CPUは3種類のブロック保護をサポートしています。

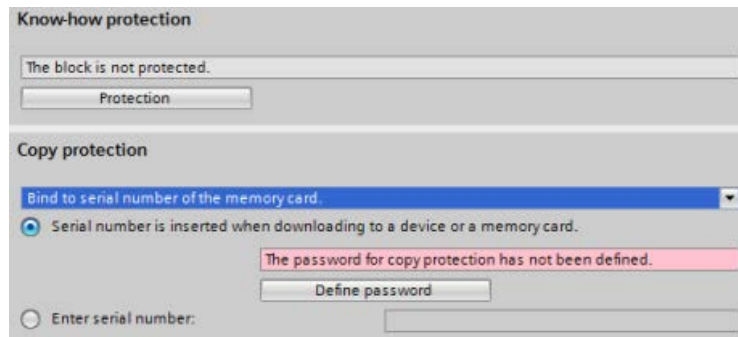
- CPUのシリアル番号をバインド
- メモリカードのシリアル番号をバインド
- パスワードによるプログラムブロックへシリアルバインド

コードブロックの[プロパティ]タスクカードを使用して、ブロックを特定の CPU またはメモ리카ードにバインドします。

1. コードブロックを開いた後で、プロパティから[保護]を選択します。



2. [コピー保護]タスクのドロップダウンリストから、使用するコピー保護のタイプを選択します。



3. CPU またはメモ리카ードのシリアル番号にバインドするには、[ダウンロード時にシリアル番号を挿入]を選択するか、メモ리카ードまたは CPU のシリアル番号を入力します。

注記

シリアル番号には大文字と小文字の区別があります。

パスワードによるシリアルバインドの場合は、ブロックのダウンロードまたはコピーに使用するパスワードを定義します。

その後に続けて動的バインドでブロックをダウンロードする場合は、ブロックをダウンロードできるようパスワードを入力する必要があります。コピー保護パスワードとノウハウプロテクト(91 ページ)パスワードは 2 つの別々のパスワードです。

カンタンプログラミング

6.1 ユーザープログラムの簡単設計

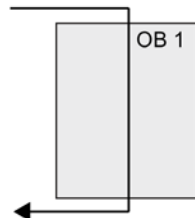
オートメーションタスク用のユーザープログラムを作成するには、そのプログラム用の命令をコードブロック(OB、FB、または FC)に挿入します。

ユーザープログラムの構造を選択する

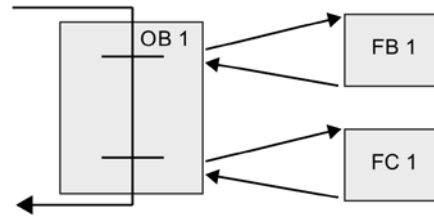
アプリケーションの要件に基づいてリニア構造またはモジュール構造のどちらかを選択し、ユーザープログラムを作成することができます。

- リニアプログラムは、オートメーションタスクのすべての命令を 1 つずつ順番に実行します。通常は、リニアプログラムでは、すべてのプログラム命令をプログラムのサイクルリック実行用の 1 つのプログラムサイクル OB(OB 1 など)に配置します。
- モジュールプログラムは、特定のタスクを実行するプログラムブロックを呼び出します。モジュール構造を作成するには、プロセスが実行する機能タスクに対応した小さなタスクに、複雑なオートメーションタスクを細分化します。それぞれのプログラムブロックが、それぞれの細分化されたタスクに対するプログラムセグメントを構成します。プログラムブロックから他のプログラムブロックを呼び出すことによって、プログラムを構造化します。

リニア構造

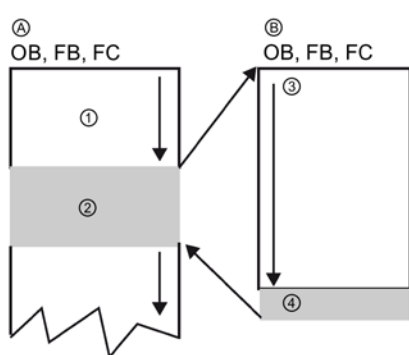


モジュール構造



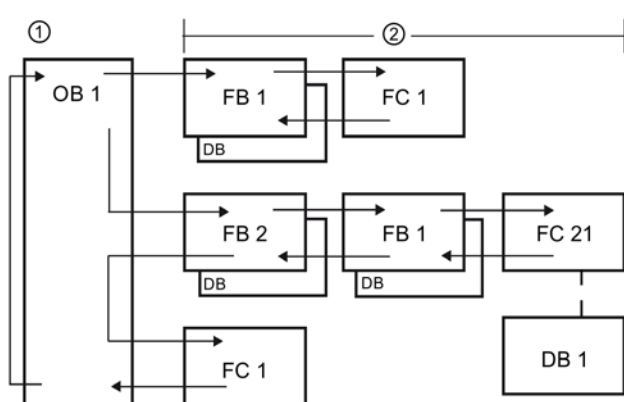
FB および FC を汎用のタスクを実行するように設計することによって、モジュール式のプログラムブロックを作成します。このように再利用できるモジュールを他のコードブロックから呼び出すようにして、ユーザープログラムを構造化します。呼び出し元のブロックから呼び出したブロックにデバイス固有のパラメータを渡します。プログラムブロックから他のプログラムブロックを呼び出すと、CPU は呼び出されたブロック内のプログラムコードを実行します。呼び出されたブロックの実行が終了すると、CPU は呼び出し元のブロックの実行に復帰します。処理は、ブロック呼び出しの後に続いている命令の実行に移ります。

OB を割り込みイベントに割り当てることもできます。イベントが発生すると、CPU は対応する OB のプログラムコードを実行します。OB の実行が完了すると、CPU は割り込みイベントが発生した時点のユーザープログラムの場所(スキャン内の任意の場所)で実行を再開します。



- A 呼び出しブロック(または割り込まれたブロック)
- B 呼び出された FB または BC (または割り込んだ OB)
- ① プログラムの実行
- ② 別のブロックの実行を開始する命令(または割り込みイベント)
- ③ プログラムの実行
- ④ ブロックの終り(呼び出し元のブロックに戻る)

ブロック呼び出しをネストしてモジュール構造を階層化することができます。以下の例では、ネストレベルは 3 です: プログラムサイクル OB とコードブロックへの 3 層の呼び出しの合計。



- ① サイクルの開始
- ② ネストレベル

ユーザープログラム内で再利用のできる汎用プログラムブロックを作成することによって、ユーザープログラムの設計および実装を単純化することができます。

- ポンプやモータの制御など、再利用のできる標準タスク用のプログラムブロックを作成することができます。また、このような汎用プログラムブロックをライブラリに格納しておき、さまざまなアプリケーションやソリューションで使用することもできます。
- 機能タスクに関連付けられたモジュール式コンポーネントを使用してユーザープログラムを構造化すると、プログラムデザインの理解および管理が簡単になります。モジュール式コンポーネントは、プログラム設計を標準化するのに役立つだけでなく、プログラムコードの更新または修正を短時間で簡単に実行できるようにするためにも役立ちます。
- モジュール式コンポーネントを作成することによって、プログラムのデバッグが単純化されます。完全なプログラムを一連のモジュール式プログラムセグメントとして構造化することによって、それぞれのプログラムブロックを開発した時点で機能試験を実行することができます。
- 特定の機能タスクに関連するモジュール方式の設計を使用することで、完成されたアプリケーションのコミショニングに必要な時間を短縮できます。

6.1.1 OBを使用したユーザープログラムの構成

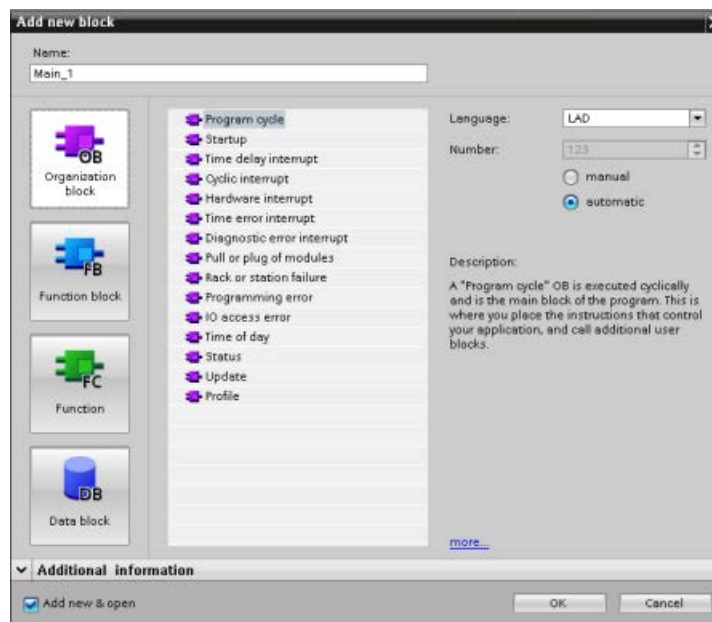
オーガニゼーションブロックは、プログラム用の構造を備えています。オペレーティングシステムとユーザープログラム間のインターフェースとして機能します。OBはイベント駆動型です。診断割り込みや時間間隔などのイベントが発生すると、CPUはOBを実行します。スタートイベントと動作が定義済みになっているOBも存在します。

作成したメインプログラムをプログラムサイクルOBに格納します。ユーザープログラム内に複数のプログラムサイクルOBを含めることができます。RUNモードでは、プログラムサイクルOBは最低のレベルの優先度で実行され、他のすべてのイベントタイプが割り込むことができます。スタートアップOBはRUNモードに移行する前に実行されるため、スタートアップOBがプログラムサイクルOBに割り込むことはありません。

CPUは、プログラムサイクルOBの処理を完了すると、そのプログラムサイクルOBの実行を直ちに繰り返します。この反復処理は、プログラマブルロジックコントローラに対して使用される「通常」の処理です。多くのアプリケーションでは、ユーザープログラム全体が単一のプログラムサイクルOB内に配置されます。

割り込みおよびエラーの処理や所定の周期で特定のプログラムコードを実行するために、特定の機能を実行するOBを作成することができます。これらのOBはプログラムサイクルOBの実行に割り込みを発生します。

ユーザープログラム内に新しいOBを作成するには[新しいブロックの追加]ダイアログを使用します。



割り込みの処理は常にイベント駆動によって行われます。このようなイベントが発生すると、CPUはユーザープログラムの実行を中断して、そのイベントを処理するように構成されているOBを呼び出します。割り込んだOBの実行が終了すると、CPUは、ユーザープログラムの実行を割り込み発生時点から再開します。

CPUが優先度によって、割り込みイベントの処理順序を決定します。複数の割り込みイベントを同じ優先度クラスに割り当てることができます。詳細は、オーガニゼーションブロック(59 ページ)およびユーザープログラムの実行(58 ページ)に関する項目を参照してください。

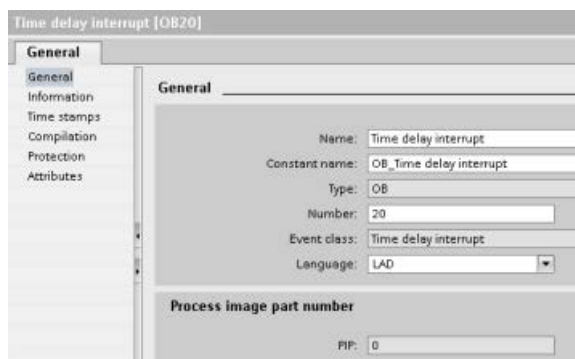
追加の OB を作成する

ユーザープログラム用、さらにはプログラムサイクルクラス、およびスタートアップ OB イベント用の OB を複数作成することができます。[新しいブロックの追加]ダイアログを使用して OB を作成し、OB の名前を入力します。

ユーザープログラム用に複数のプログラムサイクル OB を作成すると、CPU はそれぞれのプログラムサイクル OB を最下位の番号のプログラムサイクル OB (OB 1 など) から番号順に実行します。たとえば、最初のプログラムサイクル OB (OB 1 など) が終了すると、CPU は次に高い番号のプログラムサイクル OB を実行します。

OB のプロパティを設定する

OB のプロパティを変更することができます。たとえば、OB 番号やプログラミング言語を設定できます。



注記

PIP0、PIP1、PIP2、PIP3、または PIP4 に対応するプロセスイメージパート番号を OB に割り当てることができます。プロセスイメージパート番号の番号を入力すると、CPU が当該プロセスイメージパーティションを作成します。プロセスイメージパーティションの説明は、項目「ユーザープログラムの実行(58 ページ)」を参照してください。

6.1.2 FBおよびFCによりモジュール型タスクの簡単プログラミング

ファンクション(FC)はサブルーチンに似ています。FCは、一般に、一連の入力値に基づいて所定の動作を実行するコードブロックです。FCは、その動作の結果をメモリロケーションに格納します。下記のタスクを実行するにはFCを使用します。

- 数学的計算など、標準的で再使用可能な操作を実行する
- ビット論理演算を使用して、個別制御などの機能タスクを実行する

FCは、プログラム内のさまざまなポイントで何回も呼び出すことができます。FCを反復使用することによって、頻繁に反復されるタスクのプログラミングが単純化されます。

FBと異なり、FCには、関連付けられているインスタンスDBは存在しません。FCは演算の計算に使用されるデータにテンポラリメモリ(L)を使用します。このテンポラリデータは保存されません。このデータをFCの実行の終了後に使用するために保存するには、出力値をMメモリなどのグローバルメモリロケーションまたはグローバルDBに格納します。

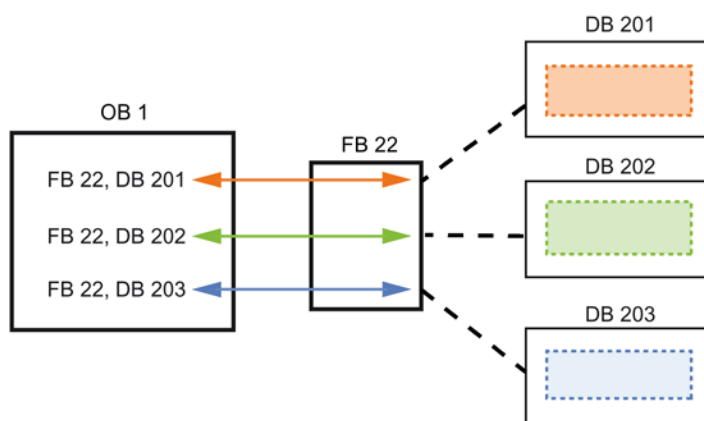
ファンクションブロック(FB)はメモリ付きのサブルーチンに似ています。FBは、呼び出しをブロックパラメータでプログラミングできるコードブロックです。FBは入力(IN)、出力(OUT)、および入力/出力(IN_OUT)パラメータを、データブロック(DB)または「インスタンス」DB内にある可変メモリに保存します。インスタンスDBには、FBのインスタンス(つまり呼び出し)に関連付けられたメモリブロックが用意され、FBの終了後にデータが格納されます。

通常は、1スキャンサイクルで動作が完了しないタスクまたはデバイスの動作を制御するために1つのFBを使用します。動作パラメータを保存しておいて次のスキャンから迅速にアクセスできるようにするために、ユーザープログラム内の各FBは複数のインスタンスDBを使用します。FBを呼び出すときに、FBのその呼び出し、つまり「インスタンス」のためのブロックパラメータの値と静的ローカルデータを格納するインスタンスDBを開きます。インスタンスDBには、FBが実行を終了した後の、これらの値が保持されます。

FBインターフェースで、パラメータに開始値を代入することができます。これらの値は関連付けられているDBに転送されます。パラメータに初期値を代入しなかった場合は、インスタンスDBに格納されている現在値が使用されます。ただし、パラメータに値を代入しなければならない場合もあります。

FBの呼び出しごとに異なるインスタンスDBを割り付けることができます。インスタンスDBを使用して、1つの汎用FBを複数のデバイスの制御に使用することができます。1つのプログラムブロックで1つのFBと1つのDBを呼び出すことによってプログラムを構造化します。このようにすれば、CPUは、そのFB内のプログラムコードを実行して、インスタンスDBにブロックパラメータと静的ローカルデータを格納します。FBの実行が終了すると、CPUは、そのFBを呼び出した元のプログラムブロックの実行に戻ります。インスタンスDBには、そのFBのインスタンスの値が保持されます。汎用制御タスク用のFBを設計することによって、FBの呼出しごとに異なるインスタンスDBを選択し、そのFBを複数のデバイスに対して再利用することができます。

呼出しごとに異なるデータブロックを使用して、同一のFBを3回呼び出すOBの例を下図に示します。この構造を使用して、それぞれのデバイスのための呼び出しごとに異なるインスタンスデータブロックを割り付けることによって、複数のモータなど、類似した複数のデバイスを1つの汎用FBで制御することができます。



それぞれの DB には、個別のデバイスのデータ(速度、起動時間、総運転時間など)が格納されます。この例では、FB 22 が 3 つのデバイスを制御し、DB 201 には最初のデバイスの運転データが格納され、DB 202 には 2 番目のデバイスの運転データが格納され、DB 203 には 3 番目のデバイスの運転データが格納されます。

6.1.3 データブロックを使用したプログラムデータの簡単保存

プログラムブロック用のデータを保存するために、ユーザープログラム内にデータブロック(DB)を作成することができます。ユーザープログラム内のすべてのプログラムブロックはグローバル DB 内のデータにアクセスできますが、インスタンス DB には特定のファンクションブロック(FB)のデータが格納されます。

ユーザープログラムは、入力(I)、出力(Q)、ビットメモリ(M)などのための CPU の特殊なメモリ領域にデータを保存することができます。また、データブロック(DB)を使用してプログラム自体に保存されているデータにすばやくアクセスすることもできます。

DB に格納されているデータは、データブロックが閉じられたり、関連付けられているコードブロックが終了しても削除されません。DB には次の 2 種類があります。

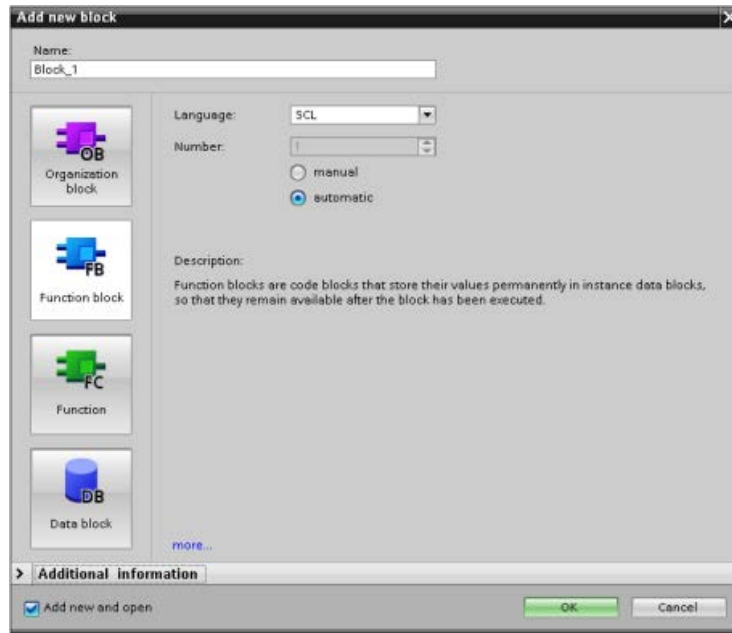
- グローバル DB - プログラム内のプログラムブロック用のデータが格納されます。グローバル DB 内のデータには、すべての OB、FB、FC からアクセスすることができます。
- インスタンス DB - 特定の FB 用のデータが格納されます。インスタンス DB 内のデータ構造は、その FB のパラメータ(Input、Output、および InOut)と静的データを反映しています。FB 用のテンポラリメモリはインスタンス DB には保存されません。

インスタンス DB は関連の FB 用のデータを反映していますが、どのプログラムブロックからもインスタンス DB 内のデータにアクセスすることができます。

6.1.4 コードブロックの新規作成

プログラムにコードブロックを新規に追加するには、次の手順を実行します。

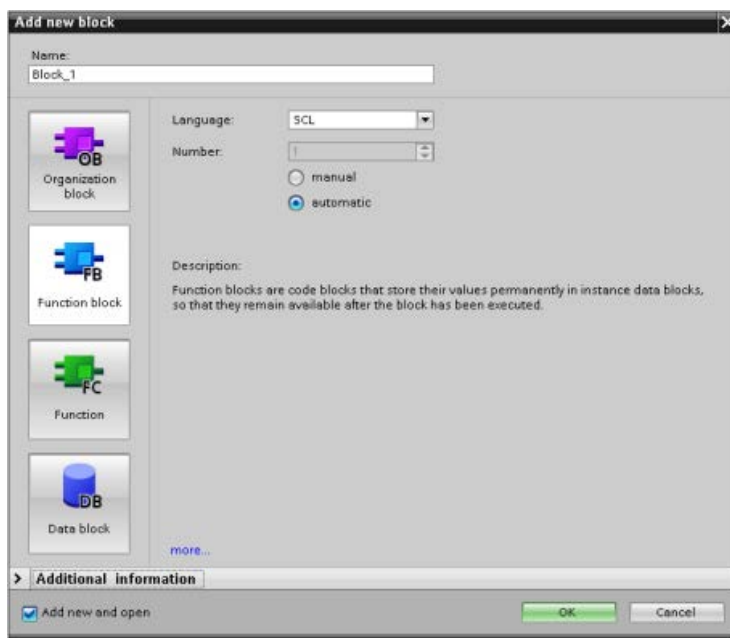
1. [プログラムブロック]フォルダを開きます。
2. [新しいブロックの追加]をダブルクリックします。
3. [新しいブロックの追加]ダイアログで、追加するブロックのタイプをクリックします。たとえば、FC を追加するには[ファンクション(FC)]アイコンをクリックします。
4. ドロップダウンメニューからコードブロックのプログラミング言語を選択します。



5. [OK]をクリックしてプロジェクトにブロックを追加します。

[新規追加して開く]オプション(デフォルト)を選択すると、STEP 7 がエディタに新規に作成されたブロックを開きます。

6.1.5 再利用できるプログラムブロックの作成



OB、FB、FC、グローバルDBを作成するには、プロジェクトナビゲータの[プログラムブロック]にある[新しいブロックの追加]ダイアログを使用します。

コードブロックを作成するときは、そのブロック用のプログラミング言語を選択します。DBの場合はデータを格納するだけですから、プログラミング言語の選択は不要です。

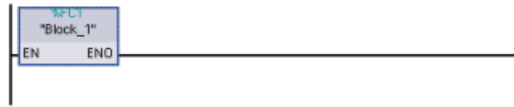
[新規追加して開く]チェックボックス(デフォルト)を選択すると、プロジェクトビューにコードブロックが開かれます。

再使用したいオブジェクトをライブラリに保存することができます。プロジェクトごとに、そのプロジェクトに接続されているプロジェクトライブラリが存在します。プロジェクトライブラリに加えて、複数のプロジェクトにわたって使用できるグローバルライブラリを好きな数だけ作成できます。ライブラリは互いに互換性があるため、ライブラリエレメントをライブラリ間でコピーして移動することができます。

ライブラリはたとえば、最初にプロジェクトライブラリにペーストしたのちに、そこでさらに開発するブロック用のテンプレートを作成するのに使用されます。最終的には、プロジェクトライブラリからグローバルライブラリにブロックをコピーします。グローバルライブラリをプロジェクトで作業する同僚が使用できるようにします。同僚はブロックを使用し、必要に応じて個々の要件に合わせてさらにブロックを調整します。

ライブラリの操作についての詳細は、STEP 7 オンラインヘルプのライブラリの項目を参照してください。

6.1.6 別のコードブロックからコードブロックの呼び出し



ユーザープログラム内のどのコードブロック(OB、FB、または FC)からでも、簡単に CPU 内の FB または FC を呼び出すことができます。

1. 他のブロックを呼び出すコードブロックを開きます。
2. プロジェクトツリーで、呼び出されるコードブロックを選択します。
3. ブロックを選択したネットワークにドラッグし、コードブロックへの呼び出しを作成します。

注記

OB はイベント駆動型(60 ページ)であるため、ユーザープログラムは OB を呼び出すことはできません。CPU はイベントの受信に応答して OB の実行を開始します。

6.2 使いやすいプログラミング言語

STEP 7 では、S7-1200 用に以下の標準プログラミング言語が用意されています。

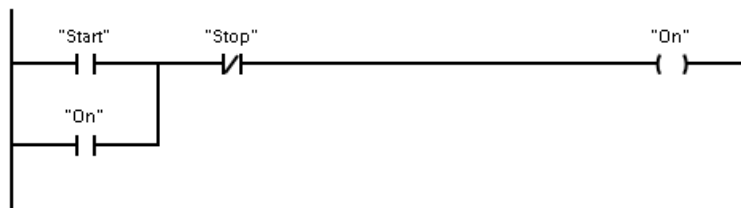
- LAD (ラダーロジック)はグラフィカルなプログラミング言語です。表現は回路図(104 ページ)がベースです。
- FBD (ファンクションブロックダイアグラム)は、Boolean 代数(105 ページ)で使用するグラフィカルロジックシンボルをベースにしたプログラミング言語です。
- SCL (構造化制御言語)はテキストベースの高度プログラミング言語(105 ページ)です。

コードブロックを作成するときは、そのブロックによって使用されるプログラミング言語を選択します。

ユーザープログラムは、一部またはすべてのプログラミング言語で作成されたコードブロックを使用できます。

6.2.1 ラダーロジック(LAD)

ノーマルクローズとノーマルオープンおよびコイルなどの回路図の要素をリンクしてネットワークを形成します。



複雑な操作を行うロジックを作成するために、分岐を挿入して並列回路のロジックを作成することができます。並列分岐は下向きに開いているか、母線に直接接続されます。分岐は上向きに終端します。

LAD には、math (演算)、timer (タイマー)、counter (カウンタ)、move (ムーブ) など、さまざまな機能のための「空ボックス」命令が用意されています。

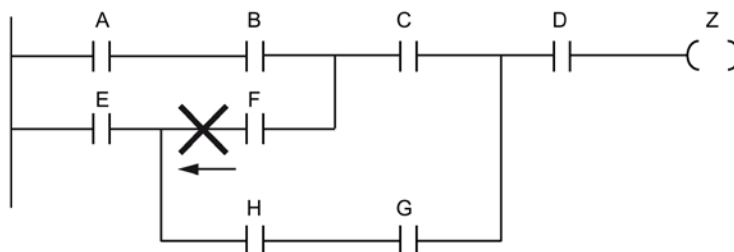
STEP 7 は LAD ネットワーク内の命令(行と列)の数を制約しません。

注記

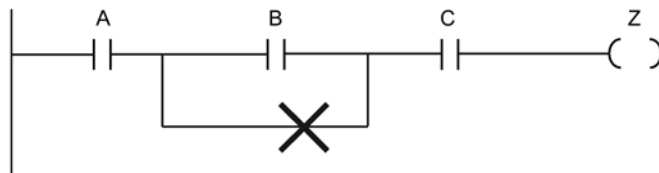
どの LAD ネットワークも coil (コイル) またはボックス命令で終端する必要があります。

LAD ネットワークを作成するときは、下記のルールを考慮する必要があります。

- パワーフローが逆向きになるような分岐を作成することはできません。

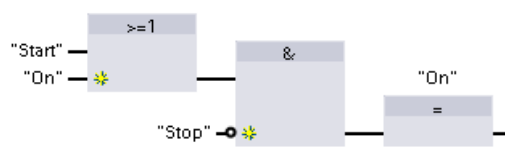


- 短絡を発生させる分岐を作成することはできません。



6.2.2 ファンクションブロックダイアグラム(FBD)

FBD も、LAD と同じようにグラフィカルなプログラミング言語です。論理表現は、ブール代数で使用されているグラフィカルな論理記号がベースになっています。



複雑な操作を行うロジックを作成するには、ボックス間に並列分岐を挿入します。

数学関数などの複雑な関数は、ロジックボックスを併用して直接表現することができます。
STEP 7 は FBD ネットワーク内の命令(行と列)の数を制約しません。

6.2.3 SCLの概要

構造化制御言語(SCL)は、SIMATIC S7 CPU 用の高度な PASCAL ベースのプログラミング言語です。SCL は STEP 7 のブロック構造をサポートします。SCL で書かれたプログラムブロックを LAD および FBD で書かれたプログラムブロックと一緒にインクルードすることもできます。

SCL 命令は標準的なプログラミング演算子(割り当て(:=)、数学関数(加算は+、減算は-、乗算は*、除算は/)を使用します。SCL は標準的な PASCAL プログラム制御演算子(IF-THEN-ELSE、CASE、REPEAT-UNTIL、GOTO、および RETURN)を使用します。SCL プログラム言語の構文エレメントには、どの PASCAL 参照でも使用できます。タイマおよびカウンタなどのその他の SCL の命令の多くは、LSD および FBD 命令に一致しています。

SCL は PASCAL と同様に条件付き処理、ループ、ネスト制御構造を提供するため、LAD または FBD よりも簡単に SCL で複雑なアルゴリズムを実現できます。

以下の例に、さまざまな用途用のさまざまな式を示します。

"C" := #A+#B;	2つのローカル変数をタグに割り当てます
"Data_block_1".Tag := #A;	データブロックタグへの割り当て
IF #A > #B THEN "C" := #A;	IF-THEN 文の条件
"C" := SQRT (SQR (#A) + SQR (#B));	SQRT 命令のパラメータ

高度なプログラミング言語として、SCL は基本タスクの標準ステートメントを使用します。

- 割り当てステートメント: :=
- 数学関数: +、-、*、および/
- グローバル変数(タグ)のアドレス指定: "<タグ名>" (二重引用符で囲ったタグ名またはデータブロック名)
- ローカル変数のアドレス指定: #<変数名> (先頭に「#」記号が付いた変数名)
- 絶対アドレス指定: %<絶対アドレス>、たとえば%I0.0または%MW10

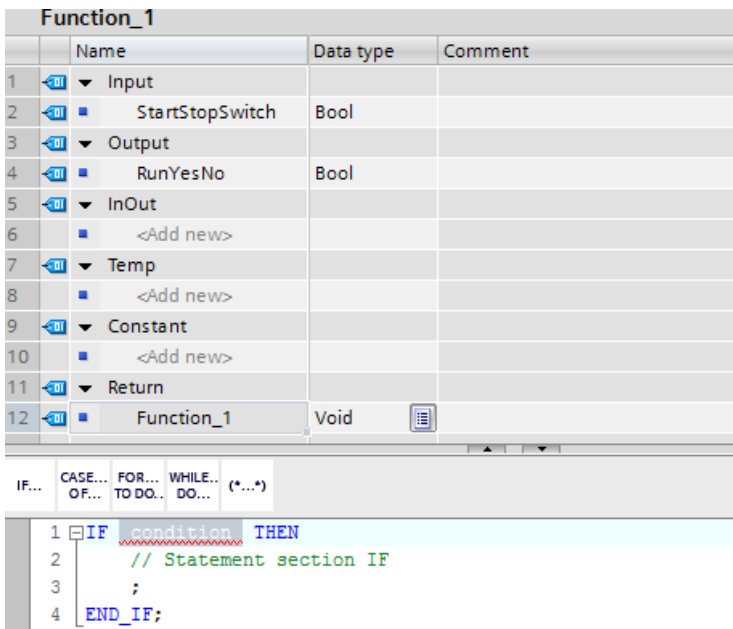
演算演算子は各種の数値データタイプを処理できます。結果のデータタイプは、最上位オペランドのデータタイプによって決まります。たとえば、INT オペランドと REAL オペランドを使用する乗算演算子により、結果として REAL 値が生成されます。

6.2.4 SCL プログラミングエディタ

ブロックを作成するときに、SCL プログラミング言語を使用する任意のタイプのブロック (OB、FB、または FC) を指定できます。STEP 7 は以下のエレメントを含む SCL プログラミングエディタを提供します。

- コードブロックのパラメータを定義するインターフェースセクション
- プログラムコードのコードセクション
- CPU がサポートする SCL 命令を含む命令ツリー

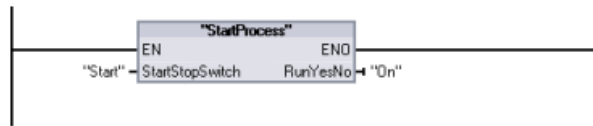
命令の SCL コードを直接、コードセクションに入力します。エディタにはコマンドコード構造とコメント用のボタンが含まれています。より複雑な命令の場合は、SCL 命令を命令ツリーからドラッグして自分のプログラムにドロップするだけです。また、任意のテキストエディタを使用して SCL プログラムを作成し、そのファイルを STEP 7 にインポートすることもできます。



SCL コードブロックのインターフェースセクションで、以下のパラメータのタイプを宣言できます。

- Input、Output、InOut、および Ret_Val: これらのパラメータは入力タグ、出力タグ、コードブロックの戻り値を定義します。ここで入力するタグ名は、コードブロックの実行中にローカルに使用されます。通常はタグテーブルのグローバルタグ名を使用しません。
- Static(FB のみ; 上の例は FC の場合です): コードブロックは静的タグを使用して静的中間結果をインスタンスデータブロックに保存します。ブロックは静的データを数サイクル後に上書きされるまで保持します。このブロックがマルチインスタンスとして呼び出すブロックの名前も静的ローカルデータに保存されます。
- Temp: このパラメータはコードブロックの実行中に使用されるテンポラリタグです。
- Constant: コードブロック用の名前のついた定数値です。

SCL コードブロックを別のコードブロックから呼び出す場合、SCL コードブロックのパラメータが入力または出力として表示されます。



この例では、「Start」および「On」のタグ（プロジェクトのタグテーブルの）が SCL プログラムの宣言テーブルの「StartStopSwitch」および「RunYesNo」に対応しています。

6.3 パワフルな命令により簡単プログラミング

6.3.1 ユーザーが期待する基本命令の提供

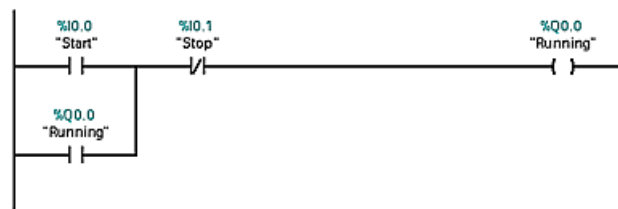
S7-1200 CPU は多くの命令をサポートします。命令は STEP 7 の命令ツリーの以下のグループから使用できます。

- 基本命令
- 拡張命令
- テクノロジー
- 通信命令

すべての命令の完全な要約は、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。このマニュアルでは多数の共通命令について説明します。

ビットロジック命令

ビットロジック命令の基本は接点とコイルです。接点はビットのステータスを読み取り、一方、コイルは演算のステータスをビットに書き込みます。



接点はビットのバイナリステータスをテストし、オン(1)の場合は結果は「パワーフロー」、オフ(0)の場合は結果は「パワーフローなし」になります。

コイルの状態は先行ロジックのステータスを反映します。

同じアドレスのコイルを複数のプログラムロケーションで使用する場合、ユーザープログラムでの最後の計算結果が、出力の更新時の物理的な出力に書き込まれる値のステータスを決定します。

常時開
接点

"IN"
—|—

常時閉
接点

"IN"
—|/—

ノーマルオープンは、割り付け済みビットの値が 1 に等しいときに閉じています (ON)。

ノーマルクローズは割り付け済みビットの値が 0 に等しいときに閉じています (ON)。

ビットロジック演算の基本命令は AND 論理または OR 論理のいずれかです。直列に接続した接点は AND 論理ネットワークを構成します。並列に接続した接点は OR 論理ネットワークを構成します。

接点と接点を接続して、独自の組合せ論理回路を作成することができます。指定した入力ビットがメモリ識別子 I (入力) または Q (出力) を使用している場合、ビット値はプロセスイメージレジスタから読み取られます。制御プロセス内の物理接点信号は PLC の入力端子に配線されています。CPU は配線されている入力信号をスキャンし、プロセスイメージ入力レジスタの対応する状態値を更新します。

入力用のタグの後ろに「:P」を使用して物理的入力の即時読み取りを指定できます (「Motor_Start:P」または「I3.4:P」など)。即時読み取りでは、ビットデータ値は、プロセスイメージではなく物理入力から直接読み取られます。即時読み取りではプロセスイメージの更新は行われません。

出力コイル

"OUT"
—()—

反転出力コイル

"OUT"
—(/)—

以下の出力が、出力および反転出力コイルを介したパワーフローの結果として得られます。

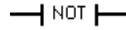
- 出力コイルを通るパワーフローが存在する場合、出力ビットが 1 にセットされます。
- 出力コイルを通るパワーフローが存在しない場合、出力ビットが 0 にセットされます。
- 反転出力コイルを通るパワーフローが存在する場合、出力ビットが 0 にセットされます。
- 反転出力コイルを通るパワーフローが存在しない場合、出力ビットが 1 にセットされます。

コイル出力命令は出力ビットに値を書き込みます。指定した出力ビットがメモリ識別子 Q を使用している場合、CPU は、プロセスイメージレジスタ内のその出力ビットをオンまたはオフにセットして、指定したビットがパワーフローステータスに等しくなるように設定します。制御アクチュエータへの出力信号は PLC の出力端子に配線されています。RUN モードでは、CPU システムは、常時入力信号のスキャンを行い、入力の状態をプログラムロジックに基づいて処理し、プロセスイメージ出力レジスタに新しい出力状態の値を設定することで応答します。プログラム実行サイクルを実行するたびに、CPU は、プロセスイメージレジスタに格納されている新しい出力状態応答を配線先の出力端子に転送します。

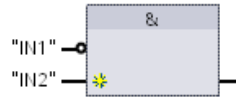
出力用のタグの後ろに「:P」を使用して物理的出力の即時書き込みを指定できます (「Motor_On:P」または「Q3.4:P」など)。即時書き込みでは、ビットデータ値がプロセスイメージ出力に書き込まれるほか、物理出力にも直接書き込まれます。

コイルはネットワークの終端に制限されません。コイルを LAD ネットワークの段の中間、接点間、または他の命令間に挿入することができます。

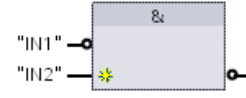
NOT 接点インバータ
(LAD)



反転論理入力付きの
AND ボックス(FBD)



反転論理入力および出力付きの
AND ボックス(FBD)



LAD での NOT 接点は、パワーフロー入力の論理状態を反転します。

- NOT 接点にパワーフローが存在しない(OFF)場合、パワーフローを出力します(ON)。
- NOT 接点にパワーフローが存在する(ON)場合、パワーフロー出力は存在しません(OFF)。

FBD プログラミングでは、[お気に入り]ツールバーまたは命令ツリーから[InvertN RLO]ツールをドラッグして入力または出力にドロップすると、そのボックスコネクタにロジックインバータが作成されます。

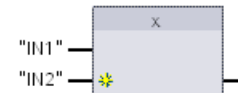
AND ボックス(FBD)



OR ボックス(FBD)



XOR ボックス(FBD)



- AND ボックスの出力が TRUE であるためには、すべての入力が TRUE でなければなりません。
- OR ボックスの出力が TRUE であるためには、どちらか 1 つの入力が TRUE でなければなりません。
- XOR ボックスの出力が TRUE であるためには、どちらか一方の入力が TRUE、もう一方の入力が FALSE でなければなりません。

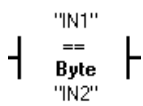

FBD プログラミングでは、LAD の接点ネットワークは AND (&)、OR (>=1)、排他的 OR (x) ボックスネットワークで表現され、ボックス入力およびボックス出力用のビット値を指定することができます。また、他のロジックボックスに接続して、独自の論理回路を構成することもできます。ネットワーク内にボックスを配置した後で、[お気に入り]ツールバーまたは命令ツリーから[入力の挿入]ツールをドラッグしてボックスの入力側にドロップし、入力を追加することができます。ボックス入力コネクタを右クリックして[入力の挿入]を選択することもできます。

ボックス入力および出力を他のロジックボックスに接続したり、接続されていない入力に対してビットアドレスやビットシンボル名を入力することができます。ボックス命令が実行されると、現在の入力の状態がバイナリボックスロジックに適用され、True の場合はボックス出力が True になります。

6.3.2 Comparator命令とMove命令

Comparator 演算は、同じデータタイプの 2 つの値の比較を実行します。

表 6- 1 Comparator 演算

命令	SCL	説明
LAD: 	<pre> out := in1 = in2; out := in1 <> in2; out := in1 >= in2; out := in1 <= in2; out := in1 > in2; out := in1 < in2; </pre>	<ul style="list-style-type: none"> 等しい(==):IN1 が IN2 と等しい場合、比較は True です。 等しくない(<>):IN1 が IN2 と等しくない場合、比較は True です。 等しいか大きい(>=):IN1 が IN2 と等しいか大きい場合、比較は True です。 等しいか小さい(<=):IN1 が IN2 と等しいか小さい場合、比較は True です。 大きい(>):IN1 が IN2 より大きい場合、比較は True です。 小さい(<):IN1 が IN2 より小さい場合、比較は True です。
FBD: 		

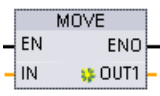
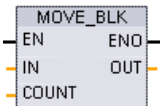
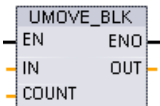
¹ LAD および FBD の場合: 比較が TRUE の場合、接点が有効になるか(LAD)、ボックス出力が TRUE になります(FBD)。

Comparator 演算についての追加情報は、『S7-1200 プログラマブルコントローラシステム マニュアル』を参照してください。

Move 演算は、データエレメントを新しいメモリアドレスにコピーして、データタイプの変換を行うことができます。ムーブ処理によって、元のデータに変更が加えられることはありません。

- MOVE は指定したアドレスに格納されているデータエレメントを、新しいアドレスにコピーしますさらに出力を追加する場合は、OUT1 パラメータの隣にあるアイコンをクリックします。
- MOVE_BLK (割り込み可能ムーブ)および UMOVE_BLK (割り込み不可能ムーブ)は、データエレメントのブロックを新しいアドレスにコピーします。MOVE_BLK 命令および UMOVE_BLK 命令には COUNT パラメータが存在します。この COUNT パラメータを使用して、コピーするデータ要素の数を指定します。コピーされる要素あたりのバイト数は、PLC タグテーブル内の IN および OUT パラメータタグの名前に割り付けられているデータタイプに依存します。

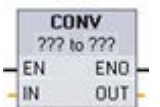
表 6-2 MOVE、MOVE_BLK、および UMOVE_BLK 命令

LAD / FBD	SCL	説明
	<pre>out1 := in;</pre>	<p>指定したアドレスに格納されているデータエレメントを、新しいアドレスまたは複数のアドレスにコピーします。さらに出力を LAD または FBD に追加する場合は、出力パラメータの隣にあるアイコンをクリックします。SCL の場合は、複数の割り当てステートメントを使用します。以下のループ構造のどれかを使用することもできます。</p>
	<pre>MOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out);</pre>	<p>データエレメントを新しいアドレスにコピーする、割り込み可能なムーブ命令。</p>
	<pre>UMOVE_BLK(in:=_variant_in, count:=_uint_in out=>_variant_out);</pre>	<p>データエレメントのブロックを新しいアドレスにコピーする割り込み不可能なムーブ命令。</p>

Move 演算についての追加情報は、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。

6.3.3 変換操作

表 6-3 変換操作

LAD / FBD	SCL	説明
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	<p>データエレメントのデータタイプを別のデータタイプに変換します。</p>

- ¹ LAD および FBD の場合: ボックス名の下をクリックし、ドロップダウンメニューでデータタイプを選択します。ソースのデータタイプを選択すると、そのデータタイプから変換することのできるデータタイプがドロップダウンリストに表示されます。
- ² SCL の場合: 入力パラメータ(in)および出力パラメータ(out)のデータタイプを指定して、変換命令を作成します。たとえば、DWORD_TO_REAL は DWord 値を Real 値に変換します。

表 6-4 Round 命令と Truncate 命令



LAD / FBD	SCL	説明
	<pre>out := ROUND (in);</pre>	<p>実数(Real または LReal)を整数に丸めます。この命令は、実数を最も近い整数に丸めます(IEEE - 近似値への丸め)。数が 2 つの整数の間値である場合は(10.5 など)、命令はその数を偶数に丸めます。たとえば、</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12 <p>LAD/FBD の場合は、命令ボックスの[??]をクリックして、出力のデータタイプ(たとえば、「DInt」)を選択します。SCL の場合、デフォルトの出力データタイプは DINT です。別の出力データタイプに丸めるには、命令名にデータタイプの明示的な名前をつけて入力します(たとえば、ROUND_REAL または ROUND_LREAL)。</p>
	<pre>out := TRUNC(in);</pre>	<p>実数(Real または LReal)を整数に丸めます。実数の小数点以下の部分は切り捨てられゼロになります(IEEE - 0 への丸め)。</p>

表 6-5 Seiling (CEIL)命令と Floor 命令



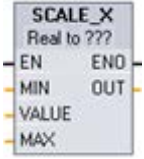
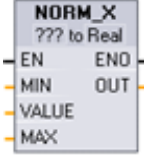
LAD / FBD	SCL	説明
	<pre>out := CEIL(in);</pre>	<p>実数(Real または LReal)を選択された実数以上の最も近い整数に変換します(IEEE 「正の無限大への丸め」)。</p>
	<pre>out := FLOOR(in);</pre>	<p>実数(Real または LReal)を選択された実数以下の最も近い整数に変換します(IEEE 「負の無限大への丸め」)。</p>

表 6-6 SCALE_X 命令と NORM_X 命令

LAD / FBD	SCL	説明
	<pre>out := SCALE_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>スケーリングされた実数パラメータ VALUE ($0.0 \leq \text{VALUE} \leq 1.0$)を、MIN および MAX パラメータで指定したデータタイプおよび値の範囲にスケーリングします。</p> <p>$\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$</p>
	<pre>out := NORM_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>パラメータ VALUE を MIN および MAX パラメータで指定した値の範囲に正規化します。</p> <p>$\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN})$、ただし($0.0 \leq \text{OUT} \leq 1.0$)</p>

¹ 等価 SCL: $\text{out} := \text{value} (\text{max} - \text{min}) + \text{min}$; ²等価 SCL: $\text{out} := (\text{value} - \text{min}) / (\text{max} - \text{min})$;

6.3.4 Calculate命令を使用した簡単数値計算

表 6-7 CALCULATE 命令

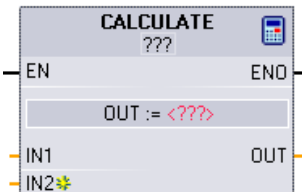
LAD / FBD	SCL	説明
	<p>標準の SCL 数式を使用して計算式を作成します。</p>	<p>CALCULATE 命令により、入力(IN1、IN2、.. INn)で動作し、定義した計算式に従って OUT の結果を作成する数学関数を作成できます。</p> <ul style="list-style-type: none"> まずデータタイプを選択します。すべての入力および出力は、同じデータタイプでなければなりません。 さらに入力を追加するには、最後の入力にあるアイコンをクリックします。

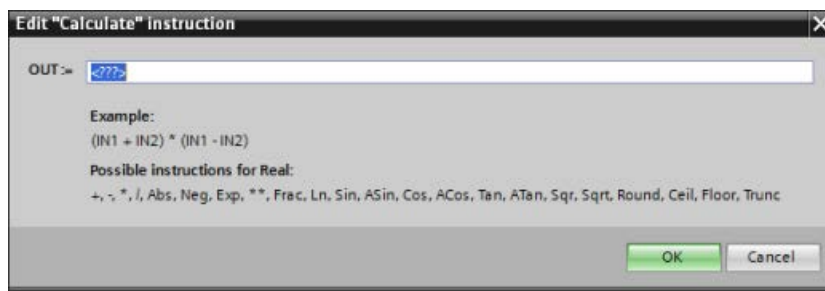
表 6-8 パラメータのデータタイプ

パラメータ	データタイプ ¹
IN1、IN2、..INn	SInt、Int、DInt、USInt、UInt、UDInt、Real、LReal、Byte、Word、DWord
OUT	SInt、Int、DInt、USInt、UInt、UDInt、Real、LReal、Byte、Word、DWord

¹ IN および OUT パラメータは同じデータタイプでなければなりません(入力パラメータの暗黙的な変換により)。たとえば、入力の SINT 値は、OUT が INT または REAL の場合、INT または REAL 値に変換されます。

計算機アイコンをクリックしてダイアログを開き、数学関数を定義します。入力(IN1 および IN2 などの)および演算として、計算式を入力します。[OK]をクリックして関数を保存すると、ダイアログが Calculate 命令用の入力を自動的に作成します。

ダイアログには、例と OUT パラメータのデータタイプに基づいて使用可能な命令のリストが表示されます。



注記

関数のすべての定数用の入力も作成する必要があります。定数値は、CALCULATE 命令用の関連づけられた入力に入力されます。

定数を入力として入力することで、CALCULATE 命令を関数を変更する必要なしにユーザープログラムの他のロケーションにコピーできます。これで、関数を変更せずに命令用の入力の値またはタグを変更できます。

CALCULATE が実行され、計算内のすべての個々の演算が正常に完了すると、ENO = 1 となります。そうでない場合は、ENO = 0 となります。

CALCULATE 命令の例は、「複雑な数学計算式用の CALCULATE 命令の使用(48 ページ)」を参照してください。

6.3.5 タイマの動作

S7-1200 は以下のタイマをサポートします。

- TP タイマはプリセットした幅のパルスを生成します。
- TON タイマはプリセットした時間後に出力(Q)をオンに設定します。
- TOF タイマは出力(Q)をオンに設定した後、プリセットした時間後にオフにリセットします。
- TONR タイマはプリセットした遅延後に出力(Q)をオンに設定します。経過した時間は、リセット(R)入力を使用して経過時間をリセットするまで、複数の時間にわたって累積されます。
- PT (プリセットタイマ)コイルは指定されたタイマに、新しいプリセットされた時間値をロードします。
- RT (リセットタイマ)コイルは指定されたタイマをリセットします。

LAD および FBD の場合、これらの命令はボックス命令または出力コイルのいずれかとして使用可能です。

ユーザープログラム内で使用できるタイマの数は、CPU のメモリ量によってのみ制約されます。それぞれのタイマはメモリの 16 バイトを使用します。

それぞれのタイマは、データブロック内に格納された構造体を使用して、タイマデータを保持します。SCL の場合、まず個々のタイマ命令用の DB を作成しないと DB を参照できません。LAD および FBD の場合、命令を挿入すると STEP 7 が自動的に DB を作成します。

DB を作成すると、マルチインスタンス DB も使用できます。タイマデータは単一の DB に含まれていて、タイマごとに個別の DB は必要ないため、タイマを処理するための処理時間が短縮されます。共有するマルチインスタンス DB 内のタイマデータ構造体間での相互作用はありません。

表 6-9 TP (パルスタイマ)

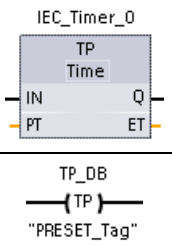
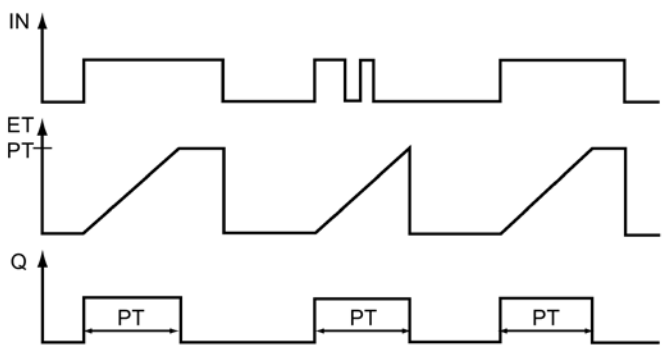
LAD / FBD	SCL	タイミングダイアグラム
	<pre>"timer_db".TP(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表 6-10 TON (オンディレイタイマ)

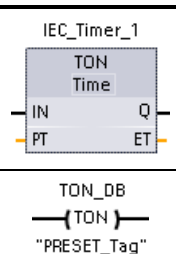
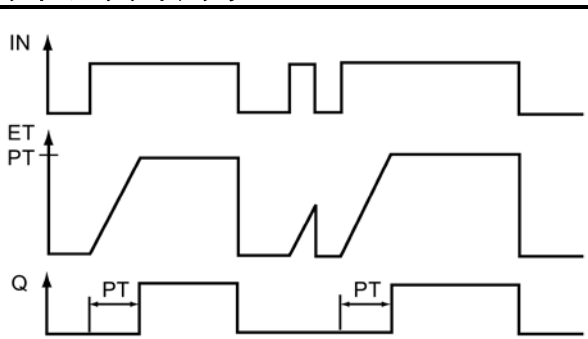
LAD / FBD	SCL	タイミングダイアグラム
<p>IEC_Timer_1</p> 	<pre>"timer_db".TON(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表 6-11 TOF (オフディレイタイマ)

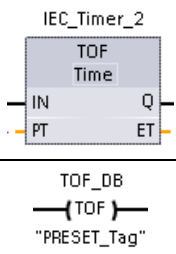
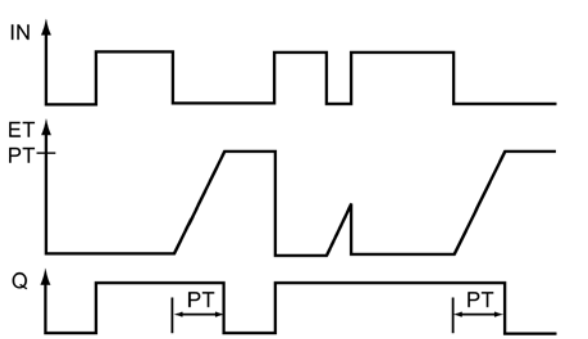
LAD / FBD	SCL	タイミングダイアグラム
<p>IEC_Timer_2</p> 	<pre>"timer_db".TOF(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表 6-12 TONR (保持型オンディレイタイマ)

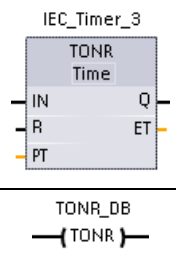
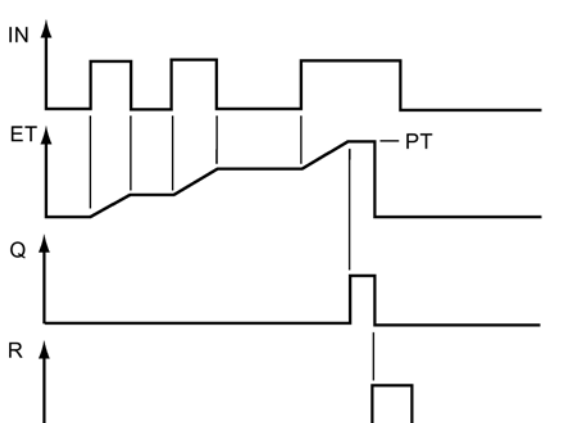
LAD / FBD	SCL	タイミングダイアグラム
<p>IEC_Timer_3</p> 	<pre>"timer_db".TONR(IN:=_bool_in_, R:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表 6- 13 プリセットタイマ-(PT)- コイル命令とリセットタイマ -(RT)- コイル命令

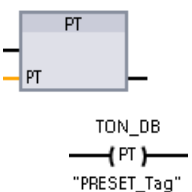
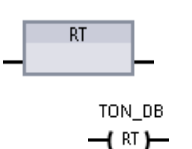
LAD / FBD	SCL	説明
	<pre>PRESET_TIMER(PT:=_time_in_, TIMER:=_iec_timer_in_);</pre>	<p>プリセットタイマ-(PT)-コイル命令とリセットタイマ-(RT)-コイル命令は、ボックスまたはコイルタイマのいずれかで使用します。これらのコイル命令はミッドライン位置に配置できます。コイル出力パワーフローステータスは常に、コイル入力ステータスと同じです。</p> <ul style="list-style-type: none"> • -(PT)-コイルが有効の場合、指定された IEC_Timer DB データの PRESET タイムエレメントは「PRESET_Tag」時間に設定されます。 • -(RT)-コイルが有効の場合、指定された IEC_Timer DB データの ELAPSED タイムエレメントは 0 にリセットされます。
	<pre>RESET_TIMER(_iec_timer_in_);</pre>	

表 6- 14 パラメータのデータタイプ

パラメータ	データタイプ	説明
ボックス: IN コイル: パワーフロー	Bool	TP、TON、および TONR: ボックス: 0=タイマの無効化、1=タイマの有効化 コイル: パワーフローなし=タイマの無効化、パワーフローあり=タイマの有効化 TOF: ボックス: 0=タイマの有効化、1=タイマの無効化 コイル: パワーフローなし=タイマの有効化、パワーフローあり=タイマの無効化
R	Bool	TONR ボックスのみ: 0=リセットなし 1= 経過時間および Q ビットを 0 にリセット
ボックス: PT コイル: 「PRESET_Tag」	Time	タイマボックスまたはコイル: プリセット時間入力
ボックス: Q コイル: DBdata.Q	Bool	タイマボックス: Q ボックス出力またはタイマ DB データの Q ビット タイマコイル: タイマ DB データの Q ビットのみをアドレス指定できます
ボックス: ET コイル: DBdata.ET	Time	タイマボックス: ET (経過時間)ボックス出力またはタイマ DB データの ET 時間値 タイマコイル: タイマ DB データの ET 時間値のみをアドレス指定できます

表 6- 15 PT パラメータと IN パラメータの値の変化の影響

タイマ	PT ボックスパラメータと IN ボックスパラメータおよび対応するコイルパラメータの値の変化
TP	<ul style="list-style-type: none"> このタイマの実行中に PT が変化しても、影響はありません。 このタイマの実行中に IN が変化しても、影響はありません。
TON	<ul style="list-style-type: none"> このタイマの実行中に PT が変化しても、影響はありません。 このタイマの実行中に IN が FALSE に変化すると、タイマはリセットされ停止します。
TOF	<ul style="list-style-type: none"> このタイマの実行中に PT が変化しても、影響はありません。 このタイマの実行中に IN が TRUE に変化すると、タイマはリセットされ停止します。
TONR	<ul style="list-style-type: none"> このタイマの実行中に PT が変化しても影響はありませんが、タイマの復帰時に影響を与えます。 このタイマの実行中に IN が FALSE に変化すると、タイマは停止しますがリセットは行われません。IN が再び TRUE に変化すると、タイマは累積された時間値から計時を再開します。

PT (プリセット時間)および ET (経過時間)の値は、ミリ秒単位の時間を表す符号付き倍精度整数として指定された IEC_TIMER DB データに格納されます。TIME データには識別子 T# が使用され、単純な時間単位(T#200ms または 200)および複合時間単位(T#2s_200ms など)として入力することができます。

表 6- 16 TIME データタイプのサイズと範囲

データタイプ	サイズ	有効な数字の範囲 ¹
TIME	32 ビット、DInt データとして保存	T#-24d_20h_31m_23s_648ms～T#24d_20h_31m_23s_647ms -2,147,483,648 ms～+2,147,483,647 ms として保存

¹ 上に示した TIME データタイプの負の値の範囲はタイマ命令には使用できません。負の PT (プリセット時間)値は、タイマ命令の実行時にゼロにセットされます。ET (経過時間)は常に正の値です。

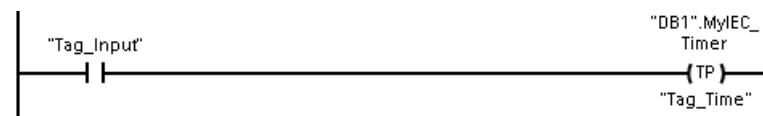
タイマのプログラミング

ユーザープログラムを計画して作成する場合は、以下のタイマ動作の結果を考慮する必要があります。

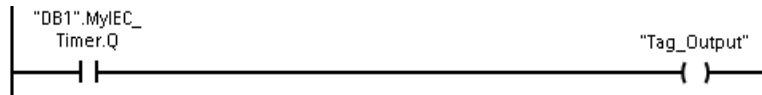
- 同じスキャンでタイマを複数回更新できます。タイマはタイマ命令(TP、TON、TOF、TONR)が実行されるたびに、およびタイマ構造の ELAPSED または Q メンバが、実行された別の命令のパラメータとして使用されるたびに更新されます。最新の時間データ(基本的にはタイマの即時読み取り値)が必要な場合、これは利点になります。ただし、プログラムスキャン全体で整合性のある値を持つ必要がある場合は、タイマ命令をこの値を必要とする他のすべての命令の前に配置し、タイマ DB 構造の ELAPSED および Q メンバではなく、タイマ命令の Q および ET 出力のタグを使用してください。
- タイマの更新が発生しないスキャンを行うこともできます。タイマをファンクションで起動し、1 つ以上のスキャンに対して当該ファンクションを再度呼び出すのをやめることが可能です。タイマ構造の ELAPSED または Q メンバを参照する他の命令が実行されない場合、タイマは更新されません。タイマ命令が再度実行されるか、タイマ構造の ELAPSED または Q をパラメータとして使用して他の命令が実行されるまで、新しい更新は発生しません。
- 通常は行いませんが、同じ DB タイマ構造を複数のタイマ命令に割り当てることもできます。通常は、予期しない相互作用を避けるために、DB タイマ構造あたり 1 つのタイマ命令(TP、TON、TOF、TONR)のみを使用してください。

セルフリセットタイマは、周期的に発生のあるアクションをトリガするのに便利です。通常、セルフリセットタイマは、タイマビットを参照するノーマルクローズをタイマ命令の前に配置して作成します。このタイマネットワークは、タイマビットを使用してアクションをトリガする 1 つ以上の依存型ネットワークに配置するのが普通です。タイマの期限が切れると(経過時間がプリセット値に達する)、タイマビットは 1 回のスキャンに対してオンになり、タイマビットで制御される依存型ネットワークロジックの実行が可能になります。タイマネットワークの次の実行時に、ノーマルクローズがオフになり、タイマがリセットされてタイマビットがクリアされます。次のスキャンでノーマルクローズがオンになり、タイマが再起動されます。このようなセルフリセットタイマを作成する場合は、タイマ DB 構造の「Q」メンバをタイマ命令の前に配置されたノーマルクローズのパラメータとして使用しないでください。代わりに、タイマ命令の「Q」出力に接続されたタグを使用してください。タイマ DB 構造の Q メンバへのアクセスを避ける理由は、それによってタイマの更新が引き起こされ、タイマがノーマルクローズによって更新された場合、接点がタイマ命令を即時にリセットするからです。タイマ命令の Q 出力は 1 回のスキャンに対してオンにならず、依存型ネットワークが実行されません。

-(TP)-、-(TON)-、-(TOF)-、および-(TONR)-タイマコイルは、ネットワーク内の最後の命令でなければなりません。タイマの例で示したように、後続のネットワーク内の接点命令はタイマコイルの IEC_Timer DB データの Q ビットを評価します。同様に、プログラムで経過時間値を使用したい場合は、IEC_timer DB データ内の ELAPSED エlement をアドレス指定する必要があります。



パルスタイマは、Tag_Input ビット値が 0 から 1 に移行するときに開始されます。タイマは Tag_Time 時間値で指定された時間のあいだ稼働します。



タイマが稼働している間は、DB1.MyIEC_Timer.Q=1、Tag_Output 値=1 の状態になります。Tag_Time 値が経過すると、DB1.MyIEC_Timer.Q=0、Tag_Output 値=0 になります。

6.3.6 カウンタの動作

内部プログラムイベントおよび外部プロセスイベントのカウントを行うにはカウンタ命令を使用します。

- 「カウントアップ」カウンタ(CTU)は、入力パラメータ CU の値が 0 から 1 に変化すると 1 カウント加算します。
- 「カウントダウン」カウンタ(CTD)は、入力パラメータ CD の値が 0 から 1 に変化すると 1 カウント減算します。
- 「カウントアップおよびダウン」カウンタ(CTUD)は、カウントアップ(CU)入力またはカウントダウン(CD)入力が 0 から 1 に変化すると 1 カウント加算または減算します。

S7-1200 には、OB 実行速度よりも速い速度で発生するイベントをカウントするための高速カウンタ(131 ページ) (HSC)もあります。

CU、CD、および CTUD 命令はソフトウェアカウンタを使用しているため、配置されている OB の実行速度によって最大カウント速度が制限されます。

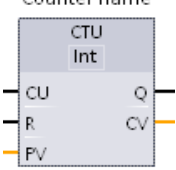
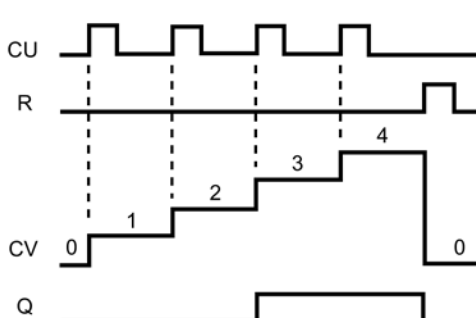
注記

カウント対象のイベントが OB の実行速度内で発生した場合は、CTU、CTD、または CTUD カウンタ命令を使用します。イベントが DB の実行速度より速い速度で発生した場合は、HSC を使用します。

それぞれのカウンタは、データブロック内に格納された構造体を使用して、カウンタデータを保持します。SCL の場合、まず個々のカウンタ命令用の DB を作成しないと命令を参照できません。LAD および FBD の場合、命令を挿入すると STEP 7 が自動的に DB を作成します。

ユーザープログラム内で使用できるカウンタの数は、CPU のメモリ量によってのみ制約されます。それぞれのカウンタは 3 バイト(SInt または USInt)、6 バイト(Int または UInt)、または 12 バイト(DInt または UDInt)を使用します。

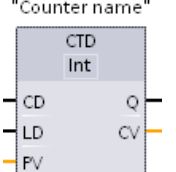
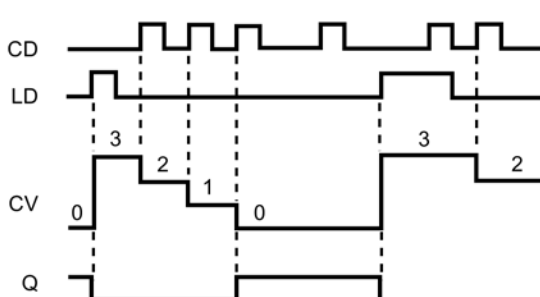
表 6- 17 CTU (カウントアップ)カウンタ

LAD / FBD	SCL	演算
<p>"Counter name"</p> 	<pre>"ctu_db".CTU(CU:=_bool_in, R:=_bool_in, PV:=_in_, Q=>_bool_out, CV=>_out_);</pre>	

タイミングダイアグラムに符号なし整数のカウンタ値を使用した CTU カウンタの動作を示します(PV = 3 の場合)。

- パラメータ CV (現在のカウンタ値)の値がパラメータ PV (プリセットカウンタ値)の値以上になった場合は、カウンタ出力パラメータ Q が 1 になります。
- リセットパラメータ R の値が 0 から 1 に変化すると、CV は 0 にリセットされます。

表 6- 18 CTD (カウントダウン)カウンタ

LAD / FBD	SCL	演算
<p>"Counter name"</p> 	<pre>"ctd_db".CTD(CD:=_bool_in, LD:=_bool_in, PV:=_in_, Q=>_bool_out, CV=>_out_);</pre>	

タイミングダイアグラムに符号なし整数のカウンタ値を使用した CTD カウンタの動作を示します(PV = 3 の場合)。

- パラメータ CV の値(現在のデータ値)が 0 以下になるとカウンタの出力パラメータ Q が 1 になります。
- パラメータ LD の値が 0 から 1 に変化すると、パラメータ PV (プリセット値)の値が新しい CV としてカウンタにロードされます。

表 6- 19 CTUD (カウントアップおよびダウン)カウンタ

LAD / FBD	SCL	演算
<div> <div> Counter name </div> <div> <div>CTUD</div> <div>Int</div> <div>CU</div> <div>QU</div> <div>CD</div> <div>QD</div> <div>R</div> <div>CV</div> <div>LD</div> <div>PV</div> </div> </div>	<div> ctud_db".CTUD(<div>CU:=_bool_in,</div> <div>CD:=_bool_in,</div> <div>R:=_bool_in,</div> <div>LD:=_bool_in,</div> <div>PV:=_in_,</div> <div>QU=>_bool_out,</div> <div>QD=>_bool_out,</div> <div>CV=>_out_);</div> </div>	<div> <div>CU</div> <div>CD</div> <div>R</div> <div>LD</div> <div>CV</div> <div>QU</div> <div>QD</div> </div>

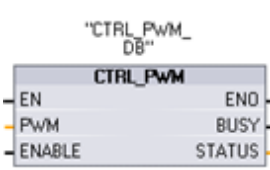
- タイミングダイアグラムに符号なし整数のカウンタ値を使用した CTUD カウンタの動作を示します(PV = 4 の場合)。

 - パラメータ CV (現在のカウンタ値)の値がパラメータ PV (プリセット値)の値以上になった場合は、カウンタ出力パラメータ QU が 1 になります。
 - パラメータ CV の値がゼロ以下になった場合は、カウンタの出力パラメータ QD が 1 になります。
 - パラメータ LD の値が 0 から 1 に変化すると、パラメータ PV の値が新しい CV としてカウンタにロードされます。
 - リセットパラメータ R の値が 0 から 1 に変化すると、CV は 0 にリセットされます。

6.3.7 パルス幅振幅(PWM)

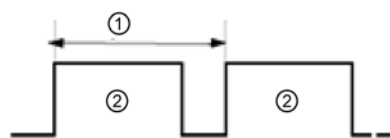
CTRL_PWM 命令は拡張命令の Pulse グループで使用可能です。

表 6- 20 CTRL_PWM 命令

LAD / FBD	SCL	説明
	<pre>"ctrl_pwm_db"(PWM:=W#16#0, ENABLE:=False, BUSY=>_bool_out_, STATUS=>_word_out_);</pre>	<p>CTRL_PWM 命令では、可変デューティサイクルによる固定サイクルタイム出力が可能です。PWM 出力は、開始後に指定された周波数(サイクルタイム)で継続的に実行されます。パルス幅は、目的の制御を行えるように必要に応じて変更可能です。</p>

CTRL_PWM 命令をコードブロックに挿入するときに、[呼び出しオプション]ダイアログから命令用の DB を作成します。CTRL_PWM 命令はパラメータ情報を DB に保存し、データブロックパラメータを制御します。

パルス幅は、CPU が最初に RUN モードに入ると、デバイスコンフィグレーションで設定された初期値に設定されます。パルス幅の変更が必要な場合は、デバイスコンフィグレーションで指定されたワード長出力(Q)に値を書き込みます([出力アドレス]/[開始アドレス])。Move、Convert、Math、または PID などの命令を使用して、指定されたパルス幅を適切なワード長の出力(Q)に書き込みます。出力値の有効な範囲を使用してください(パーセント、1000 分の 1、10,000 分の 1、または S7 アナログフォーマット)。



- ① サイクルタイム
- ② パルス幅時間

デューティサイクルはたとえば、サイクルタイムのパーセンテージ、または相対数量(0~1000 または 0~10000 など)として表現できます。パルス幅は 0 (パルスなし、常にオフ)からフルスケール(パルスなし、常にオン)まで変更可能です。

PWM 出力は 0 からフルスケールまで可変であるため、アナログ出力と似たデジタル出力を提供します。例えば、PWM 出力はモーターの速度を停止からフルスピードまで制御するために使用できます。または、バルブの位置を閉じた状態から完全に開いた状態まで制御するために使用できます。

6.4 データログの簡単作成

ユーザーの制御プログラムは、データログ命令を使用してランタイムデータ値を不揮発性ログファイルに保存することができます。データログファイルは、フラッシュメモリ(CPU またはメモリカード)に保存されます。ログファイルデータは、標準 CSV (Comma Separated Value)フォーマットで保存されます。データレコードは、あらかじめ定義されたサイズの循環ログファイルとして保存されます。

データログ命令をユーザーのプログラムで使用して、ログファイルを作成して開き、レコードを書き込んで閉じることができます。ユーザーは、単一のログデータレコードを定義するデータバッファを作成し、ロギングされるプログラム値を決定します。ユーザーのデータバッファは、新規ログレコードの一時保管場所として使用されます。新しい現在値をランタイム時にプログラムによってバッファに移動する必要があります。現在のデータ値がすべて更新されると、DataLogWrite 命令を実行してデータをバッファからデータログレコードに転送できるようになります。

Web サーバーの[フィアルブラウザ]ページから、データログファイルを開いて、編集、保存、名前の変更、削除を行うことができます。ファイルブラウザを表示するには読み取り権限が、データログファイルの編集、削除、名前の変更を行うには変更権限が必要です。

DataLog 命令を使用して、プログラムによってランタイムプロセスデータを CPU のフラッシュメモリに保存します。データレコードは、あらかじめ定義されたサイズの循環ログファイルとして保存されます。新しいレコードがデータログファイルに付加されます。データログファイルに最大数のレコードが保存されると、次のレコードの書き込みによって一番古いレコードが上書きされます。データレコードの上書きを防止するには、DataLogNewFile 命令を使用します。新しいデータレコードは新しいデータログファイルに保存され、古いデータログファイルは CPU 内にとどまります。

表 6- 21 DataLogWrite 命令

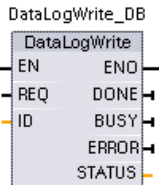
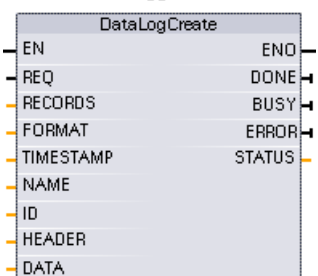
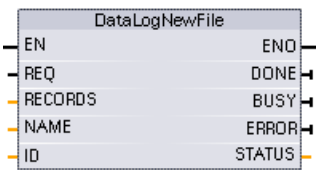
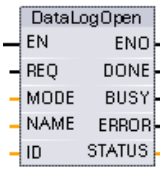
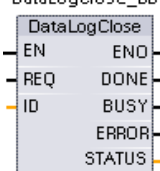
LAD/FBD	SCL	説明
	<pre>"DataLogWrite_DB"(req:=FALSE, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogWrite は指定されたデータログにデータレコードを書き込みます。事前に存在していたターゲットデータログが開いている必要があります。</p> <p>プログラムによって、現在のランタイムデータ値の入ったレコードバッファをロードし、DataLogWrite 命令を実行して新しいレコードデータをバッファからデータログに移動してください。</p> <p>DataLogWrite 操作が完了しないうちに電力障害が発生した場合、データログに転送中のデータレコードは失われるおそれがあります。</p>

表 6- 22 DataLogCreate 命令と DataLogNewFile 命令

LAD/FBD	SCL	説明
<p>DataLogCreate_DB</p> 	<pre>"DataLogCreate_DB" (req:=FALSE, records:=1, format:=1, timestamp:=1, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_variant_in_, ID:=_dword_inout_, header:=_variant_inout_, data:=_variant_inout_);</pre>	<p>DataLogCreate¹はCPUの\DataLogsディレクトリに保存されるデータログファイルを作成して初期化します。データログファイルは事前に決められた固定サイズで作成されます。</p>
<p>DataLogNewFile_DB</p> 	<pre>"DataLogNewFile_DB" (req:=FALSE, records:=1, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_variant_in_, ID:=_dword_inout_);</pre>	<p>DataLogNewFile¹を使用して、ユーザーのプログラムで既存のデータログファイルに基づいた新しいデータログファイルを作成することができます。新しいデータログが、指定された名前で作成され暗黙的に開かれます。ヘッダーレコードが元のデータログから、元のデータログプロパティとともに複製されます。元のデータログファイルは暗黙的に閉じられます。</p>

¹ DataLogCreate 動作と DataLogNewFile 動作は多数のプログラムスキャンサイクルにわたって実行されます。ログファイルの作成に必要な実際の時間は、レコードの構造と数によって異なります。新しいデータログが他のデータログ操作に対して使用可能になる前に、ユーザーのプログラムロジックが、DONE ビットの TRUE への移行をモニタする必要があります。

表 6- 23 DataLogOpen 命令と DataLogClose 命令

LAD/FBD	SCL	説明
<p>DataLogOpen_DB</p> 	<pre>"DataLogOpen_DB" (req:=FALSE, mode:=0, name:=_variant_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogOpen 命令が事前に存在していたデータログファイルを開きます。ログに新規データレコードを書き込む前に、データログを開く必要があります。データログは個別に開いたり閉じることができます。8つのデータログを同時に開くことができます。</p>
<p>DataLogClose_DB</p> 	<pre>"DataLogClose_DB" (req:=FALSE, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogClose 命令は開いているデータログファイルを閉じます。閉じているデータログに対する DataLogWrite 操作はエラーとなります。別の DataLogOpen 演算が実行されるまで、このデータログへの書き込み動作は許可されません。STOP モードに移行すると、開いているすべてのデータログファイルが閉じられます。</p>

6.5 ユーザープログラムの簡単モニタとテスト

6.5.1 ウォッチテーブルとフォーステーブル

オンライン CPU によって実行されているユーザープログラムのモニタおよび値の修正を行うには、「ウォッチテーブル」を使用します。多様な試験環境をサポートするために、プロジェクト内にさまざまなウォッチテーブルを作成および保存することができます。試運転中やサービスおよびメンテナンスのために、さまざまな試験を繰り返し実施することができます。

ウォッチテーブルを使用して、ユーザープログラムを実行しながら、CPU のモニタおよび情報のやり取りを行うことができます。コードブロックおよびデータブロックのタグだけではなく、入力および出力(I および Q)、ダイレクトアクセスの入力(I:P)、ビットメモリ(M)、データブロック(DB)など、CPU のメモリ領域の値を表示および変更することができます。

ウォッチテーブルを使用して、STOP モードの CPU の物理出力(Q:P)を有効設定にすることができます。たとえば、CPU への配線の試験時に、特定の値を出力に割り付けることができます。

STEP 7 にはタグを特定の値に「強制設定」するためのフォーステーブルもあります。この強制設定の詳細については、「オンラインおよび診断」の章の「CPU に値を強制設定する 342 ページ」のセクションを参照してください。

注記

強制値はウォッチテーブルではなく、CPU に保存されます。

入力(すなわち「I」アドレス)は強制設定できません。ただし、ダイレクトアクセスの入力は強制設定できます。ダイレクトアクセスの入力を強制設定するには、アドレスに「:P」を付加します(例:「On:P」)。

STEP 7 は、トリガ条件に基づいてプログラム変数をトレースして記録する(355 ページ)することもできます。

6.5.2 クロスリファレンスの使用状況表示

インスペクタウィンドウには、ユーザープログラム、CPU、任意の HMI デバイスなどの選択されたオブジェクトのプロジェクト全体での使用状況に関するクロスリファレンス情報が表示されます。[クロスリファレンス]タブには、選択されたオブジェクトが使用されている場所と、他のオブジェクトがそのオブジェクトを使用している場所が表示されます。インスペクタウィンドウには、オンラインでのみクロスリファレンスで利用できるブロックも含まれています。クロスリファレンスを表示するには、[クロスリファレンスを表示]コマンドを選択します。(プロジェクトビューで、[ツール]メニューでクロスリファレンスを探してください。)

注記

クロスリファレンス情報を見るためにエディタを閉じる必要はありません。

クロスリファレンスのエントリをソートすることができます。クロスリファレンスリストに、ユーザープログラム内のメモリアドレスとタグの使用状況の概要が表示されます。

- プログラムの作成および変更を行う場合は、使用したオペランド、タグ、ブロック呼び出しの概要を保持します。
- クロスリファレンスから、オペランドおよびタグを使用したポイントに直接ジャンプできます。
- プログラムテスト中またはトラブルシューティング時に、どのメモリロケーションがどのブロック内のどのコマンドで処理中であるか、どのタグがどの画面で使用中であるか、どのブロックが他のどのブロックで呼び出されているかが通知されます。

表 6- 24 クロスリファレンスのエレメント

列	説明
オブジェクト	下位レベルのオブジェクトを使用している、または、下位レベルのオブジェクトに使用されているオブジェクトの名前
番号	使用回数
使用ポイント	それぞれの使用場所、たとえばネットワーク
プロパティ	リファレンスオブジェクトの特殊プロパティ、たとえば、マルチインスタンス宣言のタグ名など
用途	インスタンス DB がテンプレートとして使用されるか、マルチインスタンスとして使用されるかなどの、オブジェクトについての追加情報
アクセス	アクセスのタイプ(オペランドへのアクセスが読み取りアクセス(R)および/または書き込みアクセス(W)であるか)
アドレス	オペランドのアドレス
タイプ	オブジェクトの作成に使用するタイプと言語の情報
パス	プロジェクトツリー内のオブジェクトのパス

インストールされている製品に応じて、クロスリファレンステーブルには追加列または別の列が表示されます。

6.5.3 呼び出し構造体を使用した呼び出し階層検査

呼び出し構造体は、ユーザープログラム内のブロックの呼び出し階層を記述します。使用されるブロック、他のブロックへの呼び出し、ブロック間の関係、それぞれのブロックのデータ要件、ブロックのステータスの概要を提供します。プログラミングエディタを開いて呼び出し構造体からブロックを編集できます。

呼び出し構造体を表示すると、ユーザープログラムで使用されるブロックのリストが表示されます。STEP 7は呼び出し構造体の最初のレベルを強調表示し、プログラム内の他のどのブロックからも呼び出されないすべてのブロックを表示します。呼び出し構造体の最初のレベルには、OB とすべての FC、FB、および OB に呼び出されない DB が表示されます。コードブロックが別のブロックを呼び出す場合、呼び出されるブロックは呼び出しブロックの下にインデントとして表示されます。呼び出し構造体はコードブロックに呼び出されるブロックのみを表示します。

呼び出し構造体内で衝突を引き起こすブロックのみを選択して表示できます。以下の条件により衝突が引き起こされます。

- 古いまたは新しいコードタイムスタンプのすべての呼び出しを実行するブロック
- インターフェースが変更されたブロックを呼び出すブロック
- アドレスおよび/またはデータタイプが変更されたタグを使用するブロック
- OB により直接的にも間接的にも呼び出されないブロック
- 存在しないブロックまたは不明のブロックを呼び出すブロック

複数のブロック呼び出しとデータブロックを 1 つのグループにグループ化することができます。ドロップダウンリストを使用して、各呼び出しロケーションへのリンクを表示します。

タイムスタンプの衝突を表示するために、整合性チェックを行うこともできます。プログラムの生成中または生成後にブロックのタイムスタンプを変更すると、タイムスタンプの衝突が発生し、それによって今度は、呼び出しブロックと呼び出されるブロックの間に不整合が引き起こされることがあります。

- タイムスタンプとインターフェースの衝突の大部分は、コードブロックの再コンパイルで修正できます。
- コンパイルで不整合を解決できない場合は、[詳細]列のリンクを使用して、プログラミングエディタで問題のソースにジャンプします。これで、すべての不整合を手動で解消できます。
- 赤でマークされたすべてのブロックを再コンパイルする必要があります。

6.5.4 診断命令でハードウェアモニタ

6.5.4.1 CPUのLED状態の読み取り

LED 命令により、ユーザープログラムが CPU の LED の状態を判定することができます。
 この情報を使用して、HMI デバイスのタグをプログラミングすることができます。


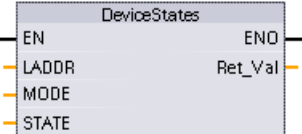
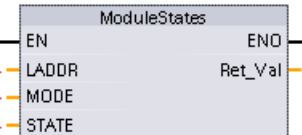
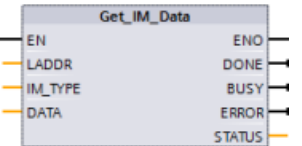
表 6- 25 LED 命令

LAD / FBD	SCL	説明
<div> <div>LED</div> <div> <div>EN</div> <div>END</div> </div> <div> <div>LADDR Ret_Val</div> <div>LED</div> </div> </div>	<pre> ret_val := LED(laddr:=_word_in_, LED:=_uint_in_); </pre>	RET_VAL は以下の CPU の LED の状態を返します <ul style="list-style-type: none"> • RUN/STOP: 緑または黄 • エラー: 赤 • MAINT (保守): 黄 • リンク: 緑 • Tx/Rx (送信/受信): 黄

6.5.4.2 デバイスの診断ステータスを読み取るための命令

STEP 7には、ネットワーク上のハードウェアデバイスから提供されるステータス情報を読み取るための命令も含まれています。

表 6- 26 診断命令

LAD / FBD	SCL	説明
	<pre>ret_val := GET_DIAG(mode:=_uint_in_, laddr:=_word_in_, cnt_diag=>_uint_out_, diag:=_variant_inout_, de- tail:=_variant_inout_);</pre>	GET_DIAG 命令は指定されたハードウェアデバイスから診断情報を読み取ります。
	<pre>ret_val := DeviceStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	DeviceStates 命令は PROFINET デバイスまたは PROFIBUS デバイスのステータスを読み取ります。
	<pre>ret_val := ModuleStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	ModuleStates 命令は PROFINET モジュールまたは PROFIBUS モジュールのステータスを読み取ります。
<p>*GET_IM_DATA_DB"</p> 	<pre>"GET_IM_DATA_DB"(LADDR:=16#0, IM_TYPE:=0, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA:=_variant_inout_);</pre>	指定されたモジュールまたはサブモジュールの識別およびメンテナンス(I&M)データをチェックするには、Get_IM_Data 命令を使用します。

6.6 高速カウンタ(HSC)

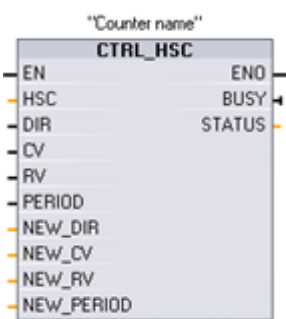
OB 実行速度よりも速い速度で発生するイベントをカウントするには、高速カウンタ(HSC)を使用します。Counting 命令は命令ツリーのテクノロジーフォルダにあります。
CTRL_HSC 命令は HSC の動作を制御します。

注記

カウント対象のイベントが OB の実行速度内で発生した場合は、CTU、CTD、または CTUD カウンタ命令を使用します。イベントが DB の実行速度より速い速度で発生した場合は、HSC を使用します。

CPU のデバイスコンフィグレーションで、それぞれの HSC のパラメータを設定します: カウンタモード、I/O 接続、割り込みの割り付け、高速カウンタとして使用するのかパルス周波数計測デバイスとして使用するのかを設定します。

表 6- 27 CTRL_HSC 命令

LAD / FBD	SCL	説明
	<pre>"counter_name" (HSC:=W#16#0, DIR:=FALSE, CV:=FALSE, RV:=FALSE, Period:=FALSE, New_DIR:=0, New_CV:=L#0, New_RV:=L#0, New_Period:=0, Busy=>_bool_out_, Status=>_word_out_);</pre>	<p>CTRL_HSC 命令のそれぞれが、データブロック内に格納されている構造体を使用してカウンタデータを保持します。</p> <p>SCL の場合、まず個々のカウンタ命令用の DB を作成しないと命令を参照できません。LAD および FBD の場合、命令を挿入すると STEP 7 が自動的に DB を作成します。</p>

一般に、CTRL_HSC 命令は、カウンタハードウェア割り込みイベントがトリガされた時に実行するハードウェア割り込み OB 内に配置します。たとえば、CV=RV イベントがカウンタ割り込みをトリガした場合、ハードウェア割り込み OB プログラムブロックが CTRL_HSC 命令を実行し、NEW_RV 値をロードすることによって基準値を変更することができます。

注記

CTRL_HSC のパラメータに現在のカウンタ値を使用することはできません。現在のカウンタ値が格納されているプロセスイメージアドレスは、高速カウンタのハードウェアコンフィグレーション中に割り付けられます。プログラムロジックを使用してカウンタ値を直接読み取ることができます。プログラムに返された値がカウンタを読み取った時点での正しいカウンタ値になります。高速カウンタのイベントは続行されます。したがって、プログラムで、このカウンタ値を使用して処理を完了する前に、実際のカウンタ値が変化している可能性があります。

HSCのパラメータのなかには、カウントプロセスのプログラム制御を行えるようユーザープログラムで変更できるものもあります。

- カウント方向を NEW_DIR 値にセットする
- 現在のカウント値を NEW_CV 値にセットする
- 基準値を NEW_RV にセットする
- 周波数計測時間を NEW_PERIOD にセットする(周波数計測モード)

CTRL_HSC の実行時に下記のフラグが 1 にセットされた場合は、対応する新しい値がカウンタにロードされます。複数の要求が(同時に複数のフラグがセットされたとき)、CTRL_HSC 命令の 1 回の実行で処理されます。以下の Boolean フラグ値を 0 に設定しても変化は起きません。

- DIR = 1 を設定すると NEW_DIR 値がロードされます。
- CV = 1 を設定すると NEW_CV 値がロードされます。
- RV = 1 を設定すると NEW_RV 値がロードされます。
- PERIOD = 1 を設定すると NEW_PERIOD 値がロードされます。

CTRL_HSC_EXT 命令(高速カウンタの制御(拡張))命令

STEP 7 および S7-1200 CPU は、拡張高速カウンタ命令 CTRL_HSC_EXT もサポートしています。この命令により、プログラムが指定された HSC の入力パルスの周期を精確に測定できます。詳細については、『S71200 プログラマブルコントローラシステムマニュアル』を参照してください。

6.6.1 高速カウンタの動作

高速カウンタ(HSC)は、サイクリック OB 実行速度よりも速い速度で発生するイベントをカウントできます。カウント対象のイベントが OB の実行速度内より遅い速度で発生した場合は、CTU、CTD、または CTUD 標準カウンタ命令を使用できます。イベントが DB の実行速度より速い速度で発生した場合は、より速い HSC デバイスを使用します。CTRL_HSC 命令により、HSC パラメータの一部をプログラムにより変更することができます。

たとえば、HSC をインクリメンタルシャフトエンコーダの入力として使用できます。このシャフトエンコーダは、1 回転あたりのカウント数を指定でき、1 回転ごとにリセットパルスが発生させることができます。シャフトエンコーダからのクロックおよびリセットパルスを HSC への入力として使用します。

HSC にはいくつかのプリセット値の初期値がロードされ、現在のカウンタ値が現在のプリセット値以下の時間はアクティブになります。HSC は、現在のカウンタ値がプリセット値と等しくなったとき、リセットが発生したとき、および方向が変化したときに、割り込みを発生させます。

現在のカウンタ値がプリセット値に等しくなったことによって発生する割り込みのたびに、新しいプリセット値がロードされ、次の出力状態がセットされます。リセットイベントが発生した場合は、最初のプリセット値と最初の出力の状態がセットされ、サイクルが反復されます。

割り込みの発生頻度は HSC のカウンタ速度よりもはるかに低いため、高速カウンタの正確な制御を実装しても、CPU のスキャンサイクルに与える影響を比較的小さく抑えることができます。割り込みを付加する方法によって、複数の新しいプリセット値のロードを別の割り込みルーチンで実行することができ、状態制御を簡単に行うことができます。また、すべての割り込みイベントを 1 つの割り込みルーチンで処理することもできます。

HSC 入力チャンネルの選択

以下のテーブルを使用して、接続する CPU および SB 入力チャンネルがプロセス信号の最大パルス速度を確実にサポートできるようにしてください。

注記

CPU および SB 入力チャンネル(V4 以降のファームウェア)の入力フィルタ時間は設定可能です

V4 以前のファームウェアバージョンでは、HSC 入力チャンネルとフィルタ時間は固定されていて変更できませんでした。

V4 以降のバージョンでは、入力チャンネルとフィルタ時間を割り当てることができます。デフォルトの入力フィルタ設定 6.4 ms では、プロセス信号に遅すぎる場合があります。デジタル入力フィルタ時間を、HSC アプリケーションの HSC 入力に合わせて最適化する必要があります。

表 6- 28 CPU 入力: 最大周波数

CPU	CPU 入力チャンネル	1 または 2 位相モード	A/B 直角位相モード
1211C	Ia.0～Ia.5	100 kHz	80 kHz
1212C	Ia.0～Ia.5	100 kHz	80 kHz
	Ia.6、Ia.7	30 kHz	20 kHz
1214C および 1215C	Ia.0～Ia.5	100kHz	80kHz
	Ia.6～Ia.5	30 kHz	20 kHz
1217C	Ia.0～Ia.5	100 kHz	80 kHz
	Ia.6～Ia.1	30 kHz	20 kHz
	Ib.2～Ib.5 (.2+、.2-～.5+、.5-)	1 MHz	1 MHz

表 6- 29 SB シグナルボード入力: 最大周波数(オプションボード)

SB シグナルボード	SB 入力チャンネル	1 または 2 位相モード	A/B 直角位相モード
SB 1221、200 kHz	Ie.0～Ie.3	200kHz	160 kHz
SB 1223、200 kHz	Ie.0、Ie.1	200kHz	160 kHz
SB 1223	Ie.0、Ie.1	30 kHz	20 kHz

HSC の機能を選択する

すべての HSC は、同じカウンタモードの動作では同じように機能します。カウンタモード、方向制御、初期方向が、CPU デバイスコンフィギュレーションで HSC ファンクションのプロパティに割り当てられています。

HSC には 4 つの基本的なタイプがあります。

- 内部方向制御式単相カウンタ
- 外部方向制御式単相カウンタ
- 2 クロック入力式 2 相カウンタ
- A/B 相直角位相カウンタ

それぞれの HSC をリセット入力付き、または無しで使用することができます。リセット入力をアクティブにすると(制限事項については下表参照)、現在値がクリアされ、リセット入力を解除するまでクリアした状態が保持されます。

- 周波数ファンクション: HSC のモードによっては、現在のパルス数ではなく周波数をカウントするように構成できます(カウンタ型)。周波数計測時間として、0.01、0.1、1.0 秒の 3 種類が用意されています。

周波数計測時間は、HSC での新しい周波数値の計算頻度を決定します。報告される周波数は、直前の計測時間のカウンタ値の合計によって決定される平均値です。周波数が高速に変化している場合は、その計測時間に発生した最高と最低の周波数の中間値になります。周波数計測時間の設定に関係なく、周波数は常にヘルツ単位(1 秒間あたりのパルスの数)で報告されます。

- カウンタのモードおよび入力 HSC に関連のあるクロック、方向制御、リセット機能用の入力を下の表に示します。
- 周期測定ファンクション: 周期測定は設定された測定間隔(10ms、100ms、または 1000ms)で行われます。HSC_Period SDT は周期測定を返し、周期測定を 2 つの値として提供します: ElapsedTime と EdgeCount です。HSC 入力 ID1000~ID1020 は周期測定に影響されません。
 - ElapsedTime は、測定間隔内の最初のカウントイベントから最後のカウントイベントまでの時間を表すナノ秒単位の符号なし倍精度整数値です。EdgeCount = 0 の場合、ElapsedTime は前の間隔での最後のカウントイベント以降の時間となります。ElapsedTime の範囲は 0~4,294,967,280 ns (0x0000 0000~0xFFFF FFF0)です。オーバーフローは値 4,294,967,295 (0xFFFF FFFF)で示されます。0xFFFF FFF1~0xFFFF FFFE の値は予約済みです。
 - EdgeCount は、測定間隔内のカウントイベントの数を表す符号なし倍精度整数値です。

2 つの異なる機能に同じ入力を使用することはできませんが、その HSC の現在のモードで使用していない入力を別の目的に使用することができます。たとえば、HSC1 が 2 つの内部入力を使用し、3 番目の外部リセット入力(I0.3 のデフォルトの割り当て)を使用しないモードのとき、I0.3 をエッジ割り込みまたは HSC 2 に使用することができます。

表 6- 30 HSC のカウンタモード

タイプ	入力 1	入力 2	入力 3	ファンクション
内部方向制御式単相カウンタ	クロック	-	-	カウンタ値または周波数
			リセット	カウント
外部方向制御式単相カウンタ	クロック	方向	-	カウンタ値または周波数
			リセット	カウント
2 クロック入力式 2 相カウンタ	クロック アップ	クロック ダウン	-	カウンタ値または周波数
			リセット	カウント
A/B 相直角位相カウンタ	A 相	B 相	-	カウンタ値または周波数
			リセット ¹	カウント

¹ エンコーダの場合: 位相 Z、原点復帰

HSCの入力アドレス

CPU を構成する場合、それぞれの HSC の「ハードウェア入力」を有効にして設定するオプションがあります。

すべての HSC 入力は、CPU モジュールまたは CPU モジュールの前面に装着されているオプションのシグナルボードの端子に接続する必要があります。

注記

以下の表に示すように、異なった HSC のオプションの信号に対するデフォルトの割り当ては重複しています。たとえば、HSC 1 のオプションの外部リセットは HSC 2 のいずれかの入力と同じ入力を使用します。

V4 CPU 以降の場合、CPU の構成時に HSC 入力を再割り当てすることができます。デフォルトの入力割り当てを使用する必要はありません。

常に、1 つの入力が 2 つの HSC で使用されないように HSC が設定されていることを確認してください。

以下の表に、CPU およびオプションの SB のオンボード I/O に対する HSC 入力のデフォルトの割り当てを示します。(選択された SB モデルに 2 つの入力しかない場合、4.0 および 4.1 入力のみが使用可能です。)

HSC 入力テーブルの定義

- **単相:** C はクロック入力、[d] は方向入力(オプション)、[R] は外部リセット入力(オプション)
(リセットは「カウント」モードでのみ使用可能です。)
- **2 相:** CU はクロックアップ入力、CD はクロックダウン入力、[R] は外部リセット入力(オプション)
(リセットは「カウント」モードでのみ使用可能です。)
- **AB 相カドラチャ:** A はクロック A 入力、B はクロック B 入力、[R] は外部リセット入力(オプション)。(リセットは「カウント」モードでのみ使用可能です。)

表 6- 31 CPU 1211C: HSC のデフォルトのアドレス割り当て

HSC カウンタモード		CPU オンボード入力 (デフォルト 0.x)						オプションの SB 入力 (デフォルト 4.x) ¹			
		0	1	2	3	4	5	0	1	2	3
HSC 1	1 相	C	[d]		[R]			C	[d]		[R]
	2 相	CU	CD		[R]			CU	CD		[R]
	AB 位相	A	B		[R]			A	B		[R]
HSC 2	1 相		[R]	C	[d]				[R]	C	[d]
	2 相		[R]	CU	CD				[R]	CU	CD
	AB 位相		[R]	A	B				[R]	A	B
HSC 3	1 相					C	[d]	C	[d]		R]
	2 相										
	AB 位相										
HSC4	1 相					C	[d]	C	[d]		R]

HSC カウンタモード		CPU オンボード入力 (デフォルト 0.x)						オプションの SB 入力 (デフォルト 4.x) ¹			
		0	1	2	3	4	5	0	1	2	3
	2 相					CU	CD				
	AB 位相					A	B				
HSC 5	1 相							C	[d]		[R]
	2 相							CU	CD		[R]
	AB 位相							A	B		[R]
HSC 6	1 相								[R]	C	[d]
	2 相								[R]	CU	CD
	AB 位相								[R]	A	B

¹ 2つのデジタル入力しかないSBは、4.0および4.1入力だけを提供します。

表 6- 32 CPU 1212C: HSC のデフォルトのアドレス割り当て

HSC カウンタモード		CPU オンボード入力 (デフォルト 0.x)								オプションの SB 入力 (デフォルト 4.x) ¹			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1	1 相	C	[d]		[R]					C	[d]		[R]
	2 相	CU	CD		[R]					CU	CD		[R]
	AB 位相	A	B		[R]					A	B		[R]
HSC 2	1 相		[R]	C	[d]						[R]	C	[d]
	2 相		[R]	CU	CD						[R]	CU	CD
	AB 位相		[R]	A	B						[R]	A	B
HSC 3	1 相					C	[d]		[R]	C	[d]		[R]
	2 相					CU	CD		[R]				
	AB 位相					A	B		[R]				
HSC 4	1 相						[R]	C	[d]	C	[d]		[R]
	2 相						[R]	CU	CD				
	AB 位相						[R]	A	B				
HSC 5	1 相									C	[d]		[R]
	2 相									CU	CD		[R]
	AB 位相									A	B		[R]
HSC 6	1 相										[R]	C	[d]
	2 相										[R]	CU	CD
	AB 位相										[R]	A	B

¹ 2つのデジタル入力しかないSBは、4.0および4.1入力だけを提供します。

表 6- 33 CPU 1214C、CPU 1215C、および CPU1217C:
HSC のデフォルトのアドレス割り当て
(オンボード入力のみ、オプションの SB アドレスは次の表を参照)

HSC カウンタモード		デジタル入力バイト 0 (デフォルト: 0.x)								デジタル入力バイト 1 (デフォルト: 1.x)					
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1	1 相	C	[d]		[R]										
	2 相	CU	CD		[R]										
	AB 位相	A	B		[R]										
HSC 2	1 相		[R]	C	[d]										
	2 相		[R]	CU	CD										
	AB 位相		[R]	A	B										
HSC 3	1 相					C	[d]		[R]						
	2 相					CU	CD		[R]						
	AB 位相					A	B		[R]						
HSC 4	1 相						[R]	C	[d]						
	2 相						[R]	CU	CD						
	AB 位相						[R]	A	B						
HSC 5	1 相									C	[d]	[R]			
	2 相									CU	CD	[R]			
	AB 位相									A	B	[R]			
HSC 6	1 相												C	[d]	[R]
	2 相												CU	CD	[R]
	AB 位相												A	B	[R]

表 6- 34 上の表の CPU のオプションの SB: HSC のデフォルトのアドレス割り当て

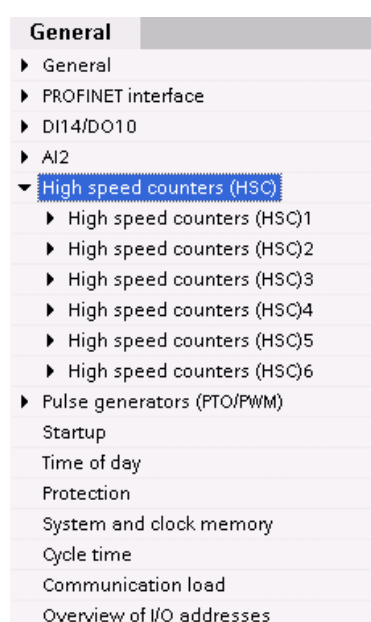
HSC		オプションの SB 入力(デフォルト: 4.x) ¹			
		0	1	2	3
HSC 1	1 相	C	[d]		[R]
	2 相	CU	CD		[R]
	AB 位相	A	B		[R]
HSC 2	1 相		[R]	C	[d]
	2 相		[R]	CU	CD
	AB 位相		[R]	A	B
HSC 5	1 相	C	[d]		[R]
	2 相	CU	CD		[R]
	AB 位相	A	B		[R]
HSC 6	1 相		[R]	C	[d]
	2 相		[R]	CU	CD
	AB 位相		[R]	A	B

¹ 2つのデジタル入力しかない SB は、4.0 および 4.1 入力だけを提供します。

注記

高速カウンタデバイスで使用するデジタル I/O 点は、CPU のデバイスコンフィグレーション時に割り付けられます。HSC デバイスにデジタル I/O 点のアドレスが割り当てられると、割り当てられた I/O 点のアドレスの値をウォッチテーブルの強制ファンクションで変更することはできません。

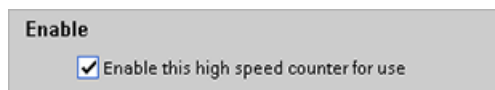
6.6.2 HSCのコンフィグレーション



最大で 6 つの高速カウンタを設定できます。CPU のデバイスコンフィグレーションを編集し、個々の HSC の HSC プロパティを割り当てます。

当該 HSC の[有効化]オプションを選択して HSC を有効にします。

ユーザープログラムで CTRL_HSC 命令および/または CTRL_HSC_EXT 命令を使用して、HSC の動作を制御します。



警告

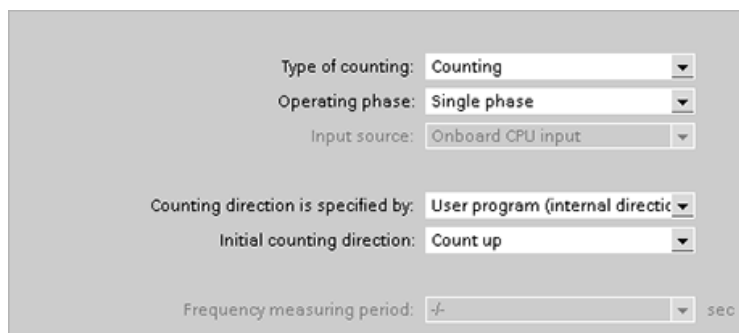
デジタル入力チャンネルのフィルタ時間設定の変更に伴うリスク

デジタル入力チャンネルのフィルタ時間を以前の設定から変更した場合、フィルタが新しい入力に完全に応答できるようになる前に、最高で 20.0 ms の累積時間の間、新しい「0」レベル入力値が存在する必要があります。この間、時間が 20 ms 未満の短い「0」パルスイベントは検出またはカウントできません。

このようなフィルタ時間の変更により、予期しない機械またはプロセスオペレーションが発生し、それが死傷事故などの重大な人的傷害や物的損害を引き起こすおそれがあります。

新しいフィルタ時間を確実に即時に有効にするには、CPU の電源を切った後に再投入します。

HSC を有効設定にした後で、カウンタ機能、初期値、リセットオプション、割り込みイベントなど、他のパラメータを設定します。

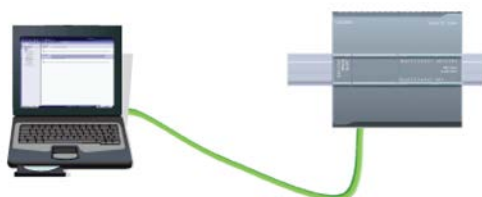


The screenshot shows a configuration window for the High-Speed Counter (HSC). It contains several dropdown menus and a text input field. The settings are as follows:

- Type of counting: Counting
- Operating phase: Single phase
- Input source: Onboard CPU input
- Counting direction is specified by: User program (internal directio
- Initial counting direction: Count up
- Frequency measuring period: -/- sec

HSC の設定についての追加情報は、CPU の構成(82 ページ)のセクションを参照してください。

カンタンデバイス間通信



プログラミングデバイスと CPU 間の直接接続の場合

- プロジェクトに CPU が含まれている必要があります。
- プログラミングデバイスはプロジェクトの一部ではありませんが、STEP 7 がインストールされている必要があります。



HMI パネルと CPU 間の直接接続の場合、プロジェクトに CPU と HMI の両方が含まれている必要があります。



2 つの CPU 間の直接接続の場合

- プロジェクトに両方の CPU が含まれている必要があります。
- 2 つの CPU 間にネットワーク接続を設定する必要があります。

S7-1200 CPU は PROFINET IO コントローラであり、プログラミングデバイス上の STEP 7、他の CPU、またはシーメンス製以外のデバイスとの通信を行います。イーサネットスイッチは、プログラミング装置または HMI と CPU の直接接続には不要です。イーサネットスイッチは、2 つ以上の CPU または HMI デバイスのあるネットワークに必要です。

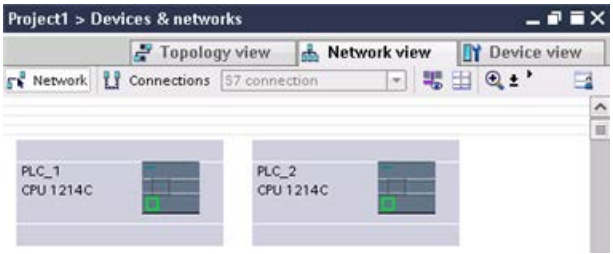
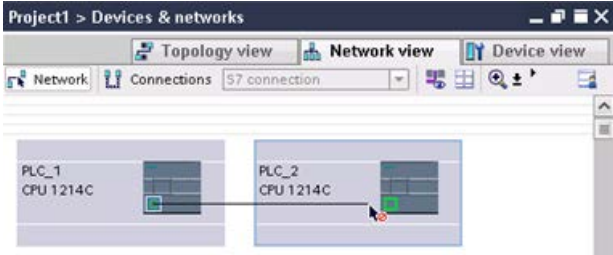
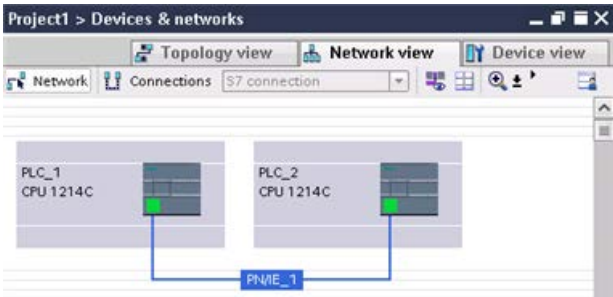
PROFIBUS CM を追加することで、CPU が PROFIBUS ネットワーク上でマスタまたはスレーブのどちらとしても機能することができます。

他の通信インターフェース(CM、CP、または CB)は、ポイントツーポイント(PTP)、Modbus、USS、GPRS (モデム)、セキュリティ CP、リモートコントロール CP などのさまざまなプロトコルをサポートします。

7.1
 ネットワーク接続の作成

プロジェクト内のデバイス間のネットワーク接続を作成するには、デバイスコンフィグレーションの[ネットワークビュー]を使用します。ネットワーク接続を作成し、インスペクタウィンドウの[プロパティ]タブを使用して、ネットワークのパラメータを構成します。

表 7- 1
 ネットワーク接続を作成する

操作	結果
[ネットワークビュー]を選択して、接続するデバイスを表示します。	
接続元のデバイスのポートを選択して、接続先のポートにドラッグします。	
マウスボタンを離すと、ネットワーク接続が作成されます。	

7.2 通信オプション

S7-1200 は、CPU とプログラミングデバイス、HMI、その他の CPU との間に複数のタイプの通信を提供します。



警告

攻撃側がネットワークに物理的にアクセスできる場合、データの読み取りまたは書き込みが可能であると思われます。

TIA ポータル、CPU、HMI (GET/PUT を使用する HMI 以外)は、反射攻撃および「中間者」攻撃を防ぐ安全な通信を使用しています。いったん通信が可能になると、署名付きのメッセージの交換が明確なテキストで行われ、攻撃側はデータを読むことはできますが、データの未許可の書き込みが防止されます。TIA ポータルは通信プロセスではなく、ノウハウプロテクトブロックのデータを暗号化します。

他のすべての通信形態(PROFIBUS、PROFINET、AS-i、またはその他の I/O バス、GET/PUT、T ブロック、通信モジュール(CM)を介した I/O 交換)にはセキュリティ機能はありません。このような通信形態は、物理的アクセスを制限することで保護する必要があります。攻撃側がこのような通信形態を使用してネットワークに物理的にアクセスできる場合、データの読み取りまたは書き込みが可能であると思われます。

セキュリティ情報と推奨事項については、シーメンスのサービス&サポートサイトにある弊社の『産業セキュリティの操作ガイドライン』

(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)を参照してください。

PROFINET

PROFINET は、ユーザープログラムを介してイーサネット経由の他の通信パートナーとデータを交換するのに使用します。

- S7-1200 の PROFINET は最大で 256 のサブモジュールを持つ 16 台の IO デバイスとの通信をサポートし、PROFIBUS は 3 つの独立した PROFIBUS DP マスタが使用でき、DP マスタ当たり 32 台のスレーブ、DP マスタ当たり最大で 512 のモジュールとの通信をサポートしています。
- S7 通信
- User Datagram Protocol (UDP) プロトコル
- ISO on TCP (RFC 1006)
- Transport Control Protocol (TCP)

PROFINET IO コントローラ

PROFINET IO を使用する IO コントローラとして、CPU はローカル PN ネットワークで、または PN/PN カプラー(リンク)経由で最大 16 台の PN デバイスと通信します。詳細は、PROFIBUS および PROFINET International、PI (www.us.profinet.com)を参照してください。

PROFIBUS

PROFIBUS は、ユーザープログラムを介して PROFIBUS ネットワーク経由の他の通信パートナーとデータを交換するのに使用します。

- CM 1242-5 では、CPU は PROFIBUS DP スレーブとして動作します。
- CM 1243-5 では、CPU は PROFIBUS DP マスタ class 1 として動作します。
- PROFIBUS DP スレーブ、PROFIBUS DP マスタ、AS-i (3つの左側の通信モジュール)および PROFINET は、互いに制約しない個別の通信ネットワークです。

AS-i

S7-1200 CM 1243-2 AS-i マスタでは、AS-i ネットワークを S7-1200 CPU に接続できます。

CPU 間 S7 通信

パートナーステーションへの通信接続を作成し、GET 命令と PUT 命令を使用して S7 CPU と通信できます。

TeleService 通信

GPRS 経由の TeleService では、STEP 7 がインストールされているエンジニアリングステーションが GSM ネットワークとインターネットを経由して、CP 1242-7 を搭載した SIMATIC S7-1200 ステーションと通信します。接続は、仲介物として機能しインターネットに接続される遠隔制御サーバーを介して行われます。

IO リンク

S7-1200 SM 1278 4xIO リンクマスタにより、IO リンクデバイスを S7-1200 CPU に接続することができます。

7.3 V4.1 非同期通信接続

通信サービスの概要

CPU は以下の通信サービスをサポートします。

通信サービス	機能	PROFIBUS DP の使用		イーサネットの使用
		CM 1243-5 DP マスタモジュール	CM 1242-5 DP スレーブモジュール	
PG 通信	コミッショニング、テスト、診断	あり	なし	あり
HMI 通信	オペレータコントロールおよびモニタリング	あり	なし	あり
S7 通信	設定された通信を使用したデータ交換	あり	なし	あり
PG ファンクションのルーティング	たとえば、ネットワーク境界外でのテストと診断	なし	なし	なし
PROFIBUS DP	マスタとスレーブ間のデータ交換	あり	あり	なし
PROFINET IO	I/O コントローラと I/O デバイス間のデータ交換	なし	なし	あり
Web サーバー	診断	なし	なし	あり
SNMP (Simple Network Management Protocol)	ネットワーク診断とパラメータ割り当て用の標準プロトコル	なし	なし	あり
TCP/IP を介した開放型通信	産業用イーサネットを介した TCP/IP プロトコルとの(ロード可能な FB との)データ交換	なし	なし	あり
ISO on TCP を介した開放型通信	産業用イーサネットを介した ISO on TCP プロトコルとの(ロード可能な FB との)データ交換	なし	なし	あり
UDP を介した開放型通信	産業用イーサネットを介した UDP プロトコルとの(ロード可能な FB との)データ交換	なし	なし	あり

使用可能な接続

CPU は PROFINET および PROFIBUS に対して、以下の同時非同期通信接続の最大数をサポートします。それぞれのカテゴリに割り当てられている接続リソースの最大数は固定です。この値を変更することはできません。ただし、6つの「自由な使用可能接続」を設定し、アプリケーションの必要に応じて任意のカテゴリの数を増やすことができます。



割り当てられたリソースに基づいて、1つのデバイスにつき以下の数の通信が可能です。

	プログラミング ターミナル(PG)	ヒューマンマシン インターフェース (HMI)	GET/PUT クライアント/ サーバー	開放型ユーザー間 通信	Web ブラウザ
接続リソース の最大数	3 (1つのPGデバイスの サポートを保証)	12 (4つのHMIデバイスの サポートを保証)	8	8	30 (3つのウェブブラウ ザのサポートを保証)

たとえば、1つのPGには3つの使用可能な接続リソースがあります。使用している現在のPGファンクションに応じて、PGは実際は、使用可能な接続リソースのうちの1つ、2つ、または3つを使用します。S7-1200では、常に最低1つのPGが保証されています。ただし、それ以上のPGは許可されません。

もう1つの例は、下の図に示すようにHMIの数です。HMIには12の使用可能な接続リソースがあります。ユーザーの所有するHMIタイプまたはモデル、およびユーザーの使用するHMIファンクションに応じて、それぞれのHMIは実際は、使用可能な接続リソースのうちの1つ、2つ、または3つを使用します。使用されている使用可能な接続リソースの数が明らかの場合、一度に4つ以上のHMIを使用することができます。ただし、保証されているのは常に最低4つのHMIです。HMIは以下のファンクションに対して使用可能な接続リソースを使用できます(それぞれが1つずつで、合計3つ)。

- 読み取り
- 書き込み
- アラームと診断

例	HMI 1	HMI 2	HMI 3	HMI 4	HMI 5	使用可能な 接続リソース の合計
使用される 接続リソース	2	2	2	3	3	12

注記

Web サーバー(HTTP)接続: CPU は複数のウェブブラウザに接続を提供します。CPU が同時にサポートできるブラウザの数は、指定されたウェブブラウザが要求/使用する接続数によって決まります。

注記

オープンユーザーコミュニケーション、S7 通信、HMI、プログラミングデバイス、Web サーバー(HTTP)通信接続は、現在使用されている機能に基づいて複数の接続リソースを使用することができます。

7.4 PROFINET 命令と PROFIBUS 命令

PROFINET 命令

TSEND_C 命令と TRCV_C 命令は、TCON 命令と TDISCON 命令の機能を TSEND 命令または TRCV 命令とを組み合わせた命令で、PROFINET 通信をより簡単にします。

- TSEND_C は、相手ステーションとの TCP または ISO on TCP の通信接続を確立し、データを送信して、接続を終了することができます。セットアップおよび確立された接続は、CPU によって自動的に維持され、モニタされます。TSEND_C は TCON、TDISCON、および TSEND 命令のファンクションを組み合わせたものです。
- TRCV_C は、パートナー CPU との TCP または ISO-on-TCP の通信接続を確立し、データを受信して接続を終了することができます。セットアップおよび確立された接続は、CPU によって自動的に維持され、モニタされます。TRCV_C 命令は、TCON、TDISCON、TRCV 命令のファンクションを組み合わせたものです。

TCON、TDISCON、TSEND、および TRCV 命令もサポートされます。

UDP 経由でデータの送受信を行うには、TUSEND 命令と TURCV 命令を使用します。TUSEND および TURCV(ならびに、TSEND、TRCV、TCON、TDISCON)は非同期に機能します。つまり、ジョブの処理は複数の命令呼び出しにわたって実行されます。

ユーザープログラムから IP 設定パラメータを変更するには、IP_CONF 命令を使用します。IP_CONF は非同期に機能します。実行は複数の呼び出しにわたって行われます。

PROFIBUS 命令

DPNRM_DG (診断読み取り)命令は、DP スレーブの現在の診断データを EN 50 170 Volume 2、PROFIBUS で指定されたフォーマットで読み取ります。

PROFINET、PROFIBUS、および AS-i 用の Distributed I/O 命令

PROFINET、PROFIBUS、GPRS で以下の命令を使用することができます。

- 基本ラック内のモジュールまたはリモートコンポーネント(PROFIBUS DP または PROFINET IO)などのコンポーネント間で指定されたデータレコードを転送するには、RDREC (レコードの読み取り)命令と WRREC (レコードの書き込み)命令を使用します。
- DP スレーブまたは PROFINET IO デバイスコンポーネントから割り込みと割り込み情報を読み取るには、RALRM (アラームの読み取り)命令を使用します。出力パラメータ内の情報には、呼び出される OB の開始情報ならびに割り込みソースの情報が含まれています。
- DP 標準スレーブとの間で 64 バイトを超える整合性のあるデータ領域をやり取りするには、DPRD_DAT (整合性のあるデータの読み取り)命令と DPWR_DAT (整合性のあるデータの書き込み)命令を使用します。
- PROFIBUS の場合に限り、DPNRM_DG 命令を使用して、DP スレーブの現在の診断データを EN 50 170 Volume 2、PROFIBUS で指定されたフォーマットで読み取ります。

7.5 PROFINET

7.5.1 オープンユーザーコミュニケーション

CPU の内蔵 PROFINET ポートは、イーサネットネットワーク経由の複数の通信規格をサポートします。

- Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)
- User Datagram Protocol (UDP)

表 7-2 それぞれに対するプロトコルと通信命令

プロトコル	使用例	受信領域へのデータ入力	通信命令	アドレスタイプ
TCP	CPU 間通信 フレームの転送	アドホックモード	TRCV_C and TRCV のみ(V4.1 およびレガシー命令)	ローカル(アクティブ)およびパートナー(パッシブ)デバイスにポート番号を割り当てます
		指定された長さのデータ受信	TSEND_C、TRCV_C、TCN、TDISCON、TSEND、および TRCV(V4.1 およびレガシー命令)	
ISO on TCP	CPU 間通信 メッセージのフラグメント化と再構築	アドホックモード	TRCV_C and TRCV のみ(V4.1 およびレガシー命令)	ローカル(アクティブ)およびパートナー(パッシブ)デバイスに TSAP を割り当てます
		プロトコル制御	TSEND_C、TRCV_C、TCN、TDISCON、TSEND、および TRCV(V4.1 およびレガシー命令)	
UDP	CPU 間通信 ユーザープログラム通信	User Datagram Protocol	TUSEND と TURCV	ローカル(アクティブ)およびパートナー(パッシブ)デバイスにポート番号を割り当てますが、専用接続ではありません
S7 通信	CPU 間通信 CPU からのデータの読み取りと CPU へのデータの書き込み	指定された長さのデータの送信と受信	GET と PUT	ローカル(アクティブ)およびパートナー(パッシブ)デバイスに TSAP を割り当てます
PROFINET IO	CPU と PROFINET IO デバイス間の通信	指定された長さのデータの送信と受信	内蔵	内蔵

7.5.1.1 アドホックモード

一般的に、TCP と ISO-on-TCP は指定された長さ(1~8192 バイトの範囲)のデータパケットを受信します。ただし、TRCV_C および TRCV 通信命令も、「アドホック」通信モードで 1~1472 バイトの可変長のデータパケットを受信できます。

注記

「最適化された」DB (シンボリックのみ)にデータを保存する場合、Arrays of Byte、Char、USInt、SInt データタイプのデータのみを受信できます。

TRCV_C 命令または TRCV 命令をアドホックモード用に設定するには、ADHOC 命令入力パラメータを設定します。

TRCV_C 命令または TRCV 命令をアドホックモードで頻繁に呼び出さない場合は、1 回の呼び出しで複数のパケットを受信できます。たとえば、1 回の呼び出しで 5 つの 100 バイトパケットを受信する場合、TCP はこれらのパケットを 1 つの 500 バイトパケットとして配信しますが、ISO-on-TCP はパケットを 5 つの 100 バイトパケットに再構築します。

7.5.1.2 オープンユーザーコミュニケーション命令用の接続ID

TSEND_C、TRCV_C、または TCON PROFINET 命令をユーザープログラムに挿入すると、STEP 7 がインスタンス DB を作成してデバイス間に通信チャンネル(すなわち接続)を構成します。接続のパラメータを設定するには、命令の「プロパティ」(154 ページ)を使用します。パラメータの中には、その接続の接続 ID が含まれています。

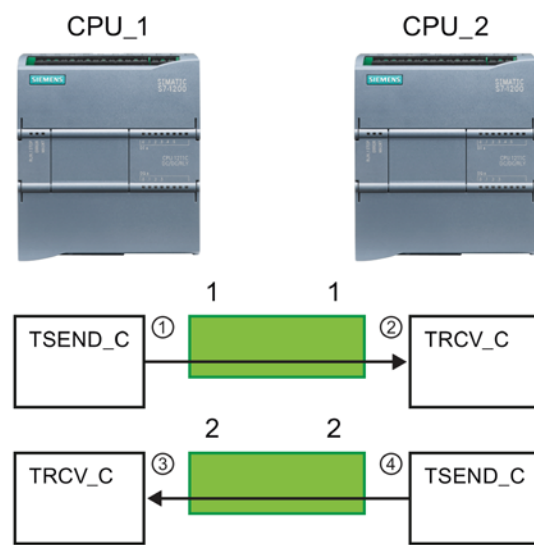
- 接続 ID は CPU に対して一意でなければなりません。作成する接続ごとに、DB および接続 ID が異なっている必要があります。
- ローカル CPU とパートナーCPU はどちらも同じ接続用の同じ接続 ID を使用できますが、接続 ID 番号が一致する必要はありません。接続 ID 番号は、個々の CPU のユーザープログラム内の PROFINET 命令に対してのみ関連があります。
- CPU の接続 ID に任意の番号を使用できます。ただし、接続 ID を「1」から順番に設定すると、特定の CPU に対して使用する接続の数を簡単にトラッキングすることができます。

注記

ユーザープログラム内の TSEND_C、TRCV_C、または TCON 命令のそれぞれが新しい接続を作成します。各接続に対して正しい接続 ID を使用することが重要です。

以下の例に、データの送受信に2つの別々の接続を使用する2つのCPU間の通信を示します。

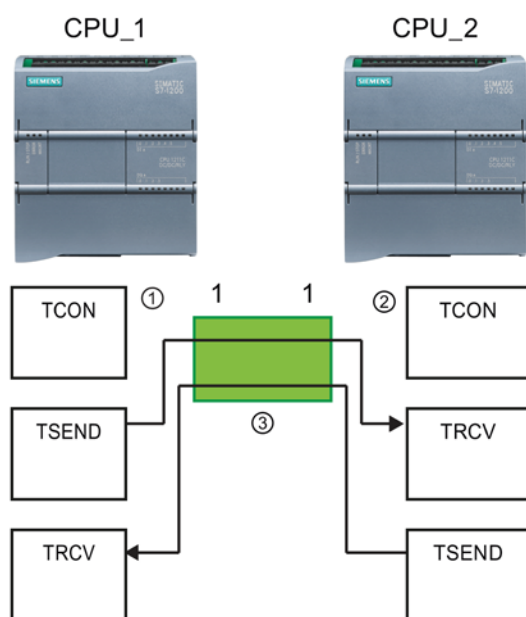
- CPU_1のTSEND_C命令は、1番目の接続によってCPU_2のTRCV_Cにリンクされています(CPU_1とCPU_2はどちらも「接続ID 1」)。
- CPU_1のTRCV_C命令は、2番目の接続によってCPU_2のTSEND_Cにリンクされています(CPU_1とCPU_2はどちらも「接続ID 2」)。



- ① CPU_1のTSEND_Cは接続を作成し、その接続に接続ID(CPU_1の接続ID 1)を割り当てます。
- ② CPU_2のTRCV_CはCPU_2に対する接続を作成し、接続ID(CPU_2の接続ID 1)を割り当てます。
- ③ CPU_1のTRCV_CはCPU_1に対する2番目の接続を作成し、その接続の別の接続ID(CPU_1の接続ID 2)を割り当てます。
- ④ CPU_2のTSEND_Cは2番目の接続を作成し、その接続の別の接続ID(CPU_2の接続ID 2)を割り当てます。

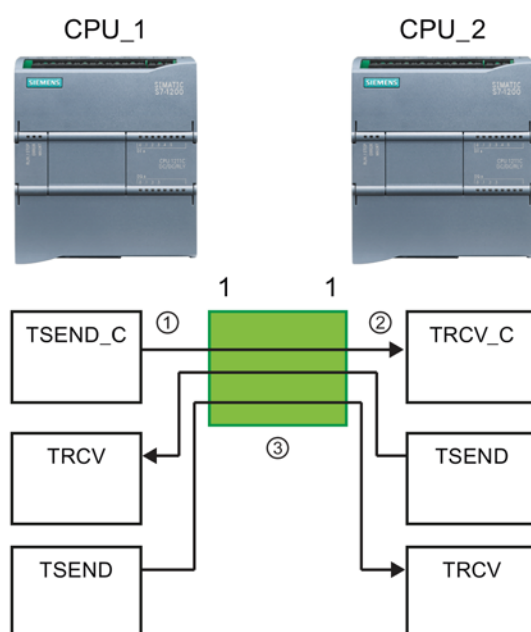
以下の例に、データの送信と受信の両方に 1 つの接続を使用する 2 つの CPU 間の通信を示します。

- それぞれの CPU は、TCON 命令を使用して 2 つの CPU 間の接続を設定します。
- CPU_1 の TSEND 命令は、CPU_1 の TCON 命令によって設定された接続 ID (「接続 ID 1」) を使用して、CPU_2 の TRCV 命令にリンクされます。CPU_2 の TRCV 命令は、CPU_2 の TCON 命令によって設定された接続 ID (「接続 ID 1」) を使用して、CPU_1 の TSEND 命令にリンクされます。
- CPU_2 の TSEND 命令は、CPU_2 の TCON 命令によって設定された接続 ID (「接続 ID 1」) を使用して、CPU_1 の TRCV 命令にリンクされます。CPU_1 の TRCV 命令は、CPU_1 の TCON 命令によって設定された接続 ID (「接続 ID 1」) を使用して、CPU_2 の TSEND 命令にリンクされます。



- ① CPU_1 の TCON は接続を作成し、CPU_1 のその接続に接続 ID(ID=1)を割り当てます。
- ② CPU_2 の TCON は接続を作成し、CPU_2 のその接続に接続 ID(ID=1)を割り当てます。
- ③ CPU_1 の TSEND および TRCV は、CPU_1 の TCON が作成した接続 ID (ID=1)を使用します。
CPU_2 の TSEND および TRCV は、CPU_2 の TCON が作成した接続 ID (ID=1)を使用します。

下の例に示すように、個々の TSEND および TRCV 命令を使用して、TSEND_C または TRCV_C 命令が作成した接続を介して通信を行うこともできます。TSEND 命令と TRCV 命令は、それ自体では新しい接続を作成しません。したがって、TSEND_C、TRCV_C、または TCON 命令で作成された DB または接続 ID を使用する必要があります。



- ① CPU_1 の TSEND_C は接続を作成し、その接続に接続 ID (ID=1) を割り当てます。
- ② CPU_2 の TRCV_C は接続を作成し、CPU_2 のその接続に接続 ID (ID=1) を割り当てます。
- ③ CPU_1 の TSEND および TRCV は、CPU_1 の TSEND_C が作成した接続 ID (ID=1) を使用します。
CPU_2 の TSEND および TRCV は、CPU_2 の TRCV_C が作成した接続 ID (ID=1) を使用します。

7.5.1.3 PROFINET接続のパラメータ

TSEND_C、TRCV_C、および TCON 命令は、パートナーデバイスへの接続のために接続関連パラメータを指定する必要があります。これらのパラメータは、TCP、ISO-on-TCP、および UDP プロトコルの場合、TCON_Param 構造によって割り当てられます。通常は、命令の[プロパティ]の[設定]タブを使用して、これらのパラメータを指定します。[設定]タブにアクセスできない場合は、プログラムによって TCON_Param 構造を指定する必要があります。

TCON_Param

表 7-3 接続記述子(TCON_Param)の構造

バイト	パラメータとデータタイプ		説明
0 ... 1	block_length	UInt	長さ: 64 バイト(固定)
2 ... 3	ID	CONN_OUC (Word)	この接続への参照: 値の範囲: 1 (デフォルト)~4095。ID の下に、TSEND_C、TRCV_C、または TCON 命令に対するこのパラメータの値を指定します。
4	connection_type	USInt	接続タイプ: <ul style="list-style-type: none"> 17: TCP (デフォルト) 18: ISO-on-TCP 19: UDP
5	active_est	Bool	接続タイプの ID: <ul style="list-style-type: none"> TCP および ISO-on-TCP: <ul style="list-style-type: none"> FALSE: パッシブ接続 TRUE: アクティブ接続 (デフォルト) UDP: FALSE
6	local_device_id	USInt	ローカル PROFINET または産業用イーサネットインターフェースの ID: 1 (デフォルト)
7	local_tsap_id_len	USInt	使用されているパラメータ local_tsap_id の長さ、バイト単位; 考えられる値: <ul style="list-style-type: none"> TCP: 0 (アクティブ、デフォルト)または 2 (パッシブ) ISO-on-TCP: 2~16 UDP: 2
8	rem_subnet_id_len	USInt	このパラメータは使用されません。
9	rem_staddr_len	USInt	パートナーエンドポイントのアドレスの長さ、バイト単位: <ul style="list-style-type: none"> 0: 未指定(パラメータ rem_staddr は対象外です) 4 (デフォルト): パラメータ rem_staddr の有効な IP アドレス (TCP および ISO-on-TCP のみ)
10	rem_tsap_id_len	USInt	使用されているパラメータ rem_tsap_id の長さ、バイト単位; 考えられる値: <ul style="list-style-type: none"> TCP: 0 (パッシブ)または 2 (アクティブ、デフォルト) ISO-on-TCP: 2~16 UDP: 0

バイト	パラメータとデータタイプ	説明
11	next_staddr_len	USInt このパラメータは使用されません。
12 ... 27	local_tsap_id	BYTE の ARRAY [1..16] 接続のローカルアドレスコンポーネント: <ul style="list-style-type: none"> • TCP および ISO-on-TCP: ローカルポート番号(使用可能な値: 1~49151; 推奨値: 2000...5000): <ul style="list-style-type: none"> – local_tsap_id[1] = 10 進数表記のポート番号の上位バイト; – local_tsap_id[2] = 10 進数表記のポート番号の下位バイト; – local_tsap_id[3-16] = 対象外 • ISO-on-TCP: ローカル TSAP-ID: <ul style="list-style-type: none"> – local_tsap_id[1] = B#16#E0; – local_tsap_id[2] = ローカルエンドポイントのラックとスロット(ビット 0~4: スロット番号、ビット 5~7: ラック番号); – local_tsap_id[3-16] = TSAP 拡張子、オプション • UDP: このパラメータは使用されません。 注記: local_tsap_id のすべての値は CPU 内で必ず一意にしてください。
28 ... 33	rem_subnet_id	USInt の配列 [1..6] このパラメータは使用されません。
34 ... 39	rem_staddr	USInt の配列 [1..6] TCP および ISO-on-TCP のみ: パートナーエンドポイントの IP アドレス。(パッシブ接続の場合は対象外) たとえば、IP アドレス 192.168.002.003 は配列の以下のエレメントに保存されます。 rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = 対象外
40 ... 55	rem_tsap_id	BYTE の ARRAY [1..16] 接続のパートナーアドレスコンポーネント: <ul style="list-style-type: none"> • TCP: パートナーポート番号。範囲: 1~49151; 推奨値: 2000~5000): <ul style="list-style-type: none"> – rem_tsap_id[1] = 10 進数表記のポート番号の上位バイト – rem_tsap_id[2] = 10 進数表記のポート番号の下位バイト; – rem_tsap_id[3-16] = 対象外 • ISO-on-TCP: パートナー TSAP-ID: <ul style="list-style-type: none"> – rem_tsap_id[1] = B#16#E0 – rem_tsap_id[2] = パートナーエンドポイントのラックとスロット(ビット 0~4: スロット番号、ビット 5~7: ラック番号) – rem_tsap_id[3-16] = TSAP 拡張子、オプション • UDP: このパラメータは使用されません。
56 ... 61	next_staddr	BYTE の ARRAY [1..6] このパラメータは使用されません。
62 ... 63	スペア	WORD 予約済み: W#16#0000

7.5.2 ローカル/パートナー接続パスの構成

ローカル/パートナー(リモート)接続は、通信サービスを確立するために、2つの通信パートナーの論理割り当てを定義します。接続は以下を定義します。

- 関係する通信パートナー(一方はアクティブ、もう一方はパッシブ)
- 接続のタイプ(たとえば、PLC、HMI、デバイス接続など)
- 接続パス

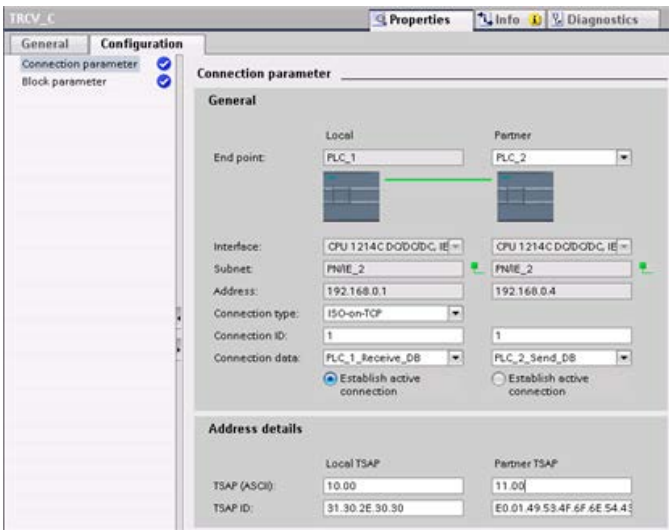
通信パートナーは通信接続をセットアップおよび確立するための命令を実行します。パラメータを使用して、アクティブ通信とパッシブ通信のエンドポイントパートナーを指定します。セットアップおよび確立された接続は、CPUによって自動的に維持され、モニタされます。

接続が終了した場合(たとえば、ライン切断などによって)、送信側のパートナーは、構成された接続の再確立を試みます。通信命令を再実行する必要はありません。

接続パス

TSEND_C、TRCV_C、または TCON 命令をユーザープログラムに挿入すると、インスペクタウィンドウには、命令のどの部分を選択しても必ず接続のプロパティが表示されます。通信命令の[プロパティ]の[設定]タブで、通信パラメータを指定します。

表 7-4 接続パスの設定(命令のプロパティを使用する)

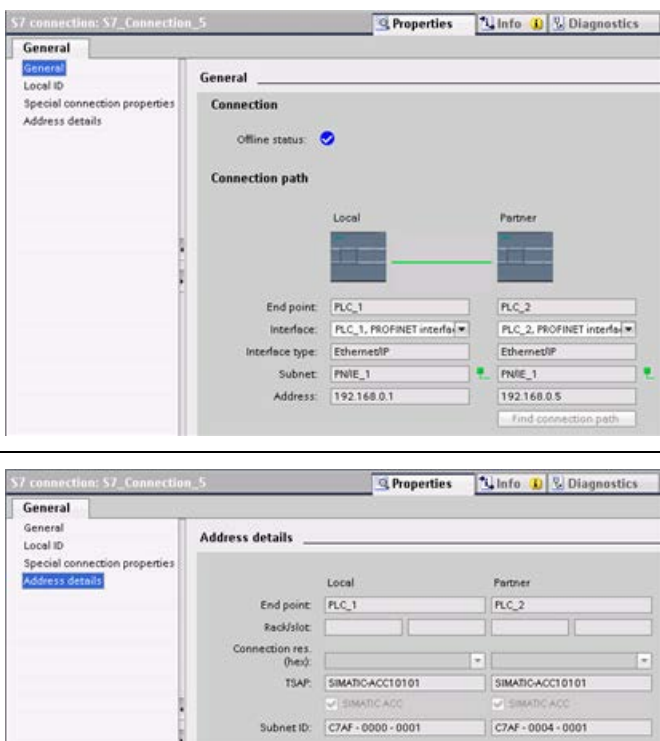
TCP、ISO-on-TCP、および UDP	接続プロパティ
<p>TCP、ISO-on-TCP、および UDP イーサネットプロトコルの場合、命令(TSEND_C、TRCV_C、または TCON)の[プロパティ]を使用して[ローカル/パートナー]接続を構成します。</p> <p>図は、ISO-on-TCP 接続の場合の[構成]タブの[接続プロパティ]を示したものです。</p>	

注記

1 つの CPU に対して接続プロパティを構成する場合、STEP 7 ではパートナー CPU で特定の接続 DB (存在する場合) を選択するか、パートナー CPU に接続 DB を作成するかのどちらかが可能です。パートナー CPU はプロジェクトに対して作成済みでなければならず、「未指定」CPU とすることはできません。

さらに、TSEND_C、TRCV_C、または TCON 命令をパートナー CPU のユーザープログラムに挿入する必要があります。命令を挿入するときに、構成で作成された接続 DB を選択します。

表 7-5 S7 通信の接続パスを構成する(デバイスコンフィグレーション)

S7 通信(GET および PUT)	接続プロパティ
<p>S7 通信の場合、ネットワークの[デバイスとネットワーク]エディタを使用して、ローカル/パートナー接続を構成します。[強調表示: 接続]ボタンをクリックして[プロパティ]にアクセスできます。</p> <p>[全般]タブに複数のプロパティが表示されます。</p> <ul style="list-style-type: none"> • [全般] (図に表示) • [ローカル ID] • [特殊な接続プロパティ] • [アドレス詳細] (図に表示) 	

詳細と使用可能な通信命令の一覧については、「PROFINET」セクションの「プロトコル」(149 ページ)または「S7 通信」セクションの「S7 接続の作成」(171 ページ)を参照してください。

表 7-6 複数の CPU 接続用のパラメータ

パラメータ		定義
アドレス		割り当てられた IP アドレス
全般	エンドポイント	パートナー(受信側) CPU に割り付ける名前
	インターフェース	インターフェースに割り付ける名前
	Subnet	サブネットに割り付ける名前
	インターフェースタイプ	S7 通信のみ: インターフェースのタイプ
	接続タイプ	イーサネットプロトコルのタイプ
	接続 ID	ID 番号
	接続データ	ローカルおよびパートナーの CPU データ保管場所
	アクティブ接続の確立	ローカルまたはパートナーCPU をアクティブな接続として選択するラジオボタン
アドレスの詳細	エンドポイント	S7 通信のみ: パートナー(受信側) CPU に割り付ける名前
	ラック/スロット	S7 通信のみ: ラックおよびスロットの場所
	接続リソース	S7 通信のみ: S7-300 または S7-400 CPU で S7 コネクションを構成するとき に使用する TSAP のコンポーネント
	ポート(10 進):	TCP と UDP: 10 進フォーマットのパートナーCPU ポート
	TSAP ¹ とサブネット ID:	ISO on TCP (RFC 1006)と S7 通信: ローカルおよびパートナーの CPU TSAP (ASCII および 16 進フォーマット)

¹ S7-1200 CPU に ISO-on-TCP の接続を設定する場合、パッシブ通信パートナーの TSAP 拡張子は ASCII 文字だけを使用してください。

トランスポートサービスアクセスポイント(TSAP)

TSAP を使用して、ISO on TCP プロトコルおよび S7 通信で複数の接続を 1 つの IP アドレスに関連付けることができます(最大 64K 接続)。TSAP は 1 つの IP アドレスへの複数のエンドポイント接続を一意に識別します。

[接続パラメータ]ダイアログの[アドレスの詳細]セクションで、使用する TSAP を定義することができます。CPU での接続 TSAP は、[ローカル TSAP]フィールドに入力します。パートナーCPU での接続に割り付けられる TSAP は、[パートナー TSAP]フィールドに入力します。

ポート番号

TCP および UDP プロトコルの場合、ローカル(アクティブ)接続 CPU の接続パラメータ設定でパートナー(パッシブ)接続 CPU のリモート IP アドレスとポート番号を指定する必要があります。

[接続パラメータ]ダイアログの[アドレスの詳細]セクションで、使用するポートを定義することができます。CPU での接続ポートは、[ローカルポート]フィールドに入力します。パートナーCPU での接続に割り当てられるポートは、[パートナーポート]フィールドに入力します。

7.6 PROFIBUS

PROFIBUS システムはバスマスタを使用して、マルチドロップ式で RS485 シリアルバス上に分散されているスレーブデバイスをポーリングします。PROFIBUS スレーブは、情報を処理して出力をマスタに送信する任意の周辺デバイスです(I/O トランスデューサ、バルブ、モータドライブ、その他の測定デバイス)。スレーブはネットワーク上にパッシブステーションを形成します。その理由は、バスアクセス権を持たず、受信したメッセージを確認するか、または要求されたときにマスタに応答メッセージを送信することしかできないからです。すべての PROFIBUS スレーブの優先度は同じで、すべてのネットワーク通信はマスタから行われます。

PROFIBUS マスタはネットワーク上に「アクティブステーション」を形成します。PROFIBUS DP は 2 つのクラスのマスタを定義します。クラス 1 のマスタ(通常は中央プログラマブルコントローラ(PLC)または特殊なソフトウェアを実行する PC)は、通常の通信の処理または割り当てられたスレーブとのデータ交換を行います。クラス 2 のマスタ(通常は、コミッシュニング、メンテナンス、診断のために使用されるラップトップまたはプログラミングコンソールなどのコンフィグレーションデバイス)は、主としてスレーブのコミッシュニングおよび診断のために使用される特別なデバイスです。

S7-1200 は CM 1242-5 通信モジュールを搭載した DP スレーブとして PROFIBUS ネットワークに接続されます。CM 1242-5 (DP スレーブ)モジュールは、DP V0/V1 マスタの通信パートナーとすることができます。モジュールをサードパーティー製のシステムで構成したい場合は、モジュールに同梱されている CD およびインターネットの Siemens Automation Customer Support

(<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=6GK72425DX300XE0&caller=view>) ページに、CM 1242 5 (DP スレーブ)に使用可能な GSD ファイルがあります。

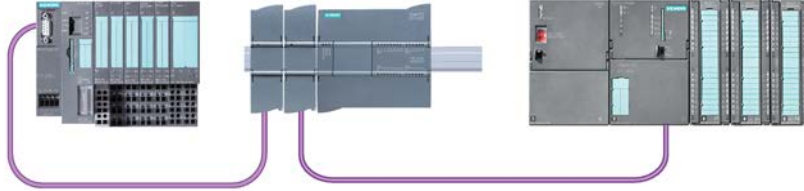
下の図では、S7-1200 は S7-300 コントローラの DP スレーブです。



S7-1200 は CM 1243-5 通信モジュールを搭載した DP マスタとして PROFIBUS ネットワークに接続されます。CM 1243-5 (DP マスタ)モジュールは、DP V0/V1 スレーブの通信パートナーとすることができます。下の図では、S7-1200 は ET200S DP スレーブを制御するマスタです。



CM 1242-5 と CM 1243-5 が一緒に設置されている場合、S7-1200 は上位レベル DP マスタシステムのスレーブおよび下位レベル DP スレーブシステムのマスタの両方として同時に機能できます。



V4.0 の場合、ステーションあたり最大 3 つの PROFIBUS CM を構成し、DP マスタ CM または DB スレーブ CM を任意に組み合わせられます。V3.0 以降の CPU ファームウェアが実装された DP マスタは、それぞれ最大 32 のスレーブを制御できます。

PROFIBUS CM の構成データはローカル CPU に保存されます。これにより、必要に応じて通信モジュールを簡単に交換できます。

7.6.1 PROFIBUS CMの通信サービス

PROFIBUS CM は PROFIBUS DP-V1 プロトコルを使用します。

DP V1 を使用した通信のタイプ

以下のタイプの通信が DP V1 を使用して行えます。

- サイクリック通信(CM 1242 5 および CM 1243 5)

両方の PROFIBUS モジュールが、DP スレーブと DP マスタ間でプロセスデータを転送するためのサイクリック通信をサポートします。

サイクリック通信は CPU のオペレーティングシステムによって処理されます。このためのソフトウェアブロックは必要ありません。I/O データは CPU のプロセスイメージから直接、読み取りまたは書き込みされます。

- 非サイクリック通信(CM 1243 5 のみ)

DP マスタモジュールは、ソフトウェアブロックを使用して非サイクリック通信もサポートします。

- [RALRM]命令を使用して割り込み処理が行えます。
- [RDREC]命令と[WRREC]命令を使用して、構成および診断データを転送できます。

CM 1243 5 にサポートされていないファンクション: SYNC/FREEZE および Get_Master_Diag

CM 1243 5 のその他の通信サービス

CM 1243 5 DP マスタモジュールは、以下のその他の通信サービスをサポートします。

- S7 通信
 - PUT/GET サービス

DP マスタは、他の S7 コントローラまたは PC からの PROFIBUS 経由の照会を行うためのクライアントおよびサーバーとして機能します。
 - PG/OP 通信

PG ファンクションを使用して、構成データとユーザープログラムを PG からダウンロードしたり、診断データを PG に転送することができます。

OP 通信が行える通信パートナーは HMI パネル、WinCC flexible を搭載した SIMATIC パネル PC、または S7 通信をサポートする SCADA システムです。

7.6.2 PROFIBUS CMユーザーマニュアルへの参照

詳細情報


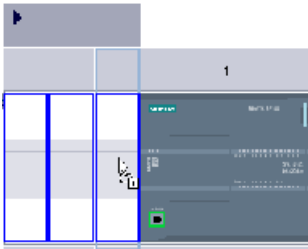
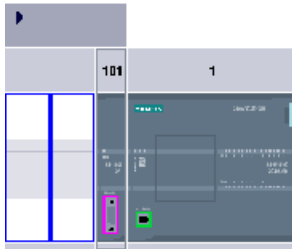
PROFIBUS CM の詳細情報は、デバイスのマニュアルを参照してください。デバイスのマニュアルは、インターネットの Siemens Industrial Automation Customer Support のページの以下のエントリ ID にあります。

- CM 1242 5 (<http://support.automation.siemens.com/WW/view/en/49852105>)
- CM 1243 5 (<http://support.automation.siemens.com/WW/view/en/49851842>)

7.6.3 CM 1243-5 (DPマスタ)モジュールとDPスレーブの追加

[デバイスとネットワーク]ポータルで、ハードウェアカタログを使用して CPU に PROFIBUS モジュールを追加します。通信モジュールは CPU の左側に接続します。ハードウェア構成にモジュールを挿入するには、ハードウェアカタログでモジュールを選択し、そのモジュールをダブルクリックするか、強調表示されているスロットにドラッグします。

表 7-7 PROFIBUS CM 1243-5 (DP マスタ)モジュールをデバイスコンフィグレーションに追加する

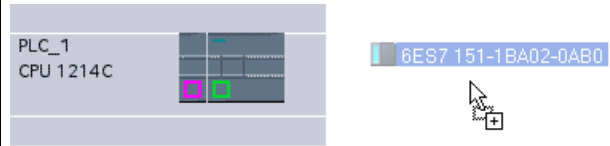

モジュール	モジュールを選択する	モジュールを挿入する	結果
CM 1243-5 (DP マスタ)			

ハードウェアカタログを使用して、DP スレーブも追加します。たとえば、ハードウェアカタログで ET200 S DP スレーブを追加するには、以下のコンテナを拡張します。

- リモート I/O
- ET200 S
- インターフェースモジュール
- PROFIBUS

次に、部品番号のリストから[6ES7 151-1BA02-0AB0] (IM151-1 HF)を選択し、下の図に示すように ET200 S DP スレーブを追加します。

表 7-8 ET200 S DP スレーブをデバイスコンフィグレーションに追加する

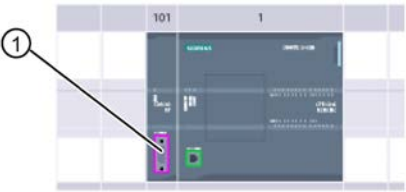
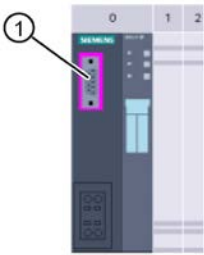
DP スレーブを挿入する	結果
	

7.6.4 CM 1243-5 モジュールとDPスレーブへPROFIBUSアドレスの割り当て

PROFIBUS インターフェースを設定する

2つの PROFIBUS デバイス間に論理ネットワーク接続を構成すると、PROFIBUS インターフェースのパラメータを設定できます。これを行うには、CM 1243-5 モジュールの紫色の PROFIBUS ボックスをクリックすると、インスペクタウィンドウの[プロパティ]タブに PROFIBUS インターフェースが表示されます。DP スレーブ PROFIBUS インターフェースも同じ方法で設定できます。

表 7-9 CM 1243-5 (DP マスタ)モジュールと ET200 S DP スレーブ PROFIBUS インターフェースを設定する

CM 1243-5 (DP マスタ)モジュール	ET200 S DP スレーブ
	

① PROFIBUS ポート

PROFIsafe アドレスを割り当てる

PROFIBUS ネットワークでは、それぞれのデバイスに PROFIBUS アドレスが割り当てられます。このアドレスは、以下の例外を除き、0～127 の範囲とすることができます。

- アドレス 0: ネットワーク構成および/またはバスに接続されるプログラミングツール用に予約済み
- アドレス 1: 最初のマスタ用にシーメンスによって予約済み
- アドレス 126: スイッチ設定がない工場出荷時のデバイス用に予約済みで、ネットワークを介して再アドレス指定が必要
- アドレス 127: ネットワーク上のすべてのデバイスに対するブロードキャストメッセージ用に予約済みで、動作中のデバイスには割り当てられません。

したがって、動作中の PROFIBUS デバイスに使用できるアドレスは 2～125 です。

7.6 PROFIBUS

[プロパティ]ウィンドウで[PROFIBUS アドレス]設定エントリを選択します。STEP 7は、デバイスの PROFIBUS アドレスの割り当てに使用される[PROFIBUS アドレスの設定]ダイアログを表示します。

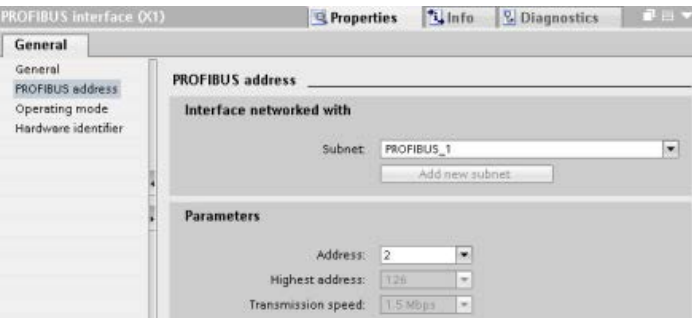


表 7- 10 PROFIBUS アドレスのパラメータ

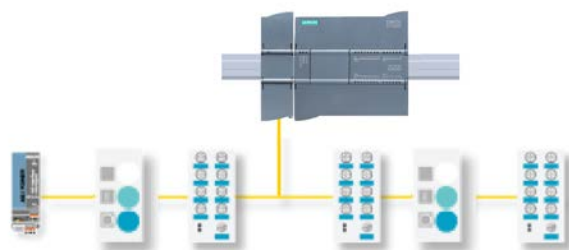
パラメータ		説明
パラメータ	サブネット	デバイスの接続先サブネット名。新しいサブネットを作成するには[新しいサブネットの追加]ボタンをクリックする。[ネットワーク未接続]がデフォルト。次の 2 種類の接続タイプが指定できる。 <ul style="list-style-type: none">デフォルトの[ネットワーク未接続]によるローカル接続ネットワーク上に 2 つ以上のデバイスが存在する場合は、サブネットが必要
	アドレス	デバイスの割り当て済みの PROFIBUS アドレス
	最上位アドレス	最上位 PROFIBUS アドレスは、PROFIBUS 上のアクティブステーション (たとえば、DP マスタ)をベースにしています。パッシブな DP スレーブは、たとえば最上位 PROFIBUS アドレスが 15 に設定されている場合でも、個別に 1～125 の PROFIBUS アドレスを持っています。最上位 PROFIBUS アドレスはトークンの転送(送信権の転送)に関係しており、トークンはアクティブステーションにのみ転送されます。最上位 PROFIBUS アドレスの指定により、バスが最適化されます。
	伝送速度	構成された PROFIBUS ネットワークの伝送速度: PROFIBUS 伝送速度の範囲は 9.6 Kbits/sec～12 Mbits/sec です。伝送速度の設定は使用されている PROFIBUS ノードのプロパティによって決まります。伝送速度は最も低速のノードがサポートする速度より大きくする必要があります。 伝送速度は通常、PROFIBUS ネットワーク上のマスタに対して設定され、すべての DP スレーブは自動的にそれと同じ伝送速度を使用します(自動ボ一)。

7.7 AS-i

S7-1200 AS-i マスタ CM 1243-2 では、AS-i ネットワークを S7-1200 CPU に接続できます。

アクチュエータ/センサインターフェース、すなわち AS-i は、オートメーションシステム内の最下位レベル用の単独のマスタネットワーク接続システムです。CM 1243-2 はネットワークの AS-i マスタとして機能します。1 本の AS-i ケーブルを使用して、センサとアクチュエータ (AS-i スレーブデバイス) を CM 1243-2 経由で CPU に接続できます。CM 1243-2 はすべての AS-i ネットワークの接続を処理し、アクチュエータおよびセンサからのデータおよびステータス情報を、CM 1243-2 に割り当てられた I/O アドレスを経由して CPU に受け渡します。スレーブタイプに応じて、バイナリまたはアナログ値にアクセスできます。AS-i スレーブは AS-i システムの入力および出力チャンネルで、CM 1243-2 に呼び出されたときにのみアクティブになります。




下の図では、S7-1200 は AS-i I/O モジュール、デジタル/アナログスレーブデバイスを制御する AS-i マスタです。



7.7.1 AS-iマスタCM 1243-2 とAS-iスレーブの追加

CPU に AS-i マスタ CM1243-2 モジュールを追加するには、ハードウェアカタログを使用します。これらのモジュールは CPU の左側に接続され、最大 3 つの AS-i マスタ CM1243-2 モジュールを使用できます。ハードウェア構成にモジュールを挿入するには、ハードウェアカタログでモジュールを選択し、そのモジュールをダブルクリックするか、強調表示されているスロットにドラッグします。

表 7- 11 AS-i マスタ CM1243-2 モジュールをデバイスコンフィグレーションに追加する



モジュール	モジュールを選択する	モジュールを挿入する	結果
CM 1243-2 AS-i マスタ			

ハードウェアカタログを使用して、AS-i スレーブも追加します。たとえば、ハードウェアカタログで「I/O モジュール、コンパクト、デジタル、入力」スレーブを追加するには以下のコンテナを拡張します。

- フィールドデバイス
- AS-interface

次に、部品番号のリストから[3RG9 001-0AA00] (AS-i SM-U、4DI)を選択し、下の図に示すように「I/O モジュール、コンパクト、デジタル、入力」スレーブを追加します。

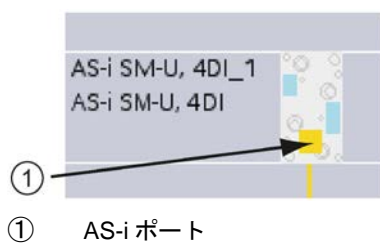
表 7- 12 AS-i スレーブをデバイスコンフィグレーションに追加する

AS-i スレーブを挿入する	結果
	

7.7.2 AS-iアドレスをAS-iスレーブに割り当て

AS-i スレーブインターフェースを設定する

AS-i インターフェースのパラメータを設定するには、AS i スレーブの黄色の AS-i ボックスをクリックすると、インスペクタウィンドウの[プロパティ]タブに AS-i インターフェースが表示されます。



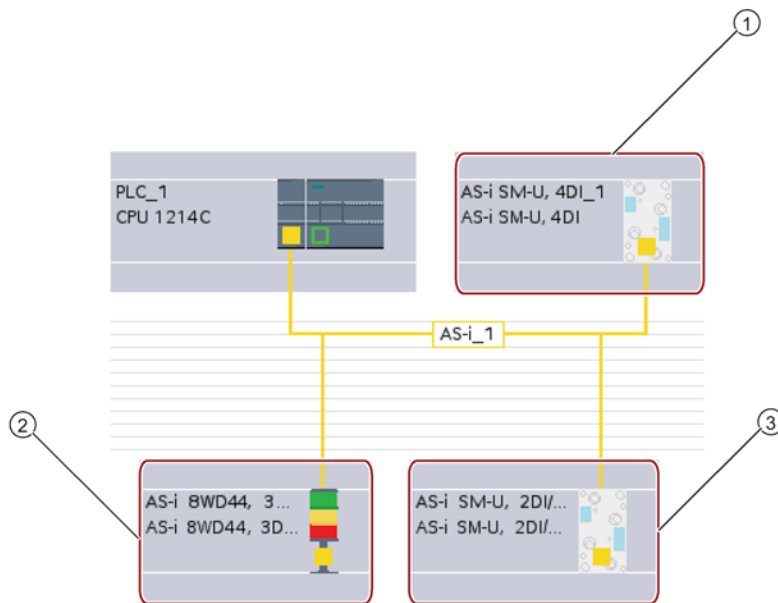
AS-i スレーブアドレスをアドレス指定する

AS-i ネットワークで、それぞれのデバイスに AS-i スレーブアドレスが割り当てられます。このアドレスは、0～31 の範囲とすることができます。ただし、アドレス 0 は新しいスレーブデバイス専用予約されています。スレーブアドレスは、合計で最大 62 のスレーブデバイスに対して 1(A または B)～31(A または B)です。

「標準の」AS-i デバイスはアドレス全体を使用し、A または B の指定なしの数字アドレスを持っています。「A/B ノード」AS-i デバイスは各アドレスの A または B の部分を使用し、31 のアドレスのそれぞれを 2 回使用することができます。アドレススペースの範囲は 1A～31A と 1B～31B です。

1～31 の範囲内のどのアドレスでも、AS-i スレーブデバイスに割り当てることができます。つまり、スレーブがアドレス 21 で始まっている場合、最初のスレーブに実際はアドレス 1 が指定されていても問題ではありません。

下の例では、3 つの AS-i デバイスが「1」(標準タイプのデバイス)、「2A」(A/B ノードタイプのデバイス)、「3」(標準タイプのデバイス)としてアドレス指定されています。



- ① AS-i スレーブアドレス 1; デバイス: AS-i SM-U、4DI; 製品番号: 3RG9 001-0AA00
- ② AS-i スレーブアドレス 2A; デバイス: AS-i 8WD44、3DO、A/B; 製品番号: 8WD4 428-0BD
- ③ AS-i スレーブアドレス 3; デバイス: AS-i SM-U、2DI/2DO; 製品番号: 3RG9 001-0AC00

ここに AS-i スレーブアドレスを入力します:

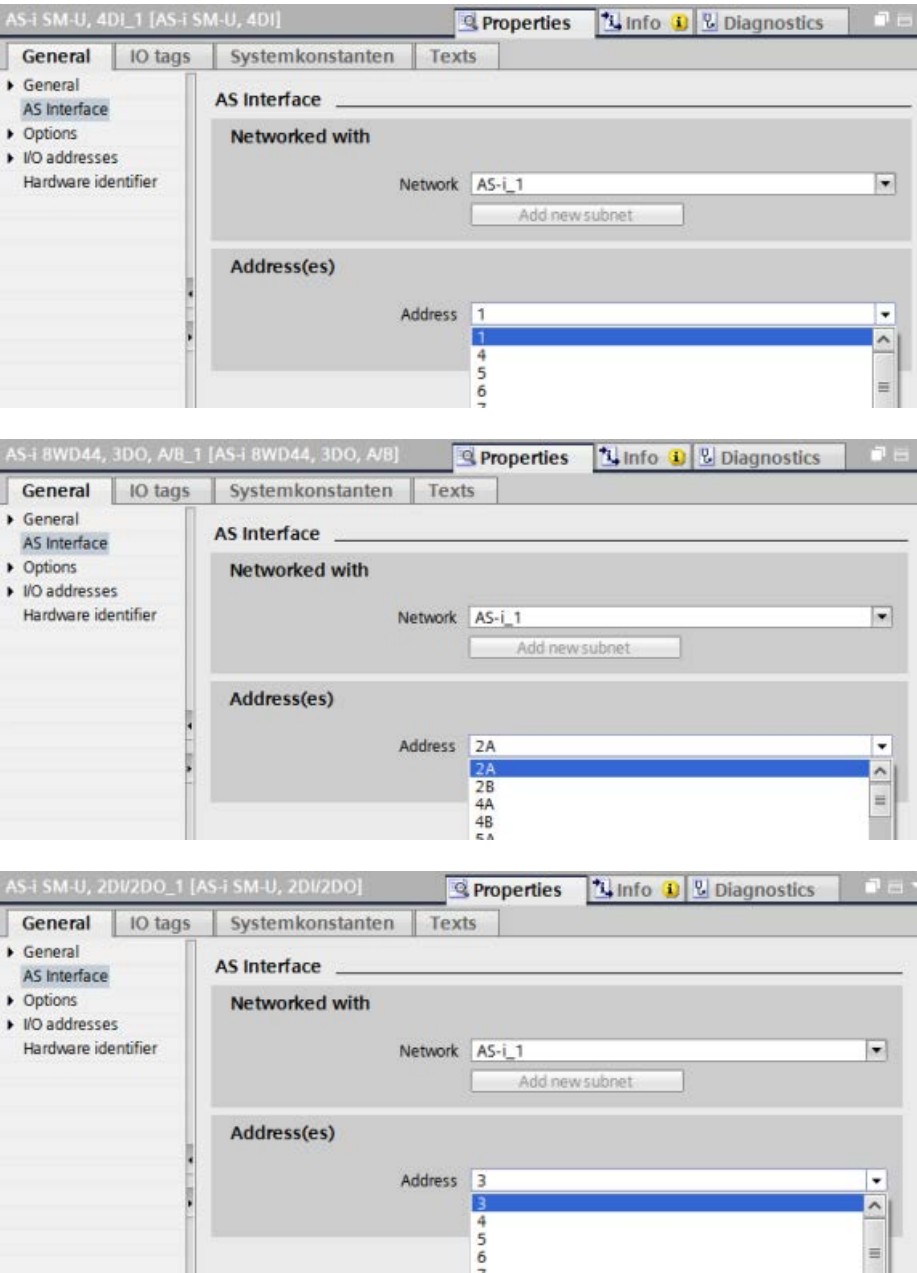


表 7- 13 AS-i インターフェースのパラメータ

パラメータ		説明
ネットワーク	デバイスの接続先ネットワーク名	
アドレス	合計で最大 62 のスレーブデバイスに対して 1(A または B)~31(A または B)の範囲で割り当てられたスレーブデバイスの AS-i アドレス	

7.8 S7 通信

7.8.1 GET命令とPUT命令

GET 命令と PUT 命令を使用して、PROFINET および PROFIBUS 接続経由で S7 CPU と通信することができます。これは、[リモートパートナーからの PUT/GET 通信によるアクセスを許可]ファンクションが、ローカル CPU プロパティの[保護]プロパティでパートナーCPU に対してアクティブになってる場合にのみ可能です。

- リモート CPU 内のデータへのアクセス: S7-1200 CPU は ADDR_x 入力フィールドの絶対アドレスのみを使用して、リモート CPU (S7-200/300/400/1200)の変数をアドレス指定することができます。
- 標準 DB 内のデータへのアクセス: S7-1200 CPU は ADDR_x 入力フィールドの絶対アドレスのみを使用して、リモート S7 CPU の標準 DB 内の DB 変数をアドレス指定することができます。
- 最適化 DB 内のデータへのアクセス: S7-1200 CPU はリモート S7-1200 CPU の最適化 DB 内の DB 変数にアクセスできません。
- ローカル CPU 内のデータへのアクセス: S7-1200 CPU は、絶対アドレスまたはシンボリックアドレスのどちらかを、GET または PUT 命令の RD_x または SD_x 入力フィールドへの入力として使用することができます。

命令を挿入すると STEP 7 が自動的に DB を作成します。

注記

データの整合性を保証するために、演算が完了したときに、データにアクセスするか別の読み取りまたは書き込み操作を開始する前に、必ず評価を行ってください(GET の場合は NDR = 1、または PUT の場合は DONE = 1)。

注記

V4.0 CPU プログラムの GET/PUT 演算は自動的に有効になりません

V3.0 CPU プログラムの GET/PUT 演算は、V4.0 CPU で自動的に有効になります。

しかしながら、V4.0 CPU プログラムの GET/PUT 演算は V4.0 CPU で自動的に有効になりません。CPU の[デバイスコンフィグレーション]でインスペクタウィンドウの[プロパティ]タブの[保護]プロパティに移動して、GET/PUT アクセス(89 ページ)を有効にする必要があります。

7.8.2 S7 コネクションの作成

接続メカニズム

PUT/GET 命令でリモート接続パートナーにアクセスするには、許可を得ている必要があります。

デフォルトでは、[リモートパートナーからの PUT/GET 通信によるアクセスを許可]オプションは無効になっています。この場合、CPU データの読み取りおよび書き込みアクセスは、ローカル CPU と通信パートナーの両方に対する設定またはプログラミングが必要な通信接続でのみ可能です。たとえば、BSEND/BRCV 命令によるアクセスが可能です。

したがって、たとえば、ローカル CPU が単なるサーバーにすぎない接続(つまり、ローカル CPU に通信パートナーとの通信の設定/プログラミングが存在しない)は、CPU の動作中では行えません。

- 通信モジュールを介した PUT/GET、FETCH/WRITE、または FTP アクセス
- 他の S7 CPU からの PUT/GET アクセス
- PUT/GET 通信を介した HMI アクセス

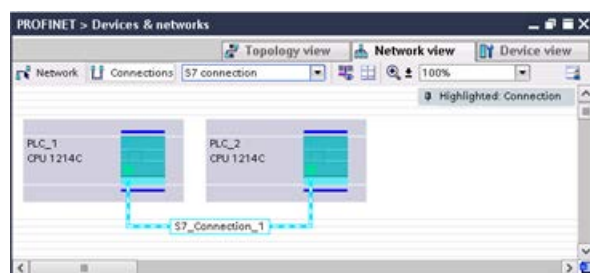
クライアント側から CPU データにアクセスできるようにする場合、つまり、CPU の通信サービスを制限したくない場合は、このセキュリティレベルに対して S7-1200 CPU のアクセス保護(89 ページ)を設定することができます。

接続タイプ

選択する接続タイプによって、パートナーステーションへの通信接続が作成されます。接続がセットアップされ、確立され、自動的にモニタされます。

[デバイスとネットワーク]で[ネットワークビュー]を使用して、プロジェクト内のデバイス間でネットワーク接続を作成します。まず、[接続]タブをクリックしてから、右隣にあるドロップダウンで接続タイプを選択します(たとえば、S7 コネクション)。最初のデバイスの緑色(PROFINET)のボックスをクリックし、2 番目のデバイスの PROFINET ボックスまで線を引きます。マウスのボタンを離すと、PROFINET 接続が作成されます。

詳細は、「ネットワーク接続を作成する」(142 ページ)を参照してください。

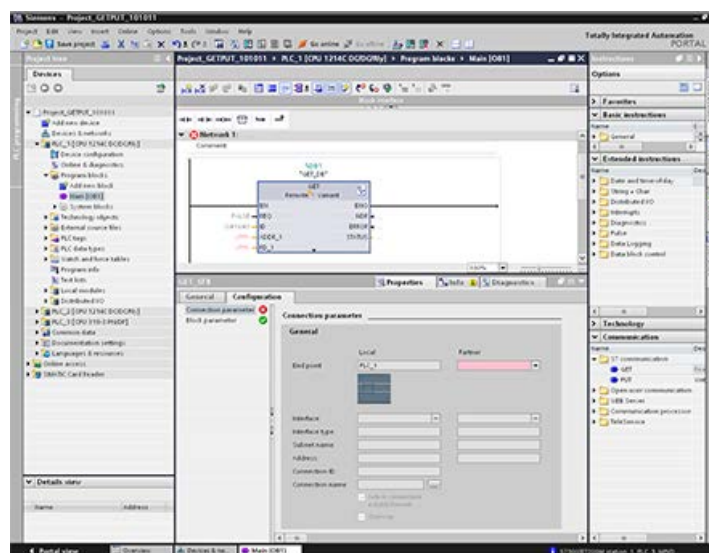


[強調表示: 接続]ボタンをクリックして通信命令の[プロパティの設定]ダイアログにアクセスします。

7.8.3 GET/PUT接続パラメータの割り当て

GET/PUT 命令の接続パラメータの割り当てにより、S7 CPU 間通信接続の設定が行いやすくなります。

GET または PUT ブロックを挿入すると、GET/PUT 命令の接続パラメータ割り当てが開始されます。



インスペクタウィンドウには、命令のどの部分を選択しても必ず接続のプロパティが表示されます。通信命令の[プロパティ]の[設定]タブで、通信パラメータを指定します。

GET または PUT ブロックを挿入すると、[設定]タブが自動的に表示され、[接続パラメータ]ページが即時に表示されます。このページを使用して、必要な S7 コネクションを設定し、ブロックパラメータ[ID]で参照されるパラメータ[接続 ID]を設定できます。[ブロックパラメータ]ページでは、その他のブロックパラメータを設定できます。

注記

V4.0 CPU プログラムの GET/PUT 演算は自動的に有効になりません

V3.0 CPU プログラムの GET/PUT 演算は、V4.0 CPU で自動的に有効になります。

しかしながら、V4.0 CPU プログラムの GET/PUT 演算は V4.0 CPU で自動的に有効になりません。CPU の[デバイスコンフィグレーション]でインスペクタウィンドウの[プロパティ]タブの[保護]プロパティに移動して、GET/PUT アクセス(89 ページ)を有効にする必要があります。

7.9 GPRS

7.9.1 GSMネットワークの接続

GPRS 経由の IP ベースの WAN 通信

CP 1242-7 通信プロセッサを使用して、S7-1200 を GSM ネットワークに接続することができます。**CP 1242-7** により、リモートステーションとコントロールセンターとの WAN 通信およびステーション間通信が可能になります。

ステーション間通信は GSM ネットワーク経由でのみ可能です。リモートステーションとコントロールルームとの間の通信では、コントロールセンターにインターネットアクセスのできる PC が必要です。

CP 1242-7 は、GSM ネットワーク経由の通信のために以下のサービスをサポートします。

- GPRS (General Packet Radio Service)

データ伝送用のパケット指向サービス「GPRS」は GSM ネットワーク経由で処理されます。

- SMS (Short Message Service)

CP 1242-7 は SMS メッセージの送受信を行うことができます。通信パートナーは携帯電話または S7-1200 です。

CP 1242-7 は世界中の産業界での使用に最適で、以下の周波数帯域をサポートします。

- 850 MHz
- 900 MHz
- 1,800 MHz
- 1,900 MHz

必要条件

ステーションまたはコントロールセンターで使用する機器は、特定のアプリケーションによって異なります。

- 中央コントロールルームとの通信または中央コントロールを介した通信の場合、コントロールセンターにはインターネットにアクセスできる PC が必要です。
- ステーションの機器とは別に、**CP 1242-7** を搭載したリモート S7-1200 ステーションは GSM ネットワーク経由の通信が可能になるように、以下の要件を満たす必要があります。

- 適切な GSM ネットワークプロバイダとの契約

GPRS を使用する場合、契約で GPRS サービスの使用が可能になっている必要があります。

GSM ネットワーク経由のみでステーション間の直接通信を行う場合、GSM ネットワークプロバイダは CP に固定の IP アドレスを割り当てる必要があります。この場合、ステーション間の通信はコントロールセンター経由で行われません。

- 契約に付属する SIM カード

SIM カードは **CP 1242-7** に挿入します。

- ステーションの範囲内での GSM ネットワークが可能であること

7.9.2 CP 1242-7 の用途

CP 1242-7 は以下の用途に使用できます。

遠隔制御アプリケーション

- SMS でメッセージを送信する

CP 1242-7 経由で、リモート S7-1200 ステーションの CPU は GSM ネットワークから SMS メッセージを受信したり、SMS によってメッセージを携帯電話や S7-1200 に送信することができます。

- コントロールセンターとの通信

リモート S7-1200 ステーションは、GSM ネットワークおよびインターネット経由でマスタステーションの**遠隔制御サーバー**と通信します。GPRS を使用したデータ転送の場合、「TELECONTROL SERVER BASIC」アプリケーションがマスタステーションの**遠隔制御サーバー**にインストールされています。遠隔制御サーバーは、統合 OPC サーバファンクションを使用する上位レベルの中央制御システムと通信します。

- GSM ネットワーク経由の S7-1200 ステーション間の通信

CP 1242-7 を搭載したリモートステーション間の通信は、以下の 2 つの方法で処理することができます。

- マスタステーション経由のステーション間通信

この構成では、相互通信する S7-1200 ステーションと**遠隔制御サーバー**との間の固定的でセキュアな接続がマスタステーションで確立されます。ステーション間の通信は**遠隔制御サーバー**経由で行われます。CP 1242-7 は「遠隔制御」モードで動作します。

- ステーション間の直接通信

迂回路なしのマスタステーション経由のステーション間の直接通信の場合、固定 IP アドレスを持つ SIM カードを使用して、ステーションが直接、相互にアドレス指定できるようにします。使用できる通信サービスとセキュリティファンクション(たとえば、VPN など)は、ネットワークプロバイダの提供するサービスによって異なります。CP 1242-7 は「GPRS 直接」モードで動作します。

GPRS 経由の TeleService

TeleService 接続は、STEP 7 を搭載したエンジニアリングステーションと CP 1242-7 を搭載したリモート S7-1200 ステーションとの間に、GSM ネットワークおよびインターネットを経由して確立できます。接続は、中間転送フレームとして機能し認証を確立する遠隔制御サーバーまたは TeleService ゲートウェイ経由で、エンジニアリングステーションから実行されます。これらの PC は「TELECONTROL SERVER BASIC」アプリケーションのファンクションを使用します。

TeleService 接続は以下の目的で使用できます。

- 構成データまたはプログラムデータを STEP 7 プロジェクトからステーションにダウンロードする
- ステーションの診断データを照会する

7.9.3 CP-1242-7 のその他のプロパティ

CP 1242-7 のその他のサービスとファンクション

- インターネット経由の CP の時刻同期

CP の時間を以下のように設定できます。

- 「遠隔制御」モードでは、時間が遠隔制御サーバーによって転送されます。CP はこれを使用して時間を設定します。
- 「GPRS 直接」モードでは、CP は SNTP を使用して時間を要求できます。

CPU の時間を同期化するために、ブロックを使用して CP から現在の時間を読み出すことができます。

- 接続に問題がある場合に送信するメッセージの中間バッファリング
- 代替遠隔制御サーバーへの接続オプションによる可用性の向上
- データ量の最適化(テンポラリ接続)

遠隔制御サーバーへの固定的な接続の代わりに、STEP 7 で遠隔制御サーバーにテンポラリ接続された CP を構成することができます。この場合、遠隔制御サーバーへの接続は必要なときにのみ確立されます。

- データ量のロギング

転送されたデータの量をログして、特定の目的で評価することができます。

7.9.4 構成と電気的接続

構成とモジュール交換

モジュールを構成するには、以下の構成ツールが必要です。

STEP 7 バージョン V11.0 SP1 以降

STEP 7 V11.0 SP1 の場合、サポートパッケージ「**CP 1242-7**」(HSP0003001)も必要です。

GPRS を使用したプロセスデータの転送には、ステーションのユーザープログラムにある遠隔制御通信命令を使用します。

CP 1242-7 の構成データはローカル CPU に保存されます。これにより、必要に応じて CP を簡単に交換できます。

S7-1200 あたり最大で 3 つの **CP 1242-7** タイプのモジュールを挿入できます。これで、たとえば、冗長化通信経路を確立することができます。

電気的接続

- **CP 1242-7 の電源**

CP には、外部 24V DC 電源用の個別の接続があります。

- GSM ネットワーク用の無線インターフェース

GSM 通信用の追加のアンテナが必要です。このアンテナは CP の SMA ソケット経由で接続されます。

7.9.5 その他の情報

詳細情報

CP 1242-7 のマニュアルに、詳細な情報が記載されています。マニュアルは、インターネットの Siemens Industrial Automation Customer Support のページの以下のエントリ ID にあります。

45605894 (<http://support.automation.siemens.com/WW/view/en/45605894>)

7.9.6 付属品

ANT794-4MR GSM/GPRS アンテナ

以下のアンテナが GSM/GPRS ネットワークで使用するために用意されており、屋内にも屋外にも設置できます。

- クワッドバンドアンテナ ANT794 4MR



略称	注文番号	説明
ANT794-4MR	6NH9 860-1AA00	クワッドバンドアンテナ(900、1800/1900 MHz、UMTS); 屋内および屋外領域用の全天候型; 5 m の接続ケーブルがアンテナに固定的に接続; SMA コネクタ、取り付け金具、ネジ、ウォールプラグ含む

- フラットアンテナ ANT794 3M



略称	注文番号	説明
ANT794-3M	6NH9 870-1AA00	フラットアンテナ(900、1800/1900 MHz); 屋内および屋外領域用の全天候型; 1.2 m の接続ケーブルがアンテナに固定的に接続; SMA コネクタ、接着パッド含む、ネジによる取り付け可能

アンテナは別途注文してください。

7.9.7 GSMアンテナマニュアルの参照

詳細情報

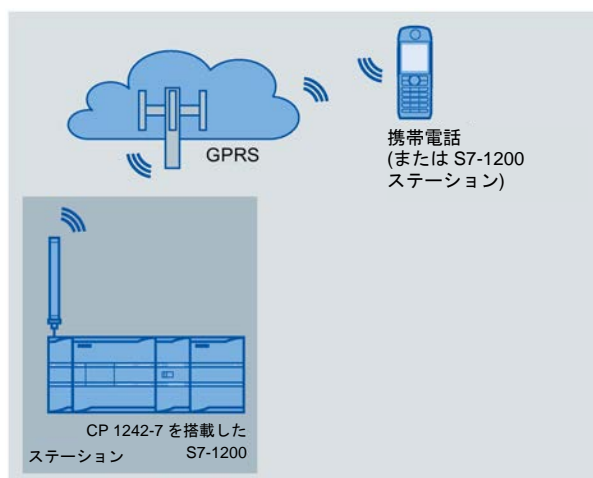
詳細はデバイスマニュアルを参照してください。マニュアルは、インターネットの Siemens Industrial Automation Customer Support のページの以下のエントリ ID にあります。

23119005 (<http://support.automation.siemens.com/WW/view/en/23119005>)

7.9.8 遠隔制御の構成例

下に、CP 1242-7 を搭載したステーションの構成例をいくつか示します。

SMS でメッセージを送信する



CP 1242-7 を搭載した SIMATIC S7-1200 は、SMS でメッセージを携帯電話または構成済みの S7-1200 ステーションに送信できます。

コントロールセンターによる遠隔制御

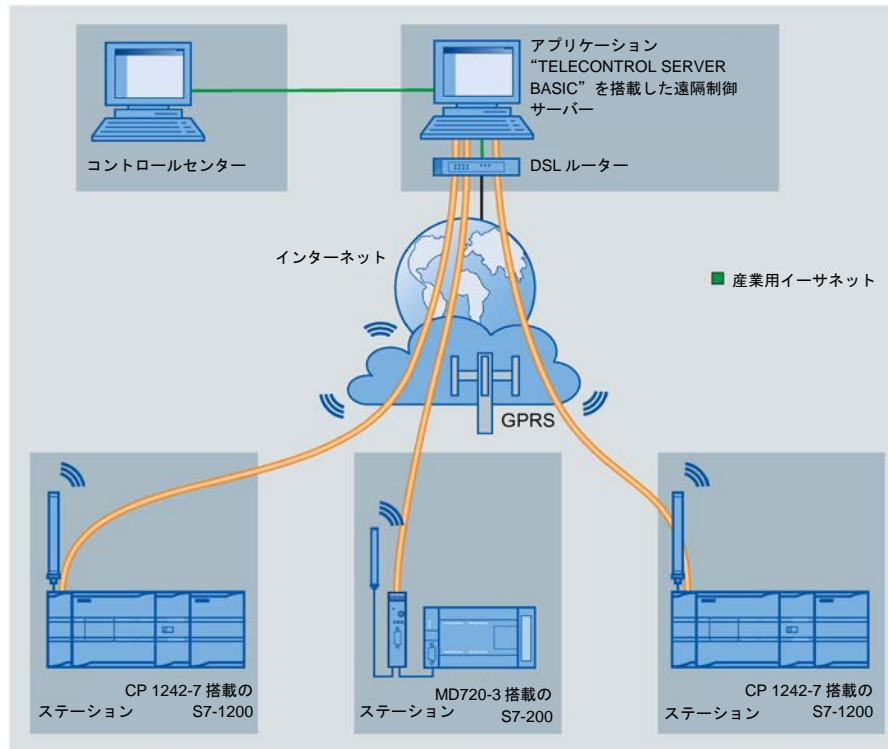


図 7-1 S7-1200 ステーションとコントロールセンター間の通信

遠隔制御アプリケーションでは、**CP 1242-7** を搭載した SIMATIC S7-1200 ステーションは GSM ネットワークおよびインターネット経由でコントロールセンターと通信します。

「TELECONTROL SERVER BASIC」アプリケーションがマスターステーションの**遠隔制御サーバー**にインストールされています。その結果として、以下のような使用事例があります。

- ステーションとコントロールセンター間の遠隔制御通信

この使用事例では、フィールドからのデータは GSM ネットワークおよびインターネット経由で、ステーションによってマスターステーションの遠隔制御サーバーに送信されます。遠隔制御サーバーはリモートステーションのモニタに使用されます。

- ステーションと OPC クライアントの設置されたコントロールルーム間の通信

最初の事例と同じように、ステーションは遠隔制御サーバーと通信します。統合 OPC サーバーを使用して、遠隔制御サーバーはコントロールルームの OPC クライアントとデータを交換します。

OPC クライアントと遠隔制御サーバーは、たとえば、TCSB が WinCC を搭載したコントロールセンターのコンピュータにインストールされている場合に、単独のコンピュータに配置できます。

- コントロールセンター経由のステーション間通信

ステーション間通信は、CP 1242-7 を搭載した S7 ステーションで可能です。

ステーション間通信を可能にするために、遠隔制御サーバーは送信側のステーションのメッセージを受信側のステーションに転送します。

ステーション間の直接通信

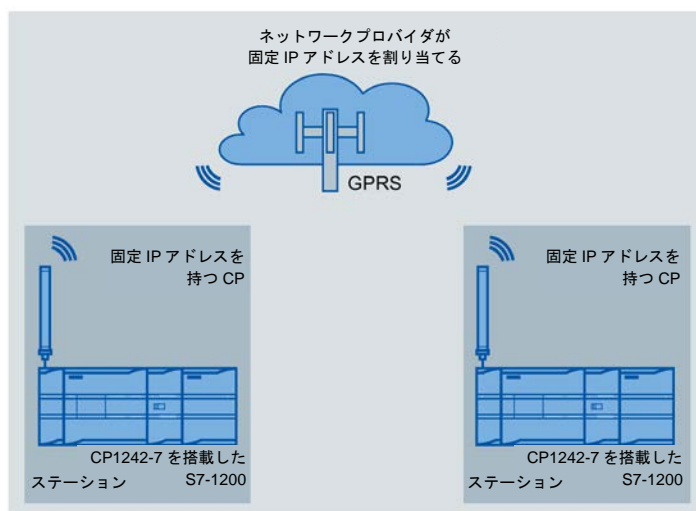


図 7-2 2つの S7-1200 ステーション間の直接通信

この構成では、2つの SIMATIC S7-1200 ステーションが **CP 1242-7** を使用して GSM ネットワーク経由で直接、相互通信します。それぞれの **CP 1242-7** には固定の IP アドレスがあります。GSM ネットワークプロバイダの関連するサービスで、これが許可されている必要があります。

GPRS 経由の TeleService

GPRS 経由の TeleService では、STEP 7 がインストールされているエンジニアリングステーションが GSM ネットワークとインターネットを経由して、S7-1200 の **CP 1242-7** と通信します。

ファイアウォールは通常、外部からの接続要求に対して閉じられているため、リモートステーションとエンジニアリングステーションとの間のスイッチングステーションが必要です。このスイッチングステーションは遠隔制御サーバーとすることができます。または、遠隔制御サーバーが構成にない場合は、TeleService ゲートウェイとすることができます。

遠隔制御サーバーによる TeleService

接続は、遠隔制御サーバーを介して行われます。

- エンジニアリングステーションと遠隔制御サーバーは、イントラネット(LAN)またはインターネット経由で接続されます。
- 遠隔制御サーバーとリモートステーションは、インターネットおよび GSM ネットワーク経由で接続されます。

エンジニアリングステーションと遠隔制御サーバーは同じコンピュータとすることもできます。つまり、STEP 7 と TCSB が同じコンピュータにインストールされます。

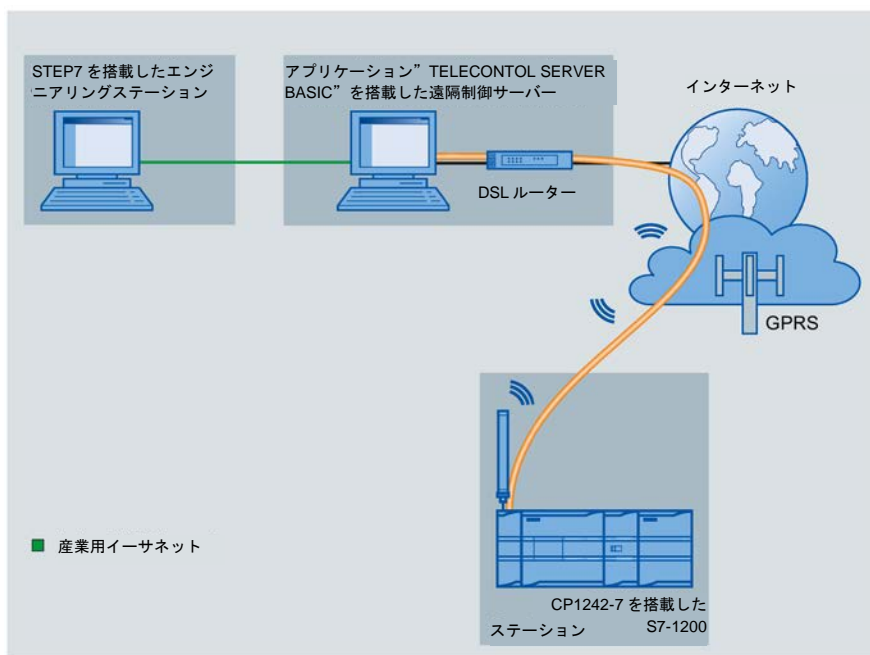


図 7-3 遠隔制御サーバーのある構成での GPRS 経由の TeleService

遠隔制御サーバーなしの TeleService

接続は、TeleService ゲートウェイを介して行われます。

エンジニアリングステーションと TeleService ゲートウェイ間の接続は、LAN またはインターネット経由のローカル接続とすることができます。

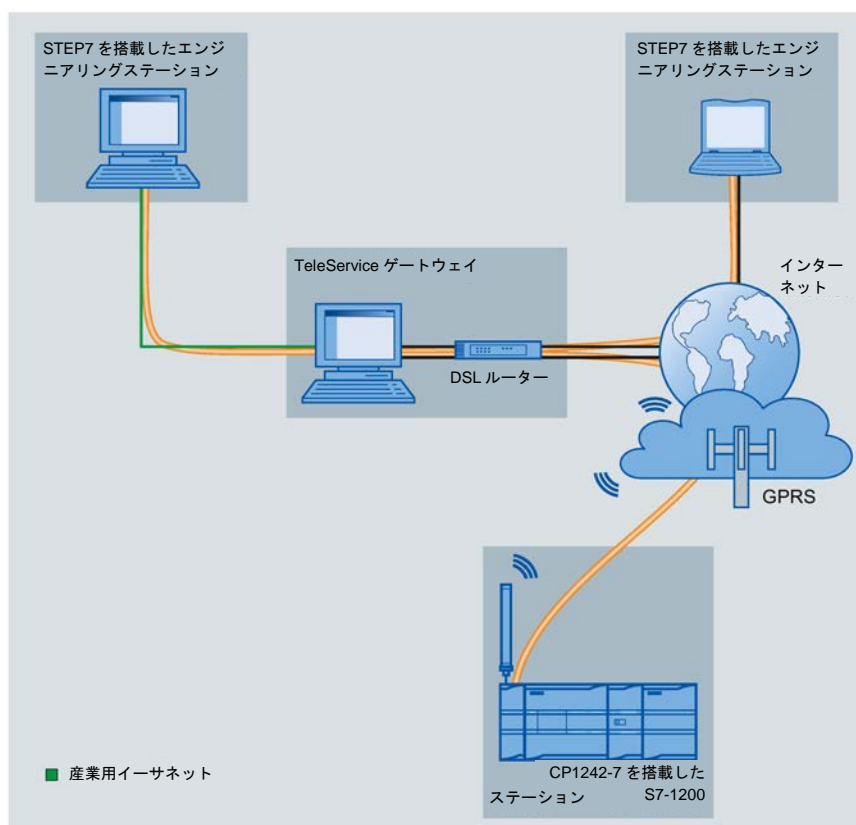


図 7-4 TeleService ゲートウェイのある構成での GPRS 経由の TeleService

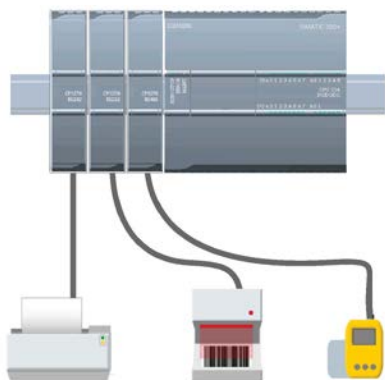
7.10 PtP、USS、および Modbus 通信プロトコル

7.10.1 ポイントツーポイント通信

CPU は文字ベースのシリアルプロトコル用に以下のポイントツーポイント通信(PtP)をサポートしています。

- PtP (187 ページ)
- USS (188 ページ)
- Modbus (190 ページ)

PtP では最大限の自由と柔軟性が得られますが、ユーザープログラム側で多数の実装が必要になります。



PtP により幅広い可能性が提供されます。

- プリンタなどの外部装置に情報を直接送信する機能
- バーコードリーダー、RFID リーダー、サードパーティ製カメラ、表示システム、その他の各種装置など他の装置から情報を読み取る機能
- GPS デバイス、サードパーティ製カメラや表示システム、無線モデムなど、その他の装置とデータの送受信を行って情報を交換する機能

このタイプの PtP 通信は、標準の UART を使用して各種のボーレートとパリティオプションをサポートするシリアル通信です。RS232 および RS422/485 通信モジュール(CM 1241)と RS485 通信ボード(CB 1241)は、PtP 通信を行うための電気インターフェースを提供します。

PROFIBUS または PROFINET 経由の PtP

バージョン V4.1 の S7-1200 CPU に STEP 7 V13 SP1 を搭載すると、PtP の機能が拡張され、PROFINET または PROFIBUS リモート I/O ラックを使用してさまざまなデバイス (RFID リーダー、GPS デバイス、その他) と通信することができます。

- PROFINET (149 ページ): S7-1200 CPU のイーサネットインターフェースを PROFINET インターフェースモジュールに接続します。これで、インターフェースモジュールのラック内の PtP 通信モジュールが、PtP デバイスへのシリアル通信を行えます。
- PROFIBUS (159 ページ): PROFIBUS 通信モジュールを S7-1200 CPU の左側のラックに挿入します。PROFIBUS 通信モジュールを PROFIBUS インターフェースモジュールの入ったラックに接続します。これで、インターフェースモジュールのラック内の PtP 通信モジュールが、PtP デバイスへのシリアル通信を行えます。

上記の理由で、S7-1200 は 2 セットの PtP 命令をサポートしています。

- レガシーポイントツーポイント命令: この命令は S7-1200 の V4.0 以前に存在していたもので、CM 1241 通信モジュールまたは CB 1241 通信ボードを使用したシリアル通信でのみ機能します。
- ポイントツーポイント命令 (187 ページ): この命令はレガシー命令のすべての機能に加えて、PROFINET および PROFIBUS リモート I/O への接続機能も提供します。ポイントツーポイント命令を使用して、リモート I/O ラック内の PtP 通信モジュールと PtP デバイス間に通信を設定することができます。

注記

S7-1200 のバージョン V4.1 では、すべてのタイプのポイントツーポイント通信にポイントツーポイント命令を使用することができます: シリアル、PROFINET を介したシリアル、PROFIBUS を介したシリアル。STEP 7 は、既存のプログラムをサポートするためだけにレガシーポイントツーポイント命令を提供します。ただし、レガシー命令は V4.1 CPU だけでなく、V4.0 以前の CPU でも機能します。以前のプログラムの命令セットを別の命令セットに変換する必要はありません。

7.10.2 シリアル通信インターフェースの使用

2つの通信モジュール(CM)と1つの通信ボード(CB)が、PtP 通信のインターフェースになります。

- CM 1241 RS232 (428 ページ)
- CM 1241 RS422/485 (427 ページ)
- CB 1241 RS485 (425 ページ)

最大3つのCM(任意のタイプ)と1つのCBを接続して、合計で4つの通信インターフェースとすることができます。CPUまたは他のCMの左側にCMを取り付けます。CBはCPUの前面に取り付けます。モジュールの取り付けと取り外しについては、『設置ガイドライン』(21 ページ)を参照してください。

シリアル通信インターフェースには以下の特性があります。

- 独立したポートを持つ
- ポイントツーポイントプロトコルをサポート
- ポイントツーポイント通信プロセッサ命令によって構成およびプログラミングされる
- LEDによって送受信動作を表示する
- 診断LEDを表示する(CMのみ)
- CPUから電源を供給される:外部電源接続が不要。

通信インターフェースの技術仕様(416 ページ)を参照してください。

LED インジケータ

通信モジュールには3つのLEDインジケータがあります。

- 診断LED (DIAG): このLEDはCPUによってアドレス指定されるまで赤く点滅します。CPUの電源投入後、CMをチェックしてアドレス指定します。診断LEDが緑に点滅し始めます。これは、CPUがCMをアドレス指定したがコンフィグレーションを提供していないことを示します。プログラムがCPUにダウンロードされると、CPUは構成されたCMにコンフィグレーションをダウンロードします。CPUにダウンロードされると、通信モジュールの診断LEDは緑に点灯します。
- 送信LED (Tx): 送信LEDは、データが通信ポートから送出されているときに点灯します。
- 受信LED (Rx): このLEDは、データが通信ポートによって受信されているときに点灯します。

通信ボードには送信(TxD)LEDと受信(RxD)LEDがあります。診断LEDはありません。

7.10.3 PtP命令

Port_Config、Send_Config、および Receive_Config 命令を使用して、ユーザープログラムから設定を変更することができます。

- Port_Config はボーレートなどのポートパラメータを変更します。
- Send_Config はシリアル送信パラメータの設定を変更します。
- Receive_Config は通信ポートのシリアル受信パラメータの設定を変更します。この命令は、受信メッセージの開始と終了を知らせる条件を設定します。この条件を満たすメッセージが、Receive_P2P 命令によって受信されます。

動的設定の変更は、CPU に永続的には保存されません。電源を切った後に再投入すると、デバイスコンフィギュレーションの初期の静的設定が使用されます。

Send_P2P、Receive_P2P、および Receive_Reset 命令は PtP 通信を制御します。

- Send_P2P は指定されたバッファを CM または CB に転送します。CPU はユーザープログラムの実行を続行し、一方、モジュールは指定されたボーレートでデータを送信します。
- Receive_P2P は、CM または CB で受信されたメッセージをチェックします。メッセージがある場合、CPU に転送されます。
- Receive_Reset は受信バッファをリセットします。

それぞれの CM または CB は最大 1K バイトをバッファできます。このバッファは複数の受信メッセージにわたって割り当てることができます。

Signal_Set 命令と Signal_Get 命令は RS232 CM に対してのみ有効です。この命令を使用して、RS232 通信信号の読み取りまたは設定を行います。

Get_Features 命令と Set_Features 命令を使用して、プログラムがモジュールの機能を読み取って設定できるようにすることができます。

7.10.4 USS命令

S7-1200 は USS プロトコルをサポートし、CM または CB の RS485 ポートを経由してドライブと通信するために特別に設計された命令を提供します。USS 命令を使って、物理ドライブや読み書きドライブパラメータを制御できます。各 RS485 CM または CB は、最大 16 のドライブをサポートします。

- USS_Port_Scan 命令は、CPU と 1 つの CM または CB に接続されているすべてのドライブとの間の実際の通信を処理します。アプリケーションの CM または CB ごとに別々の USS_Port_Scan 命令を挿入します。ユーザープログラムが、ドライブによる通信タイムアウトを防止するのに十分な速さで USS_Port_Scan 命令を確実に実行するようにしてください。USS_Port_Scan 命令はプログラムサイクルまたは任意の割り込み OB で使用します。
- USS_Drive_Control 命令は USS ネットワーク上の指定されたドライブにアクセスします。USS_Drive_Control 命令の入力および出力パラメータは、ドライブのステータスと制御です。ネットワーク上に 16 台のドライブがある場合、ユーザープログラムは、各ドライブについて 1 つずつ、少なくとも 16 の USS_Drive_Control 命令が必要です。

CPU が USS_Drive_Control 命令をドライブのファンクションを制御するのに必要な速度で確実に実行するようにしてください。USS_Drive_Control 命令はプログラムサイクル OB でのみ使用します。

- USS_Read_Param 命令と USS_Write_Param 命令は、リモートドライブの動作パラメータの読み取りと書き込みを行います。これらのパラメータは、ドライブの内部動作を制御します。これらのパラメータの定義については、ドライブマニュアルを参照してください。

プログラムにこれらの命令を必要な数だけ含ませることができます。ただし、1 つのドライブに対していつでもアクティブにできるのは 1 つの読み取りまたは書き込み要求だけです。USS_Read_Param 命令と USS_Write_Param 命令はプログラムサイクル OB でのみ使用します。

インスタンス DB には、各 CM または CB に接続されている USS ネットワーク上のすべてのドライブに対するテンポラリ記憶領域とバッファが含まれています。ドライブの USS 命令はインスタンス DB を使用して情報を共有します。

ドライブとの通信に必要な時間の計算

ドライブとの通信は、CPU スキャンに対して非同期です。通常、CPU は、1 つのドライブ通信トランザクションが完了する前に複数のスキャンを完了します。

USS_PORT_Scan 間隔は、1 つのドライブトランザクションに必要な時間です。以下の表は、各通信ボーレートに対する最小の USS_Port_Scan 間隔を示しています。

USS_Port_Scan ファンクションを USS_Port_Scan 間隔よりも高い頻度で呼び出しても、トランザクションの数は増えません。ドライブタイムアウト間隔は、通信エラーによりトランザクション完了のため 3 回の試行が行われた場合、あるトランザクションにかかったと思われる時間の量です。デフォルトで、USS プロトコルライブラリは各トランザクションで自動的に最大 2 回の試行を行います。

表 7- 14 時間要件の計算

ボーレート	計算された最少 USS_Port_Scan 呼び出し 間隔(ミリ秒)	ドライブ当たりのドライブメッセー ジ間隔タイムアウト(ミリ秒)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

7.10.5 Modbus命令

CPU は異なったネットワークを介した Modbus 通信をサポートしています。

- Modbus RTU (Remote Terminal Unit)は、Modbus ネットワークデバイス間のシリアルデータ転送のための RS232 または RS485 電気接続を使用する、標準のネットワーク通信プロトコルです。RS232 または RS485 CM、または RS485 CB を搭載した CPU に PtP (ポイントツーポイント)ネットワークポートを追加することができます。

Modbus RTU は、すべての通信が単独のマスタデバイスによって開始され、スレーブはマスタの要求に応答するのみのマスタ/スレーブネットワークを使用します。マスタは 1 つのスレーブアドレスに要求を送信し、そのスレーブアドレスのみが命令に応答します。

- Modbus TCP (Transmission Control Protocol)は、TCP/IP 通信用の CPU の PROFINET コネクタを使用する標準のネットワーク通信プロトコルです。追加のハードウェアモジュールは必要ありません。

Modbus TCP は Modbus 通信経路としてクライアント-サーバ間接続を使用します。STEP 7 と CPU 間の接続に加えて、複数のクライアント-サーバ間接続が存在できます。クライアント接続とサーバ接続の組み合わせは、CPU が許可する接続の最大数までサポートされます。各 MB_SERVER 接続は、一意のインスタンス DB と IP ポート番号を使用する必要があります。1 つの IP ポートあたり 1 つの接続しかサポートされません。各 MB_SERVER (一意のインスタンス DB および IP ポートを持つ)は、接続ごとに個別に実行する必要があります。



警告

攻撃側がネットワークに物理的にアクセスできる場合、データの読み取りまたは書き込みが可能であると思われる。

TIA ポータル、CPU、HMI (GET/PUT を使用する HMI 以外)は、反射攻撃および「中間者」攻撃を防ぐ安全な通信を使用しています。いったん通信が可能になると、署名付きのメッセージの交換が明確なテキストで行われ、攻撃側はデータを読むことはできませんが、データの未許可の書き込みが防止されます。TIA ポータルは通信プロセスではなく、ノウハウプロテクトブロックのデータを暗号化します。

他のすべての通信形態(PROFIBUS、PROFINET、AS-i、またはその他の I/O バス、GET/PUT、T ブロック、通信モジュール(CM)を介した I/O 交換)にはセキュリティ機能はありません。このような通信形態は、物理的アクセスを制限することで保護する必要があります。攻撃側がこのような通信形態を使用してネットワークに物理的にアクセスできる場合、データの読み取りまたは書き込みが可能であると思われる。

セキュリティ情報と推奨事項については、サービス&サポートサイトにある弊社の『産業セキュリティの操作ガイドライン』を参照してください。

www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf
(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

注記

Modbus TCP は、CPU ファームウェアリリース V1.02 以降でのみ正しく動作します。それ以前のファームウェアバージョンで Modbus 命令の実行を試みると、エラーが発生します。

表 7- 15 Modbus 命令

通信のタイプ	命令
Modbus RTU (RS232 または RS485)	Modbus_Comm_Load: Modbus_Comm_Load を 1 回使用すると、ボーレート、パリティ、フロー制御などの PtP ポートパラメータが設定されます。Modbus RTU プロトコルに対して CPU ポートを設定すると、Modbus_Master 命令または Modbus_Slave 命令のいずれかのみがポートを使用できます。
	Modbus_Master: Modbus master 命令により、CPU が Modbus RTU マスタデバイスとして機能可能となり、複数の Modbus スレーブデバイスと通信できます。
	Modbus_Slave: Modbus slave 命令により、CPU が Modbus RTU スレーブデバイスとして機能可能となり、Modbus マスタデバイスと通信できます。
Modbus TCP (PROFINET)	MB_CLIENT: クライアント-サーバー間 TCP 通信の作成、コマンドメッセージの送信、応答の受信、サーバーからの切断の制御を行います。
	MB_SERVER: 要求に応じて Modbus TCP クライアントに接続し、Modbus メッセージを受信し、応答を送信します。

Modbus 命令は、通信プロセスの制御に通信割り込みイベントを使用しません。ユーザーのプログラムは、完全な条件を送信および受信するために、Modbus_Master / Modbus_Slave または MB_CLIENT/ MB_SERVER 命令をポーリングする必要があります。

Modbus TCP クライアント(マスタ)は、DISCONNECT パラメータでクライアント-サーバー間接続を制御する必要があります。基本的な Modbus クライアントの動作を以下に示します。

1. 特定のサーバー(スレーブ)の IP アドレスおよび IP ポート番号への接続を開始する
2. クライアントからの Modbus メッセージの送信とサーバーからの応答の受信を開始する
3. 必要に応じて、別のサーバーとの接続を可能にするために、クライアントおよびサーバーの接続を開始する

カンタン PID

STEP 7 は S7-1200 CPU 用の以下の PID 命令を提供します。

- PID_Compact 命令は、連続する入力変数および出力変数でテクニカルプロセスを制御するのに使用します。
- PID_3Step 命令は、開作動および閉作動用の個別の信号を必要とするバルブなどのモータ駆動型デバイスの制御に使用します。
- PID_Temp 命令は、温度制御の特定の要件の処理を可能にする汎用 PID コントローラを提供します。

注記

PID 設定に行った変更と RUN 中のダウンロードは、CPU が STOP モードから RUN モードに移行するまで有効になりません。[開始値の制御]を使用して[PID パラメータ]ダイアログで行った変更は、即時に有効になります。

3つのすべての PID 命令(PID_Compact、PID_3Step、PID_Temp)は、スタートアップ時に P 成分、I 成分、D 成分の計算ができます(「プレチューニング」用に設定されている場合)。また、「ファインチューニング」用の命令を設定してパラメータを最適化することもできます。手動でパラメータを決定する必要はありません。

注記

PID 命令は、サンプリング時間の一定の間隔で実行します(定周期 OB での実行を推奨)。

PID ループは制御値の変更に応答するのに一定の時間が必要なため、サイクルごとに出力値を計算しないでください。PID 命令をメインプログラムサイクル OB (OB 1 など)で実行しないでください。

PID アルゴリズムのサンプリング時間は、出力値(制御値)の 2つの計算の間の時間を表しています。出力値はセルフチューニングの間に計算され、サイクルタイムの倍数に丸められます。PID 命令の他のすべてのファンクションは呼び出しごとに実行されます。

PID アルゴリズム

PID (比例/積分/微分)コントローラは、2つの呼び出し間の時間間隔を測定し、サンプリング時間のモニタリングの結果を評価します。サンプリング時間の平均値は、各モードの切り替え時と最初のスタートアップ時に生成されます。この値は、モニタリングファンクションの参照として使用され、計算に使用されます。モニタリングには、2つの呼び出し間の最新の測定時間と定義されたコントローラのサンプリング時間の平均値が含まれます。

PID コントローラの出力値は、3つの成分で構成されます。

- P (比例): 「P」成分で計算された場合、出力値はセットポイントとプロセス値(入力値)の間の差分に比例します。
- I (積分): 「I」成分で計算された場合、出力値はセットポイントとプロセス値(入力値)の間の差分の時間に比例して増大し、最終的には差分を修正します。
- D (微分): 「D」成分で計算された場合、出力値はセットポイントとプロセス値(入力値)の間の差分の変化の増大率の関数として増大します。出力値はできるだけ迅速にセットポイントに修正されます。

PID コントローラは以下の式を使用して、PID_Compact 命令の出力値を計算します。

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	出力値	x	プロセス値
W	セットポイント値	s	Laplace 演算子
K _p	比例ゲイン (P 成分)	A	微分遅延係数 (D 成分)
T _i	積分動作時間 (I 成分)	B	比例動作の重みづけ (P 成分)
T _d	微分動作時間 (D 成分)	C	微分動作の重みづけ (D 成分)

PID コントローラは以下の式を使用して、PID_3Stept 命令の出力値を計算します。

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	出力値	x	プロセス値
W	セットポイント値	s	Laplace 演算子
K _p	比例ゲイン (P 成分)	A	微分遅延係数 (D 成分)
T _i	積分動作時間 (I 成分)	B	比例動作の重みづけ (P 成分)
T _d	微分動作時間 (D 成分)	C	微分動作の重みづけ (D 成分)

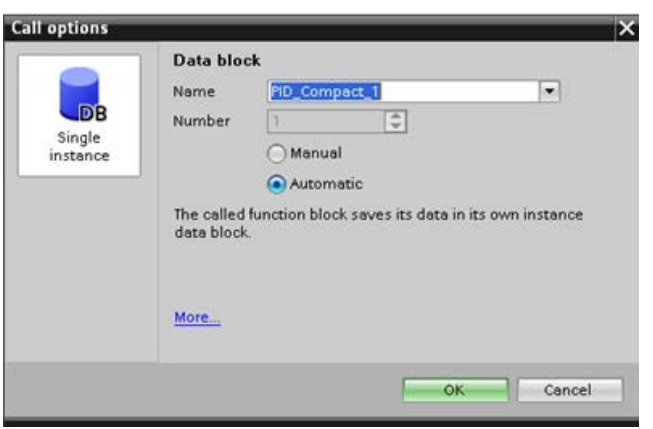
8.1 PID 命令とテクノロジーオブジェクトの挿入

STEP 7 は PID 制御用の 2 つの命令を提供します。

- PID_Compact 命令とその関連するテクノロジーオブジェクトは、汎用 PID コントローラをチューニングします。テクノロジーオブジェクトには制御ループのすべての設定が含まれています。
- PID_3Step 命令とその関連するテクノロジーオブジェクトは、PID コントローラのモータ駆動型バルブの特定の設定を提供します。テクノロジーオブジェクトには制御ループのすべての設定が含まれています。PID_3Step コントローラには、さらに 2 つの Boolean 出力があります。

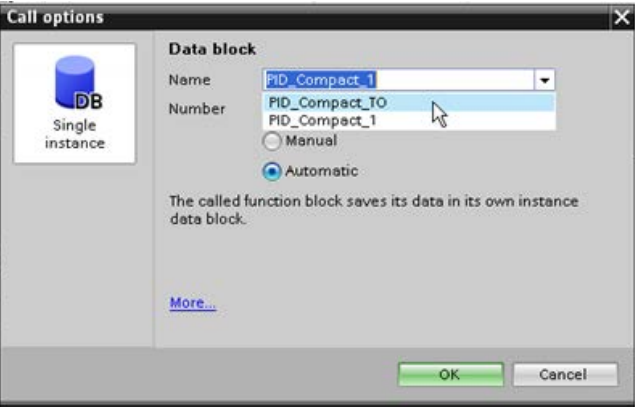
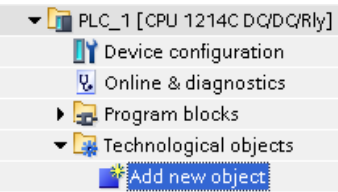
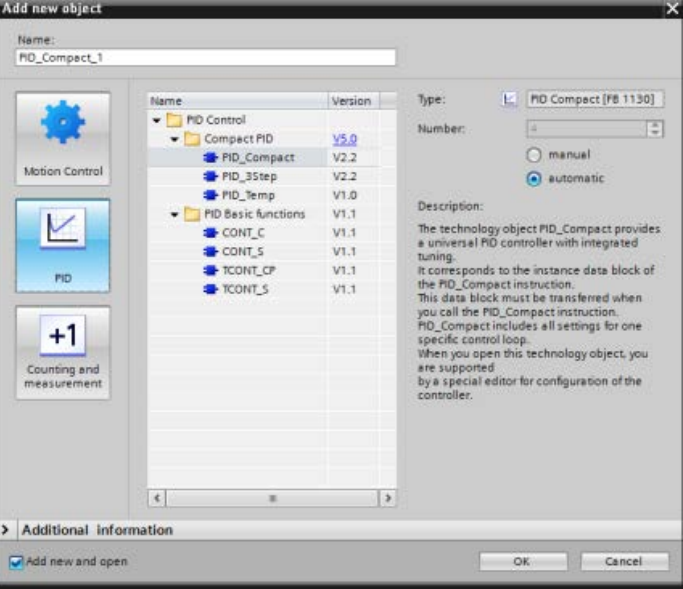
テクノロジーオブジェクトを作成した後で、パラメータを設定する(224 ページ)必要があります。オートチューニングパラメータを調整して、(スタートアップ時の「プレチューニング」または手動の「ファインチューニング」、PID コントローラの動作をコミショニング(241 ページ)することもできます。

表 8-1 PID 命令とテクノロジーオブジェクトを挿入する

<p>PID 命令をユーザープログラムに挿入すると、STEP 7 が命令用のテクノロジーオブジェクトとインスタンス DB を自動的に作成します。インスタンス DB には、PID 命令が使用するすべてのパラメータが含まれています。各 PID 命令には、正しく動作するために独自の一意のインスタンス DB が必要となります。</p> <p>PID 命令を挿入してテクノロジーオブジェクトとインスタンス DB を作成した後で、テクノロジーオブジェクトのパラメータを設定します(224 ページ)。</p>	
--	---

8.1 PID 命令とテクノロジーオブジェクトの挿入

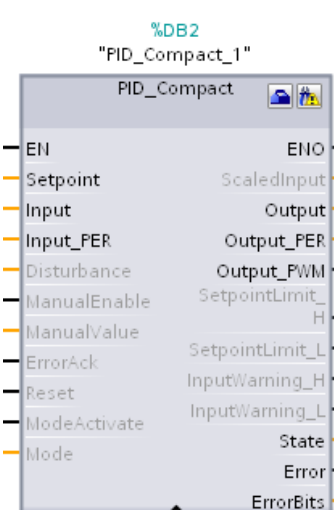
表 8-2 (オプション)プロジェクトナビゲータからテクノロジーオブジェクトを作成する

<p>PID 命令を挿入する前に、プロジェクトにテクノロジーオブジェクトを作成することもできます。PID 命令をユーザープログラムに挿入する前にテクノロジーオブジェクトを作成することで、PID 命令の挿入時にテクノロジーオブジェクトを選択できるようになります。</p>	 <p>The 'Call options' dialog box is shown. It has a 'Data block' section with a 'Name' field containing 'PID_Compact_1' and a 'Number' dropdown menu with 'PID_Compact_TO' selected. Below these are radio buttons for 'Manual' and 'Automatic', with 'Automatic' selected. A note states: 'The called function block saves its data in its own instance data block.' There is a 'More...' link and 'OK' and 'Cancel' buttons at the bottom.</p>
<p>テクノロジーオブジェクトを作成するには、プロジェクトナビゲータで[新しいオブジェクトの追加]アイコンをダブルクリックします。</p>	 <p>The 'Project Navigator' tree is shown. It includes a tree view with 'PLC_1 [CPU 1214C DQ/DQ/Rly]' expanded, showing 'Device configuration', 'Online & diagnostics', 'Program blocks', and 'Technological objects'. Under 'Technological objects', the 'Add new object' icon (a blue square with a white plus sign) is highlighted.</p>
<p>[PID]アイコンをクリックして、PID コントローラのタイプ(PID_Compact または PID_3Step)に合ったテクノロジーオブジェクトを選択します。テクノロジーオブジェクトにオプションの名前を作成することができます。</p> <p>[OK]をクリックしてテクノロジーオブジェクトを作成します。</p>	 <p>The 'Add new object' dialog box is shown. It has a 'Name' field with 'PID_Compact_1'. On the left is a sidebar with icons for 'Motion Control', 'PID', and 'Counting and measurement'. The main area shows a tree of objects under 'PID Control', with 'Compact PID' expanded, listing 'PID_Compact' (V2.2), 'PID_3Step' (V2.2), and 'PID_Temp' (V1.0). The 'PID_Compact' object is selected. On the right, the 'Type' is 'PID Compact (FB 1130)', and the 'Number' is '1'. Below this is a description of the object. At the bottom, there is an 'Additional information' section with a checked 'Add new and open' option, and 'OK' and 'Cancel' buttons.</p>

8.2 PID_Compact 命令

PID_Compact 命令によって、汎用 PID コントローラに、自動モードと手動モードのセルフチューニングが組み込まれます。

表 8-3 PID_Compact 命令

LAD / FBD	SCL	説明
	<pre> "PID_Compact_1"(Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, Output=>_real_out_, Output_PER=>_word_out_, Output_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarning_H=>_bool_out_, , InputWarning_L=>_bool_out_, , State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_); </pre>	<p>PID_Compact は、自動モードと手動モードのセルフチューニングが可能な PID コントローラです。PID_Compact は、アンチwindアップと P 成分と D 成分の重みづけを備えた PID T1 コントローラです。</p>

¹ 命令を挿入すると、STEP 7 は自動的にテクノロジーオブジェクトとインスタンス DB を作成します。インスタンス DB には、テクノロジーオブジェクトのパラメータが含まれています。

² SCL の例では、「PID_Compact_1」がインスタンス DB の名前です。

表 8-4 パラメータのデータタイプ

パラメータとタイプ		データタイプ	説明
Setpoint	IN	REAL	自動モードでの PID コントローラのセットポイント (デフォルト値: 0.0)
Input	IN	REAL	プロセス値(ユーザープログラムのタグ) (デフォルト値: 0.0) 入力パラメータを使用している場合は、Config.InputPerOn = FALSE を設定する必要があります。
Input_PER	IN	WORD	プロセス値(アナログ入力) (デフォルト値: W#16#0) Input_PER パラメータを使用している場合は、Config.InputPerOn = TRUE を設定する必要があります。
Distrubance	IN	REAL	外乱変数または事前制御値
ManualEnable	IN	Bool	手動操作モードを有効または無効 (デフォルト値: FALSE): <ul style="list-style-type: none"> FALSE から TRUE エッジは「手動モード」を有効にし、State = 4 ではモードは変更されません。 ManualEnable = TRUE の間は、ModeActivate で立ち上がりエッジを使用したり、コミッシュニングダイアログを使用して動作モードを変更することはできません。 TRUE から FALSE エッジは、モードごとに割り当てられている動作モードを有効にします。 注記: 動作モードは ModeActivate のみを使用して変更することをお奨めします。
ManualValue	IN	REAL	手動操作のための出力値 (デフォルト値: 0.0) Config.OutputLowerLimit~Config.OutputUpperLimit の値を使用できます。
ErrorAck	IN	Bool	ErrorBits と警告出力リセット FALSE から TRUE エッジ
Reset	IN	Bool	コントローラ再起動 (デフォルト値: FALSE): <ul style="list-style-type: none"> FALSE から TRUE エッジ <ul style="list-style-type: none"> 「非アクティブ」モードに切り替えます ErrorBits と警告出力をリセットします 積分動作をクリアします PID パラメータを維持します Reset = TRUE の間は、PID_Compact は「非アクティブ」モード (State = 0) のままです。 TRUE から FALSE エッジ <ul style="list-style-type: none"> PID_Compact は Mode パラメータに保存されている動作モードに切り替わります。
ModeActivate	IN	Bool	Mode パラメータに保存されている動作モードに切替 FALSE から TRUE エッジ
Mode	IN	Int	目的の PID モード; Mode Activate 入力の立ち上がりエッジで有効になります。
ScaledInput	OUT	REAL	スケーリングされたプロセス値 (デフォルト値: 0.0)
Output ¹	OUT	REAL	REAL フォーマットの出力値 (デフォルト値: 0.0)
Output_PER ¹	OUT	WORD	アナログ出力値 (デフォルト値: W#16#0)

パラメータとタイプ		データタイプ	説明
Output_PWM ¹	OUT	Bool	パルス幅変調の出力値 (デフォルト値: FALSE) オンおよびオフ時間が出力値を形成します。
SetpointLimit_H	OUT	Bool	セットポイントの上限値 (デフォルト値: FALSE) SetpointLimit_H = TRUE の場合、セットポイントの絶対上限値に達しています(セットポイント ≥ Config.SetpointUpperLimit)。 セットポイントは Config.SetpointUpperLimit に制限されます。
SetpointLimit_L	OUT	Bool	セットポイントの下限値 (デフォルト値: FALSE) SetpointLimit_L = TRUE の場合、セットポイントの絶対下限値に達しています(セットポイント ≤ Config.SetpointUpperLimit)。 セットポイントは Config.SetpointLowerLimit に制限されます。
InputWarning_H	OUT	Bool	InputWarning_H = TRUE の場合、プロセス値は警告上限値に達している、もしくは超過 (デフォルト値: FALSE)
InputWarning_L	OUT	Bool	InputWarning_L = TRUE の場合、プロセス値は警告下限値に達している、もしくは下回っている (デフォルト値: FALSE)
State	OUT	Int	PID コントローラの現在の動作モード (デフォルト値: 0) Mode 入力パラメータおよび ModeActivate の立ち上がりエッジを使用して、動作モードを変更することができます。 <ul style="list-style-type: none"> • State = 0: 非アクティブ • State = 1: プレチューニング • State = 2: 手動のファインチューニング • State = 3: 自動モード • State = 4: 手動モード • State = 5: エラーモニタリングによる代替出力値
Error	OUT	Bool	Error = TRUE の場合、このサイクルで最低でも 1 つのエラーメッセージが保留中 (デフォルト値: FALSE) 注記: V1.x PID の Error パラメータは、エラーコードが含まれている ErrorBits フィールドでした。今では、エラーが発生していることを示す Boolean フラグになっています。
ErrorBits	OUT	DWord	PID_Compact 命令の ErrorBits パラメータテーブル(201 ページ)で、保留中のエラーメッセージが定義されています (デフォルト値: DW#16#0000 (エラーなし))。ErrorBits は保持され、Reset または ErrorAck の立ち上がりエッジでリセットされます。 注記: V1.x では、ErrorBits パラメータは Error パラメータとして定義され、存在していませんでした。

¹ Output、Output_PER、および Output_PWM パラメータの出力を並行して使用することができます。

PID_Compact コントローラの動作

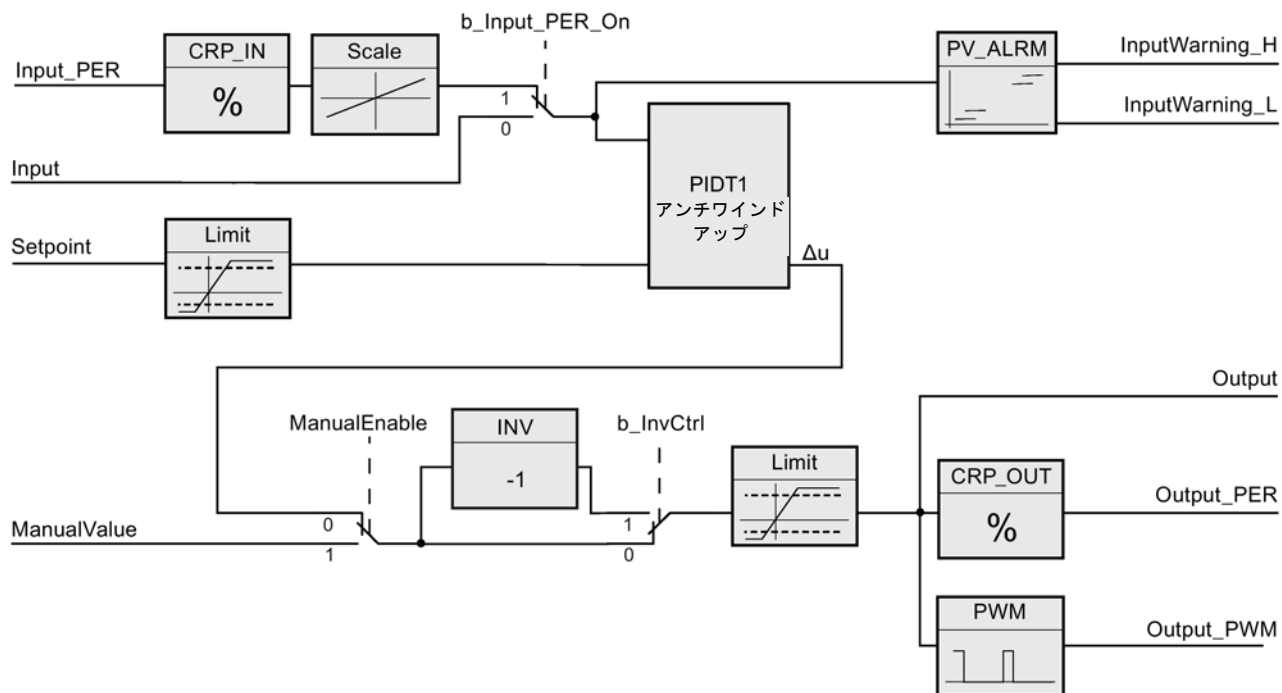


図 8-1 PID_Compact コントローラの動作

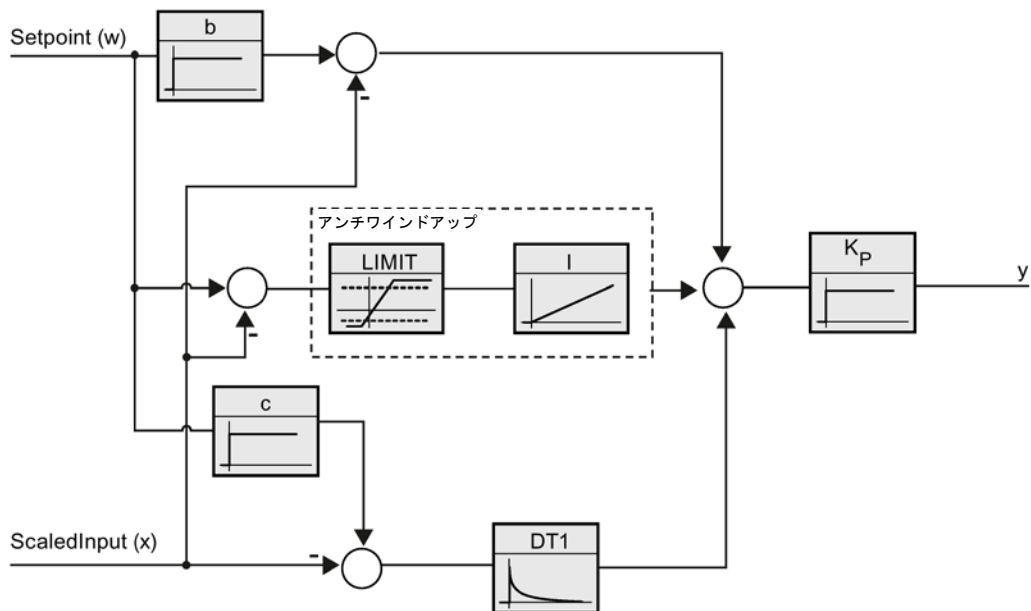


図 8-2 アンチwindアップ機能付き PID1 コントローラとしての PID_Compact コントローラの動作

8.3 PID_Compact 命令の ErrorBit パラメータ

複数のエラーが保留中の場合、エラーコードの値がバイナリ加算で表示されます。たとえば、エラーコード 0003 の表示は、エラー 0001 および 0002 も保留中であることを示します。

表 8- 5 PID_Compact 命令の ErrorBit パラメータ

ErrorBit (DW#16#...)	説明
0000	エラーは発生していません。
0001 ^{1, 2}	Input パラメータがプロセス値の限界を超えています。 Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Input_PER パラメータに無効な値があります。アナログ入力エラーが保留中かどうかをチェックします。
0004 ⁴	ファインチューニング中のエラー。プロセス値の振動を維持できませんでした。
0008 ⁴	プレチューニングの開始時のエラー。プロセス値がセットポイントに近すぎます。ファインチューニングを開始してください。
0010 ⁴	チューニング中にセットポイントが変更されました。 注記: CancelTuningLevel タグでセットポイントに許可された変動を設定できます。
0020	ファインチューニング中にプレチューニングは行えません。 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_Compact はファインチューニングモードのままとなります。
0080 ⁴	プレチューニング中のエラー。出力値の限界の不正な設定。 出力値の限界が正しく設定されていて、制御ロジックと一致しているかどうかをチェックします。
0100 ⁴	ファインチューニング中のエラーによりパラメータが無効になりました。
0200 ^{2, 3}	Input パラメータに無効な値があります: 値の数字フォーマットが無効です。
0400 ^{2, 3}	出力値の計算が失敗しました。PID パラメータをチェックします。
0800 ^{1, 2}	サンプリング時間エラー:PID_Compact が周期割り込み OB のサンプリング時間内に呼び出されません。
1000 ^{2, 3}	Setpoint パラメータに無効な値があります: 値の数字フォーマットが無効です。
10000	ManualValue パラメータに無効な値があります: 値の数字フォーマットが無効です。 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_Compact は SubstituteOutput を出力値として使用します。ManualValue パラメータに有効な値を割り当てるとすぐに、PID_Compact はそれを出力値として使用します。

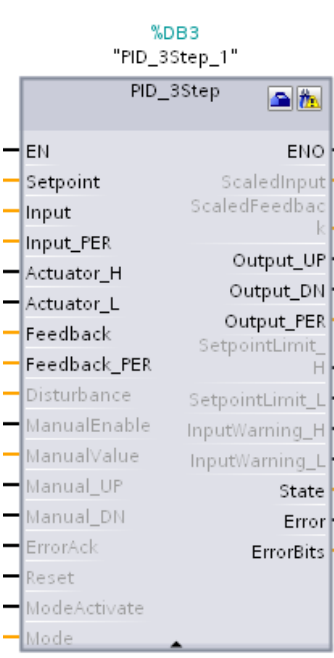
ErrorBit (DW#16#...)	説明
20000	<p>SubstituteValue タグに無効な値があります: 値の数字フォーマットが無効です。</p> <p>PID_Compact は出力の下限值を出力値として使用します。</p> <p>注記: エラーが発生する前に自動モードが有効だった場合、つまり ActivateRecoverMode = TRUE だった場合、エラーはそれ以上保留されず、PID_Compact は自動モードに戻ります。</p>
40000	<p>Disturbance パラメータに無効な値があります: 値の数字フォーマットが無効です。</p> <p>注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = FALSE だった場合、Disturbance は 0 に設定されます。PID_Compact は自動モードのままです。</p> <p>注記: エラーが発生する前にプレチューニングモードまたはファインチューニングモードが有効で、ActivateRecoverMode = TRUE だった場合、PID_Compact は Mode パラメータに保存されている動作モードに切り替わります。現在のフェーズの Disturbance が出力値に影響しない場合、チューニングはキャンセルされます。</p>

- ¹ 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = TRUE の場合、PID_Compact は自動モードのままとなります。
- ² 注記: エラーが発生する前にプレチューニングモードまたはファインチューニングモードが有効で、ActivateRecoverMode = TRUE の場合、PID_Compact は Mode パラメータに保存されている動作モードに切り替わります。
- ³ 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = TRUE の場合、PID_Compact は設定された代替出力値を出力します。エラーが保留中でなくなるとすぐに、PID_Compact は自動モードに戻ります。
- ⁴ 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_Compact はチューニングをキャンセルし、Mode パラメータに保存されている動作モードに切り替わります。

8.4 PID_3Step 命令

PID_3Step 命令は、モータ制御型バルブおよびアクチュエータに最適化されている、セルフチューニング機能付きの PID コントローラを設定します。

表 8-6 PID_3Step 命令

LAD / FBD	SCL	説明
	<pre> "PID_3Step_1"(SetpoInt:=_real_in_, Input:=_real_in_, ManualValue:=_real_in_, Feedback:=_real_in_, InputPer:=_word_in_, FeedbackPer:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualUP:=_bool_in_, ManualDN:=_bool_in_, ActuatorH:=_bool_in_, ActuatorL:=_bool_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, ScaledFeedback=>_real_out_, ErrorBits=>_dword_out_, OutputPer=>_word_out_, State=>_int_out_, OutputUP=>_bool_out_, OutputDN=>_bool_out_, SetpoIntLimitH=>_bool_out_, SetpoIntLimitL=>_bool_out_, InputWarningH=>_bool_out_, InputWarningL=>_bool_out_, Error=>_bool_out_, ErrorBits=>_dword_out_); </pre>	<p>PID_3Step は、モータ制御型バルブおよびアクチュエータに最適化されている、セルフチューニング機能付きの PID コントローラを設定します。2つの Boolean 出力を提供します。</p> <p>PID_3Step は、アンチwindアップと P 成分および D 成分の重み付けを備えた PID T1 コントローラです。</p>

- 命令を挿入すると、STEP 7 は自動的にテクノロジーオブジェクトとインスタンス DB を作成します。インスタンス DB には、テクノロジーオブジェクトのパラメータが含まれています。
- SCL の例では、「PID_3Step_1」がインスタンス DB の名前です。

表 8-7 パラメータのデータタイプ

パラメータとタイプ		データタイプ	説明
セットポイント	IN	REAL	自動モードでの PID コントローラのセットポイント (デフォルト値: 0.0)
Input	IN	REAL	プロセス値(ユーザープログラムのタグ) (デフォルト値: 0.0) 入力パラメータを使用している場合は、Config.InputPerOn = FALSE を設定する必要があります
Input_PER	IN	WORD	プロセス値(アナログ入力) (デフォルト値: W#16#0) Input_PER パラメータを使用している場合は、Config.InputPerOn = TRUE を設定する必要があります。
Actuator_H	IN	Bool	ハイエンドストップのバルブのデジタル位置フィードバック Actuator_H = TRUE の場合、バルブはハイエンドストップにあり、この方向にそれ以上移動しません。(デフォルト値: FALSE)
Actuator_L	IN	Bool	ローエンドストップのバルブのデジタル位置フィードバック Actuator_L = TRUE の場合、バルブはローエンドストップにあり、この方向にそれ以上移動しません。(デフォルト値: FALSE)
Feedback	IN	REAL	バルブの位置フィードバック (デフォルト値: 0.0) Feedback パラメータを使用している場合は、Config.FeedbackPerOn = FALSE を設定する必要があります。
Feedback_PER	IN	Int	バルブ位置のアナログフィードバック (デフォルト値: W#16#0) Feedback_PER パラメータを使用している場合は、Config.FeedbackPerOn = TRUE を設定する必要があります。 Feedback_PER は以下のタグに基づいてスケーリングされます。 <ul style="list-style-type: none"> Config.FeedbackScaling.LowerPointIn Config.FeedbackScaling.UpperPointIn Config.FeedbackScaling.LowerPointOut Config.FeedbackScaling.UpperPointOut
Distrubance	IN	REAL	外乱変数または事前制御値
ManualEnable	IN	Bool	手動操作モードの有効または無効 (デフォルト値: FALSE): <ul style="list-style-type: none"> FALSE から TRUE エッジは「手動モード」を有効にし、State = 4 ではモードは変更されません。 ManualEnable = TRUE の間は、ModeActivate で立ち上がりエッジを使用したり、コミショニングダイアログを使用して動作モードを変更することはできません。 TRUE から FALSE エッジは、モードごとに割り当てられている動作モードを有効にします。 注記: 動作モードは ModeActivate のみを使用して変更することをお奨めします。
ManualValue	IN	REAL	手動操作のためのプロセス値 (デフォルト値: 0.0) 手動モードで、バルブの絶対位置を指定します。ManualValue は、OutputPer を使用しているか、または位置フィードバックが使用可能な場合にのみ評価されます。

パラメータとタイプ		データタイプ	説明
ManualUP	IN	Bool	<ul style="list-style-type: none"> Manual_UP = TRUE: <ul style="list-style-type: none"> Output_PER または位置フィードバックを使用している場合でも、バルブは開かれます。バルブはエンドストップに達すると、それ以上移動しません。 Config.VirtualActuatorLimit も参照してください。 Manual_UP = FALSE: <ul style="list-style-type: none"> Output_PER または位置フィードバックを使用している場合、バルブは ManualValue に移動します。使用していない場合、バルブはそれ以上移動しません。 <p>注記: Manual_UP および Manual_DN が同時に TRUE に設定されている場合、バルブは移動しません。</p>
ManualDN	IN	Bool	<ul style="list-style-type: none"> Manual_DN = TRUE: <ul style="list-style-type: none"> Output_PER または位置フィードバックを使用している場合でも、バルブは開かれます。バルブはエンドストップに達すると、それ以上移動しません。 Config.VirtualActuatorLimit も参照してください。 Manual_DN = FALSE: <ul style="list-style-type: none"> Output_PER または位置フィードバックを使用している場合、バルブは ManualValue に移動します。使用していない場合、バルブはそれ以上移動しません。
ErrorAck	IN	Bool	ErrorBits と警告出力のリセット FALSE から TRUE エッジ
Reset	IN	Bool	<p>コントローラの再起動 (デフォルト値: FALSE):</p> <ul style="list-style-type: none"> FALSE から TRUE エッジ <ul style="list-style-type: none"> 「非アクティブ」モードに切り替えます ErrorBits と警告出力をリセットします 積分動作をクリアします PID パラメータを維持します Reset = TRUE の間は、PID_3Step は「非アクティブ」モード (State = 0)のままです。 TRUE から FALSE エッジ <ul style="list-style-type: none"> PID_3Step は Mode パラメータに保存されている動作モードに切り替わります。
ModeActivate	IN	Bool	Mode パラメータに保存されている動作モードに切替 FALSE から TRUE エッジ
Mode	IN	Int	目的の PID モード; Mode Activate 入力の立ち上がりエッジで有効になります。
ScaledInput	OUT	REAL	スケーリングされたプロセス値
ScaledFeedback	OUT	REAL	<p>スケーリングされたバルブの位置フィードバック</p> <p>注記: 位置フィードバックなしのアクチュエータの場合、ScaledFeedback で示されたアクチュエータの位置は極めて不正確です。この場合、ScaledFeedback は現在の位置の概算にしか使用できません。</p>
Output_UP	OUT	Bool	<p>バルブを開くためのデジタル出力値 (デフォルト値: FALSE)</p> <p>Config.OutputPerOn = FALSE の場合、パラメータ Output_UP が使用されます。</p>

パラメータとタイプ		データタイプ	説明
Output_DN	OUT	Bool	バルブを閉じるためのデジタル出力値 (デフォルト値: FALSE) Config.OutputPerOn = FALSE の場合、パラメータ Output_DN が使用されます。
Output_PER	OUT	WORD	アナログ出力値 Config.OutputPerOn = TRUE の場合、パラメータ Output_PER が使用されます。
SetpointLimitH	OUT	Bool	セットポイントの上限値 (デフォルト値: FALSE) SetpointLimitH = TRUE の場合、セットポイントの絶対上限値に達しています (セットポイント \geq Config.SetpointUpperLimit)。 注記: セットポイントは (セットポイント \geq Config.SetpointUpperLimit) に制限されます。
SetpointLimitL	OUT	Bool	セットポイントの下限値 (デフォルト値: FALSE) SetpointLimitL = TRUE の場合、セットポイントの絶対下限値に達しています (セットポイント \geq Config.SetpointLowerLimit)。 注記: セットポイントは (セットポイント \geq Config.SetpointLowerLimit) に制限されます。
InputWarningH	OUT	Bool	InputWarningH = TRUE の場合、入力値は警告上限値に達しているか超過 (デフォルト値: FALSE)
InputWarningL	OUT	Bool	InputWarningL = TRUE の場合、入力値は警告下限値に達しているか下回っている (デフォルト値: FALSE)
State	OUT	Int	PID コントローラの現在の動作モード (デフォルト値: 0) Mode 入力パラメータおよび ModeActivate の立ち上がりエッジを使用して、動作モードを変更することができます。 <ul style="list-style-type: none"> • State = 0: 非アクティブ • State = 1: プレチューニング • State = 2: 手動のファインチューニング • State = 3: 自動モード • State = 4: 手動モード • State = 5: 代替出力値アプローチ • State = 6: 移行時間の測定 • State = 7: エラーモニタリング • State = 8: エラーモニタリングによる代替出力値アプローチ • State = 10: エンドストップ信号なしの手動モード
Error	OUT	Bool	Error = TRUE の場合、最低でも 1 つのエラーメッセージが保留中 (デフォルト値: FALSE) 注記: V1.x PID の Error パラメータは、エラーコードが含まれている ErrorBits フィールドでした。今では、エラーが発生していることを示す Boolean フラグになっています。
ErrorBits	OUT	DWord	PID_3Step 命令の ErrorBits パラメータテーブル(210 ページ)で、保留中のエラーメッセージが定義されています。(デフォルト値: DW#16#0000 (エラーなし))。ErrorBits は保持され、Reset または ErrorAck の立ち上がりエッジでリセットされます。 注記: V1.x では、ErrorBits パラメータは Error パラメータとして定義され、存在していませんでした。

PID_3Step コントローラの動作

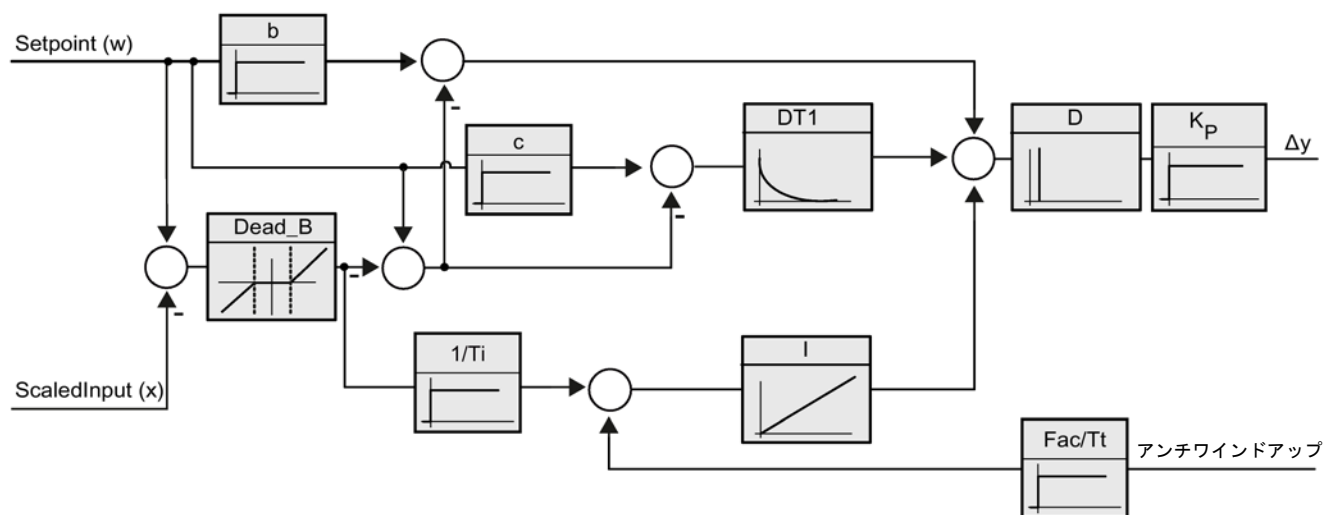


図 8-3 アンチwindアップ機能付き PID T1 コントローラとしての PID_3Step コントローラの動作

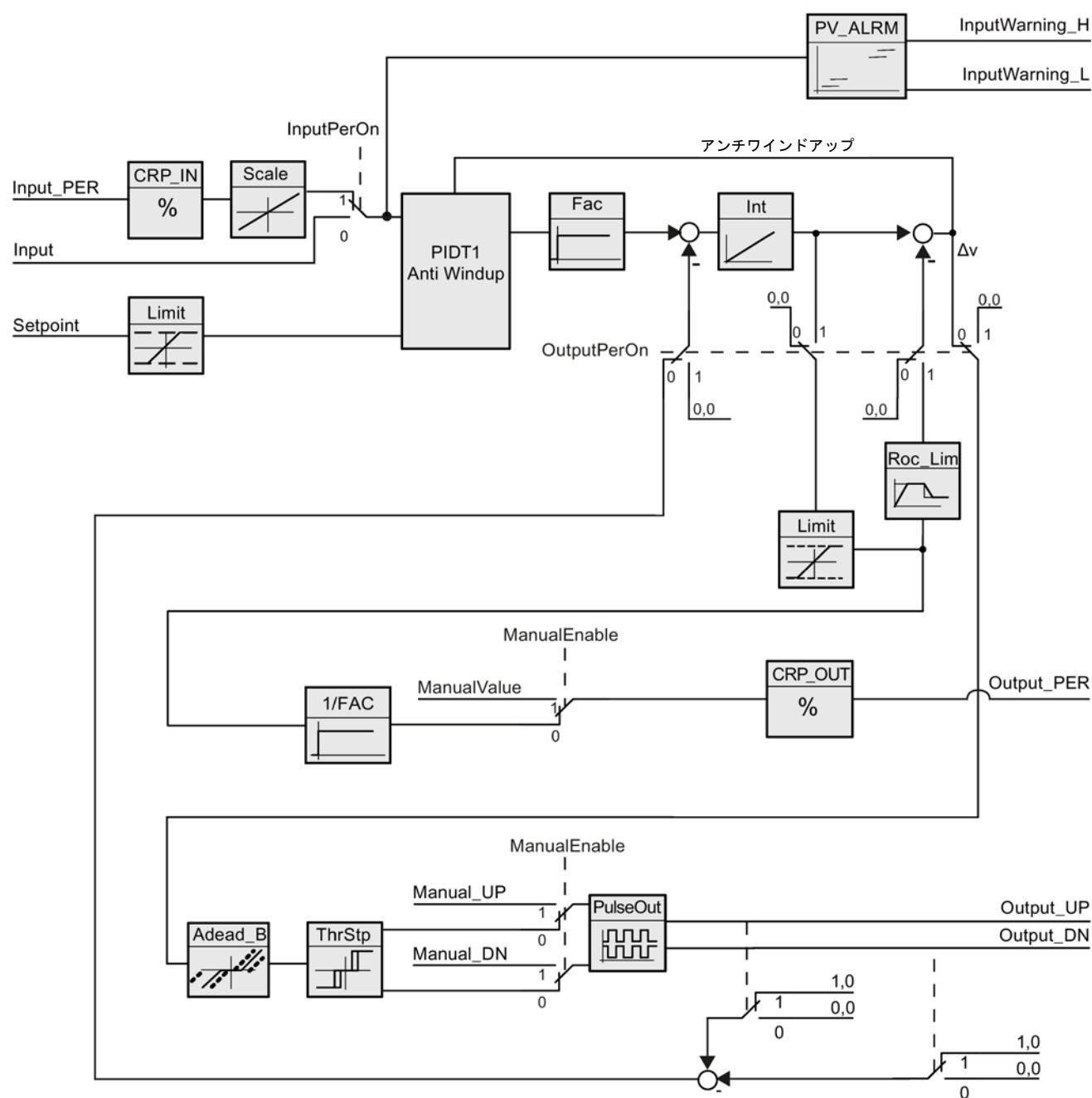


図 8-4 位置フィードバックなしの PID_3Step コントローラの動作

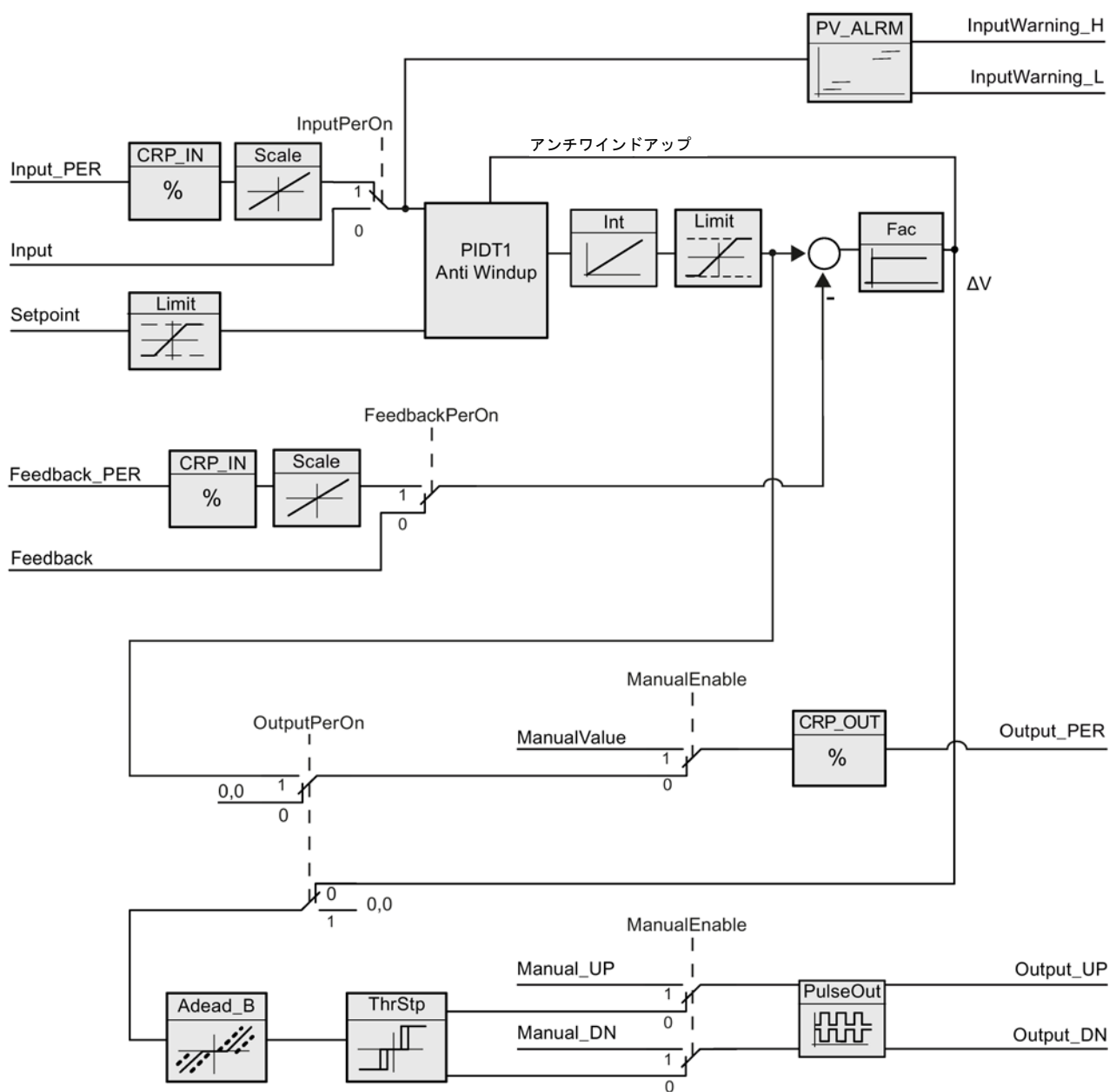


図 8-5 位置フィードバックが有効の PID_3Step コントローラの動作

8.5 PID_3Step 命令の ErrorBit パラメータ

複数のエラーが保留中の場合、エラーコードの値がバイナリ加算で表示されます。たとえば、エラーコード 0003 の表示は、エラー 0001 および 0002 も保留中であることを示します。

表 8-8 PID_3STEP 命令の ErrorBit パラメータ

ErrorBit (DW#16#...)	説明
0000	エラーは発生していません。
0001 ^{1, 2}	Input パラメータがプロセス値の限界を超えています。 Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Input_PER パラメータに無効な値があります。アナログ入力エラーが保留中かどうかをチェックします。
0004 ⁴	ファインチューニング中のエラー。プロセス値の振動を維持できませんでした。
0010 ⁴	チューニング中にセットポイントが変更されました。 注記: CancelTuningLevel タグでセットポイントに許可された変動を設定できます。
0020	ファインチューニング中にプレチューニングは行えません。 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_3Step はファインチューニングモードのままとなります。
0080 ⁴	プレチューニング中のエラー。出力値の限界の不正な設定。 出力値の限界が正しく設定されていて、制御ロジックと一致しているかどうかをチェックします。
0100 ⁴	ファインチューニング中のエラーによりパラメータが無効になりました。
0200 ^{2, 3}	Input パラメータに無効な値があります: 値の数字フォーマットが無効です。
0400 ^{2, 3}	出力値の計算が失敗しました。PID パラメータをチェックします。
0800 ^{1, 2}	サンプリング時間エラー:PID_3Step が周期割り込み OB のサンプリング時間内に呼び出されません。
1000 ^{2, 3}	Setpoint パラメータに無効な値があります: 値の数字フォーマットが無効です。
2000 ^{1, 2, 5}	Feedback_PER パラメータに無効な値があります。 アナログ入力エラーが保留中かどうかをチェックします。
4000 ^{1, 2, 5}	Feedback パラメータに無効な値があります。値の数字フォーマットが無効です。
8000 ^{1, 2}	デジタル位置フィードバック時のエラー。Actuator_H = TRUE および Actuator_L = TRUE。 アクチュエータは代替出力値に移動できず、現在の位置にとどまります。この状態では手動モードは使用できません。 アクチュエータをこの状態から移動するには、[アクチュエータのエンドストップ](Config.ActuatorEndStopOn = FALSE)を無効にするか、またはエンドストップ信号なしの手動モード(モード = 10)に切り替える必要があります。

ErrorBit (DW#16#...)	説明
10000	ManualValue パラメータに無効な値があります: 値の数字フォーマットが無効です。 アクチュエータは手動値に移動できず、現在の位置にとどまります。 ManualValue に有効な値を割り当てるか、アクチュエータを Manual_UP および Manual_DN で手動モードに移します。
20000	SavePosition タグに無効な値があります: 値の数字フォーマットが無効です。 アクチュエータは代替出力値に移動できず、現在の位置にとどまります。
40000	Disturbance パラメータに無効な値があります: 値の数字フォーマットが無効です。 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = FALSE だった場合、Disturbance は 0 に設定されます。PID_3Step は自動モードのままです。 注記: エラーが発生する前にプレチューニングモードまたはファインチューニングモードが有効で、ActivateRecoverMode = TRUE だった場合、PID_3Step は Mode パラメータに保存されている動作モードに切り替わります。現在のフェーズの Disturbance が出力値に影響しない場合、チューニングはキャンセルされます。 移行時間測定中は、エラーは影響を及ぼしません。

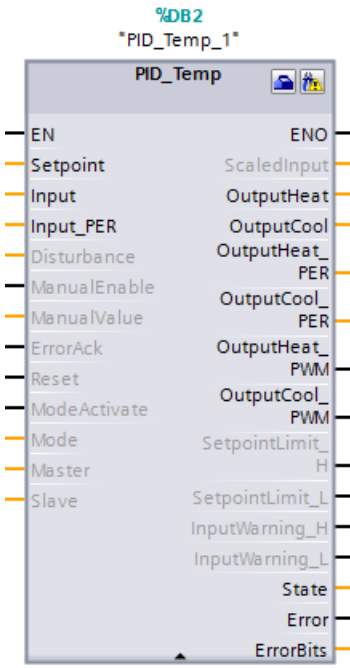
- ¹ 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = TRUE の場合、PID_3Step は自動モードのままとなります。
- ² 注記: エラーが発生する前にプレチューニングモード、ファインチューニングモード、または移行時間測定モードが有効で、ActivateRecoverMode = TRUE だった場合、PID_3Step は Mode パラメータに保存されている動作モードに切り替わります。
- ³ 注記: エラーが発生する前に自動モードがアクティブになっていて、ActivateRecoverMode = TRUE の場合、PID_3Step は[エラーモニタリングで代替出力値にアプローチ]モードまたは[エラーモニタリング]モードに切り替わります。エラーが保留中でなくなるとすぐに、PID_3Step は自動モードに戻ります。
- ⁴ 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_3Step はチューニングをキャンセルし、Mode パラメータに保存されている動作モードに切り替わります。
- ⁵ アクチュエータは代替出力値に移動できず、現在の位置にとどまります。手動モードでは、アクチュエータの位置は ManualValue ではなく、Manual_UP および Manual_DN でのみ変更できます。

8.6 PID_Temp 命令

8.6.1 概要

PID_Temp 命令は、温度制御の特定の要件の処理を可能にする汎用 PID コントローラを提供します。

表 8- 9 PID_Temp 命令

LAD / FBD	SCL	説明
	<pre> "PID_Temp_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_int_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, Master:=_dword_in Save:=_dword_in ScaledInput=>_real_out_, OutputHeat=>_real_out_, OutputCool=>_real_out_, OutputHeat_PER=>_int_out_, OutputCool_PER=>_int_out_, OutputHeat_PWM=>_bool_out_, OutputCool_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarning_H=>_bool_out_, InputWarning_L=>_bool_out_, State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_); </pre>	<p>PID_Temp には以下の機能があります。</p> <ul style="list-style-type: none"> 異なったアクチュエータによるプロセスの加熱と冷却 温度プロセスを処理するための内蔵オートチューニング 同じアクチュエータに依存する複数の温度を処理するためのカスケード

¹ 命令を挿入すると、STEP 7 は自動的にテクノロジーオブジェクトとインスタンス DB を作成します。インスタンス DB には、テクノロジーオブジェクトのパラメータが含まれています。

² SCL の例では、「PID_Temp_1」がインスタンス DB の名前です。

表 8- 10 パラメータのデータタイプ

パラメータとタイプ		データタイプ	説明
Setpoint	IN	REAL	自動モードでの PID コントローラのセットポイント。(デフォルト値: 0.0)
Input	IN	REAL	プロセス値(ユーザープログラムのタグ) (デフォルト値: 0.0) 入力パラメータを使用している場合は、Config.InputPerOn = FALSE を設定する必要があります。
Input_PER	IN	Int	プロセス値(アナログ入力) (デフォルト値: 0) Input_PER パラメータを使用している場合は、Config.InputPerOn = TRUE を設定する必要があります。
Distrubance	IN	REAL	外乱変数または事前制御値
ManualEnable	IN	Bool	手動操作モードを有効または無効 (デフォルト値: FALSE): <ul style="list-style-type: none"> FALSE から TRUE エッジは「手動モード」を有効にし、State = 4 ではモードは変更されません。 ManualEnable = TRUE の間は、ModeActivate で立ち上がりエッジを使用したり、コミッシュニングダイアログを使用して動作モードを変更することはできません。 <ul style="list-style-type: none"> TRUE から FALSE エッジは、モードごとに割り当てられている動作モードを有効にします。 注記: 動作モードは ModeActivate のみを使用して変更することをお奨めします。
ManualValue	IN	REAL	手動操作のための出力値 (デフォルト値: 0.0) Config.OutputLowerLimit~Config.OutputUpperLimit の値を使用できます。
ErrorAck	IN	Bool	ErrorBits と警告出力を FALSE から TRUE エッジでリセット (デフォルト値: FALSE)
Reset	IN	Bool	コントローラを再起動 (デフォルト値: FALSE): <ul style="list-style-type: none"> FALSE から TRUE エッジ <ul style="list-style-type: none"> 「非アクティブ」モードに切り替えます ErrorBits と警告出力をリセットします 積分動作をクリアします PID パラメータを維持します Reset = TRUE の間は、PID_Temp は非アクティブモード(State = 0)のままです。 TRUE から FALSE エッジ <ul style="list-style-type: none"> PID_Temp は Mode パラメータに保存されている動作モードに切り替わります。
ModeActivate	IN	Bool	Mode パラメータに保存されている動作モードに切替 FALSE から TRUE エッジ (デフォルト値: FALSE)

パラメータとタイプ		データタイプ	説明
Mode	IN/OUT	Int	Mode Activate 入力の立ち上がりエッジで有効になります。 動作モードの選択(デフォルト値: 0.0): <ul style="list-style-type: none"> モード = 0: 非アクティブ モード = 1: プレチューニング モード = 2: ファインチューニング モード = 3: 自動モード モード = 4: 手動モード 「エラーモニタリングによる代替出力値」(State = 5)これはユーザーが有効にすることはできません。単なる自動エラー応答です。
Master	IN/OUT	DWord	マスタへのカスケード接続(AntiWindUp およびチューニング条件) (デフォルト値: DW#16#0000) <ul style="list-style-type: none"> ビット 0~15: PID_Temp 命令では未使用 ビット 16~-23: 限界値カウンタ: スレーブは限界値に達すると、この値を大きくします。限界の数のスレーブが、アンチwindアップ機能のための処理されます(Config.Cascade.AntiWindUpMode パラメータを参照)。 ビット 24: IsAutomatic: 当該コントローラのすべてのスレーブが自動モードで、カスケードでのチューニングのための条件をチェックするために処理される場合、このビットは「1」に設定されます。このビットは AllSlaveAutomaticState パラメータと同じです。 ビット 25: [IsReplacement-Setpoint]: 当該コントローラのスレーブが[交換セットポイント]が有効になっていて、カスケードでのチューニングのための条件をチェックするために処理される場合、このビットは「1」に設定されます。反転された値は NoSlaveReplacementSetpoint パラメータに保存されます。
Slave	IN/OUT	DWord	
ScaledInput	OUT	REAL	スケーリングされたプロセス値 (デフォルト値: 0.0)
OutputHeat ¹	OUT	REAL	REAL フォーマットの加熱用の出力値 (デフォルト値: 0.0) この出力値は、出力の選択とは無関係に、Config.Output.Heat.Select パラメータを使用して計算されます。
OutputCool ¹	OUT	REAL	REAL フォーマットの冷却用の出力値 (デフォルト値: 0.0) この出力値は、出力の選択とは無関係に、Config.Output.Cool.Select パラメータを使用して計算されます。
OutputHeat_PER ¹	OUT	Int	ダイレクトフォーマットの加熱用の出力値 (デフォルト値:0) この出力値は、Config.Output.Heat.Select = 2 パラメータを使用して選択された場合にのみ計算されます。選択されない場合、この出力は常に「0」です。
OutputCool_PER ¹	OUT	Int	ダイレクトフォーマットの冷却用の出力値(デフォルト値:0) この出力値は、Config.Output.Cool.Select = 2 パラメータを使用して選択された場合にのみ計算されます。選択されない場合、この出力は常に「0」です。
OutputHeat_PWM ¹	OUT	Bool	加熱用のパルス幅変調出力値 (デフォルト値: FALSE) この出力値は、Config.Output.Heat.Select = 1 (デフォルト値)パラメータを使用して選択された場合にのみ計算されます。選択されない場合、この出力は常に「FALSE」です。

パラメータとタイプ		データタイプ	説明
OutputCool_PWM ¹	OUT	Bool	冷却用のパルス幅変調出力値 (デフォルト値: FALSE) この出力値は、Config.Output.Cool.Select = 1 (デフォルト値)パラメータを使用して選択された場合にのみ計算されます。選択されない場合、この出力は常に「FALSE」です。
SetpointLimit_H	OUT	Bool	セットポイントの上限値 (デフォルト値: FALSE) SetpointLimit_H = TRUE の場合、セットポイントの絶対上限値に達しています(セットポイント ≥ Config.SetpointUpperLimit)。 セットポイントは Config.SetpointUpperLimit に制限されます。
SetpointLimit_L	OUT	Bool	セットポイントの下限値 (デフォルト値: FALSE) SetpointLimit_L = TRUE の場合、セットポイントの絶対下限値に達しています(セットポイント ≤ Config.SetpointUpperLimit)。 セットポイントは Config.SetpointLowerLimit に制限されます。
InputWarning_H	OUT	Bool	InputWarning_H = TRUE の場合、プロセス値は警告上限値に達しているか超過(デフォルト値: FALSE)
InputWarning_L	OUT	Bool	InputWarning_L = TRUE の場合、プロセス値は警告下限値に達しているか下回っている (デフォルト値: FALSE)
State	OUT	Int	PID コントローラの現在の動作モード (デフォルト値: 0) Mode 入力パラメータおよび ModeActivate の立ち上がりエッジを使用して、動作モードを変更することができます。 <ul style="list-style-type: none"> • State = 0: 非アクティブ • State = 1: プレチューニング • State = 2: ファインチューニング • State = 3: 自動モード • State = 4: 手動モード • State = 5: エラーモニタリングによる代替出力値
Error	OUT	Bool	Error = TRUE の場合、このサイクルで最低でも 1 つのエラーメッセージが保留中 (デフォルト値: FALSE) 注記: V1.x PID の Error パラメータは、エラーコードが含まれている ErrorBits フィールドでした。今では、エラーが発生していることを示す Boolean フラグになっています。
ErrorBits	OUT	DWord	PID_Temp 命令の ErrorBits パラメータテーブル(222 ページ)で、保留中のエラーメッセージが定義されています。(デフォルト値: DW#16#0000 (エラーなし))。ErrorBits は保持され、Reset または ErrorAck の立ち上がりエッジでリセットされます。 注記: V1.x では、ErrorBits パラメータは Error パラメータとして定義され、存在していませんでした。
Warning	OUT	DWord	PID_Temp 命令の Warning パラメータテーブルで、保留中のユーザー関連の警告メッセージが定義されています。(デフォルト値: DW#16#0000 (警告なし))。
WarningInternal	OUT	DWord	PID_Temp 命令の WarningInternal パラメータテーブルで、保留中の警告内部メッセージが定義されています(すべての警告を含む)。(デフォルト値: DW#16#0000 (内部警告なし))。

¹ Output、Output_PER、および Output_PWM パラメータの出力を並行して使用することができます。

8.6.2 PID_Tempコントローラの動作

加熱および/または冷却制御を選択する

まず、パラメータ[ActivateCooling]で、加熱出力に加えて冷却デバイスも必要かどうかを選択する必要があります。その後で、2つのPIDパラメータセットを使用するか(詳細モード)、またはパラメータ[AdvancedCooling]で加熱/冷却ファクタを追加して1つのPIDパラメータセットだけを使用するかを定義します。

CoolFactor を使用する

加熱/冷却ファクタを適用する場合、手動で値を定義する必要があります。使用するアプリケーションの技術データから値を特定し(アクチュエータの比例ゲインの比率(たとえば、アクチュエータの最大加熱および冷却パワーの比率))、パラメータ[CoolFactor]に割り当てます。加熱/冷却ファクタ2.0は、加熱デバイスが冷却デバイスの2倍の効果があることを意味します。冷却ファクタを使用する場合、PID_Tempは出力信号を計算し、その符号に応じて出力信号と加熱/冷却ファクタを乗算(符号が負の場合)、または乗算しません(符号が正の場合)。

2つのPIDパラメータセットを使用する

加熱と冷却用の別々のPIDパラメータセットをコミッショニング中に自動的に検出できます。加熱/冷却ファクタに比べて高い制御パフォーマンスを期待できます。別々の比例ゲインが得られるだけでなく、2つのパラメータセットで別々の遅延時間を考慮することができます。ただし、チューニングプロセスよりも時間がかかるという欠点があります。PIDパラメータの切り替えが有効の場合(Config.AdvancedCooling = TRUE)、PID_Tempコントローラは「自動モード」(制御が有効)で、該当時に過熱または冷却のどちらが必要かを検出し、PIDパラメータセットを使用して制御します。

ControlZone

PID_Tempコントローラでは、パラメータ[ControlZone]でパラメータセットごとに制御ゾーンを定義できます。制御偏差(セットポイント - 入力)が制御ゾーン内の場合、PID_TempはPIDアルゴリズムを使用して出力信号を計算します。しかしながら、制御偏差が定義された範囲から離れている場合、出力は最大加熱出力値または最大冷却出力値(冷却出力が有効)/最小加熱出力値(冷却出力が無効)に設定されます。この機能を使用して、特に低速の温度プロセスの初期の加熱時に、目的のセットポイントに到達する時間を速めることができます。

DeadZone

[DeadZone]パラメータで、PIDアルゴリズムが無視する加熱および冷却の制御偏差の幅を定義することができます。つまり、この範囲内の制御偏差は抑制され、PID_Tempコントローラはセットポイントとプロセス値が同一の場合と同様に動作します。したがって、セットポイント周辺のコントローラによる不要な介入を減らし、アクチュエータを節約することができます。DeadZoneを適用する場合は、値を手動で定義する必要があります。オートチューニングでは、DeadZone値は自動的に設定されません。DeadZoneは、冷却を使用しない加熱コントローラ、またはCoolFctorを使用する加熱/冷却コントローラに対して対称的です(-Retain.CtrlParams.Heat.DeadZone と +Retain.CtrlParams.Heat.DeadZone の間)。DeadZoneは、2つのPIDパラメータセットを使用する加熱/冷却コントローラに対して非対称とすることができます(-Retain.CtrlParams.Cool.DeadZone と +Retain.CtrlParams.Heat.DeadZone の間)。

PID_Temp コントローラの動作

以下のブロックダイアグラムに PID_Temp 命令の標準的なカスケード動作を示します。

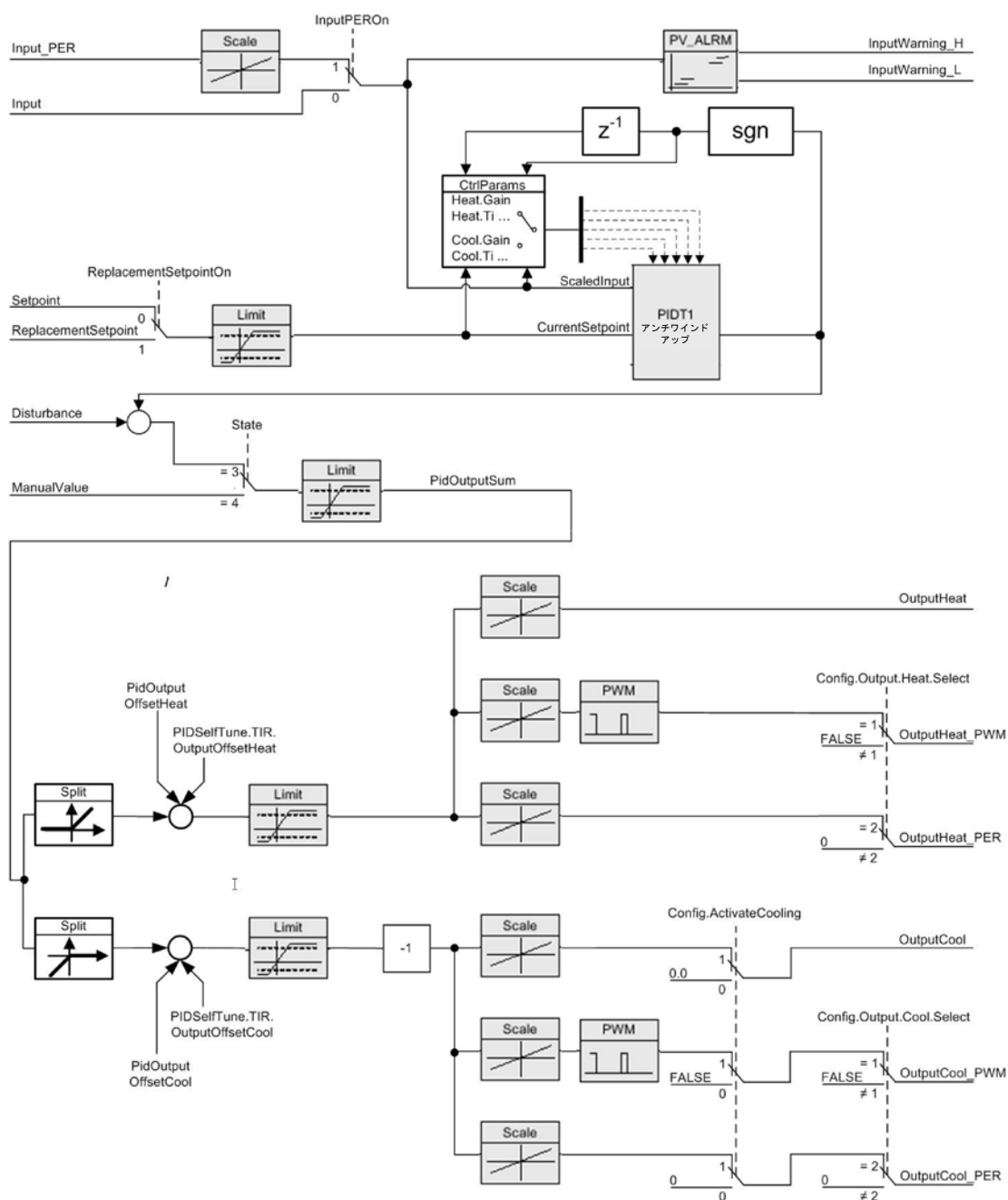


図 8-6 PID_Temp_Operation_Block_Diagram

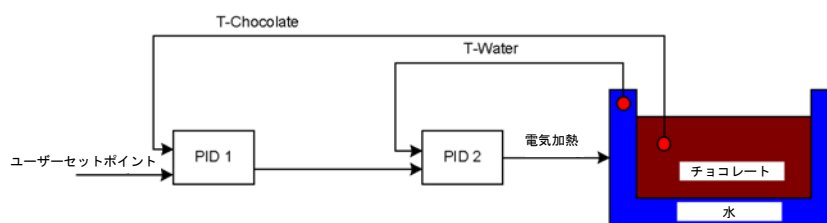


図 8-7 PID_Temp_Cascade_Operation_Block_Diagram

8.6.3 カスケードコントローラ

同じアクチュエータに依存する複数の温度を処理するために、温度 PID コントローラをカスケードすることができます。

呼び出し順序

カスケードされた PID コントローラは同一の OB サイクルで呼び出す必要があります。最初にマスタを呼び出し、次に制御信号フロー内の次のスレーブを呼び出し、最終的にはカスケードの最後のスレーブを呼び出します。PID_Temp 命令は呼び出し順序を自動的にチェックしません。

通信接続

コントローラをカスケードする場合は、マスタとスレーブが互いに情報を共有できるように接続する必要があります。スレーブの「マスタ」IN/OUT パラメータをマスタの「スレーブ」IN/OUT に、信号フロー方向に接続します。

以下に、2つのサブカスケードを持つカスケード内での PID_Temp コントローラの接続を示します。「PID_Temp1」はセットポイントを提供します。この構成では、「PID_Temp2」、「PID_Temp3」、「PID_Temp5」、「PID_Temp6」、「PID_Temp8」の出力がプロセスに接続されています。

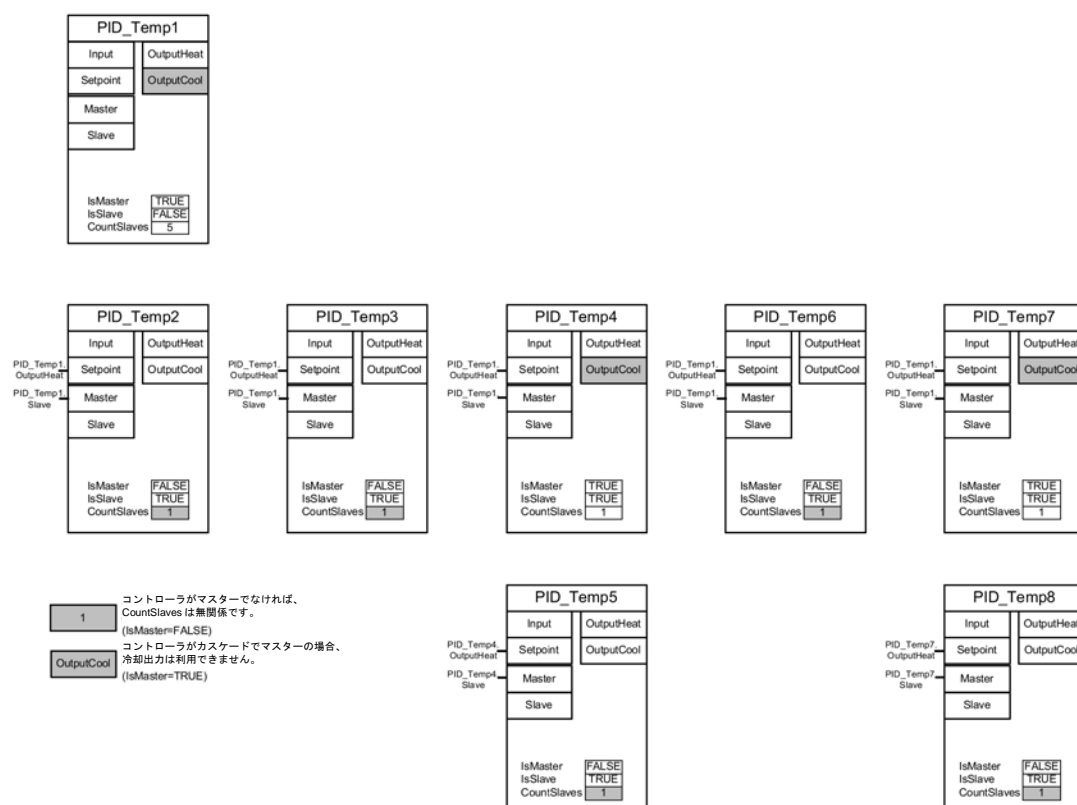


図 8-8 PID_Temp_Cascading_communication_connection

交換セットポイント

PID_Temp 命令は、[ReplacementSetpoint]パラメータで2番目のセットポイント入力を提供し、ユーザーはパラメータ[ReplacementSetpointOn] = TRUEを設定してそれを有効にできます。コミッシング中またはスレーブコントローラのチューニング中に、マスタとスレーブ間の出力-セットポイント接続を切断せずに、[ReplacementSetpoint]をセットポイント入力として使用できます。この接続はカスケードの通常操作に必要です。

このように、一時的にマスタからスレーブを分離したい場合に、プログラムを変更してダウンロードする必要はありません。[ReplacementSetpoint]を有効にし、終了したらもう一度無効にするだけです。セットポイント値は、[CurrentSetpoint]パラメータに値が表示されると、PID アルゴリズムに対して有効になります。

オートチューニング

カスケードされたマスタコントローラのオートチューニングは、以下の要件を満たす必要があります。

- 内部スレーブから 1 番目のマスタにコミッショニングされていること。
- マスタのすべてのスレーブが「自動モード」であること。
- マスタの出力がスレーブのセットポイントであること。

PID_Temp 命令は、カスケードでのオートチューニングで以下のサポートを行います。

- オートチューニングをマスタコントローラに対して開始する場合、マスタはすべてのスレーブが「自動モード」であるかどうかと、すべてのスレーブに対して交換セットポイント機能が無効になっていること ([ReplacementSetpointOn] = FALSE) をチェックします。この条件を満たしていない場合、マスタをオートチューニングすることはできません。マスタはチューニングをキャンセルし、「非アクティブモード」になります ([ActivateRecoverMode] = FALSE の場合)。または、[モード] パラメータに保存されているモードに戻ります ([ActivateRecoverMode] = TRUE の場合)。マスタはエラーメッセージ 200000hex (「カスケードのマスタにエラーがあります。スレーブが自動モードでないか、代替セットポイントが有効になっていて、マスタのチューニングが行えません」) が表示されます。
- すべてのスレーブが「自動モード」の場合、システムはパラメータ [AllSlaveAutomaticState] = TRUE を設定します。このパラメータをプログラムに適用するか、エラー 200000hex の原因を特定することができます。
- [ReplacementSetpoint] がすべてのスレーブに対して無効になっている場合、システムはパラメータ [NoSlaveReplacementSetpoint] = TRUE を設定します。このパラメータをプログラムに適用するか、エラー 200000hex の原因を特定することができます。

PID_Temp 命令の [コミッショニング] ダイアログを使用すると、さらにカスケードチューニングのサポート (243 ページ) を受けることができます。

動作モードとエラー処理

PID_Temp コントローラでは、マスタまたはスレーブによる動作モードの切り替えは行えません。つまり、カスケード内部のマスタはスレーブにエラーが発生すると現在のモードにとどまります。これは、2 つ以上の並行スレーブがこのマスタコントローラで動作している場合に利点になります。1 つのチェーン内のエラーによって、並行チェーンがシャットダウンされることはありません。

同様に、カスケード内部のスレーブもマスタにエラーが発生した場合、現在の動作モードにとどまります。ただし、スレーブのそれ以降の動作は、スレーブのセットポイントがマスタの出力であるため、マスタの設定によって異なります。つまり、マスタを [ActivateRecoverMode] = TRUE で設定していてエラーが発生した場合、マスタは最後の値または代替出力値をスレーブのセットポイントとして出力します。マスタを [ActivateRecoverMode] = FALSE で設定している場合、マスタは「非アクティブモード」に切り替わり、スレーブがセットポイントとして「0.0」を使用できるよう、すべての出力を「0.0」に設定します。

スレーブコントローラのみがアクチュエータへのダイレクトアクセスが行え、マスタにエラーが発生した場合は自身の動作モードにとどまるため、プロセスへの損傷を防止できます。たとえば、プラスチックの加工デバイスの場合、マスタコントローラにエラーが発生したというだけでスレーブが動作を停止し、アクチュエータをシャットダウンして、プラスチックがデバイス内部で固まってしまうと致命的です。

アンチwindアップ

カスケード内部のスレーブは、マスタの出力からセットポイントを取得します。マスタがまだ制御偏差(セットポイント - 入力)を認識している間にスレーブが独自の出力限界値に達すると、マスタはいわゆる「windアップ」を防止するために、積分寄与をフリーズするか小さくします。「windアップ」の場合、マスタは積分寄与をきわめて大きな値にまで大きくし、コントローラが再度、通常の応答ができるようになる前に、この値を小さくする必要があります。このような「windアップ」は制御のダイナミック性にマイナスの影響を与えます。PID_Temp は、マスタコントローラのパラメータ [Config.Cascade.AntiWindUpMode] を設定することで、カスケード内でのこの影響を防止することができます。

値	説明
0	アンチwindアップ機能を無効にします。
1	マスタコントローラの積分寄与を「限界内のスレーブ」の比率で「既存スレーブ」にまで小さくします(パラメータ [CountSlaves])。
2	スレーブが限界値に達すると同時に、マスタの積分寄与をフリーズします。 [Config.Cascade.IsMaster] = TRUE の場合にのみ関係します。

8.7 PID_Temp 命令の ErrorBit パラメータ

PID コントローラに複数の保留中の警告がある場合、エラーコードの値をバイナリ加算で表示します。たとえば、エラーコード 0003 の表示は、エラー 0001 および 0002 が保留中であることを示します。

表 8- 11 PID_Temp 命令の ErrorBit パラメータ

ErrorBit (DW#16#...)	説明
0000	エラーは発生していません。
0001 ^{1, 2}	Input パラメータがプロセス値の限界を超えています。 Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Input_PER パラメータに無効な値があります。アナログ入力エラーが保留中かどうかをチェックします。
0004 ⁴	ファインチューニング中のエラー。プロセス値の振動を維持できませんでした。
0008 ⁴	プレチューニングの開始時のエラー。プロセス値がセットポイントに近すぎます。ファインチューニングを開始してください。
0010 ⁴	チューニング中にセットポイントが変更されました。 注記: CancelTuningLevel タグでセットポイントに許可された変動を設定できます。
0020	ファインチューニング中にプレチューニングは行えません。 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_Temp はファインチューニングモードのままとなります。
0040 ⁴	プレチューニング中のエラー。冷却がプロセス値を小さくできませんでした。
0080 ⁴	プレチューニング中のエラー。出力値の限界の不正な設定。 出力値の限界が正しく設定されていて、制御ロジックと一致しているかどうかをチェックします。
0100 ⁴	ファインチューニング中のエラーによりパラメータが無効になりました。
0200 ^{2, 3}	Input パラメータに無効な値があります: 値の数字フォーマットが無効です。
0400 ^{2, 3}	出力値の計算が失敗しました。PID パラメータをチェックします。
0800 ^{1, 2}	サンプリング時間エラー:PID_Temp が周期割り込み OB のサンプリング時間内に呼び出されません。
1000 ^{2, 3}	Setpoint パラメータに無効な値があります: 値の数字フォーマットが無効です。
10000	ManualValue パラメータに無効な値があります: 値の数字フォーマットが無効です。 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_Temp は SubstituteOutput を出力値として使用します。ManualValue パラメータに有効な値を割り当てるとすぐに、PID_Temp はそれを出力値として使用します。

ErrorBit (DW#16#...)	説明
20000	SubstituteValue タグに無効な値があります: 値の数字フォーマットが無効です。 PID_Temp は出力の下限值を出力値として使用します。 注記: エラーが発生する前に自動モードが有効だった場合、つまり ActivateRecoverMode = TRUE だった場合、エラーはそれ以上保留されず、PID_Temp は自動モードに戻ります。
40000	Disturbance パラメータに無効な値があります: 値の数字フォーマットが無効です。 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = FALSE だった場合、Disturbance は 0 に設定されます。PID_Temp は自動モードのままです。 注記: エラーが発生する前にプレチューニングモードまたはファインチューニングモードが有効で、ActivateRecoverMode = TRUE だった場合、PID_Temp は Mode パラメータに保存されている動作モードに切り替わります。現在のフェーズの Disturbance が出力値に影響しない場合、チューニングはキャンセルされます。
200000	カスケード内のマスタにエラーがあります。スレーブが自動モードでないか、代替セットポイントが有効になっていて、マスタのチューニングが行えません。
400000	PID コントローラが、冷却が有効の間、加熱のためのプレチューニングを許可しません。
800000	冷却のためのプレチューニングを開始するために、プロセス値をセットポイントに近い値にする必要があります。
1000000	チューニングの開始でエラーが発生しました。[Heat.EnableTuning]および[Cool.EnableTuning]が設定されていないか、構成と一致していません。
2000000	冷却のためのプレチューニングには、加熱のためのプレチューニングが正常に行われている必要があります。
4000000	ファインチューニングの開始でエラーが発生しました。 [Heat.EnableTuning]と[Cool.EnableTuning]を同時に設定できません。
8000000	PID パラメータの計算中にエラーが発生し、パラメータが無効になりました(たとえば、負のゲイン; 現在の PID パラメータは変更されず、チューニングは有効になりません)。

- ¹ 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = TRUE の場合、PID_Temp は自動モードのままとなります。
- ² 注記: エラーが発生する前にプレチューニングモードまたはファインチューニングモードが有効で、ActivateRecoverMode = TRUE の場合、PID_Temp は Mode パラメータに保存されている動作モードに切り替わります。
- ³ 注記: エラーが発生する前に自動モードが有効で ActivateRecoverMode = TRUE の場合、PID_Compact は設定された代替出力値を出力します。エラーが保留中でなくなるとすぐに、PID_Temp は自動モードに戻ります。
- ⁴ 注記: エラーが発生する前に ActivateRecoverMode = TRUE だった場合、PID_Temp はチューニングをキャンセルし、Mode パラメータに保存されている動作モードに切り替わります。

8.8 PID_Compact および PID_3Step コントローラの構成

テクノロジーオブジェクトのパラメータによって PID コントローラの動作が決まります。アイコンを使用して、構成エディタを開きます。

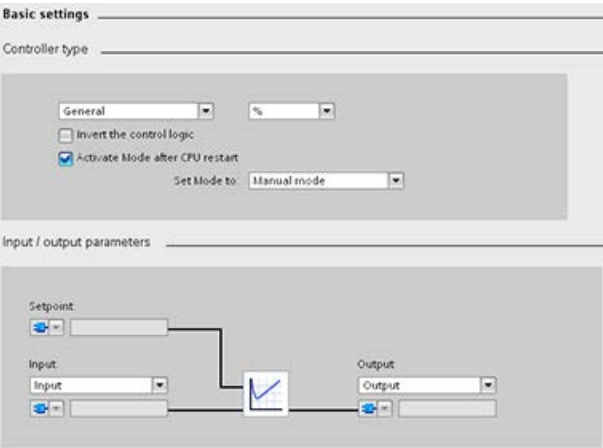


表 8- 12 PID_Compact 命令の構成の設定例

設定		説明
基本	コントローラのタイプ	エンジニアリング単位を選択します。
	制御ロジックの反転	反転動作 PID ループの選択を可能にします。 <ul style="list-style-type: none">• 選択しない場合、PID ループは直接動作モードとなり、PID ループの出力は入力値 < セットポイントの場合に大きくなります。• 選択する場合、PID ループの出力は入力値 > セットポイントの場合に大きくなります。
	CPU 再起動後に最後のモードを有効化	リセット後、または入力限界値を超えて有効な範囲に戻った場合に、PID ループを再開します。
	Input	プロセス値の入力パラメータまたは Input_PER パラメータ(アナログの場合)を選択します。Input_PER はアナログ入力モジュールから直接取得できます。
	出力	出力値の Output パラメータまたは Output_PER パラメータ(アナログの場合)を選択します。Output_PER は直接、アナログ出力モジュールに入力できます。
プロセス値	プロセス値の範囲と限界値の両方をスケーリングします。プロセス値が下限値を下回るか上限値を上回った場合、PID ループは非アクティブモードになり、出力値を 0 に設定します。 Input_PER を使用するには、アナログプロセス値(入力値)をスケーリングする 必要があります 。	

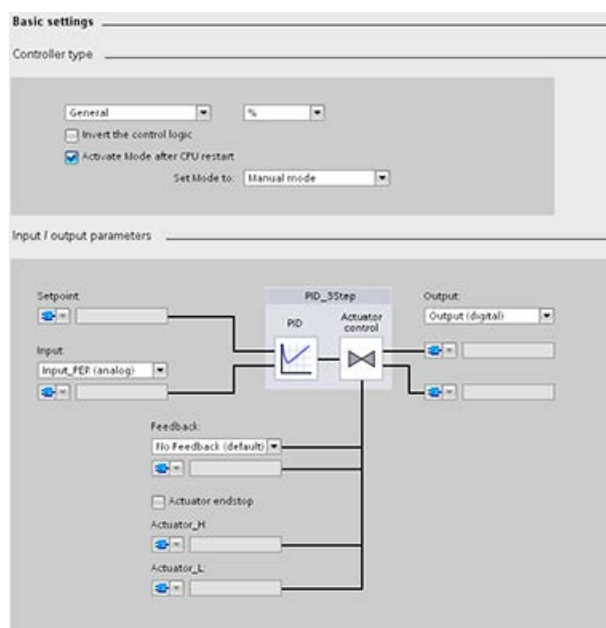


表 8- 13 PID_ 3 Stept 命令の構成の設定の例

設定	説明
基本	<p>コントローラのタイプ エンジニアリング単位を選択します。</p> <p>制御ロジックの反転 反転動作 PID ループの選択を可能にします。</p> <ul style="list-style-type: none"> 選択しない場合、PID ループは直接動作モードとなり、PID ループの出力は入力値 < セットポイントの場合に大きくなります。 選択する場合、PID ループの出力は入力値 > セットポイントの場合に大きくなります。 <p>CPU 再起動後にモードを有効化 リセット後、または入力限界値を超えて有効な範囲に戻った場合に、PID ループを再開します。 モードの設定: 再起動後の PID のモードを定義します。</p> <p>Input プロセス値の入力パラメータまたは Input_PER パラメータ(アナログの場合)を選択します。Input_PER はアナログ入力モジュールから直接取得できます。</p> <p>出力 出力値にデジタル出力を使用するか(Output_UP および Output_DN)、アナログ出力を使用するか(Output_PER)を選択します。</p> <p>Feedback PID ループに返されるデバイスのステータスのタイプを選択します。</p> <ul style="list-style-type: none"> フィードバックなし(デフォルト) Feedback Feedback_PER
プロセス値	<p>プロセス値の範囲と限界値の両方をスケーリングします。プロセス値が下限値を下回るか上限値を上回った場合、PID ループは非アクティブモードになり、出力値を 0 に設定します。 Input_PER を使用するには、アナログプロセス値(入力値)をスケーリングする必要があります。</p>
アクチュエータ	<p>モータ遷移時間 バルブが開いてから閉じるまでの時間を設定します。(この値は、データシートまたはバルブのフェースプレートで探してください。)</p> <p>最小オン時間 バルブの最少移動時間を設定します。(この値は、データシートまたはバルブのフェースプレートで探してください。)</p>

設定		説明
	最小オフ時間	バルブの最少一時停止時間を設定します。(この値は、データシートまたはバルブのフェースプレートで探してください。)
	エラーへの応答	エラーが検出されたとき、または PID ループがリセットされたときのバルブの動作を定義します。代替位置の使用を選択する場合は、[安全位置]を入力します。アナログフィードバックまたはアナログ出力の場合は、出力の上限値と下限値の間の値を選択します。デジタル出力の場合は、0% (オフ)または 100% (オン)のどちらかしか選択できません。
	位置フィードバックのスケールリング ¹	<ul style="list-style-type: none"> • [ハイエンドストップ]と[ローエンドストップ]は、正の最大位置(全開)と負の最大位置(全閉)を定義します。[ハイエンドストップ]は[ローエンドストップ]より大きくなければなりません。 • [プロセス値の上限値]と[プロセス値の下限値]は、チューニングおよび自動モード時のバルブの上側と下側の位置を定義します。 • [FeedbackPER] ([Low および High])は、バルブ位置のアナログフィードバックを定義します。[FeedbackPER High]は[FeedbackPER Low]より大きくなければなりません。
詳細設定	プロセス値のモニタリング	プロセス値の警告上限値と警告下限値を設定します。
	PID パラメータ	ユーザーが望む場合は、このウィンドウでユーザー独自の PID チューニングパラメータを入力することができます。これを可能にするには、[手動入力を有効化]チェックボックスをチェックする必要があります。

¹ [位置フィードバックのスケールリング]は、[基本]設定で[フィードバック]を有効にしている場合にのみ編集可能です。

8.9 PID_Temp コントローラの構成

テクノロジーオブジェクトのパラメータによって PID コントローラの動作が決まります。アイコンを使用して、構成エディタを開きます。

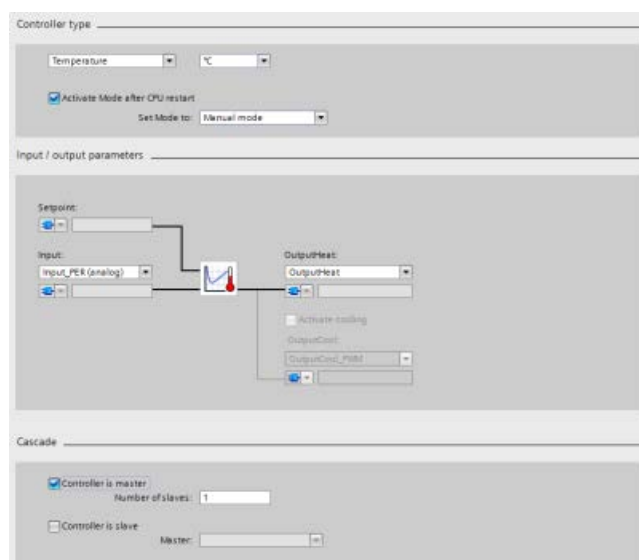


表 8- 14 PID_Temp 命令の構成の設定例

	設定	説明
基本	コントローラのタイプ	単位を選択します。
	CPU 再起動後にモードを有効化	リセット後、または入力限界値を超えて有効な範囲に戻った場合に、PID ループを再開します。 モードの設定: 再起動後の PID のモードを定義します。
	入力	プロセス値の入力パラメータまたは Input_PER パラメータ(アナログの場合)を選択します。Input_PER はアナログ入力モジュールから直接取得できます。
	Output Heat	出力値にデジタル出力を使用するか(OutputHeat および OutputHeat_PWM)、アナログ出力を使用するか(OutputHeat_PER (アナログ))を選択します。
	Output Cool	出力値にデジタル出力を使用するか(OutputCool および OutputCool_PWM)、アナログ出力を使用するか(OutputCool_PER (アナログ))を選択します。
プロセス値	プロセス値の範囲と限界値の両方をスケーリングします。プロセス値が下限値を下回るか上限値を上回った場合、PID ループは非アクティブモードになり、出力値を 0 に設定します。 Input_PER を使用するには、アナログプロセス値(入力値)をスケーリングする 必要があります 。	
カスケード	コントローラがマスタ	コントローラをマスタとして設定し、スレーブの数を選択します。
	コントローラがスレーブ	コントローラをスレーブとして設定し、マスタの数を選択します。

コントローラのタイプ

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
物理量	[PhysicalQuantity]	Int (Enum)	<ul style="list-style-type: none"> 全般 温度(=デフォルト) 	物理量値の事前選択 ファンクショナル ビューのオンライン モードではマルチ値制 御なし、編集不可。
測定単位	[PhysicalUnit]	Int (Enum)	<ul style="list-style-type: none"> 全般: 単位 = % 温度: 単位(可能な選択肢) = <ul style="list-style-type: none"> °C (=デフォルト) °F K 	物理量を変更すると、 ユーザーの単位の選択 は「0」に設定し直さ れます。
CPU 再 起動後に モードを 有効化	[RunModeByStartup]	Bool	チェックボックス	TRUE (=デフォルト) に設定されている場 合、コントローラは電 源を切断して再投入し た後(電源オン - オフ - オン)または PLC が STOP から RUN に移 行した後で、[Mode]変 数に保存されている状 態に切り替わります。 設定されていない場 合、PID_Temp は「非 アクティブ」モードの ままです。
モードの 設定	[Mode]	Int (Enum)	モード(可能な選択肢) <ul style="list-style-type: none"> 0: 非アクティブ 1: プレチューニング 2: ファインチューニング 3: 自動モード 4: 手動モード (=デフォルト) 	エンジニアリングス テーション(ES)は [Mode]変数の開始値を ユーザーの選択に従っ て設定します。Mode のデフォルト値(TO- DB に格納)は、手動 モードです。

入力/出力パラメータ

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
セット ポイント	Setpoint	Real)	REAL	[プロパティ]ページでのみアクセス可能です。 ファンクショナルビューのオンラインモードでのマルチ値制御なし。
入力の 選択	Config.InputPerOn	Bool (Enum)	Bool	使用する入力の種類を選択します。 可能な選択肢 <ul style="list-style-type: none"> FALSE: [Input] (Real) TRUE: [Input_PER (アナログ)]
入力	Input または Input_PER	Real または Int	Real または Int	[プロパティ]ページでのみアクセス可能です。 ファンクショナルビューのオンラインモードでのマルチ値制御なし。
出力の選 択(加熱)	Config.Output.Heat.Select	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	加熱に使用する出力の種類を選択します。 可能な選択肢 <ul style="list-style-type: none"> [OutputHeat] (Real) [OutputHeat_PWM] (Bool) (=デフォルト) [OutputHeat_PER (アナログ)] (Word) [カスケード]セクションの[このコントローラはマスタ]チェックボックスをユーザーが有効にしている場合、一回 [OutputHeat]に設定されます。
出力(加熱)	OutputHeat、OutputHeat_PER、または OutputHeat_PWM	Real または Int または Bool	Real、Int、または Bool	[プロパティ]ページでのみアクセス可能です。 ファンクショナルビューのオンラインモードでのマルチ値制御なし。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
出力の有効化(冷却)	Config.ActivateCooling	Bool	Bool	<p>このチェックボックスをチェック</p> <ul style="list-style-type: none"> • [Config.Output.Heat.PidLowerLimit] = 0.0 を一回設定します。 • 以下を設定します チェックしない場合(=デフォルト)、[Config.ActivateCooling] パラメータを FALSE ではなく TRUE に設定します。 • 他のすべての[出力(冷却)]制御を有効にします([基本設定]およびその他のビュー)。 • PID シンボルから制御までのラインを灰色から黒に変更します。 • [カスケード]セクションの[このコントローラはマスタ]チェックボックスが無効になります。 <p>注記: コントローラをカスケード用のマスタとして設定していない場合にのみ使用可能です([カスケード]セクションの[このコントローラはマスタ]チェックボックスが無効; [Config.Cascade.IsMaster] = FALSE)。</p>

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
出力の選 択(冷却)	[Config.Output.Cool.Select]	Int (Enum)	2 >= Config.Output. Heat.Select >= 0	冷却に使用する出力の種 類を選択します。 可能な選択肢 <ul style="list-style-type: none"> • [OutputCool] (Real) • [OutputCool_PWM] (Bool) (=デフォルト) • [OutputCool_PER (アナログ)] (Word) [出力(冷却)を有効化]を チェックしている場合に のみ使用可能です; (Config.ActivateCooling = TRUE)。
出力(冷却)	OutputCool、 OutputCool_PER、または OutputCool_PWM	Real ま たは Int または Bool	Real、Int、または Bool	[プロパティ]ページでの みアクセス可能です。 ファンクショナルビュー のオンラインモードでの マルチ値制御なし。

カスケードパラメータ

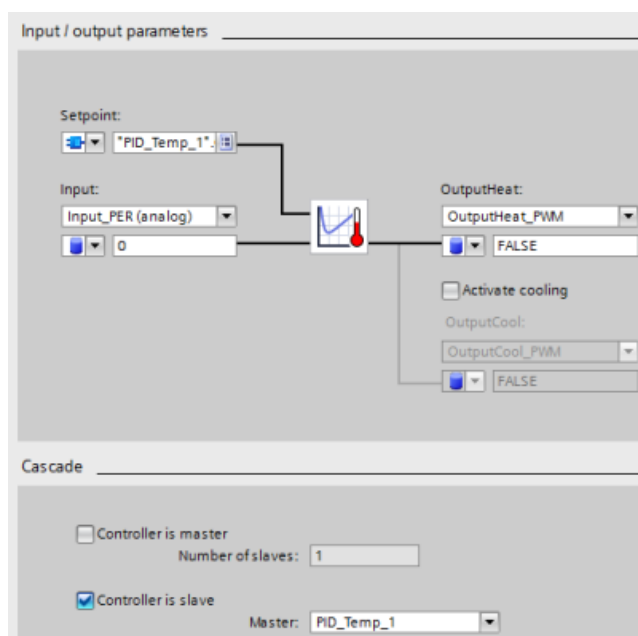
以下のパラメータを使用して、コントローラをマスタまたはスレーブとして選択し、マスタコントローラからセットポイントを直接受け取るスレーブの数を決定することができます。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
このコントローラはマスタ	[Config.Cascade.IsMaster]	Bool	Bool	<p>当該コントローラがカスケード内でマスタかどうかを示します。このチェックボックスをチェックすると、以下を実行できます。</p> <ul style="list-style-type: none"> チェックしない場合 (=デフォルト)、パラメータ [Config.Cascade.IsMaster] を FALSE ではなく TRUE に設定します。 [入力/出力パラメータ]セクションの[出力(加熱)の選択]を一回 [OutputHeat]に設定します (Config.Output.Heat.Select = 0)。 [スレーブの数]入力フィールドを有効にします。 [入力/出力パラメータ]セクションの[出力(冷却)の有効化]チェックボックスを無効にします。 <p>注記: 当該コントローラの冷却出力が無効の場合にのみ使用可能です([入力/出力パラメータ]セクションの[出力(冷却)の有効化]チェックボックスが無効 (Config.ActivateCooling = FALSE))。</p>

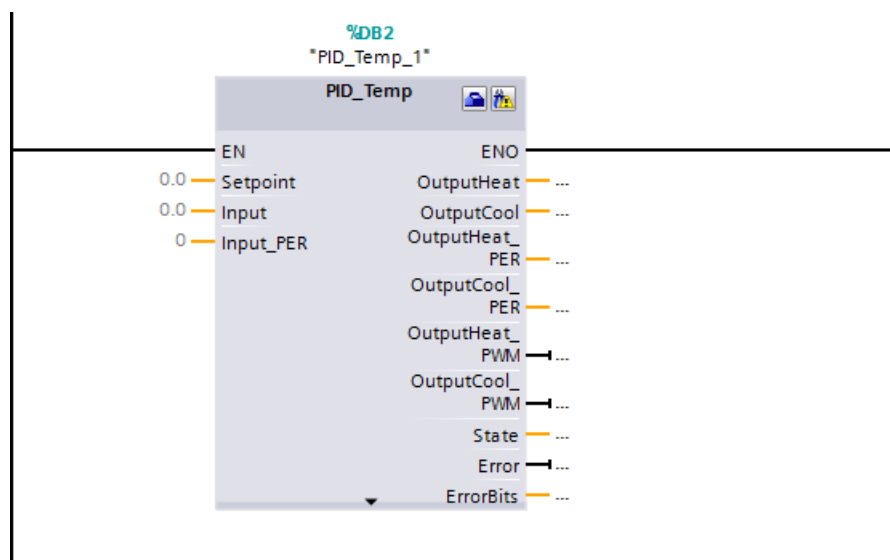
設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
スレーブ の数	[Config.Cascade.CountSlaves]	Int	255 >= Config.Cascade. CountSlaves >= 1	当該マスタコントローラ から直接、セットポイン トを受け取るスレーブコ ントローラの数。 PID_Temp 命令は、他の 値とともにこの値を処理 して、アンチwind アップ処理を行います。 [スレーブの数]は[このコ ントローラがマスタ] チェックボックスが有効 (Config.Cascade. IsMaster = TRUE)の場合 にのみ使用可能です。
この コント ローラは スレーブ	[Config.Cascade.IsSlave]	Bool	Bool	当該コントローラがカス ケード内でスレーブかど うかを示します。この チェックボックスを チェックする場合は、パ ラメータ [Config.Cascade.IsSlave] を TRUE に設定します。 チェックしない場合(=デ フォルト)は FALSE に設 定します。[プロパティ] ページでこのチェック ボックスをチェックし て、[SelectionMaster]ド ロップダウンリストを 有効にする必要があり ます。

例: カスケードコントローラ

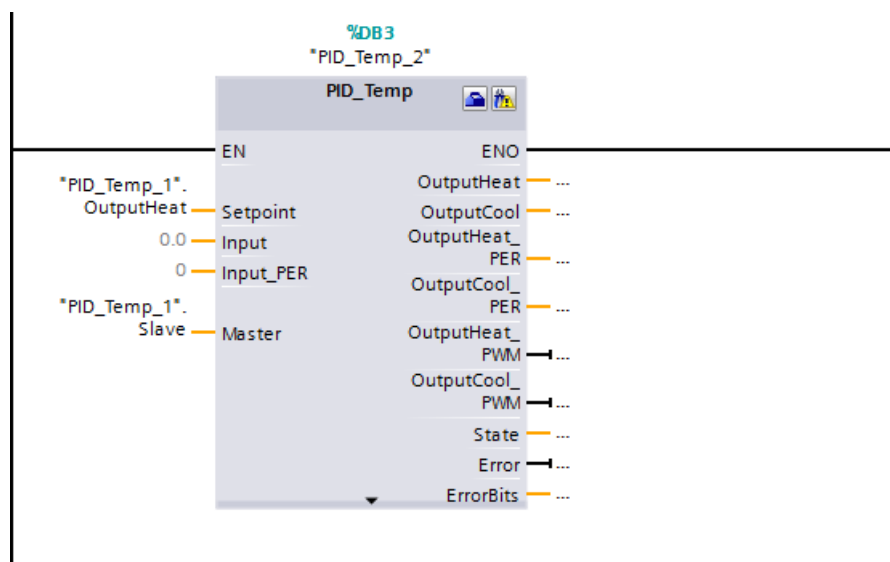
[PID_Temp_1]をマスタとして選択すると、下の[基本設定]ダイアログに、[入力/出力パラメータ]セクションとスレーブコントローラ[PID_Temp_2]用の[カスケード]セクションが表示されます。マスタコントローラとスレーブコントローラ間の接続を作成します。



ネットワーク 1: このネットワークでは、[PID_Temp_1]マスタと[PID_Temp_2]スレーブとの間にプログラミングエディタで接続を作成します。



ネットワーク 2: [PID_Temp_1]マスタの[OutputHeat]および[Slave]パラメータを、[PID_Temp_2]スレーブの[Setpoint]および[Master]パラメータとの間の接続をそれぞれ作成します。



温度プロセスのオートチューニング

PID_Temp 命令はオートチューニング用の 2 つのモードを提供します。

- [プレチューニング] (パラメータ[Mode] = 1)
- [ファインチューニング] (パラメータ[Mode] = 2)

コントローラの設定に従って、以下のさまざまな種類のチューニング方法が使用可能です。

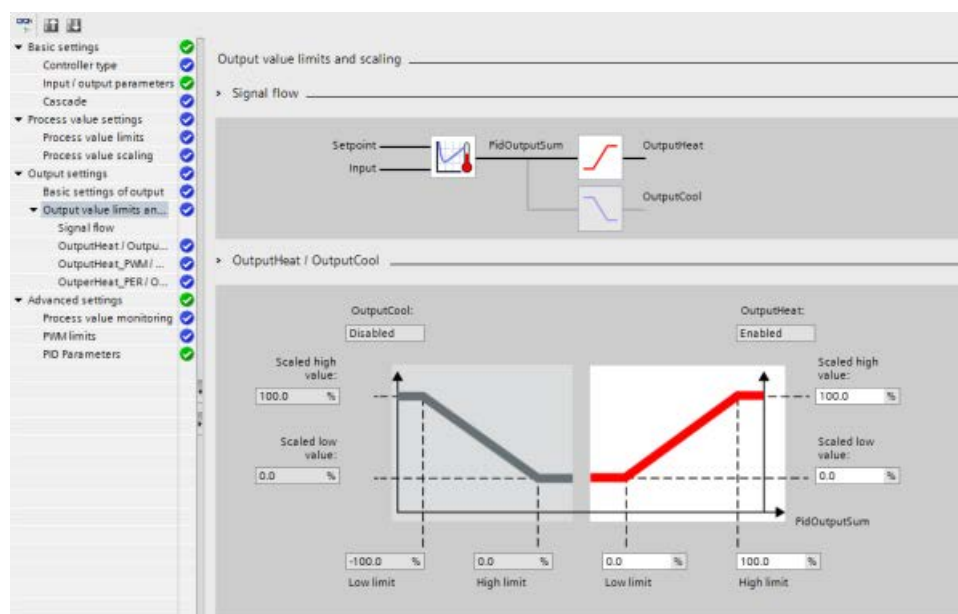
コンフィグレーション	加熱出力を持つコントローラ	冷却ファクタを使用する加熱および冷却出力を持つコントローラ	2つのPIDパラメータセットを使用する加熱および冷却出力を持つコントローラ
関連するTO-DB 値	<ul style="list-style-type: none"> Config.ActivateCooling = FALSE Config.AdvancedCooling = irrelevant 	<ul style="list-style-type: none"> Config.ActivateCooling = TRUE Config.AdvancedCooling = FALSE 	<ul style="list-style-type: none"> Config.ActivateCooling = TRUE Config.AdvancedCooling = TRUE
使用可能なチューニング方法	<ul style="list-style-type: none"> [プレチューニング加熱] [ファインチューニング加熱] (冷却オフセットは使用できません) 	<ul style="list-style-type: none"> [プレチューニング加熱] [ファインチューニング加熱] (冷却オフセットを使用できます) 	<ul style="list-style-type: none"> [プレチューニング加熱および冷却] [プレチューニング加熱] [プレチューニング冷却] [ファインチューニング加熱] (冷却オフセットを使用できます) [ファインチューニング冷却] (加熱オフセットを使用できます)

出力限界値とスケーリング

冷却の有効化が無効

PID_Temp 命令をカスケードのマスタとして設定し、[基本設定]ビューの[出力(冷却)を有効化]チェックボックスのチェックがはずされていて無効になっている場合、冷却の有効化に依存する[出力設定]ビューのすべての設定も無効になります。

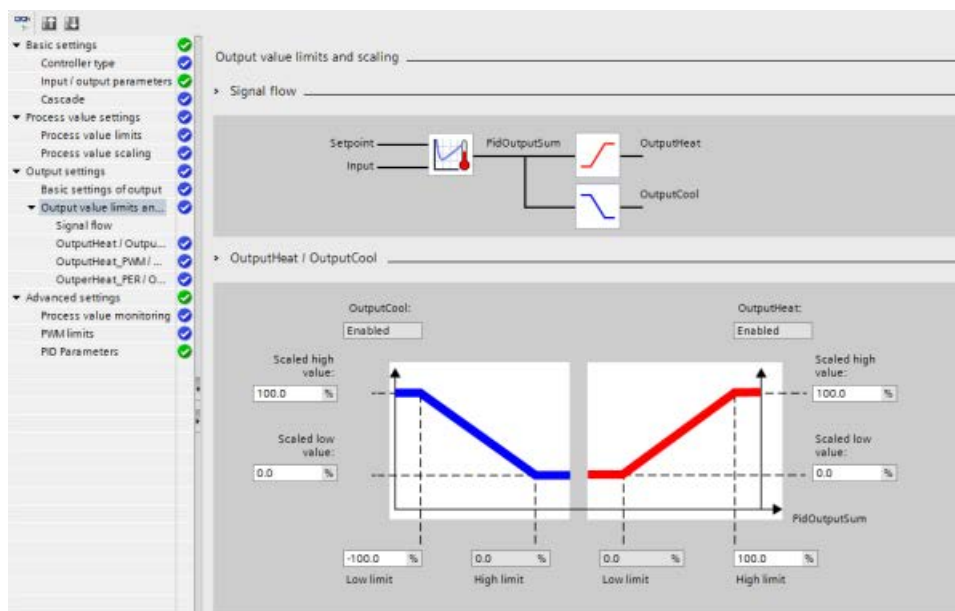
下の図に、冷却が無効になっている[出力設定]ビューの[出力限界値とスケーリング]セクションを示します(OutputHeat_PWM が[入力/出力パラメータ]ビューで選択されていて、OutputHeat が常に有効)。

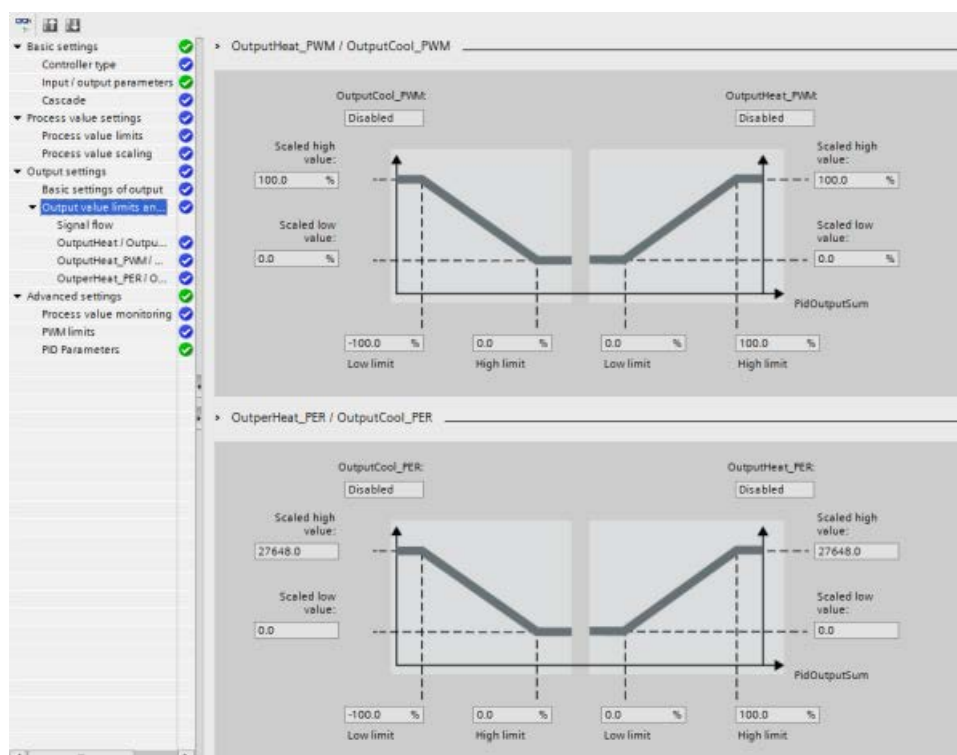




冷却の有効化が有効

下の図に、冷却が有効になっている[出力設定]ビューの[出力限界値とスケール]セクションを示します(OutputCool_PER および OututHeat_PWM が[入力/出力パラメータ]ビューで選択されています;OutputCool と OutputHeat は常に有効)。





動作モード

動作モードを手動で変更するには、コントローラの[Mode] in-out パラメータを設定し、[ModeActivate]を FALSE から TRUE に設定して(立ち上がりエッジがトリガされます)それを有効にする必要があります。次のモードを変更する前に、[ModeActivate]をリセットしてください。[ModeActivate]は自動的にリセットされません。

出力パラメータ[State]は現在の動作モードを示し、可能な場合は要求された[モード]に設定されます。[State]パラメータは直接変更できません。[Mode]パラメータを使用するか、コントローラによる自動動作モード変更によってのみ変更されます。

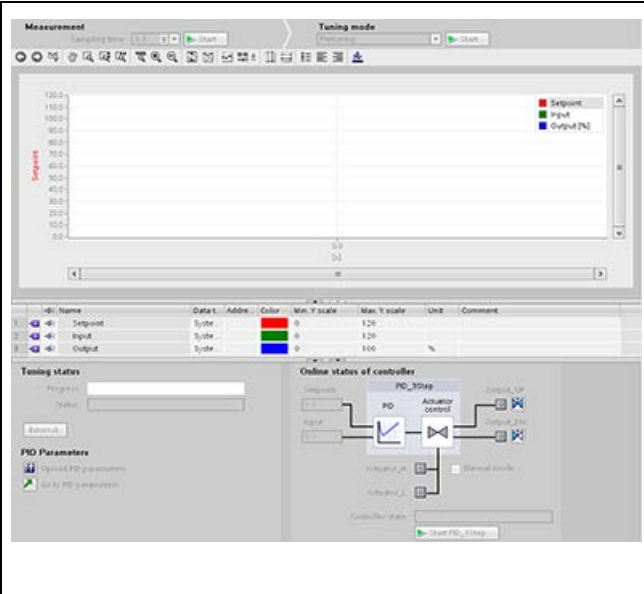
[Mode] / [State]	名前	説明
0	非アクティブ	<p>PID_Temp 命令</p> <ul style="list-style-type: none"> PID アルゴリズムとパルス幅変調を無効にします 設定されている出力限界値またはオフセットとは無関係に、すべてのコントローラ出力(OutputHeat、OutputCool、OutputHeat_PWM、OutputCool_PWM、OutputHeat_PER、OutputCool_PER)を「0」(FALSE)に設定します。[Mode] = 0、[Reset] = TRUE の設定、またはエラーによりこのモードになります。
1	プレチューニング (スタートアップ チューニング/SUT)	<p>このモードはコントローラの最初のスタートアップ時にパラメータを決定します。PID_Compact とは異なり、PID_Temp では、加熱チューニング、冷却チューニング、またはその両方が必要かどうかを[Heat.EnableTuning]および[Cool.EnableTuning]パラメータを使用して選択する必要があります。</p> <p>非アクティブ、自動モード、または手動モードから[プレチューニング]を有効にすることができます。</p> <p>チューニングが正常に終了した場合、PID_Temp は自動モードに切り替わります。チューニングが正常に終了しなかった場合、動作モードの切り替えは[ActivateRecoverMode]によって異なります。</p>

[Mode] / [State]	名前	説明
2	ファインチューニング(RUN/TIR でのチューニング)	<p>このモードは、セットポイントでの PID コントローラの最適なパラメータ割り当てを決定します。</p> <p>PID_Compact とは異なり、PID_Temp では、加熱チューニングまたは冷却チューニングが必要かどうかを[Heat.EnableTuning]および[Cool.EnableTuning]パラメータを使用して選択する必要があります。</p> <p>非アクティブ、自動モード、または手動モードから[ファインチューニング]を有効にすることができます。</p> <p>チューニングが正常に終了した場合、PID_Temp は自動モードに切り替わります。</p> <p>チューニングが正常に終了しなかった場合、動作モードの切り替えは[ActivateRecoverMode]によって異なります。</p>
3	自動モード	<p>自動モード(標準 PID 制御モード)では、PID アルゴリズムの結果によって出力値が決まります。</p> <p>エラーが発生して[ActivateRecoverMode] = FALSE の場合、PID_Temp は非アクティブに切り替わります。エラーが発生して[ActivateRecoverMode] = TRUE の場合、動作モードの切り替えはエラーによって異なります。詳細は、PID_Temp 命令の ErrorBit パラメータ(222 ページ)を参照してください。</p>
4	手動モード	<p>このモードでは、PID コントローラはパラメータ[ManualValue]の値をスケーリングおよび制約し、出力に転送します。</p> <p>PID コントローラは PID アルゴリズムのスケーリングで[ManualValue]を割り当てるため([PidOutputSum]と同様に)、[ManualValue]の値が加熱出力と冷却出力のどちらに有効であるかは値によって決まります。</p> <p>このモードになるには、[Mode] = 4 または[ManualEnable] = TRUE を設定します。</p>
5	エラーモニタリングによる代替出力値(復元モード)	<p>このモードは[Mode] = 5 の設定によって有効になります。このモードは、自動モードがエラーの発生時に有効になっている場合の、コントローラの自動エラー応答です。</p> <ul style="list-style-type: none"> SetSubstituteOutput = FALSE (最後に有効だった出力値) SetSubstituteOutput = TRUE (パラメータ[SubstituteOutput]に保存されている値) <p>PID_Temp が[自動モード]で、[ActivateRecoverMode]パラメータ = TRUE の場合、PID_Temp は以下のエラーの場合にこのモードに変わります。</p> <ul style="list-style-type: none"> 「[Input_PER]パラメータに無効な値があります。アナログ入力でエラーをチェックします(たとえば、断線など)。」 (ErrorBits = DW#16#0002) 「[Input]パラメータに無効な値があります。値が数値ではありません。」 (ErrorBits = DW#16#0200) 「出力値の計算が失敗しました。PID パラメータをチェックします。」 (ErrorBits = DW#16#0400) 「[Setpoint]パラメータに無効な値があります。値が数値ではありません。」 (ErrorBits = DW#16#1000) <p>エラーが保留中でなくなると、PID_Temp は自動的に自動モードに戻ります。</p>

8.10 PID_Compact および PID_3Step コントローラのコミッショニング

PID コントローラをスタートアップ時のオートチューニングおよび動作中のオートチューニング用に設定するには、コミッショニングエディタを使用します。コミッショニングエディタを開くには、命令またはプロジェクトナビゲータのどちらかのアイコンをクリックします。

表 8- 15 コミッショニング画面の例(PID_3Step)

	<ul style="list-style-type: none"> 測定: セットポイント、プロセス値(入力値)、出力値をリアルタイムトレンドで表示するには、サンプリング時間を入力して[Start]ボタンをクリックします。 チューニングモード: PID ループをチューニングするには、[プレチューニング]または[ファインチューニング](手動)のいずれかを選択して、[Start]ボタンをクリックします。PID コントローラは複数のフェーズにわたって稼働し、システム応答と更新時間を計算します。該当するチューニングパラメータがこの値から計算されます。 <p>チューニングプロセスが完了すると、コミッショニングエディタの[PID パラメータ]セクションの[PID パラメータのアップロード]ボタンをクリックして、新しいパラメータを保存できます。</p> <p>チューニング中にエラーが発生した場合、PID の出力値は「0」になります。その後、PID モードは[非アクティブ]モードに設定されます。ステータスインジケータでエラーが示されます。</p>
--	---

PID 開始値の制御

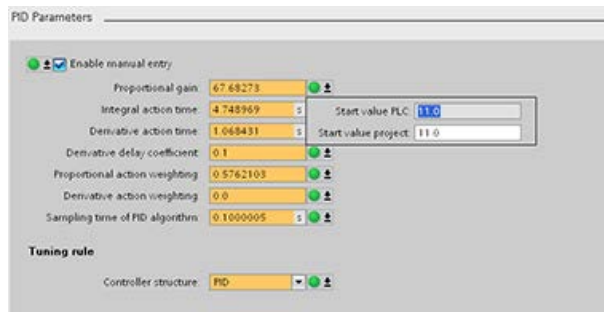
PID 構成パラメータの現在値を、PID コントローラの動作をオンラインモードで最適化できるように編集することができます。

PID コントローラの「テクノロジーオブジェクト」と、その「構成」オブジェクトを開きます。開始値の制御にアクセスするには、ダイアログの左上隅にある[眼鏡アイコン]をクリックします。



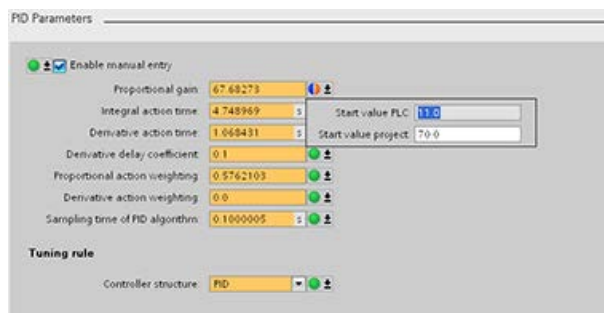
これで、下の図に示すように、すべての PID 構成パラメータの値を変更することができます。

現在値を各パラメータのプロジェクト(オフライン)開始値および PLC(オンライン)開始値と比較できます。これは、テクノロジーオブジェクトデータブロック(TO-DB)のオンライン/オフライン差分を比較し、PLC の次の Stop から Start への移行で現在値として使用される値を知るのに必要です。さらに、オンライン/オフラインを簡単に識別できる比較アイコンが表示されます。



上の図は、比較アイコンによってオンラインプロジェクトとオフラインプロジェクト間でどの値が異なっているかが表示された PID パラメータ画面を示しています。緑のアイコンは値が同じであることを示しています。青/オレンジのアイコンは値が異なっていることを示しています。

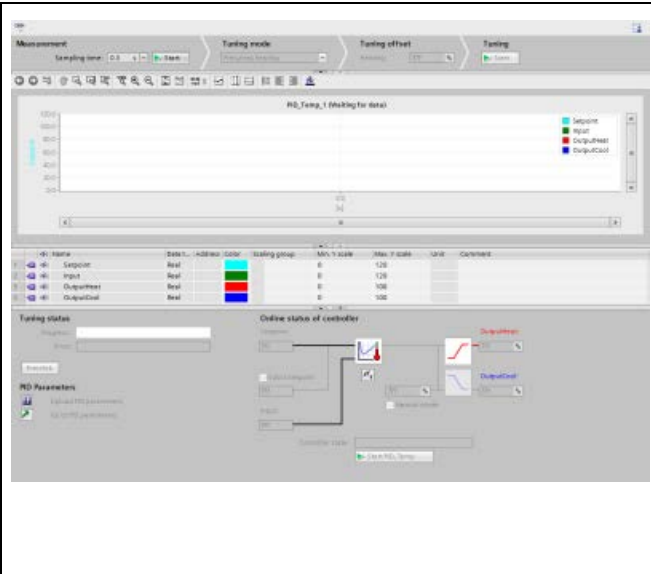
さらに、下矢印のついたパラメータボタンをクリックすると小さいウィンドウが開き、各パラメータのプロジェクト(オフライン)開始値と PLC(オンライン)開始値が表示されます。



8.11 PID_Temp コントローラのコミッショニング

PID コントローラをスタートアップ時のオートチューニングおよび動作中のオートチューニング用に設定するには、コミッショニングエディタを使用します。コミッショニングエディタを開くには、命令またはプロジェクトナビゲータのどちらかのアイコンをクリックします。

表 8- 16 コミッショニング画面の例(PID_Temp)

	<p>測定: セットポイント、プロセス値(入力値)、出力値をリアルタイムトレンドで表示するには、サンプリング時間を入力して [Start]ボタンをクリックします。</p> <p>チューニングモード: PID_Temp ループをチューニングするには、[プレチューニング]または[ファインチューニング](手動)のいずれかを選択して、[Start]ボタンをクリックします。PID コントローラは複数のフェーズにわたって稼働し、システム応答と更新時間を計算します。該当するチューニングパラメータがこの値から計算されます。</p> <p>チューニングプロセスが完了すると、コミッショニングエディタの[PID パラメータ]セクションの[PID パラメータのアップロード]ボタンをクリックして、新しいパラメータを保存できます。</p> <p>チューニング中にエラーが発生した場合、PID の出力値は「0」になります。その後、PID モードは[非アクティブ]モードに設定されます。ステータスインジケータでエラーが示されます。</p>
--	--

PWM 制限値

PID_Temp のソフトウェア PWM ファンクションで制御されるアクチュエータは、短すぎるパルス持続時間から保護する必要があります(たとえば、サイリスタリレーは、完全に反応できるようになる前に 20 ms 以上オンにする必要があります); 最小オン時間を割り当てます。アクチュエータは短いインパルスも無視するため、制御品質が損なわれます。最小オフ時間が必要な場合もあります(たとえば、過熱防止のため)。

PWM 制限値ビューを表示するには、テクノロジーオブジェクト(TO)構成でファンクショナルビューを開き、ナビゲーションツリーの[詳細設定]ノードから[PWM 限界値]を選択します。

ファンクショナルビューで[PWM 限界値]を開き、モニタリングを有効にすると([眼鏡]ボタン)、すべてのコントロールに TO-DB からのオンラインモニタ値がオレンジ色の背景色とマルチ値制御で表示され、値を編集することができます(構成条件が満たされている場合; 下の表を参照してください)。



設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
最小オン 時間(加熱) ^{1,2}	[Config.Output.Heat. MinimumOnTime]	REAL	100000.0 ≥ "Config.Output. Heat. MinimumOnTime ≥ 0.0	[OutputHeat_PWM]の パルスはこの値より短 くではありません。
最小オフ 時間(加熱) ^{1,2}	[Config.Output.Heat. MinimumOffTime]	REAL	100000.0 ≥ "Config.Output. Heat. MinimumOffTime ≥ 0.0	[OutputHeat_PWM]の ブレークはこの値より 短くではありません。
最小オン 時間(冷却) ^{1,3,4}	[Config.Output.Cool. MinimumOnTime]	REAL	100000.0 ≥ Config.Output. Cool. MinimumOnTime ≥ 0.0	[OutputCool_PWM]の パルスはこの値より短 くではありません。
最小オフ 時間(冷却) ^{1,3,4}	[Config.Output.Cool. MinimumOffTime]	REAL	100000.0 ≥ Config.Output. Cool. MinimumOffTime ≥ 0.0	[OutputCool_PWM]の ブレークはこの値より 短くではありません。

¹ フィールドに「s」(秒)が時間単位として表示されます。

² [基本設定]ビューでの選択出力(加熱)が[OutputHeat_PWM] (Config.Output.Heat.Select = TRUE)でない場合、この値を「0.0」に設定する必要があります。

³ [基本設定]ビューでの選択出力(冷却)が[OutputCool_PWM] (Config.Output.Cool.Select = TRUE)でない場合、この値を「0.0」に設定する必要があります。

⁴ [基本設定]ビューで[出力(冷却)を有効化]をチェックしている場合(Config.ActivateCooling = TRUE)にのみ使用可能です。

PID パラメータ

[詳細設定]ビュー、つまり[PID パラメータ]セクションを、冷却および/または[PID パラメータの切り替え]機能を無効にした状態で下に示します。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
手動入力の有効化	[Retain.CtrlParams. SetByUser]	Bool	Bool	PID パラメータを手動で入力するには、このチェックボックスをチェックする必要があります。
比例ゲイン(加熱) ²	[Retain.CtrlParams. Heat.Gain]	REAL	Gain >= 0.0	加熱のための PID 比例ゲイン
積分動作時間(加熱) ^{1,2}	[Retain.CtrlParams. Heat.Ti]	REAL	100000.0 >= Ti >= 0.0	加熱のための PID 積分動作。
微分動作時間(加熱) ^{1,2}	[Retain.CtrlParams. Heat.Td]	REAL	100000.0 >= Td >= 0.0	加熱のための PID 微分動作時間。
微分遅延係数(加熱) ²	[Retain.CtrlParams. Heat.TdFiltRatio]	REAL	TdFiltRatio >= 0.0	微分遅延時間を PID 微分時間からの係数として定義する加熱用の PID 微分遅延係数。
比例動作の重み付け(加熱) ²	[Retain.CtrlParams. Heat.PWeighting]	REAL	1.0 >= PWeighting >= 0.0	加熱用 PID 比例ゲインの直接制御経路またはループバック制御経路のいずれかでの重みづけ。
微分動作の重み付け(加熱) ²	[Retain.CtrlParams. Heat.DWeighting]	REAL	1.0 >= DWeighting >= 0.0	加熱用の PID 微分部分の直接制御経路またはループバック制御経路のいずれかでの重みづけ。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
PID アルゴリズム のサンプリング時 間(加熱) ^{1,2}	[Retain.CtrlParams. Heat.Cycle]	REAL	100000.0 >=Cycle > 0.0	加熱用の PID コント ローラの内部呼び出し サイクル。 FB 呼び出しサイクルタ イムの整数倍に丸めら れます。
デッドゾーン幅 (加熱) ^{2,3}	[Retain.CtrlParams. Heat.DeadZone]	REAL	DeadZone>= 0.0	加熱制御偏差のデッド バンドの幅。
制御ゾーン(加熱) ^{2,3}	[Retain.CtrlParams. Heat.ControlZone]	REAL	ControlZone> 0.0	PID 制御が有効な加熱 用の制御偏差ゾーンの 幅。制御偏差がこの範 囲から離れると、出力 は最大出力値に切り替 えられます。 デフォルト値は [MaxReal]なので、制御 ゾーンはオートチュー ニングが実行されない 限り無効です。 値「0.0」は制御ゾー ンには使用できません。 値「0.0」の場合、 PID_Temp は常にフル パワーで加熱または冷 却を行う 2 位置コント ローラと同じように動 作します。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
コントローラ構造 (加熱)	[PIDSelfTune.SUT. TuneRuleHeat]、 [PIDSelfTune.TIR. TuneRuleHeat]	Int	[PIDSelfTune.SUT. TuneRuleHeat] = 0..2、 [PIDSelfTune.TIR. TuneRuleHeat] = 0..5	<p>加熱のためのチューニングアルゴリズムを選択できます。</p> <p>可能な選択肢</p> <ul style="list-style-type: none"> PID (温度) (=デフォルト) <ul style="list-style-type: none"> ([PIDSelfTune.SUT. TuneRuleHeat] = 2) ([PIDSelfTune.TIR. TuneRuleHeat] = 0) PID <ul style="list-style-type: none"> ([PIDSelfTune.SUT. TuneRuleHeat] = 0) ([PIDSelfTune.TIR. TuneRuleHeat] = 0) PI <ul style="list-style-type: none"> ([PIDSelfTune.SUT. TuneRuleHeat] = 1) ([PIDSelfTune.TIR. TuneRuleHeat] = 4) <p>他のどの組み合わせでも [ユーザー定義] が表示されますが、[ユーザー定義] はデフォルトでは提供されません。</p> <p>[PID (温度)] は、温度プロセスのための特定のプレチューニング (SUT) 方法を備えた PID_Temp の新しい機能です。</p>
比例ゲイン(冷却) ⁴	[Retain.CtrlParams. Cool.Gain]	REAL	Gain >= 0.0	冷却のための PID 比例ゲイン
積分動作時間 (冷却) ^{1,4}	[Retain.CtrlParams. Cool.Ti]	REAL	100000.0 >=Ti >= 0.0	冷却のための PID 積分動作。
微分動作時間 (冷却) ^{1,4}	[Retain.CtrlParams. Cool.Td]	REAL	100000.0 >=Td >= 0.0	冷却のための PID 微分動作時間
微分遅延係数(冷却) ⁴	Retain.CtrlParams. Cool.TdFiltRatio]	REAL	TdFiltRatio >= 0.0	微分遅延時間を PID 微分時間からの係数として定義する冷却用の PID 微分遅延係数。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
比例動作の重み付け(冷却) ⁴	[Retain.CtrlParams. Cool.PWeighting]	REAL	1.0 >=PWeighting >= 0.0	冷却用の PID 比例ゲインの直接制御経路またはループバック制御経路のいずれかでの重みづけ。
微分動作の重み付け(冷却) ⁴	Retain.CtrlParams. Cool.DWeighting]	REAL	1.0 >=DWeighting >= 0.0	冷却用の PID 微分部分の直接制御経路またはループバック制御経路のいずれかでの重みづけ。
PID アルゴリズムのサンプリング時間(冷却) ^{1,4}	[Retain.CtrlParams. Cool.Cycle]	REAL	100000.0 >=Cycle > 0.0	冷却用の PID コントローラの内部呼び出しサイクル。 FB 呼び出しサイクルタイムの整数倍に丸められます。
デッドゾーン幅(冷却) ^{3,4}	[Retain.CtrlParams. Cool.DeadZone]	REAL	DeadZone>= 0.0	冷却制御偏差のデッドバンドの幅。

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
制御ゾーン (冷却) ^{3,4}	[Retain.CtrlParams. Cool.ControlZone]	REAL	ControlZone> 0.0	<p>PID 制御が有効な冷却用の制御偏差ゾーンの幅。制御偏差がこの範囲から離れると、出力は最大出力値に切り替えられます。</p> <p>デフォルト値は [MaxReal] なので、制御ゾーンはオートチューニングが実行されない限り無効です。</p> <p>値「0.0」は制御ゾーンには使用できません。</p> <p>値「0.0」の場合、PID_Temp は常にフルパワーで加熱または冷却を行う 2 位置コントローラと同じように動作します。</p>

設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
コントローラ構造 (冷却)	[PIDSelfTune.SUT. TuneRuleCool]、 [PIDSelfTune.TIR. TuneRuleCool]	Int	[PIDSelfTune.SUT. TuneRuleHeat] = 0..2、 [PIDSelfTune.TIR. TuneRuleHeat] = 0..5	<p>冷却のためのチューニングアルゴリズムを選択できます。</p> <p>可能な選択肢</p> <ul style="list-style-type: none"> PID (温度) (=デフォルト) <ul style="list-style-type: none"> ([PIDSelfTune.SUT. TuneRuleCool] = 2) ([PIDSelfTune.TIR. TuneRuleCool] = 0) PID <ul style="list-style-type: none"> ([PIDSelfTune.SUT. TuneRuleCool] = 0) ([PIDSelfTune.TIR. TuneRuleCool] = 0) PI <ul style="list-style-type: none"> ([PIDSelfTune.SUT. TuneRuleCool] = 1) ([PIDSelfTune.TIR. TuneRuleCool] = 4) <p>他のどの組み合わせでも [ユーザー定義] が表示されますが、[ユーザー定義] はデフォルトでは提供されません。</p> <p>[PID (温度)] は、温度プロセスのための特定のプレチューニング (SUT) 方法を備えた PID_Temp の新しい機能です。</p> <p>以下の項目をチェック/選択した場合のみ使用可能です。[基本設定] ビューの [出力(冷却)の有効化] ([Config.ActivateCooling] = TRUE)、および [出力の設定] ビューの [PID パラメータの切り替え] (Config.AdvancedCooling = TRUE)。</p>

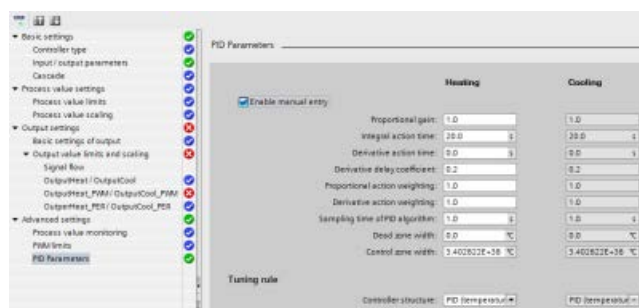
設定	TO-DB パラメータ	データ タイプ	値の範囲	説明
----	-------------	------------	------	----

- ¹ フィールドに「s」(秒)が時間単位として表示されます。
- ² PID パラメータの[手動入力の有効化] ([Retain.CtrlParams.SetByUser] = TRUE)をチェックしている場合にのみ使用可能です。
- ³ 測定単位が[基本設定]ビューでの選択に従ってフィールドの最後に表示されます。
- ⁴ 以下の項目をチェック/選択した場合のみ使用可能です。PID パラメータの[手動入力の有効化] ([Retain.CtrlParams.SetByUser] = TRUE)、[基本設定]ビューの[出力(冷却)の有効化] ([Config.ActivateCooling] = TRUE)、および[出力の設定]ビューの[PID パラメータの切り替え] ([Config.AdvancedCooling] = TRUE)。

PID 開始値の制御

PID 構成パラメータの現在値を、PID コントローラの動作をオンラインモードで最適化できるように編集することができます。

PID コントローラの「テクノロジーオブジェクト」と、その「構成」オブジェクトを開きます。開始値の制御にアクセスするには、ダイアログの左上隅にある[眼鏡アイコン]をクリックします。



これで、下の図に示すように、すべての PID 構成パラメータの値を変更することができます。

現在値を各パラメータのプロジェクト(オフライン)開始値および PLC(オンライン)開始値と比較できます。これは、テクノロジーオブジェクトデータブロック(TO-DB)のオンライン/オフライン差分を比較し、PLC の次の Stop から Start への移行で現在値として使用される値を知るのに必要です。さらに、オンライン/オフラインを簡単に識別できる比較アイコンが表示されます。

Parameter	Heating Value	Cooling Value
Proportional gain	1.0	1.0
Integral action time	20.0 s	20.0 s
Derivative action time	0.0 s	0.0 s
Derivative delay coefficient	0.2	0.2
Proportional action weighting	1.0	1.0
Derivative action weighting	1.0	1.0
Sampling time of PID algorithm	1.0 s	1.0 s
Dead zone width	0.0 °C	0.0 °C
Control zone width	3.402822E+38 °C	3.402822E+38 °C

Tuning rule

Controller structure: PD (temperaturi)

上の図は、比較アイコンによってオンラインプロジェクトとオフラインプロジェクト間でどの値が異なっているかが表示された PID パラメータ画面を示しています。緑のアイコンは値が同じであることを示しています。青/オレンジのアイコンは値が異なっていることを示しています。

さらに、下矢印のついたパラメータボタンをクリックすると小さいウィンドウが開き、各パラメータのプロジェクト(オフライン)開始値と PLC(オンライン)開始値が表示されます。

PID Parameters

☒ Enable manual entry

	Heating	Cooling
Proportional gain:	1.0	1.0
Integral action time:	20.0 s	20.0 s
Derivative action time:	0.0 s	0.0 s
Derivative delay coefficient:	0.2	0.2
Proportional action weighting:	1.0	1.0
Derivative action weighting:	1.0	1.0
Sampling time of PID algorithm:	1.0 s	1.0 s
Dead zone width:	0.0 °C	0.0 °C
Control zone width:	3.402822E+38 °C	3.402822E+38 °C

Tuning rule

Controller structure: PD (temperatur) PD (temperatur)

カンタン Web サーバ接続

Web サーバーは、使用している CPU に関するデータと CPU 内部のプロセスデータへの Web ページアクセスを提供します。この Web ページを使用して、PC またはモバイルデバイスの Web ブラウザで CPU (または、Web 対応 CP) にアクセスします。標準 Web ページでは、認可されたユーザーが以下の機能や、それ以外にも多くの機能を実行できます。

- CPU の動作モードを変更する (STOP から RUN)
- PLC タグ、データブロックタグ、I/O 値をモニタリングして変更する
- データログを表示してダウンロードする
- CPU の診断バッファを表示する
- CPU のファームウェアを更新する

Web サーバーでは、CPU データにアクセスできるユーザー定義の Web ページを作成することもできます。このようなページを自分が選択した HTML 作成ソフトウェアを使用して開発することができます。CPU 内のデータにアクセスするには、事前に定義した「AWP」(Automation Web Programming) コマンドを HTML コードに挿入します。

STEP 7 の CPU のデバイスコンフィグレーションで、Web サーバーのユーザーおよび権限レベルを設定します。

Web ブラウザの要件

Web サーバーは以下の PC Web ブラウザをサポートしています。

- Internet Explorer 8.0
- Internet Explorer 9.0
- Mozilla Firefox 17.0.1
- Google Chrome 23.0
- Apple Safari 5.1.7 (Windows)
- Apple Safari 6.0.2 (Mac)

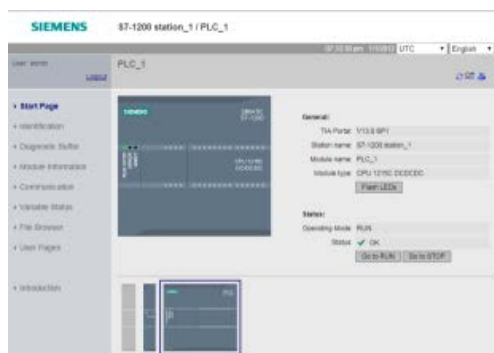
Web サーバーは以下のモバイルデバイスの Web ブラウザをサポートしています。

- Internet Explorer 6.0 まで、HMI パネルの場合
- Mobile Safari 7534.48.3 (iOS 5.0.1)
- Mobile Android Browser 2.3.4
- Mobile Google Chrome 23.0

標準 Web ページまたはユーザー定義 Web ページの表示を妨げるおそれのあるブラウザ関連の制約事項については、制約事項に関する項目 (258 ページ) を参照してください。

9.1 標準 Web ページの簡単使用

標準 Web ページを簡単に使用できます! CPU を設定するときに Web サーバーを有効にして、必要なタスクを実行するための権限を持つユーザーを設定するだけです。



開始ページには、接続先となる CPU の図が表示され、CPU に関する全般的な情報リンクがリスト表示されます。Web サーバー対応 CP を使用している場合、開始ページにはその CP も表示され、CP 経由で Web ページに接続することができます。

ユーザーが必要な権限を持っている場合、CPU の動作モード(STOP および RUN)を変更したり、LED を点滅させることができます。



変数ステータスページでは、CPU 内のすべての I/O またはメモリデータをモニタまたは変更することができます。値をモニタするには「タグステータス読み取り」権限、値を変更するには「タグステータス書き込み」権限を持っている必要があります。ダイレクトアドレス(I0.0 など)、PLC タグ名、特定のプログラムブロックのタグを入力することができます。データ値は、自動リフレッシュオプションを無効にするまで、自動的にリフレッシュされます。

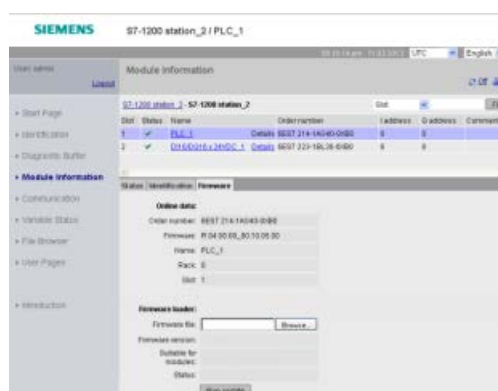


診断バッファページには診断バッファが表示され、診断の照会権限を持つユーザーがアクセスできます。表示される診断エントリの範囲を選択できます。

診断エントリには、発生したイベントとイベントが発生した CPU 時刻と日付がリストされます。個々のイベントを選択して、当該イベントに関する詳細情報を表示できます。



ファイルブラウザページでは、CPU のロードメモリ内のファイル、たとえばデータログ (124 ページ) およびレシピなどを表示、ダウンロード、編集することができます。CPU の保護レベルが 4 の場合を除き、すべてのユーザーがファイルブラウザページのファイルを見ることができます。ファイルの変更権限を持つユーザーは、ファイルの削除、編集、名前の変更が行えます。



モジュール情報ページでは、ステーション内のモジュールに関する情報を表示するほかに、ファームウェア更新をサポートする CPU または他のモジュールのファームウェアのバージョンを更新できます。診断の照会権限を持つユーザーは、モジュール情報ページを見ることができます。ファームウェアの更新の実行権限を持つユーザーは、ファームウェアを更新できます。

その他の標準 Web ページには、CPU に関する情報(シリアル番号、バージョン、製品番号など)と通信パートナーに関する情報(通信インターフェースのネットワークアドレスおよび物理的プロパティなど)が表示されます。



警告

Web サーバ経由の CPU への未許可のアクセス

CPU に未許可でアクセス、または PLC 変数を無効な値に変更すると、プロセスオペレーションが混乱し、死傷事故などの重大な人的傷害や物的損害が発生するおそれがあります。

Web サーバを有効にすることで未許可のユーザーが動作モードの変更、PLC データへの書き込み、ファームウェアの更新を行えるようになるため、シーメンスは以下のセキュリティ慣行を守ることを推奨します。

- Web サーバへのアクセスは、HTTPS プロトコルを使用したものだけを可能にします。
- パワフルなパスワードを持つパスワード保護 Web サーバユーザー ID。パワフルなパスワードは文字、数字、特殊文字を組合わせた最低 10 文字の長さで、辞書にある言葉や個人情報から取得できる名前や識別子でないものです。パスワードは秘密とし、頻繁に変更してください。
- 「Everybody」ユーザーのデフォルトの最低限の権限を拡張しない。
- Web ページユーザーは PLC 変数を無効な値に変更できるため、プログラムロジックの変数のエラーチェックと範囲チェックを行ってください。
- セキュアな仮想プライベートネットワーク (VPN) を使用して、保護されているネットワーク外部のロケーションから S7-1200 PLC Web サーバに接続する。

9.2 Web サーバーの使用に影響する制約事項

以下の IT 制約事項は Web サーバーの使用に影響を及ぼすおそれがあります。

- 通常は、標準 Web ページまたはユーザー定義 Web ページにアクセスするには、CPU の IP アドレスを使用するか、無線ルーターの IP アドレスとポート番号を使用する必要があります。Web ブラウザが IP アドレスへの直接接続を許可しない場合は、IT 管理者に相談してください。ローカルポリシーが DNS をサポートしている場合、当該アドレスへの DNS エントリ経由で IP アドレスに接続することができます。
- ファイアウォール、プロキシ設定、およびその他のサイト固有の制約事項により、CPU へのアクセスが制約されることもあります。このような問題を解決するには、IT 管理者に相談してください。
- 標準 Web ページは JavaScript とクッキーを使用します。Web ブラウザの設定で JavaScript またはクッキーが無効になっている場合は、有効にしてください。有効にできない場合、一部の機能が制約されます。ユーザー定義 Web ページでの JavaScript とクッキーの使用はオプションです。使用する場合、自分のブラウザで有効にする必要があります。
- Web サーバーは Secure Sockets Layer (SSL) をサポートしています。標準 Web ページおよびユーザー定義 Web ページには、`http://ww.xx.yy.zz` または `https://ww.xx.yy.zz` のいずれかの URL でアクセスできます。「ww.xx.yy.zz」は CPU の IP アドレスを表しています。
- シーメンスは Web サーバーへの安全なアクセスのためにセキュリティ証明書を用意しています。概要標準 Web ページから、この証明書をダウンロードして Web ブラウザのインターネットオプションにインポートできます。証明書のインポートを選択しない場合、`https://` で Web サーバーにアクセスするたびにセキュリティ検証プロンプトが表示されます。

接続数

Web サーバーは最大 30 のアクティブな HTTP 接続をサポートします。使用する Web ブラウザとページあたりの各オブジェクト(.css ファイル、イメージ、その他の.html ファイル)の数に応じて、各種アクションは 30 接続数を仮定しています。Web サーバーがページを表示中でも存続する接続もあります。それ以外の接続は、最初の接続後は存続しません。

たとえば、最大 6 つの永続的な接続をサポートする Mozilla Firefox 8 を使用している場合、Web サーバーが接続の切断を開始する前に、5 つのブラウザまたはブラウザタブインスタンスを使用できます。ページで 6 つの接続のすべてが使用されていない場合は、追加のブラウザまたはブラウザタブインスタンスを持つことができます。

アクティブ接続の数がページの性能に影響することにも気をつけてください。

注記

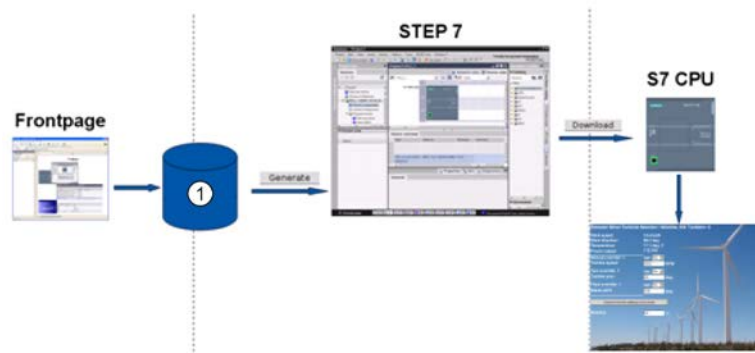
Web サーバーを閉じる前にログオフする

Web サーバーにログインした場合は、必ず Web ブラウザを閉じる前にログオフしてください。Web サーバーは最大で 7 つの同時ログインをサポートします。

9.3 ユーザー定義 Web ページの簡単作成

9.3.1 カスタム「ユーザー定義」Webページの簡単作成

S7-1200 Web サーバーでは、PLC からのデータを組み込んだ独自のアプリケーション固有 HTML ページを作成することもできます。選択した HTML エディタを使用してこのようなページを作成し、標準 Web ページからそのページにアクセスできる場所から CPU にダウンロードします。



① AWP コマンドが埋め込まれた HTML ファイル

このプロセスには複数のタスクが関係します。

- HTML エディタで HTML ページを作成する。
- AWP コマンドを HTML コードで HTML コメントに組み込む: AWP コマンドは CPU 情報にアクセスするためのコマンドの固定セットです。
- STEP 7 が HTML ページを読み取って処理できるよう設定する。
- HTML ページからプログラムブロックを生成する。
- STEP 7 が HTML ページの使用を制御できるようプログラミングする。
- プログラムブロックをコンパイルして CPU にダウンロードする。
- PC またはモバイルデバイスからユーザー定義 Web ページにアクセスする。

選択したソフトウェアパッケージを使用して、Web サーバーで使用する独自の HTML ページを作成します。HTML コードが W3C (World Wide Web Consortium) の HTML 規格を遵守していることを確認してください。STEP 7 は HTML 構文の検証を行いません。

ユーザーは WYSIWYG での設計またはレイアウトモデルの設計が可能なソフトウェアパッケージを使用できますが、純粋な HTML フォームで HTML コードを編集可能でなければなりません。ほとんどの Web 作成ツールはこのタイプの編集が可能です。そうでない場合は、常に簡単なテキストエディタを使用して HTML コードを編集することができます。ページの文字セットを UTF-8 に設定するには、以下の行を HTML ページに組み込みます。

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

また、必ずエディタからのファイルを UTF-8 文字コード化で保存してください。

STEP 7 を使用して、HTML ページ内のすべてのものを STEP 7 データブロックにコンパイルします。このデータブロックは、Web ページの表示を指示する 1 つの制御データブロックと、コンパイルされた Web ページの入った 1 つ以上のフラグメントデータブロックで構成されます。HTML ページの大きなセット、特に多数のイメージの入ったセットは、フラグメント DB 用に大量のロードメモリスペースを必要とすることに注意してください。CPU の内部ロードメモリがユーザー定義 Web ページに不十分な場合は、メモリカードを使用して外部ロードメモリを用意してください。

S7-1200 のデータを使用するよう HTML コードをプログラミングするには、AWP コマンドを HTML コマンドとして組み込みます。終了したら、HTML ページを PC に保存し、保存したフォルダパスを書きとめてください。

注記

AWP コマンドの入った HTML ファイルのファイルサイズの限界値は、64 キロバイトです。ファイルサイズをこの限界値より下に保つ必要があります。

ユーザー定義 Web ページをリフレッシュする

ユーザー定義 Web ページは自動的にリフレッシュされません。HTML がページをリフレッシュするかどうかをプログラミングするのはユーザーの選択です。PLC データを表示するページの場合、リフレッシュによって定期的にデータが最新に保たれます。データ入力用のフォームとして機能する HTML ページの場合、リフレッシュはユーザーのデータ入力を妨げるおそれがあります。ページ全体が自動的にリフレッシュされるようにしたい場合は、以下の行を HTML ヘッダーに追加します。ここで、「10」はリフレッシュ間の秒数です。

```
<meta http-equiv="Refresh" content="10">
```

JavaScript または他の HTML 技術を使用して、ページまたはデータのリフレッシュを制御することもできます。これについては、HTML と JavaScript に関する文書を参照してください。

9.3.2 ユーザー定義 Web ページに固有の制約事項

標準 Web ページの制約事項は、ユーザー定義 Web ページにも適用されます。また、ユーザー定義 Web ページにはいくつかの固有の制約事項があります。

ロードメモリスペース

ユーザー定義 Web ページは、[ブロックの生成]をクリックするとデータブロックになりますが、これにはロードメモリスペースが必要です。メモリカードをインストールしている場合、メモリカードの最大容量をユーザー定義 Web ページ用の外部ロードメモリスペースとして使用できます。

メモリカードをインストールしていない場合、これらのブロックは内部ロードメモリスペースを占有します。これは使用する CPU モデルに応じて制約があります。

使用されているロードメモリスペースの量と、オンラインおよび診断ツールで使用可能な量を STEP 7 でチェックすることができます。また、STEP 7 がユーザー定義 Web ページから生成する個々のブロックのプロパティと、ロードメモリ消費量を調べることができます。

注記

ユーザー定義 Web ページに必要なスペースを減らす必要がある場合は、イメージの使用を適宜、減らしてください。

テキスト文字列内の引用符

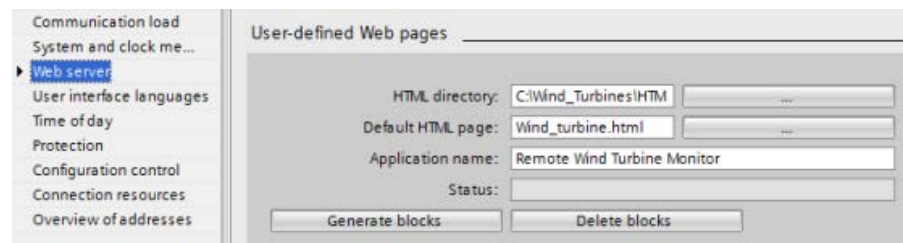
ユーザー定義 Web ページでは、どのような目的で使用するにしても、データブロックタグに一重引用符または二重引用符が埋め込まれているテキスト文字列を使用しないでください。HTML 構文は区切り文字として一重引用符または二重引用符を使用することが多いため、テキスト文字列内に引用符があるとユーザー定義 Web ページの表示が中断されるおそれがあります。

ユーザー定義 Web ページで使用する String タイプのデータブロックタグの場合、以下の規則を守ってください。

- STEP 7 でデータブロックタグのストリング値に一重引用符または二重引用符を入力しない。
- ユーザープログラムで引用符を含む文字列を上記のデータブロックタグに割り当てない。

9.3.3 ユーザー定義ウェブページの設定

ユーザー定義 Web ページを設定するには、CPU の[Web サーバー]プロパティを編集します。



Web サーバーの機能を有効にしたら、以下の情報を入力します。

- ユーザー定義 Web ページの DB を生成するための、HTML デフォルト開始ページの名前と現在のロケーション。
- 使用しているアプリケーションの名前(オプション)。アプリケーションの名前は、Web ページをさらに細かく分類したりグループ化するのに使用します。アプリケーション名を提供すると、Web サーバーは以下のフォーマットでユーザー定義ページの URL を作成します。


http[s]://ww.xx.yy.zz/awp/<アプリケーション名>/<ページ名>.html
- AWP コマンドの入っているファイルのファイル名拡張子。デフォルトでは、STEP 7 は.htm、.html、または.js 拡張子のファイルを分析します。他のファイル拡張子がある場合は、それを追加します。
- 制御 DB 番号と最初のフラグメント DB の識別番号。

Web サーバーを設定したら、[ブロックの生成]ボタンをクリックして、HTML ページから DB を生成します。DB を生成すると、Web ページがユーザープログラムの一部になります。Web ページを操作するための制御データブロックと「フラグメント」DB には、すべての HTML ページが含まれています。

9.3.4 WWW命令の使用

WWW 命令を使用して、ユーザー定義 Web ページに標準 Web ページからアクセスすることができます。ユーザー定義 Web ページにアクセスできるようにするには、ユーザープログラムが WWW 命令を一回実行するだけですみます。ただし、特定の状況でのみユーザー定義 Web ページが使用可能になることを選択してください。そうすれば、ユーザープログラムはアプリケーション要件に従って WWW 命令を呼び出すことができます。

表 9- 1 WWW 命令

LAD / FBD	SCL	説明
	<pre>ret_val := #WWW(ctrl_db:=_uint_in_);</pre>	ユーザー定義 Web ページに使用する制御 DB を識別します。 制御データブロックは WWW 命令の入力パラメータで、ページの内容をフラグメントデータブロックならびに状態および制御情報で示されているとおりに指定します。

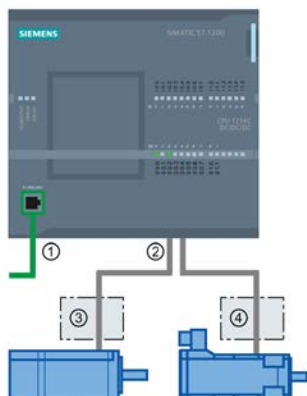
ユーザープログラムは通常、制御 DB を「ブロックの生成」プロセスで作成されたとおりに使用し、余分な操作は行いません。ただし、ユーザープログラムは制御 DB にグローバルコマンドを設定し、Web サーバーを無効にしたり、その後に再起動することができます。また、手動フラグメント DB として作成するユーザー定義ページの場合、ユーザープログラムは『S7-1200 プログラマブルコントローラシステムマニュアル』に記述されているように、ページの動作を制御 DB のリクエストテーブルを介して制御する必要があります。

CPU は、パルスインターフェースによってステッピングモータおよびサーボモータを操作するモーションコントロール機能を提供します。このモーションコントロール機能は、ドライブのコントロールとモニタリングを取り扱います。

- 「軸」テクノロジーオブジェクトを使用して、機械的なドライブデータ、ドライブインターフェース、動的パラメータ、およびその他のドライブプロパティを設定できます。
- ドライブを制御するための CPU のパルスおよび方向出力を設定します。
- ユーザープログラムで、軸を制御し、モーションタスクを開始するモーションコントロール命令を使用します。
- PROFINET インターフェースを使用して、CPU とプログラミングデバイスの間のオンライン接続を確立します。モーションコントロールでは、CPU のオンラインファンクションに加えて、コミッショニングおよび診断ファンクションも使用できます。

注記

RUN モードでのモーションコントロール設定の変更とダウンロードは、CPU が STOP から RUN モードに移行するまで有効にはなりません。



- ① PROFINET
- ② パルスおよび方向出力
- ③ ステッピングモータ用電源セクション
- ④ サーボモータ用電源セクション

CPU S7-1200 の DC/DC/DC バリエーションには、ドライブの直接コントロール用のオンボード出力があります。CPU のリレーバリエーションは、ドライブコントロール用の DC 出力付きのシグナルボードを必要とします。

シグナルボード(SB)はオンボード I/O を拡張して、数個の追加 I/O ポイントを含めます。2 つのデジタル出力付きの SB は、1 つのモータを制御するパルスおよび方向出力として使用できます。4 つのデジタル出力付きの SB は、2 つのモータを制御するパルスおよび方向出力として使用できます。内蔵リレー出力は、モータを制御するパルス出力としては使用できません。オンボード I/O、SB I/O、またはそれらの両方を使用するかどうかに関わりなく、最大 4 つのパルスジェネレータを持つことができます。

4つのパルスジェネレータにはデフォルトの I/O 割り当てがあります; ただし、CPU または SB の他のどのようなデジタル出力としても構成することができます。CPU 上のパルスジェネレータを SM またはリモート I/O に割り当ててはできません。

注記

パルス列出力は、ユーザープログラムの他の命令によって使用することはできません。

CPU またはシグナルボードの出力をパルスジェネレータとして設定すると(PWM またはモーションコントロール命令での使用のために)、対応出力アドレスはもう出力を制御しません。ユーザープログラムがパルスジェネレータとして使用されている出力に値を書き込むと、その CPU はこの値を物理出力に書き込みません。

表 10-1 制御可能なドライブの最大数

CPU のタイプ		オンボード I/O; SB が設置されていません		SB 付き (2 x DC 出力)		SB 付き (4 x DC 出力)	
		方向あり	方向なし	方向あり	方向なし	方向あり	方向なし
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

注記

パルスジェネレータの最大数は 4 です。

オンボード I/O、SB I/O、またはそれらの両方を使用するかどうかに関わりなく、最大 4 つのパルスジェネレータを持つことができます。

表 10-2 CPU 出力: 最大周波数

CPU	CPU 出力チャンネル	パルスおよび方向出力	A/B、カドラチャ、アップ/ダウン、およびパルス/方向
1211C	Qa.0~Qa.3	100 kHz	100 kHz
1212C	Qa.0~Qa.3	100 kHz	100 kHz

CPU	CPU 出力チャンネル	パルスおよび方向出力	A/B、カドラチャ、アップ/ダウン、およびパルス/方向
	Qa.4、Qa.5	20 kHz	20 kHz
1214C および 1215C	Qa.0～Qa.3	100kHz	100kHz
	Qa.4～Qb.1	20 kHz	20 kHz
1217C	DQa.0～DQa.3 (.0+、.0-～.3+、.3-)	1 MHz	1 MHz
	DQa.4～DQb.1	100 kHz	100 kHz

表 10-3 SB シグナルボード出力: 最大周波数(オプションボード)

SB シグナルボード	SB 出力チャンネル	パルスおよび方向出力	A/B、カドラチャ、アップ/ダウン、およびパルス/方向
SB 1222、200 kHz	DQe.0～DQe.3	200kHz	200 kHz
SB 1223、200 kHz	DQe.0、DQe.1	200kHz	200 kHz
SB 1223	DQe.0、DQe.1	20 kHz	20 kHz

表 10-4 パルス出力の周波数限界値

パルス出力	周波数
オンボード	4 PTO: $2 \text{ Hz} \leq f \leq 1 \text{ MHz}$ 、4 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ 、または 4 PTO に関するこれらの値の任意の組み合わせ。 ¹²
標準 SB	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
高速 SB	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$

¹ CPU 1217C の場合の 4 つの可能な出力速度の組み合わせについては、下の表を参照してください。

² CPU 1211C、CPU 1212C、CPU 1214C、または CPU 1215C の場合の 4 つの可能な出力速度の組み合わせについては、下の表を参照してください。

例: CPU 1217C パルス出力速度の設定

注記

CPU 1217C では、オンボード差動出力を使用して、最大 1 MHz のパルス出力を生成できます。

下の例に、4 つの可能な出力速度の組み合わせを示します。

- 例 1: 4 つの 1 MHz PTO、方向出力なし
- 例 2: 1 つの 1 MHz、2 つの 100 kHz、および 1 つの 20 kHz PTO、すべてが方向出力付き
- 例 3: 4 つの 200 kHz PTO、方向出力なし
- 例 4: 2 つの 100 kHz PTO および 2 つの 200 kHz PTO、すべてが方向出力付き

P = パルス D = 方向		CPU オンボード出力										高速 SB 出力				標準 SB 出力	
		1 MHz 出力(Q)				100 kHz 出力(Q)						200 kHz 出力(Q)				20 kHz 出力(Q)	
		0.0+	0.1+	0.2+	0.3+	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		0.0-	0.1-	0.2-	0.3-												
例 1: 4 つの 1 MHz (方向出力なし)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
例 2: 1 つの 1 MHz; 2 つの 100 kHz および 1 つの 20 kHz (すべて が方向出力付き)	PTO1	P	D														
	PTO2					P	D										
	PTO3							P	D								
	PTO4															P	D
例 3: 4 つの 200kHz (方向出力なし)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
例 4: 2 つの 100kHz; 2 つの 200kHz (すべて が方向出力付き)	PTO1					P	D										
	PTO2							P	D								
	PTO3											P	D				
	PTO4													P	D		

例: CPU 1211C、CPU 1212C、CPU 1214C、および CPU 1215C のパルス出力速度の設定

下の例に、4 つの可能な出力速度の組み合わせを示します。

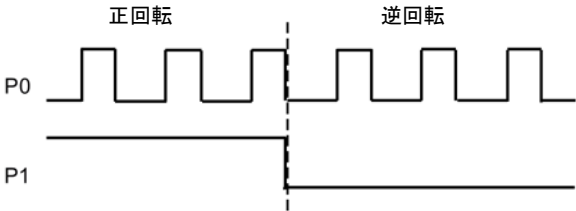
- 例 1: 4 つの 100 kHz PTO、方向出力なし
- 例 2: 2 つの 100 kHz PTO および 2 つの 20 kHz PTO、すべてが方向出力付き
- 例 3: 4 つの 200 kHz PTO、方向出力なし
- 例 4: 2 つの 100 kHz PTO および 2 つの 200 kHz PTO、すべてが方向出力付き

P = パルス D = 方向		CPU オンボード出力										高速 SB 出力				低速 SB 出力	
		100 kHz 出力(Q)				20 kHz 出力(Q)						200 kHz 出力(Q)				20 kHz 出力(Q)	
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		CPU 1211C				CPU 1212C						CPU 1214C				CPU 1215C	
		CPU 1212C				CPU 1214C						CPU 1214C				CPU 1215C	
		CPU 1214C				CPU 1214C						CPU 1214C				CPU 1215C	
		CPU 1215C				CPU 1215C						CPU 1215C				CPU 1215C	
例 1: 4 つの 100 kHz (方向出力なし)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
例 2: 2 つの 100 kHz; 2 つの 20kHz (すべてが方向出力付き)	PTO1	P	D														
	PTO2			P	D												
	PTO3					P	D										
	PTO4							P	D								
例 3: 4 つの 200kHz (方向出力なし)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
例 4: 2 つの 100kHz; 2 つの 200kHz (すべてが方向出力付き)	PTO1	P	D														
	PTO2			P	D												
	PTO3											P	D				
	PTO4													P	D		

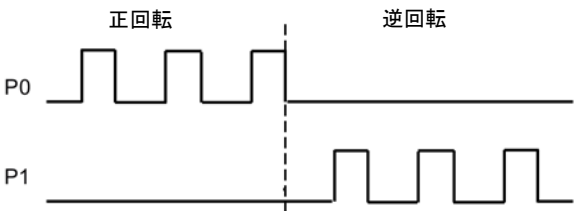
10.1 位相調整

ステッパ/サーボドライブの「位相調整」インターフェースには 4 つのオプションがあります。これらのオプションを以下に示します。

- PTO (パルス A および方向 B): PTO (パルス A および方向 B)オプションを選択すると、1 つの出力(P0)がパルス生成を制御し、1 つの出力(P1)が方向を制御します。パルス生成が正の方向である場合、P1 は高(アクティブ)です。パルス生成が負の方向である場合、P1 は低(非アクティブ)です。



- PTO (カウントアップ A およびカウントダウン B): PTO (カウントアップ A およびカウントダウン B)オプションを選択すると、1 つの出力(P0)が正の方向のパルスを生成し、もう 1 つの出力(P1)が負の方向のパルスを生成します。



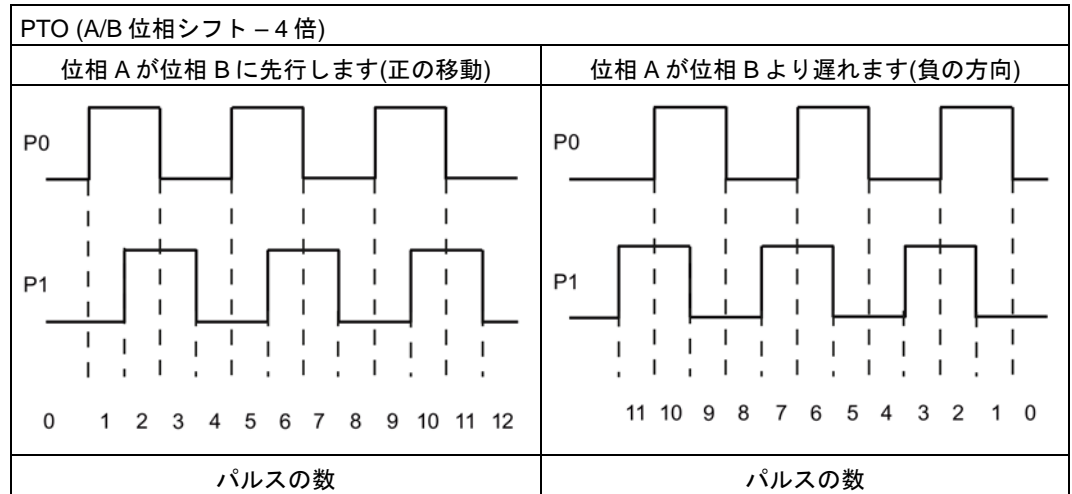
- PTO (A/B 位相シフト): PTO (A/B 位相シフト)オプションを選択すると、両方の出力が指定された速度でパルスを生成しますが、位相が 90 度異なります。それは 1X 設定で、1 つのパルスが、P0 の立ち上がり遷移間の時間です。この場合、方向は最初に高になる出力によって決まります。正の方向では、P0 が P1 に先行します。負の方向では、P1 が P0 に先行します。

生成されるパルス数は、位相 A の 0~1 への遷移数によって決まります。位相関係が移動の方向を決定します。

PTO (A/B フェースシフト)	
位相 A が位相 B に先行します(正の移動)	位相 A が位相 B より遅れます(負の方向)
パルスの数	パルスの数

- PTO (A/B 位相シフト - 4 倍): PTO (A/B 位相シフト - 4 倍)オプションを選択すると、両方の出力が指定された速度でパルスを生成しますが、位相が 90 度異なります。4 重は 4X 設定で、1 つのパルスは各出力の遷移(立ち上がりと立ち下がり)の両方です。この場合、方向は最初に高になる出力によって決まります。正の方向では、P0 が P1 に先行します。負の方向では、P1 が P0 に先行します。

4 重は、位相 A と位相 B の両方の立ち上がりおよび立ち下がり遷移に基づきます。遷移の数を設定します。位相関係(A が B に先行するか、または B が A に先行する)が移動の方向を決定します。



- PTO (パルスおよび方向(方向は選択解除)): PTO (パルスおよび方向(方向は選択解除))で方向出力を選択解除すると、出力(P0)がパルス生成を制御します。出力 P1 は使用されず、プログラムでの他の使用のために利用できます。このモードでは、正のモーションコマンドだけが CPU によって受け入れられます。このモードを選択すると、無効な正の設定を行うことがモーションコントロールによって防止されます。モーション用途が一方向のみである場合、1 つの出力を節約できます。単一の位相(1 つの出力)を下の図に示します(正の極性を仮定しています)。



10.2 パルスジェネレータの構成

1. テクノロジーオブジェクトの追加:

- プロジェクトツリーで、[テクノロジーオブジェクト]を展開し、[新しいオブジェクトの追加]を選択します。
- 「軸」直下の[TO_PositioningAxis]を選択し(必要に応じて、名前を変更し)、[OK]をクリックして軸オブジェクトの構成エディタを開きます。
- [基本パラメータ]のプロパティを表示し、[パルスジェネレータの選択]にて必要なパルスを選択します。

注記

CPU プロパティで PTO が設定済みでない場合は、オンボード出力の 1 つを PTO に設定します。

出力シグナルボードを使用する場合は、[デバイス構成]ボタンを選択して CPU プロパティにジャンプします。[パラメータ割り当て]の[ハードウェア出力]で、シグナルボード出力に設定します。

- 残りの基本および拡張パラメータを設定します。

2. アプリケーションプログラミング: ブロックに MC_Power 命令を挿入します。

- 軸入力のために、作成し、構成した軸テクノロジーオブジェクトを選択します。
- イネーブル入力を TRUE に設定すると、他のすべてのモーション命令を使用できます。
- イネーブル入力を FALSE に設定すると、他のすべてのモーション命令がキャンセルされます。

注記

軸ごとに 1 つの MC_Power 命令が必要です。

3. 必要なモーションを生成するために、他のすべてのモーション命令を挿入します。

注記

シグナルボード出力に対する 1 つのパルスジェネレータの構成: CPU の[パルスジェネレータ(PTO/PWM)]プロパティを選択し(デバイス構成で)、1 つのパルスジェネレータを有効にします。S7-1200 CPU V1.0、V2.0、V2.1、および V2.2 ごとに、2 つのパルスジェネレータが使用できます。S7-1200 CPU V3.0 および V4.0 CPU では、4 つのパルスジェネレータを使用できます。[パルスオプション]のこの同一の構成領域で、「PTO」として使用するパルスジェネレータを選択します。

注記

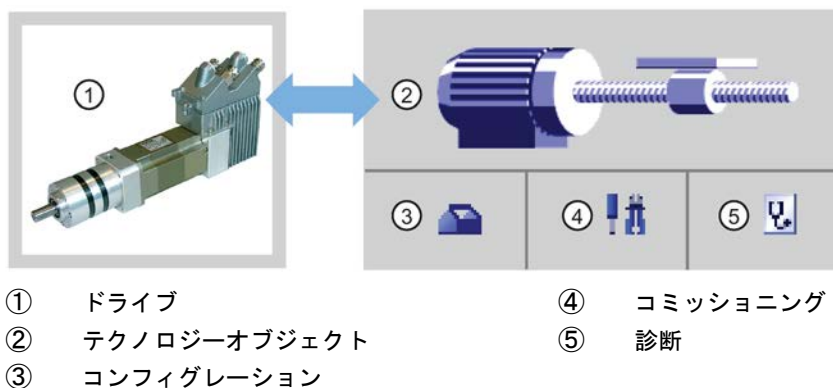
CPU は、モーションタスクを「スライス」またはセグメント(10 ミリ秒)単位で計算します。1 つのスライスが実行されているときは、次のスライスがキュー内で実行を待機します。1 つの軸のモーションタスクを中断すると(その軸に対して別の新しいモーションタスクを実行することによって)、その新しいモーションタスクは最大 20 ミリ秒間(現在のスライスの残りの時間 + キューインぐされたスライス)実行できません。

10.3 開ループモーションコントロール

10.3.1 軸の構成

PLC 上の開ループ軸を 1 つの PTO (パルス列出力) 経由でドライブに接続します。

STEP 7 は、「軸」テクノロジーオブジェクト用の構成ツール、コミッシングツール、および診断ツールを提供します。



注記

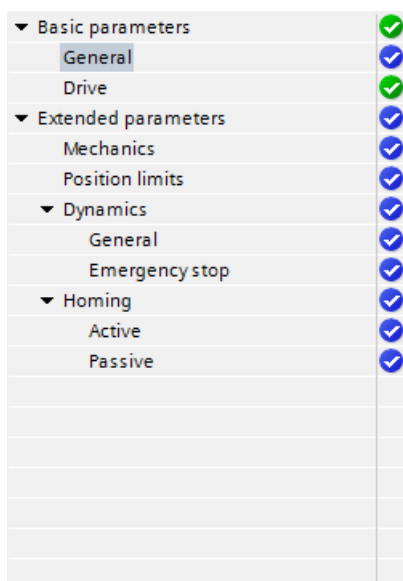
CPU ファームウェアリリース V2.2 以前の場合、PTO は、高速カウンタ(HSC)内部機能を必要とします。これは、該当 HSC を他の場所で使用できないことを意味します。

PTO と HSC の間の割り当ては固定されています。有効化された PTO1 は、HSC1 に接続されます。有効化された PTO2 は、HSC2 に接続されます。パルスが生成されているとき、現在値(たとえば、ID1000 内の現在値)をモニタできません。

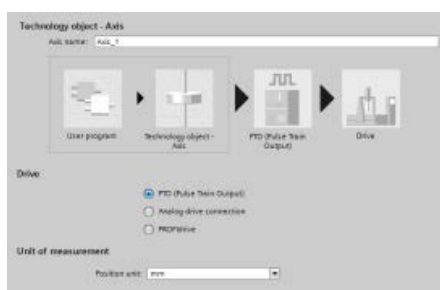
S7-1200 V3.0 以降の CPU ではこの制限はありません; これらの CPU でパルス出力が設定済みのとき、プログラムでの使用のために、すべての HSC が利用できます。

表 10-5 モーションコントロール用 STEP 7 ツール

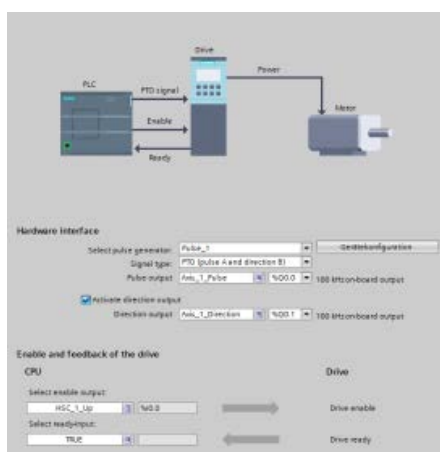
ツール	説明
構成	<p>「軸」テクノロジーオブジェクトの以下のプロパティを設定します。</p> <ul style="list-style-type: none"> 使用する PTO の選択とドライバインターフェースの構成 ドライバ(またはマシン/システム)の構造および伝達比率のプロパティ 位置制限値、ダイナミクス、および原点復帰のプロパティ <p>その構成をテクノロジーオブジェクトのデータブロックに保存します。</p>
コミッショニング	<p>軸のファンクションをテストします。この場合、ユーザープログラムを作成する必要はありません。このツールを開始すると、コントロールパネルが表示されます。コントロールパネルでは以下のコマンドを使用できます。</p> <ul style="list-style-type: none"> 軸の有効化/無効化 ジョグモードでの軸の移動 軸の絶対的および相対的な位置決め 軸の原点復帰 エラーの確認 <p>モーションコマンドでは、速度および加速/減速を指定できます。コントロールパネルには現在の軸ステータスも表示されます。</p>
診断	<p>軸およびドライバの現在のステータスとエラー情報をモニタします。</p>



PTO 軸のツリーセレクトには、エンコーダ、モジュロ、位置モニタ、およびコントロールループの構成メニューは含まれていません。

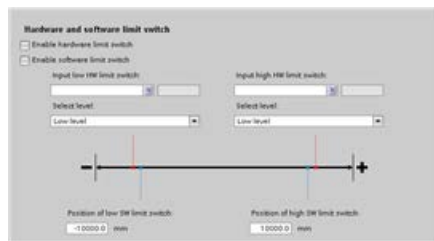


軸のテクノロジーオブジェクトを作成した後、PTO およびドライブインターフェースの構成などの基本パラメータを定義して軸を構成します。さらに、位置制限値、ダイナミクス、および原点復帰などの軸のその他のプロパティを設定します。

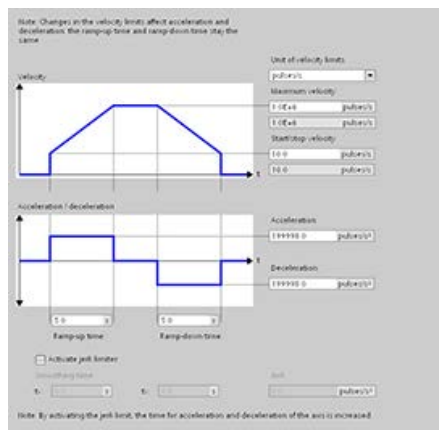


注記

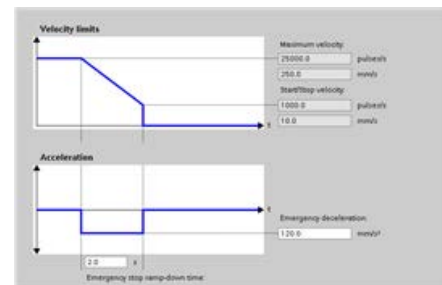
ユーザープログラム内の新しい寸法単位に合わせたモーションコントロール命令の入力パラメータ値の調整が必要になる場合があります。



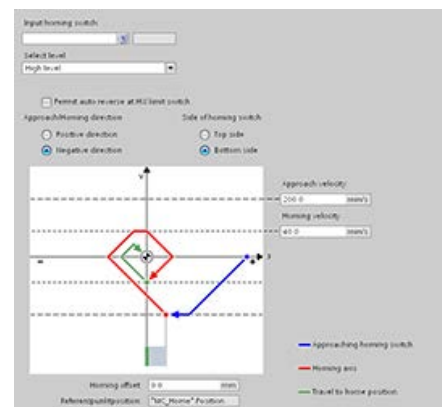
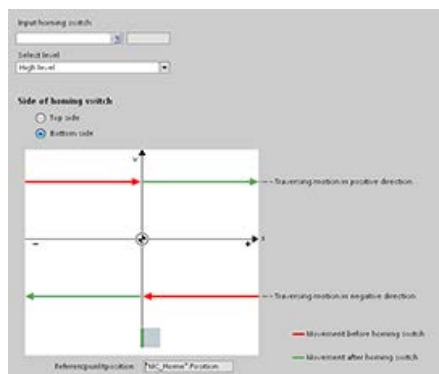
ドライブ信号、ドライブメカニクス、および位置モニタ(ハードウェアおよびソフトウェアリミットスイッチ)のプロパティを設定します。




モーションダイナミクスと、緊急停止コマンドの動作を設定します。



さらに、原点復帰動作(パッシブおよびアクティブ)を設定します。



[コミッシング]コントロールパネルを使用して、ユーザープログラムとは独立して、その機能をテストします。

 [コミッシング]アイコンをクリックして、軸のコミッシングを行います。

このコントロールパネルには、軸の現在のステータスが表示されます。軸を有効化/無効化するだけでなく、軸の位置決め(絶対値と相対値の両方)もテストでき、速度、加速、および減速を指定できます。さらに、原点復帰およびジョギングタスクをテストできます。このコントロールパネルを使用して、エラーを確認することもできます。

10.3.2 コミッショニング

[ステータスおよびエラービット]診断ファンクション

[ステータスおよびエラービット]診断ファンクションを使用して、軸の最も重要なステータスとエラーメッセージをモニタします。診断ファンクション表示は、軸が有効であるとき、オンラインモードの「手動コントロール」および「自動コントロール」モードで使用できます。

表 10-6 軸のステータス

ステータス	説明
有効	軸が有効で、モーションコントロールタスクによって制御される準備が完了しています。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Enable)
原点復帰済み	軸が原点復帰済みで、モーションコントロール命令「MC_MoveAbsolute」の絶対位置決めタスクを実行できます。相対原点復帰の場合、軸を原点復帰する必要はありません。特別な状況: <ul style="list-style-type: none"> アクティブ原点復帰中は、ステータスは FALSE です。 原点復帰済み軸がパッシブ原点復帰を受けると、パッシブ原点復帰中にステータスが FALSE にセットされます。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.HomingDone)
軸エラー	「軸」テクノロジーオブジェクトでエラーが発生しました。自動コントロールでは、エラーに関する詳細は、モーションコントロール命令の ErrorID および ErrorInfo パラメータで入手できます。手動モードでは、コントロールパネルの[最後のエラー]フィールドにエラーの原因に関する詳細が表示されます。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Error)
コントロールパネルが有効です	コントロールパネルで「手動コントロール」モードが有効化されました。コントロールパネルが、「軸」テクノロジーオブジェクトに対するコントロール優先度を持ちます。ユーザープログラムから軸を制御することはできません。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.ControlPanelActive)

表 10-7 ドライブのステータス

ステータス	説明
準備完了	ドライブの動作の準備が完了しています。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.DriveReady)
ドライブエラー	ドライブが、その準備完了信号の異常の後、エラーを報告しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.DriveFault)

表 10- 8 軸モーションのステータス

ステータス	説明
停止	軸が停止しています。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.StandStill)
加速	軸が加速します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Acceleration)
一定速度	軸が一定速度で移動します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.ConstantVelocity)
減速	軸が減速します(遅くなります)。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Deceleration)

表 10- 9 モーションタイプのステータス

ステータス	説明
位置決め	軸は、モーションコントロール命令「MC_MoveAbsolute」または「MC_MoveRelative」の位置決めタスクを実行するか、コントロールパネルの位置決めタスクを実行します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.PositioningCommand)
あらかじめ定義された速度で移動	軸は、モーションコントロール命令「MC_MoveVelocity」または「MC_MoveJog」の設定された速度でタスクを実行するか、コントロールパネルの設定された速度でタスクを実行します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.SpeedCommand)
原点復帰が有効です	軸は、モーションコントロール命令「MC_Home」の原点復帰タスクを実行するか、コントロールパネルの原点復帰タスクを実行します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Homing)

表 10- 10 エラービット

エラー	説明
下限 SW リミットスイッチがアプローチされました	下限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SWLimitMinActive)
上限 SW リミットスイッチがアプローチされました	上限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SWLimitMaxActive)
下限 HW リミットスイッチがアプローチされました	下限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HWLLimitMinActive)
上限 HW リミットスイッチがアプローチされました	上限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HWLLimitMaxActive)
SW リミットスイッチがアプローチされました	ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SWLimit)
HW リミットスイッチがアプローチされました	ハードウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HWLLimit)
PTO が既に使用中です	2 番目の軸が同一の PTO を使用していて、「MC_Power」で有効化されています。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HwUsed)

エラー	説明
構成エラー	「軸」テクノロジーオブジェクトが不正に構成されたか、ユーザープログラムのランタイム中に編集可能な構成データが不正に変更されました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.ConfigFault)
内部エラー	内部エラーが発生しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SystemFault)

「モーションステータス」診断ファンクション

「モーションステータス」診断ファンクションを使用して、軸のモーションステータスをモニタします。診断ファンクション表示は、軸が有効であるとき、オンラインモードの「手動コントロール」および「自動コントロール」モードで使用できます。

表 10-11 モーションステータス

ステータス	説明
ターゲット位置	[ターゲット位置]フィールドは、モーションコントロール命令「MC_MoveAbsolute」または「MC_MoveRelative」の有効な位置決めタスクか、またはコントロールパネルの有効な位置決めタスクの現在のターゲット位置を示します。「ターゲット位置」の値は、位置決めタスクの実行中にのみ有効です。 (テクノロジーオブジェクトのタグ: <軸名>.MotionStatus.TargetPosition)
現在の位置	「現在の位置」フィールドは、現在の軸位置を示します。軸が原点復帰しない場合は、その値は、軸のイネーブル位置に対して相対的な位置の値を示します。 (テクノロジーオブジェクトのタグ: <軸名>.MotionStatus.Position)
現在の速度	[現在の速度]フィールドは、現在の軸速度を示します。 (テクノロジーオブジェクトのタグ: <軸名>.MotionStatus.Velocity)

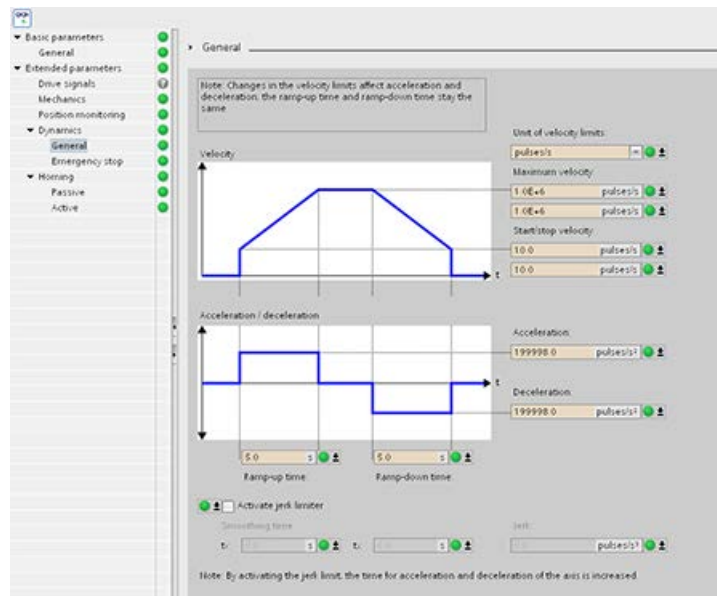
表 10-12 ダイナミクス設定

動的な制限値	説明
速度	[速度]フィールドは、軸の設定されている最大速度を示します。 (テクノロジーオブジェクトのタグ: <軸名>.Config.DynamicLimits.MaxVelocity)
加速	[加速]フィールドは、軸の現在設定されている加速を示します。 (テクノロジーオブジェクトのタグ: <軸名>.Config.DynamicDefaults.Acceleration)
減速	[減速]フィールドは、軸の現在設定されている減速を示します。 (テクノロジーオブジェクトのタグ: <軸名>.Config.DynamicDefaults.Deceleration)

モーション開始値コントロール

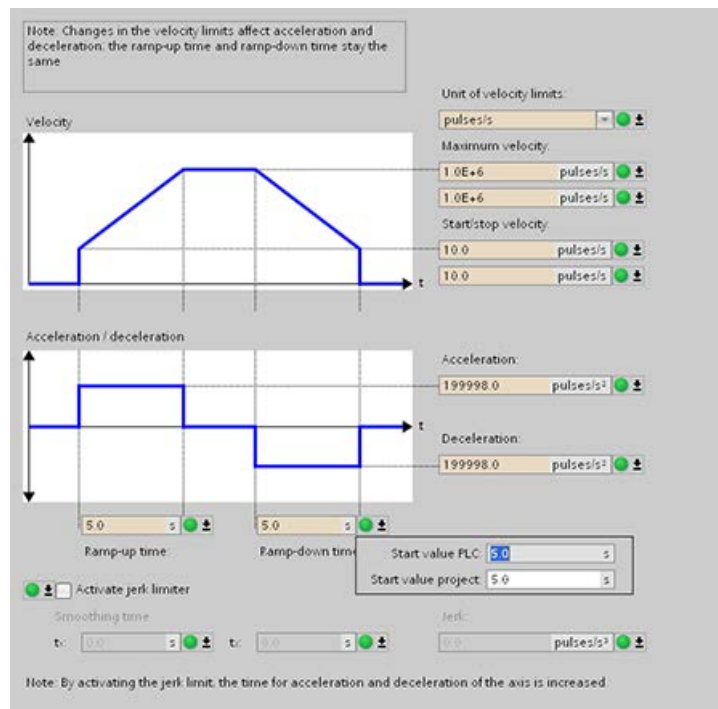
オンラインモードでプロセスの動作を最適化できるように、モーション構成パラメータの現在値を編集できます。

モーションコントロールの「テクノロジーオブジェクト」とその「構成」を開きます。開始値の制御にアクセスするには、ダイアログの左上隅にある[眼鏡アイコン]をクリックします。



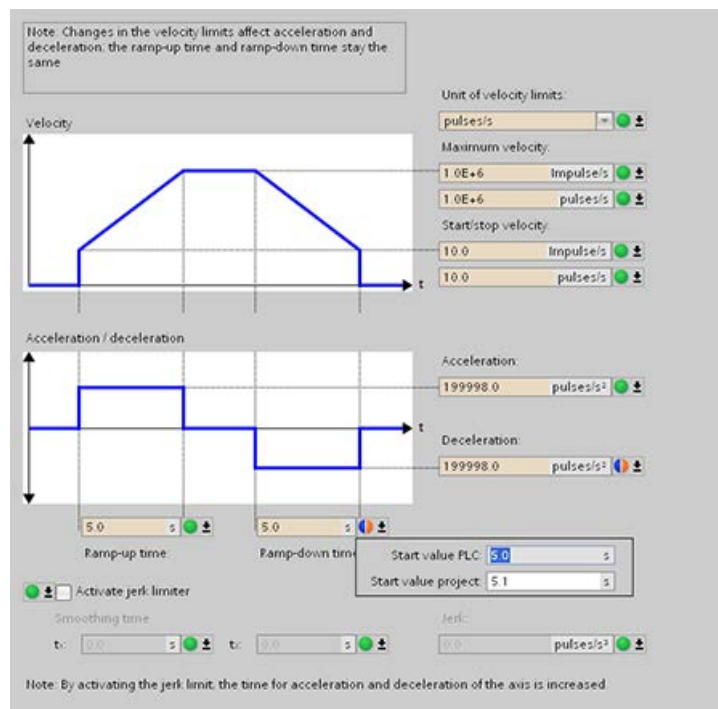
この後、下の図に示すように、任意のモーションコントロール構成パラメータの値を変更できます。

現在値を各パラメータのプロジェクト(オフライン)開始値および PLC(オンライン)開始値と比較できます。これは、テクノロジーオブジェクトデータブロック(TO-DB)のオンライン/オフライン差分を比較し、PLCの次の Stop から Start への移行で現在値として使用される値を知るのに必要です。さらに、オンライン/オフラインを簡単に識別できる比較アイコンが表示されます。



上の図は、オンラインプロジェクトとオフラインプロジェクト間で相違する値を示す比較アイコン付きのモーションパラメータ画面です。緑のアイコンは値が同じであることを示しています。青/オレンジのアイコンは値が異なっていることを示しています。

さらに、下矢印のついたパラメータボタンをクリックすると小さいウィンドウが開き、各パラメータのプロジェクト(オフライン)開始値と PLC(オンライン)開始値が表示されます。



10.4 閉ループモーションコントロール

10.4.1 軸の構成

PLC 上の閉ループ軸とドライブを、アナログドライブまたは PROFIdrive 経由で接続します。閉ループ軸にはエンコーダも必要です。

STEP 7 は、「軸」テクノロジーオブジェクト用の構成ツール、コミッショニングツール、および診断ツールを提供します。

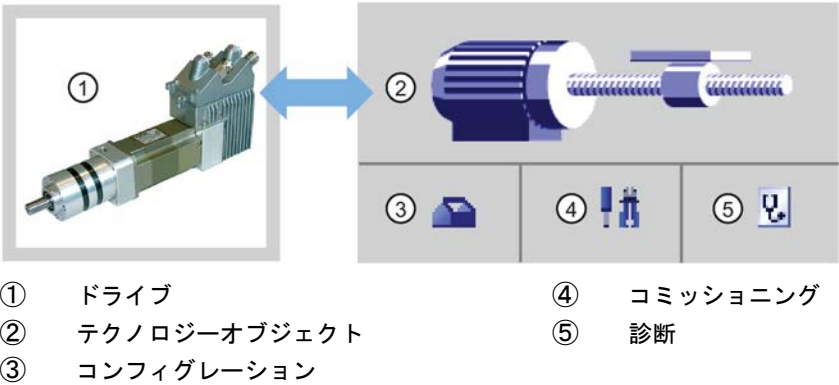


表 10- 13 閉ループモーションコントロール用 STEP 7 ツール

ツール	説明
構成	「軸」テクノロジーオブジェクトの以下のプロパティを設定します。 <ul style="list-style-type: none">使用するアナログドライブ接続または PROFIdrive の選択と、ドライブおよびエンコーダインターフェースの構成ドライブおよびエンコーダ(またはマシン/システム)の構造および伝達比率のプロパティ位置制限値、ダイナミクス、および原点復帰のプロパティ その構成をテクノロジーオブジェクトのデータブロックに保存します。
コミッショニング	軸のファンクションをテストします。この場合、ユーザープログラムを作成する必要はありません。このツールを開始すると、コントロールパネルが表示されます。コントロールパネルでは以下のコマンドを使用できます。 <ul style="list-style-type: none">軸の有効化/無効化ジョグモードでの軸の移動軸の絶対的および相対的な位置決め軸の原点復帰エラーの確認 モーションコマンドでは、速度および加速/減速を指定できます。コントロールパネルには現在の軸ステータスも表示されます。
診断	軸およびドライブの現在のステータスとエラー情報をモニタします。

注記

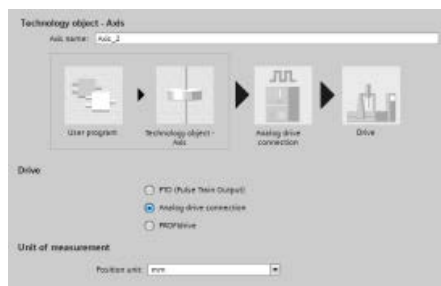
ユーザープログラム内の新しい寸法単位に合わせたモーションコントロール命令の入力パラメータ値の調整が必要になる場合があります。

軸のテクノロジーオブジェクトを作成した後、アナログドライブまたは PROFIdrive 接続、ドライブおよびエンコーダの構成などの基本パラメータを定義して軸を構成します。

▼ Basic parameters	✓
General	✓
Drive	✓
Encoder	✓
▼ Extended parameters	✓
Mechanics	✓
Modulo	✓
Position limits	✓
▼ Dynamics	✓
General	✓
Emergency stop	✓
▼ Homing	✓
Active	✓
Passive	✓
▼ Position monitoring	✓
Position monitoring	✓
Following error	✓
Standstill signal	✓
Control loop	✓

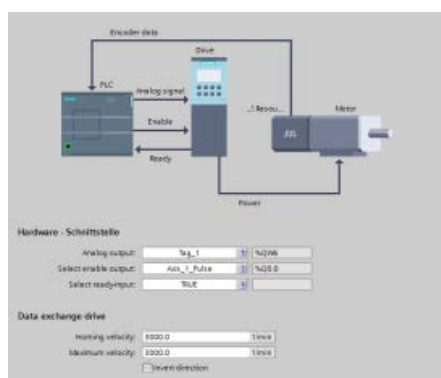
アナログドライブまたは PROFIdrive 接続のツリーセレクトには、エンコーダ、モジュロ、位置モニタ、およびコントロールループの構成メニューが含まれています。

アナログドライブ接続の構成



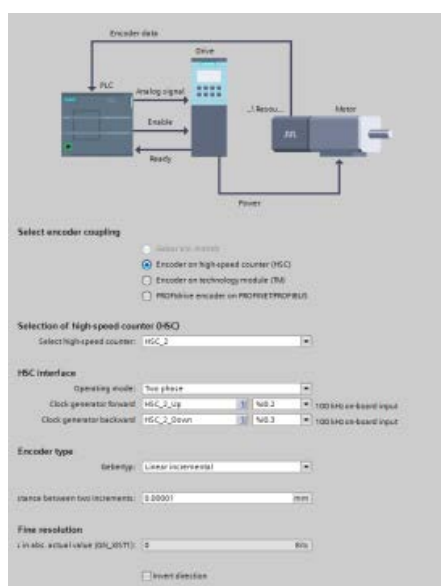
全般コンフィグレーションダイアログで、以下のパラメータを選択します。

- [アナログドライブ接続]ラジオボタン
- 測定単位



ドライブコンフィグレーションダイアログで、以下のパラメータを選択します。

- アナログドライブハードウェア出力
- データ交換ドライブ速度



エンコーダコンフィグレーションダイアログで、以下のパラメータを選択します。

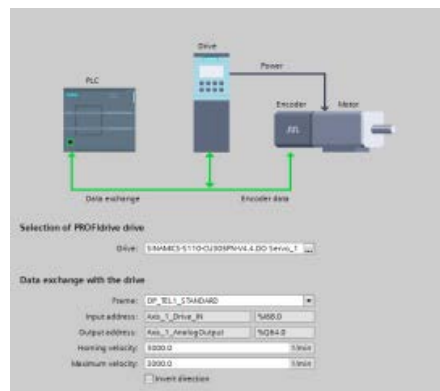
- アナログドライブエンコーダ結合(たとえば、高速カウンタ(HSC))
- HSC インターフェース
- エンコーダタイプ
- 分解能

PROFdrive の構成



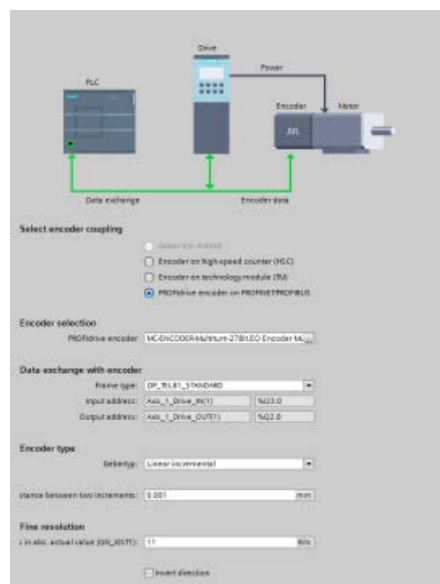
全般コンフィグレーションダイアログで、以下のパラメータを選択します。

- [PROFdrive]ラジオボタン
- 測定単位



ドライブコンフィグレーションダイアログで、以下のパラメータを選択します。

- PROFdrive ドライブの選択
- ドライブとのデータ交換



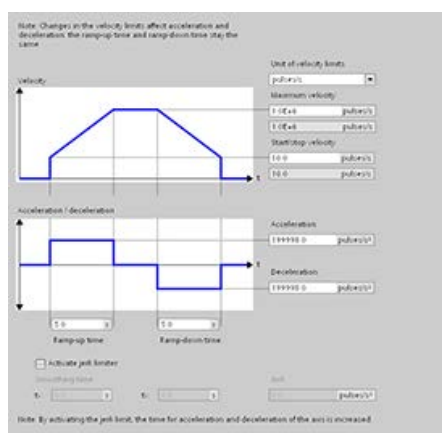
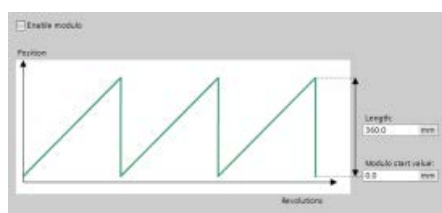
エンコーダコンフィグレーションダイアログで、以下のパラメータを選択します。

- PROFdrive エンコーダ結合(たとえば、PROFINET 上の PROFdrive エンコーダ)
- PROFdrive エンコーダ
- エンコーダとのデータ交換
- エンコーダタイプ
- 分解能

拡張パラメータ

さらに、閉ループ軸の以下のプロパティを設定することもできます。

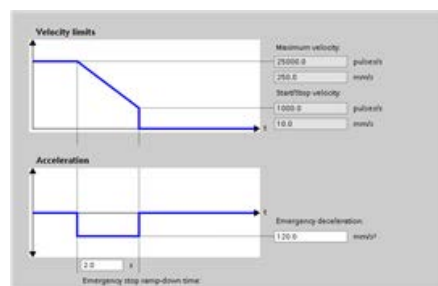
- モジユロ
- 位置制限値
- ダイナミクス
- 原点復帰
- 位置モニタ
- 追従誤差
- 停止信号
- コントロールループ



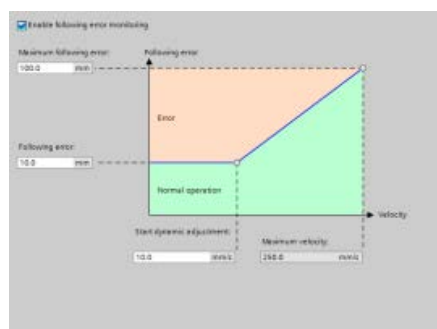
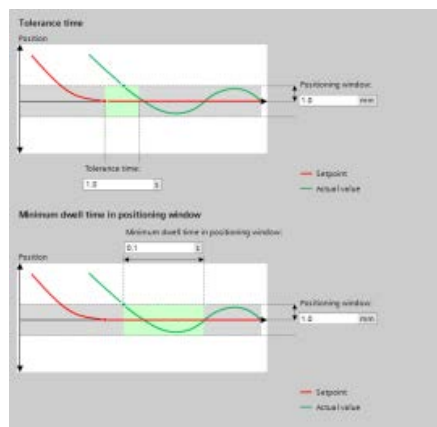
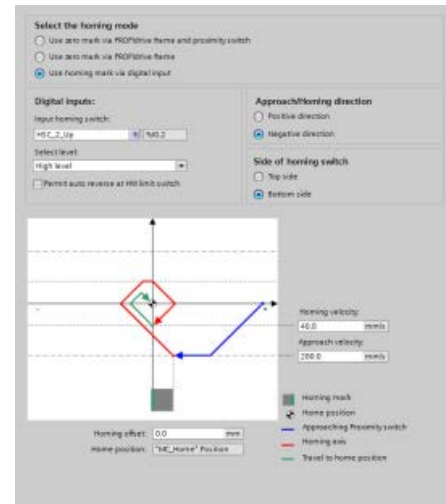
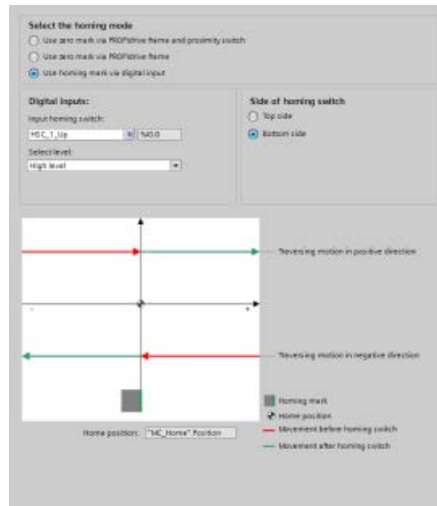
モジユロ: 「モジユロ」軸を構成して、開始値/開始位置と特定の長さを持つ繰り返し領域で負荷を移動できます。負荷の位置がこの領域の終わりに達すると、負荷の位置が自動的に開始値に再び設定されます。[モジユロの有効化]チェックボックスをチェックするとき、[長さ]および[モジユロ開始値]フィールドを有効にします。

位置制限値: ドライブ信号、ドライブメカニクス、および位置モニタ(ハードウェアおよびソフトウェアリミットスイッチ)のプロパティを設定できます。

ダイナミクス: モーションダイナミクスと、緊急停止コマンドの動作を設定できます。



原点復帰: 原点復帰動作(パッシブおよびアクティブ)を設定できます。



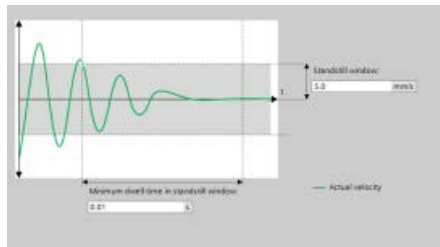
「位置決めモニタ」: 位置決めウィンドウの最大許容時間と最小滞留時間を設定できます。

システムは、以下の3つのパラメータを軸TO-DBと直接に接続します。

- 位置決めウィンドウ
- 最大許容時間
- 位置決めウィンドウでの最小滞留時間

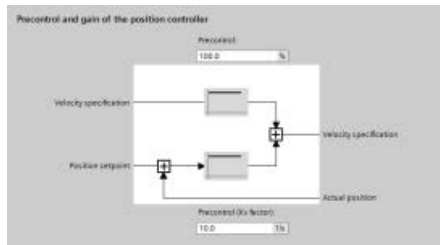
「追従誤差」: 一定の速度範囲での許可される誤差距離の相違を設定できます。[追従誤差モニタの有効化]チェックボックスをチェックして、追従誤差を有効化します。以下のパラメータを設定できます。

- 最大追従誤差
- 追従誤差
- 動的な調整の開始
- 最大速度




「停止信号」以下のパラメータを設定できます。

- 停止ウィンドウでの最小滞留時間
- 停止ウィンドウ



「コントロールループ」：「プリコントロール(Kv 係数)」と呼ばれる速度ゲインを設定できます。

[コミッショニング]コントロールパネルを使用して、ユーザープログラムとは独立して、その機能をテストします。

 [コミッショニング]アイコンをクリックして、軸のコミッショニングを行います。

このコントロールパネルには、軸の現在のステータスが表示されます。軸を有効化/無効化するだけでなく、軸の位置決め(絶対値と相対値の両方の)もテストでき、速度、加速、および減速を指定できます。さらに、原点復帰およびジョギングタスクをテストできます。このコントロールパネルを使用して、エラーを確認することもできます。

10.4.2 コミッショニング

[ステータスおよびエラービット]診断ファンクション

[ステータスおよびエラービット]診断ファンクションを使用して、軸の最も重要なステータスとエラーメッセージをモニタします。診断ファンクション表示は、軸が有効であるとき、オンラインモードの「手動コントロール」および「自動コントロール」モードで使用できます。

表 10- 14 軸のステータス

ステータス	説明
有効	軸が有効で、モーションコントロールタスクによって制御される準備が完了しています。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Enable)
原点復帰済み	軸が原点復帰済みで、モーションコントロール命令「MC_MoveAbsolute」の絶対位置決めタスクを実行できます。相対原点復帰の場合、軸を原点復帰する必要はありません。特別な状況: <ul style="list-style-type: none"> アクティブ原点復帰中は、ステータスは FALSE です。 原点復帰済み軸がパッシブ原点復帰を受けると、パッシブ原点復帰中にステータスが FALSE にセットされます。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.HomingDone)
軸エラー	「軸」テクノロジーオブジェクトでエラーが発生しました。自動コントロールでは、エラーに関する詳細は、モーションコントロール命令の ErrorID および ErrorInfo パラメータで入手できます。手動モードでは、コントロールパネルの[最後のエラー]フィールドにエラーの原因に関する詳細が表示されます。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Error)
コントロールパネルが有効です	コントロールパネルで「手動コントロール」モードが有効化されました。コントロールパネルが、「軸」テクノロジーオブジェクトに対するコントロール優先度を持ちます。ユーザープログラムから軸を制御することはできません。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.ControlPanelActive)

表 10- 15 ドライブのステータス

ステータス	説明
ドライブ準備完了	ドライブの動作の準備が完了しています。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.DriveReady)
ドライブエラー	ドライブが、その準備完了信号の異常の後、エラーを報告しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.DriveFault)

表 10- 16 軸モーションのステータス

ステータス	説明
停止	軸が停止しています。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.StandStill)
加速	軸が加速します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Acceleration)
一定速度	軸が一定速度で移動します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.ConstantVelocity)
減速	軸が減速します(遅くなります)。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Deceleration)

表 10- 17 モーションモードのステータス

ステータス	説明
位置決め	軸は、モーションコントロール命令「MC_MoveAbsolute」または「MC_MoveRelative」の位置決めタスクを実行するか、コントロールパネルの位置決めタスクを実行します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.PositioningCommand)
あらかじめ定義された速度で移動	軸は、モーションコントロール命令「MC_MoveVelocity」または「MC_MoveJog」の設定された速度でタスクを実行するか、コントロールパネルの設定された速度でタスクを実行します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.SpeedCommand)
原点復帰が有効です	軸は、モーションコントロール命令「MC_Home」の原点復帰タスクを実行するか、コントロールパネルの原点復帰タスクを実行します。 (テクノロジーオブジェクトのタグ: <軸名>.StatusBits.Homing)

表 10- 18 エラービット

エラー	説明
下限 SW リミットスイッチがアプローチされました	下限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SWLimitMinActive)
上限 SW リミットスイッチがアプローチされました	上限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SWLimitMaxActive)
下限 HW リミットスイッチがアプローチされました	下限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HWLLimitMinActive)
上限 HW リミットスイッチがアプローチされました	上限ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HWLLimitMaxActive)
SW リミットスイッチがアプローチされました	ソフトウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SWLimit)
HW リミットスイッチがアプローチされました	ハードウェアリミットスイッチに到達しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HWLLimit)
PTO が既に使用中です	2 番目の軸が同一の PTO を使用していて、「MC_Power」で有効化されています。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.HwUsed)

エラー	説明
構成エラー	「軸」テクノロジーオブジェクトが不正に構成されたか、ユーザープログラムのランタイム中に編集可能な構成データが不正に変更されました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.ConfigFault)
内部エラー	内部エラーが発生しました。 (テクノロジーオブジェクトのタグ: <軸名>.ErrorBits.SystemFault)

「モーションステータス」診断ファンクション

「モーションステータス」診断ファンクションを使用して、軸のモーションステータスをモニタします。診断ファンクション表示は、軸が有効であるとき、オンラインモードの「手動コントロール」および「自動コントロール」モードで使用できます。

表 10- 19 モーションステータス

ステータス	説明
ターゲット位置	[ターゲット位置]フィールドは、モーションコントロール命令「MC_MoveAbsolute」または「MC_MoveRelative」の有効な位置決めタスクか、またはコントロールパネルの有効な位置決めタスクの現在のターゲット位置を示します。「ターゲット位置」の値は、位置決めタスクの実行中にのみ有効です。 (テクノロジーオブジェクトのタグ: <軸名>.MotionStatus.TargetPosition)
現在の位置	「現在の位置」フィールドは、現在の軸位置を示します。軸が原点復帰しない場合は、その値は、軸のイネーブル位置に対して相対的な位置の値を示します。 (テクノロジーオブジェクトのタグ: <軸名>.MotionStatus.Position)
現在の速度	[現在の速度]フィールドは、現在の軸速度を示します。 (テクノロジーオブジェクトのタグ: <軸名>.MotionStatus.Velocity)

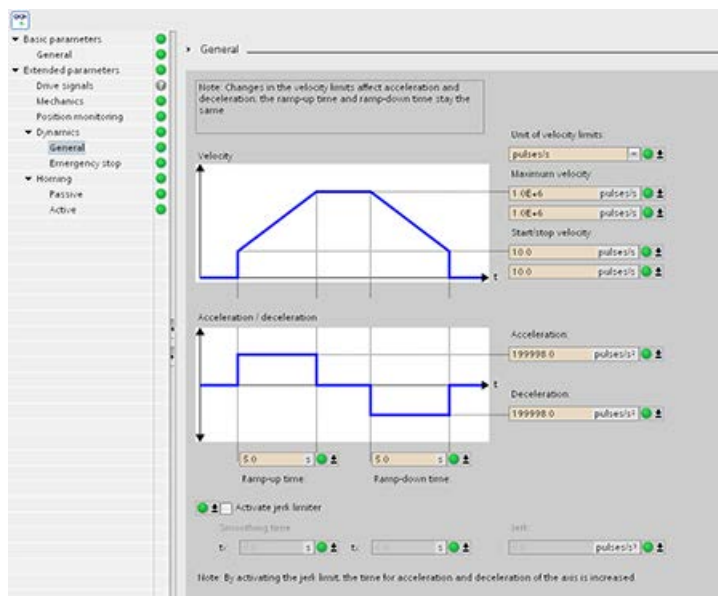
表 10- 20 ダイナミクス設定

動的な制限値	説明
速度	[速度]フィールドは、軸の設定されている最大速度を示します。 (テクノロジーオブジェクトのタグ: <軸名>.Config.DynamicLimits.MaxVelocity)
加速	[加速]フィールドは、軸の現在設定されている加速を示します。 (テクノロジーオブジェクトのタグ: <軸名>.Config.DynamicDefaults.Acceleration)
減速	[減速]フィールドは、軸の現在設定されている減速を示します。 (テクノロジーオブジェクトのタグ: <軸名>.Config.DynamicDefaults.Deceleration)

モーション開始値コントロール

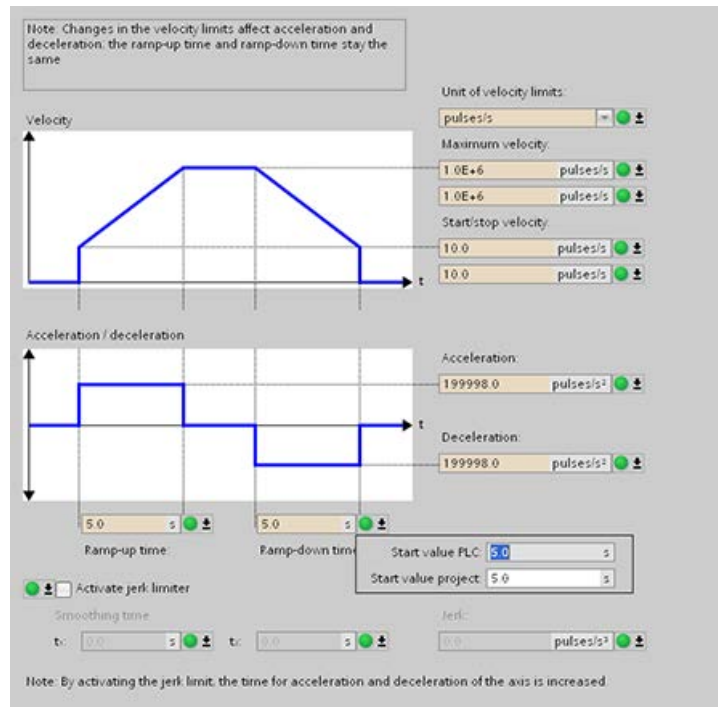
オンラインモードでプロセスの動作を最適化できるように、モーション構成パラメータの現在値を編集できます。

モーションコントロールの「テクノロジーオブジェクト」とその「構成」を開きます。開始値の制御にアクセスするには、ダイアログの左上隅にある[眼鏡アイコン]をクリックします。



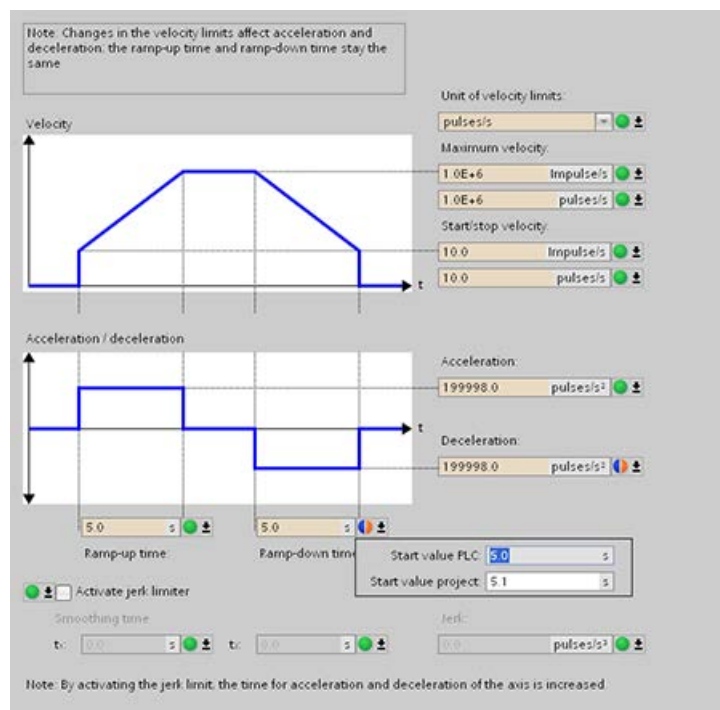
この後、下の図に示すように、任意のモーションコントロール構成パラメータの値を変更できます。

現在値を各パラメータのプロジェクト(オフライン)開始値および PLC(オンライン)開始値と比較できます。これは、テクノロジーオブジェクトデータブロック(TO-DB)のオンライン/オフライン差分を比較し、PLC の次の Stop から Start への移行で現在値として使用される値を知るのに必要です。さらに、オンライン/オフラインを簡単に識別できる比較アイコンが表示されます。



上の図は、オンラインプロジェクトとオフラインプロジェクト間で相違する値を示す比較アイコン付きのモーションパラメータ画面です。緑のアイコンは値が同じであることを示しています。青/オレンジのアイコンは値が異なっていることを示しています。

さらに、下矢印のついたパラメータボタンをクリックすると小さいウィンドウが開き、各パラメータのプロジェクト(オフライン)開始値と PLC(オンライン)開始値が表示されます。

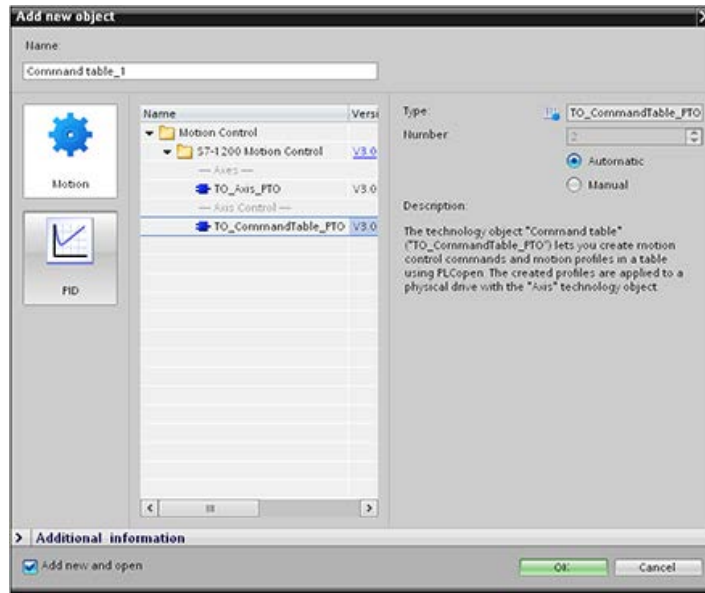


10.5 TO_CommandTable_PTO の構成

テクノロジーオブジェクトを使用して、MC_CommandTable 命令を設定できます。次の例に、この設定方法を示します。

1 つのテクノロジーオブジェクトの追加

1. プロジェクトツリーで、ノード[テクノロジーオブジェクト]を展開し、[新しいオブジェクトの追加]を選択します。
2. 「CommandTable」アイコンを選択し(必要に応じて、名前を変更し)、[OK]をクリックして CommandTable オブジェクトの構成エディタを開きます。



ユーザーの用途用ステップの計画

[コマンドテーブル]構成ウィンドウで必要な移動シーケンスを作成し、その結果をトレンドダイアグラムのグラフィックビューと比較してチェックします。

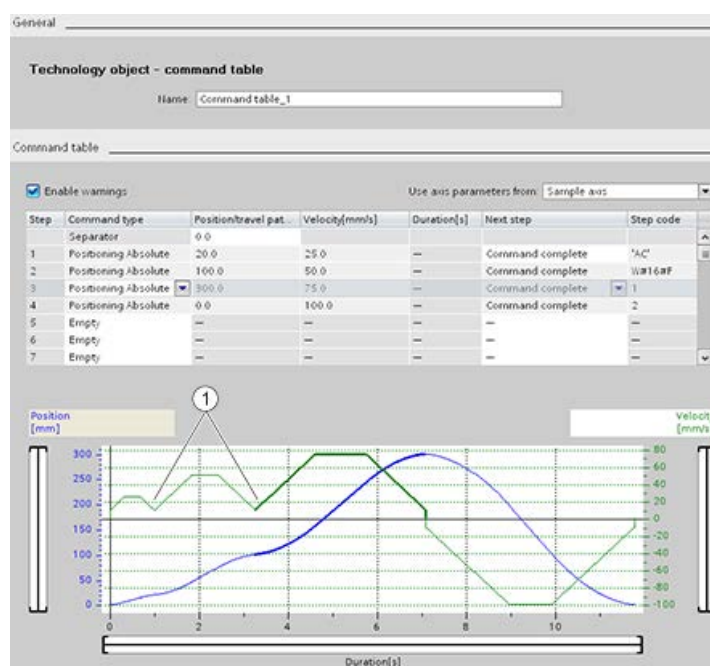
コマンドテーブルの処理で使用するコマンドタイプを選択できます。最大 32 のステップを入力できます。コマンドは順々に処理され、簡単に複雑なモーションプロファイルを生成します。

表 10- 21 MC_CommandTable コマンドタイプ

コマンドタイプ	説明
Empty	空タイプは、任意のコマンドを追加するプレースホルダとして使用されます。空エントリは、コマンドテーブルを処理するとき、無視されます。
Halt	軸を一時停止します。注記: このコマンドは、「速度セットポイント」コマンドの後にのみ実行されます。
Positioning Relative	距離に基づいて軸を位置決めします。このコマンドは、指定された距離および速度によって軸を移動します。

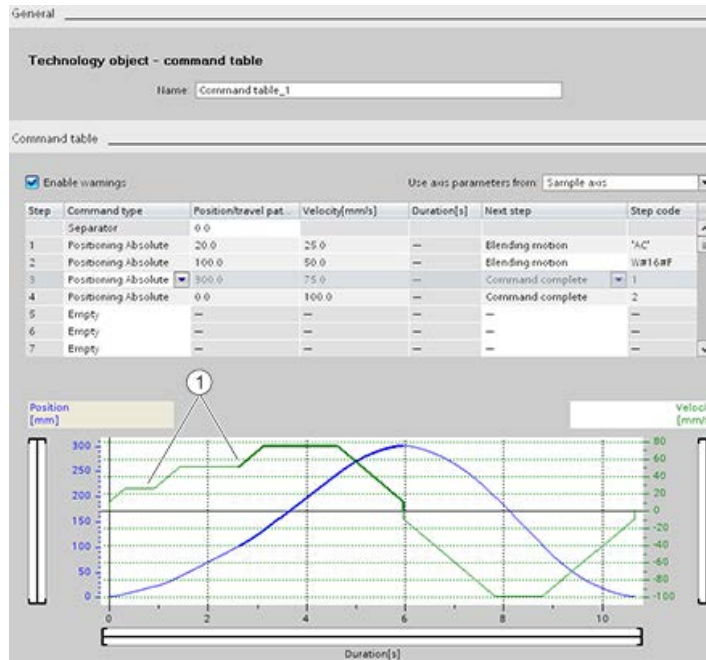
コマンドタイプ	説明
Positioning Absolute	位置に基づいて軸を位置決めします。このコマンドは、指定された速度を使用して、指定された位置まで軸を移動します。
Velocity setpoint	指定された速度で軸を移動します。
Wait	指定された時間間隔が経過するまで待機します。「待機」は、有効な移動モーションを停止しません。
Separator	選択した行の上に 1 つの「セパレータ」行を追加します。セパレータ行によって、単一のコマンドテーブル内に複数のプロファイルを定義できます。

下の図では、「コマンドの完了」が次のステップへの移行として使用されます。このタイプの移行では、デバイスが開始/停止速度まで減速した後、次のステップの開始時にもう 1 度加速できます。



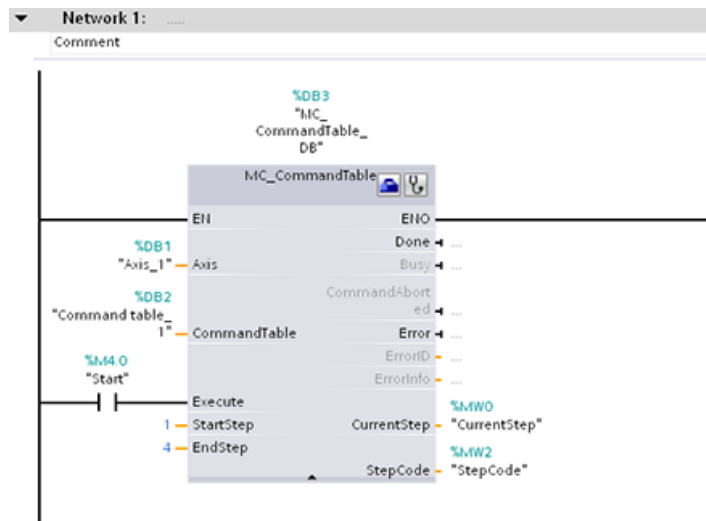
① 軸が、ステップ間で開始速度/停止速度まで減速します。

下の図では、「ブレンドモーション」が次のステップへの移行として使用されます。このタイプの移行では、デバイスの速度を次のステップの開始まで保持することによって、1つのステップから次のステップへのデバイスの移行がスムーズになります。ブレンディングを使用すると、1つのプロファイルを完全に実行するための合計時間を短縮できます。ブレンディングを使用しないと、この例の実行に7秒間かかります。ブレンディングを使用すると、実行時間が1秒間減り、合計で6秒間になります。



① 軸が移動し続け、次のステップ速度まで加速/減速することによって、時間および機械的摩耗が節約されます。

下に示すように、CommandTable の動作は MC_CommandTable 命令によって制御されます。



10.6 S7-1200 のモーションコントロールの動作

10.6.1 モーションコントロールに使用するCPU出力

CPU は、4 つのパルス出力ジェネレータを提供します。それぞれのパルス出力ジェネレータは、パルスインターフェース付きのステッピングモータドライブまたはサーボモータドライブを制御するために 1 つのパルス出力と 1 つの方向出力を提供します。パルス出力は、モータモーションに必要なパルスを提供します。方向出力は、ドライブの移動方向を制御します。

PTO は、可変周波数の方形波出力を生成します。パルス生成は、H/W コンフィグレーションおよび/または SFC/SFB によって提供される構成および実行情報によって制御されます。

CPU が RUN モード中のユーザーの選択に基づいて、イメージレジスタに保存された値か、パルスジェネレータ出力がデジタル出力を駆動します。STOP モードでは、PTO は出力を制御しません。

オンボード CPU 出力とシグナルボードの出力をパルスおよび方向出力として使用できます。[プロパティ] タブのパルスジェネレータ (PTO/PWM) でのデバイス構成中に、オンボード CPU 出力かシグナルボードの出力のどちらかを選択します。PTO (パルス列出力) だけがモーションコントロールに適用されます。

下の表に、既定の I/O 割り当てを示します。ただし、任意のデジタル出力に対して 4 つのパルスジェネレータを構成できます。

注記

パルス列出力は、ユーザープログラムの他の命令によって使用することはできません。

CPU またはシグナルボードの出力をパルスジェネレータとして設定すると (PWM またはモーションコントロール命令での使用のために)、対応出力アドレスはもう出力を制御しません。ユーザープログラムがパルスジェネレータとして使用されている出力に値を書き込むと、その CPU はこの値を物理出力に書き込みません。

注記

PTO 方向出力は、ユーザープログラムの他の場所での使用のために割り当て解除することができます。

各 PTO は 2 つの出力の割り当てを必要とします: 1 つはパルス出力として使用され、もう 1 つは方向出力として使用されます。パルス出力だけを使用し、方向出力を使用しないことも可能です。この場合、ユーザープログラムの他の目的のために方向出力を割り当て解除できます。この出力は、PTO 出力とユーザープログラムの他の目的の両方のために同時に使用することはできません。

表 10-22 パルスおよび方向出力の既定のアドレス割り当て

モーションコントロール用出力の使用		
	パルス	方向
PTO1		
内蔵 I/O	Q0.0	Q0.1
SB I/O	Q4.0	Q4.1
PTO2		
内蔵 I/O	Q0.2	Q0.3
SB I/O	Q4.2 ¹	Q4.3 ¹
PTO3		
内蔵 I/O	Q0.4 ²	Q0.5 ²
SB I/O	Q4.0	Q4.1
PTO4		
内蔵 I/O	Q0.6 ³	Q0.7 ³
SB I/O	Q4.2	Q4.3

¹ 出力 Q4.2 および Q4.3 は、SB1222 DQ4 だけで使用できます。

² CPU 1211C には出力 Q0.4、Q0.5、Q0.6、または Q0.7 はありません。このため、CPU 1211C ではこれらの出力を使用できません。

³ CPU 1212C には出力 Q0.6、または Q0.7 がありません。このため、CPU 1212C ではこれらの出力を使用できません。

⁴ この表は、CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C、および CPU 1217C PTO ファンクションに適用されます。

ドライバインターフェース

モーションコントロールでは、「ドライブ有効」および「ドライブ準備完了」用のドライバインターフェースをオプションで構成できます。このドライバインターフェースを使用するとき、ドライバイネーブル用のデジタル出力と「ドライブ準備完了」用のデジタル入力を自由に選択できます。

注記

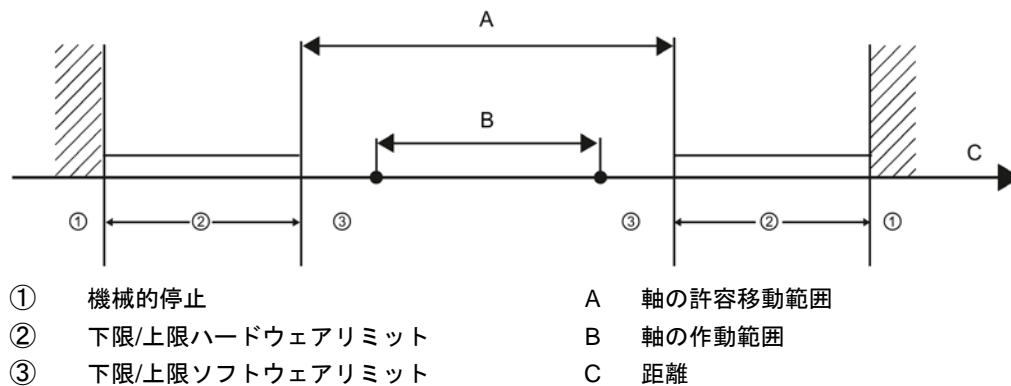
PTO (パルス列出力)が選択され、軸に割り当てられた場合、ファームウェアは該当のパルスおよび方向出力によって制御を行います。

制御ファンクションのこの引き継ぎを行うと、プロセスイメージと IO 出力間の接続も切断されます。ユーザープログラムまたはウォッチテーブルを介してパルスおよび方向出力のプロセスイメージを書き込むことができる間は、プロセスイメージが I/O 出力へ転送されることはありません。その結果、ユーザープログラムまたはウォッチテーブルを介して I/O 出力をモニタすることもできません。読み出される情報はプロセスイメージの値を反映するだけで、I/O 出力の現在のステータスと全く一致しません。

CPU ファームウェアによって持続的に使用されない他のすべての CPU 出力の場合は、I/O 出力のステータスを通常通りプロセスイメージを介して制御またはモニタすることができます。

10.6.2 モーションコントロール用のハードウェアおよびソフトウェアリミットスイッチ

ハードウェアおよびソフトウェアリミットスイッチを使用して、軸の「許容移動範囲」および「作動範囲」を制限します。



ハードウェアおよびソフトウェアリミットスイッチを使用するには、その前に、構成またはユーザープログラムでそれらを有効化する必要があります。ソフトウェアリミットスイッチは、軸の原点復帰の後にのみ有効になります。

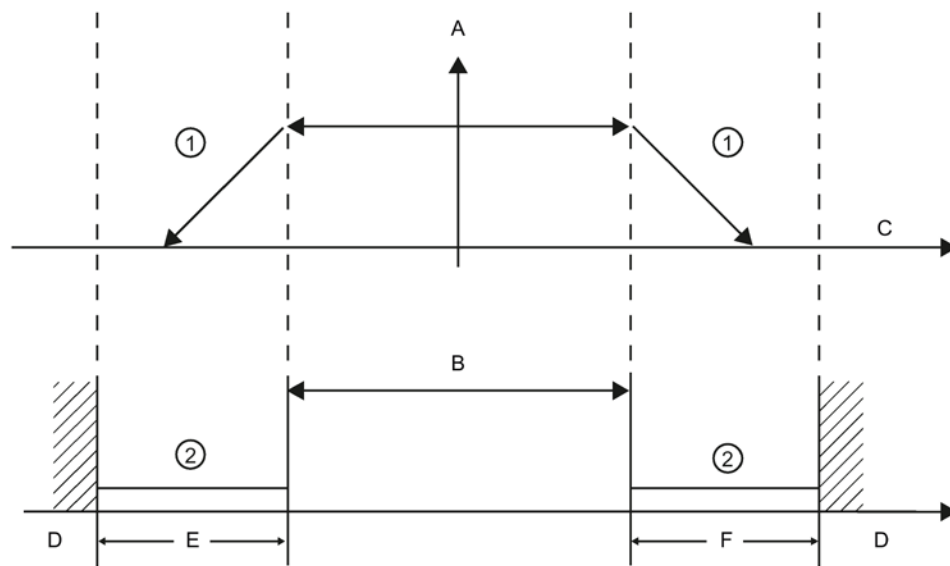
ハードウェアリミットスイッチ

ハードウェアリミットスイッチは、軸の最大移動範囲を決定します。ハードウェアリミットスイッチは、CPU の割り込み可能入力に接続する必要がある物理的スイッチングエレメントです。アプローチされた後、持続的にスイッチされたままであるハードウェアリミットスイッチだけを使用します。このスイッチングステータスは、許容移動範囲に戻った後にのみ無効にすることができます。

表 10-23 ハードウェアリミット用の使用可能な入力

説明	RPS	LIM-	LIM+
内蔵 I/O		I0.0~I1.5	
SB I/O		I4.0~I4.3	

ハードウェアリミットスイッチにアプローチすると、軸が、設定された緊急減速で制動され、停止します。指定する緊急減速は、機械的停止に達する前に軸が確実に停止する大きさでなければなりません。次のダイアグラムは、ハードウェアリミットスイッチにアプローチした後の軸の動作を示します。



- ① 軸が、設定された緊急減速で制動され、停止します。
- ② ハードウェアリミットスイッチがステータス「アプローチされています」を通知する範囲。
- A [速度]
- B 許容移動範囲
- C 距離
- D 機械的停止
- E 下限ハードウェアリミットスイッチ
- F 上限ハードウェアリミットスイッチ



警告

デジタル入力チャンネル用フィルタ時間の変更によるリスク

デジタル入力チャンネル用フィルタ時間を以前の設定から変更すると、このフィルタが新しい入力に対して完全に有効になる前に、合計で最大 20 ミリ秒の時間の間、新しい「0」レベルの入力値を提供する必要があります。この間、時間が 20 ms 未満の短い「0」パルスイベントは検出またはカウントできません。

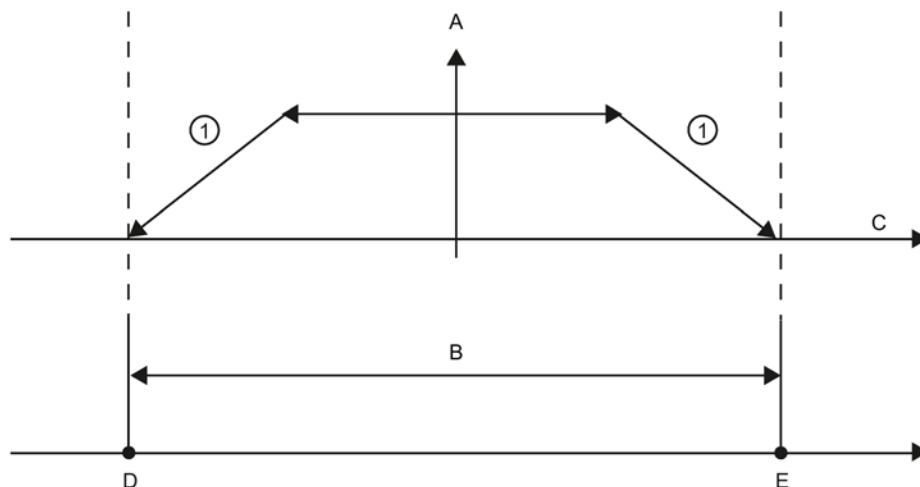
このフィルタ時間の変更は、予期しないマシンまたはプロセスの動作を引き起こし、その結果、要員の死または深刻な傷害や設備への破壊的影響を招く場合があります。

この新しいフィルタ時間を直ちに有効にするには、CPU の電源のオフ/オンを実行する必要があります。

ソフトウェアリミットスイッチ

ソフトウェアリミットスイッチは、軸の「作動範囲」を制限します。ソフトウェアリミットスイッチは、移動範囲に対するハードウェアリミットスイッチの範囲内でなければなりません。ソフトウェアリミットスイッチの位置は自由に設定できるため、軸の作動範囲は、現在の移動プロファイルに依存する個別基準に基づいて制限される場合があります。ハードウェアリミットスイッチとは対照的に、ソフトウェアリミットスイッチはソフトウェアによってのみ実装され、それら自身のスイッチングエレメントを必要としません。

ソフトウェアリミットスイッチが有効化されると、アクティブモーションがソフトウェアリミットスイッチの位置で停止されます。軸は、設定された減速で制動されます。次のダイアグラムは、ハードウェアリミットスイッチに到達するまでの軸の動作を示します。



① 軸が、設定された減速で制動され、停止します。

A [速度]

B 作動範囲

C 距離

D 下限ソフトウェアリミットスイッチ

E 上限ソフトウェアリミットスイッチ

機械的停止がソフトウェアリミットスイッチの後に存在し、機械的損傷のリスクがある場合は、追加ハードウェアリミットスイッチを使用します。

追加情報

ユーザープログラムが、ハードウェアおよびソフトウェアリミット機能の両方を有効/無効にすることによって、ハードウェアおよびソフトウェア位置リミットの現在の状態を無効にできます。有効化/無効化の選択は軸 DB を参照して行います。

- ハードウェアリミット機能を有効/無効にするには、DB パス「<軸名
>/Config/**PositonLimits_HW**」の「アクティブ」タグ(Bool)にアクセスします。「アクティブ」タグの状態が、ハードウェア位置リミットの使用を有効/無効にします。
- ソフトウェア位置リミット機能を有効/無効にするには、DB パス「<軸名
>/Config/**PositonLimits_SW**」の「アクティブ」タグ(Bool)にアクセスします。この「アクティブ」タグの状態が、ソフトウェア位置リミットを有効/無効にします。

ユーザープログラムでソフトウェア位置リミットを変更することもできます(たとえば、マシンセットアップ用のフレキシビリティを高めるか、マシン切り替え時間を短縮するために)。ユーザープログラムで、新しい値(Real 形式の工学単位で)を DB「<軸名
>/Config/**PositionLimits_SW**」内の「MinPosition」および「MaxPosition」タグに書き込むこともできます。

10.6.3 原点復帰

10.6.3.1 軸の原点復帰

原点復帰は、軸座標をドライブの実際の物理的位置に一致させることを示します。(ドライブが現在位置 x に存在する場合、軸は位置 x に合わせられます。) 位置制御される軸の場合、位置に関する入力と表示は正確にこれらの軸座標を基準にした値です。

注記

軸座標と実際の場所との一致は非常に重要です。このステップは、軸の絶対ターゲット位置がドライブによって正確に実現されるためにも必要です。

MC_Home 命令は軸の原点復帰を開始します。

4つの異なる原点復帰ファンクションがあります。最初の2つのファンクションではユーザーが軸の現在の位置を設定でき、2番目の2つのファンクションでは原点参照センサに対して軸を位置決めできます。

- モード0 - 絶対直接リファレンシング: このモードを実行すると、軸の正確な位置が指示されます。それは、内部の位置変数に原点復帰命令の「Position」入力の値を設定します。これは、マシン校正およびセットアップに使用されます。

軸位置は、原点スイッチに関わりなく、設定されます。アクティブ移動モーションは中止されません。MC_Home 命令の「Position」入力パラメータの値は、すぐに、軸の原点として設定されます。この原点を正確な機械的位置に割り当てるには、原点復帰操作時に軸がこの位置で停止していなければなりません。

- モード1 - 相対直接リファレンシング: このモードを実行すると、内部の位置変数が使用され、原点復帰命令の「Position」入力の値が加算されます。これは、通常、マシンオブセットのために使用されます。

軸位置は、原点スイッチに関わりなく、設定されます。アクティブ移動モーションは中止されません。原点復帰後の軸位置に対して次のステートメントが適用されます: 新しい軸位置 = 現在の軸位置 + MC_Home 命令の「Position」パラメータの値。

- モード2 - パッシブリファレンシング: 軸が動いていて、原点スイッチを通過すると、現在の位置が原点位置として設定されます。この機能によって、通常の機械の摩耗やギアのきしみの原因が明白になり、摩耗を防ぐための手動による位置の補正の必要がなくなります。前記と同様に、原点復帰命令の「Position」入力が、原点スイッチによって指示された位置に加算され、原点位置の簡単なオフセットを可能にします。

パッシブ原点復帰中は、MC_Home 命令は原点復帰モーションを全く実行しません。このステップで必要な移動モーションは、ユーザーが他のモーションコントロール命令を使用して実行する必要があります。原点スイッチが検出されると、軸は設定に従って原点復帰します。パッシブ原点復帰の開始時に、アクティブ移動モーションは中止されません。

- モード 3 - アクティブリファレンシング: このモードは、軸の原点復帰の最も正確な方法です。移動の最初の方向と速度は、テクノロジーオブジェクト構成拡張パラメータ - 原点復帰で設定されます。これは、マシン構成によって異なります。原点スイッチ信号の立ち上がりエッジと立ち下がりエッジのどちらが原点位置であるかどうかを決定する機能も存在します。事実上すべてのセンサが有効な範囲を持ち、安定状態オン位置が原点信号として使用された場合、オン信号有効範囲は一定の範囲の距離に渡るため、原点復帰位置に誤差が生じる可能性があります。該当信号の立ち上がりエッジまたは立ち下がりエッジを使用することによって、さらに正確な原点位置が得られます。他のすべてのモードと同様に、原点復帰命令の「Position」入力がハードウェア基準位置に加算されます。

アクティブ原点復帰モードの場合、MC_Home 命令は必要な原点アプローチを実行します。原点スイッチが検出されると、軸は設定に従って原点復帰します。アクティブ移動モーションは中止されます。

モード 0 および 1 では、軸の移動は不要です。それらは、通常、セットアップおよび校正で使用されます。モード 2 および 3 では、軸が移動し、「軸」テクノロジーオブジェクトで原点スイッチとして設定されているセンサを通過することが必要です。原点は、軸のワークエリアに配置するか、通常のワークエリアの外部の移動範囲内に配置できます。

10.6.3.2 原点復帰パラメータの設定

[原点復帰]構成ウィンドウで、アクティブおよびパッシブ原点復帰用のパラメータを設定します。原点復帰方法は、モーションコントロール命令の「Mode」入力パラメータを使用して設定されます。この場合、Mode = 2 はパッシブ原点復帰を示し、Mode = 3 はアクティブ原点復帰を示します。

注記

以下の手段の 1 つを使用して、方向反転の場合にマシンが機械的停止まで移動しないようにします。

- アプローチ速度を低速に保つこと
- 設定された加速/減速を増やすこと
- ハードウェアリミットスイッチと機械的停止の間の距離を増やすこと

表 10- 24 軸の原点復帰用設定パラメータ

パラメータ	説明
原点復帰入力 (アクティブおよびパッシブ原点復帰)	<p>ドロップダウンリストボックスから、原点スイッチ用デジタル入力を選択します。この入力は割り込み可能でなければなりません。オンボード CPU 入力および挿入されたシグナルボードの入力を、原点スイッチ用入力として選択できます。</p> <p>デジタル入力の既定のフィルタ時間は 6.4 ミリ秒です。デジタル入力を原点スイッチとして使用すると、不要な減速が生じ、その結果、位置が不正確になる場合があります。低減速度および原点スイッチの範囲によっては、原点を検出できません。フィルタ時間は、デジタル入力のデバイス構成の「入力フィルタ」で設定できます。</p> <p>指定するフィルタ時間は、原点スイッチでの入力信号の持続時間未満でなければなりません。</p>
HW リミットスイッチでの自動反転の許可 (アクティブ原点復帰のみ)	<p>ハードウェアリミットスイッチを原点アプローチ用反転カムとして使用するためのチェックボックスを有効化します。方向反転のために、ハードウェアリミットスイッチを設定し、有効化する必要があります。</p> <p>アクティブ原点復帰中にハードウェアリミットスイッチに到達すると、軸は設定された減速(緊急減速ではありません)で制動され、その方向が反転します。この後、反転方向で原点スイッチが検出されます。</p> <p>方向反転が有効でなく、軸がアクティブ原点復帰中にハードウェアリミットスイッチに到達すると、原点アプローチはエラーで中止され、軸は緊急減速で制動されます。</p>
アプローチ/原点復帰の方向 (アクティブおよびパッシブ原点復帰)	<p>方向選択では、原点スイッチを検出するためにアクティブ復帰中に使用する「アプローチ方向」と、原点復帰方向を決定します。原点復帰方向としては、原点復帰操作を実行するために、軸が原点スイッチの設定された側にアプローチするために使用する移動方向を指定します。</p>
原点復帰スイッチの側 (アクティブおよびパッシブ原点復帰)	<ul style="list-style-type: none"> アクティブ原点復帰: 軸を原点スイッチの左側に移動するか、右側に移動するかを選択します。軸の開始位置と原点復帰パラメータの設定に応じて、原点アプローチシーケンスは、構成ウィンドウのダイアグラムと異なる場合があります。 パッシブ原点復帰: パッシブ原点復帰では、原点復帰のための移動モーションは、ユーザーがモーションコマンドを使用して実行する必要があります。原点復帰が行われる原点スイッチの側は、以下のファクタに応じて異なります。 <ul style="list-style-type: none"> 「アプローチ方向」設定 「原点スイッチ」設定 パッシブ原点復帰中の現在の移動方向
アプローチ速度 (アクティブ原点復帰のみ)	<p>原点アプローチ中に原点スイッチが検索される速度を指定します。</p> <p>限界値(選択されたユーザー単位とは無関係です): $\text{開始/停止速度} \leq \text{アプローチ速度} \leq \text{最大速度}$</p>

パラメータ	説明
原点復帰速度 (アクティブ原点復帰のみ)	軸が原点復帰のために原点スイッチにアプローチする速度を指定します。 限界値(選択されたユーザー単位とは無関係です): 開始/停止速度 ≤ 低減速度 ≤ 最大速度
原点復帰位置オフセット (アクティブ原点復帰のみ)	必要な基準位置が原点スイッチの位置と異なる場合、このフィールドに原点位置オフセットを指定できます。 その値が 0 と等しくない場合、軸は、原点スイッチでの原点復帰の後に以下のアクションを実行します。 1. 軸を低減速度で原点位置オフセットの値だけ移動します。 2. 原点位置オフセットの位置に到達したら、軸位置として、絶対基準位置の値が設定されます。絶対基準位置は、モーションコントロール命令「MC_Home」のパラメータ「Position」に指定されます。 限界値(選択されたユーザー単位とは無関係です): -1.0e12 ≤ 原点位置オフセット ≤ 1.0e12

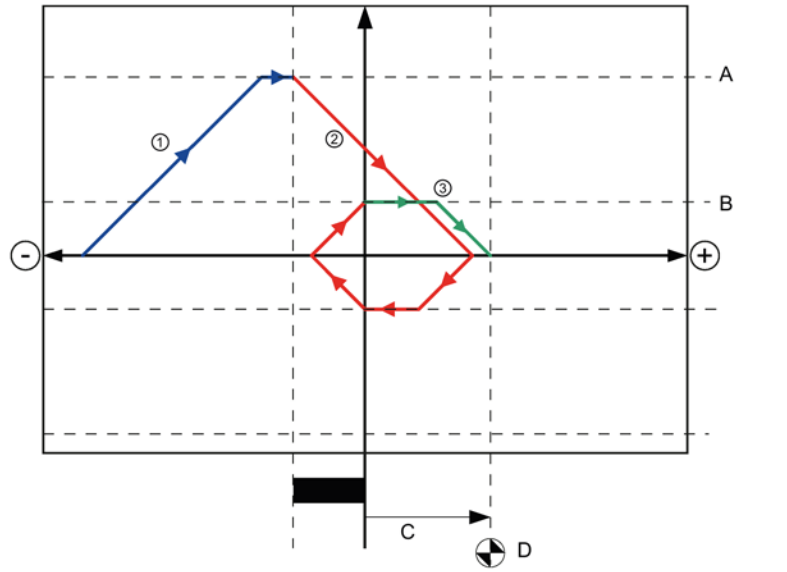
表 10- 25 原点復帰に影響するファクタ

影響するファクタ:			結果:
コンフィグレーション アプローチ方向	コンフィグレーション 原点スイッチ	現在の移動方向	原点復帰 原点スイッチ
正	「左(負の)側」	正の方向	左
		負の方向	右
正	「右(正の)側」	正の方向	右
		負の方向	左
負	「左(負の)側」	正の方向	右
		負の方向	左
負	「右(正の)側」	正の方向	左
		負の方向	右

10.6.3.3 アクティブ原点復帰のシーケンス

- アクティブ原点復帰は、モーションコントロール命令「MC_Home」(入力パラメータ「Mode」= 3)で開始します。入力パラメータ「Position」には、この場合、絶対原点座標を指定します。上記の代わりに、アクティブ原点復帰をテストの目的のためにコントロールパネルで開始できます。
- 次のダイアグラムは、以下のパラメータを設定した場合のアクティブ原点アプローチの特性曲線の例を示します。
- 「アプローチ方向」=「正のアプローチ方向」
 - 「原点スイッチ」=「右(正の)側」
 - 「原点位置オフセット」の値 > 0

表 10- 26 MC 原点復帰の速度特性

演算		注記	
		A	アプローチ速度
		B	低減速度
		C	原点位置座標
		D	原点位置オフセット
①	サーチフェーズ(青の曲線セグメント): アクティブ原点復帰が開始されると、軸が設定された「アプローチ速度」まで加速し、この速度で原点スイッチを検索します。		
②	原点アプローチ(赤の曲線セグメント): 原点スイッチが検出されると、この例では、軸は制動され、反転し、原点スイッチの設定された側方向へ設定された「低減速度」で原点復帰します。		
③	原点位置への移動(戻りの曲線セグメント): 原点スイッチで原点復帰した後、軸は、「低減速度」で、「原点座標」まで移動します。「原点座標」に到達したら、軸は、MC_Home 命令の「Position」入力パラメータで指定された位置の値で、停止されます。		

注記

原点復帰検索が予測通りに動作しない場合は、ハードウェアリミットまたは原点に割り当てられた入力をチェックしてください。これらの入力のエッジ割り込みがデバイス構成で無効化されている可能性があります。

「下限 HW リミットスイッチ入力」、「上限 HW リミットスイッチ入力」、および「原点スイッチ入力」に割り当てられている入力(存在する場合)を調べるために、該当の軸テクノロジーオブジェクトの構成データをチェックします。この後、CPU のデバイス構成を開き、割り当てられた各入力を調べます。「立ち上がりエッジ検出の有効化」と「立ち下がりエッジ検出の有効化」の両方が選択されていることを確認します。これらのプロパティが選択されていない場合は、軸構成で指定された入力を削除し、それらを再び選択します。

10.7 モーションコントロール命令

10.7.1 MC命令の概要

モーションコントロール命令は、CPU の関連付けられたテクノロジーデータブロックと専用 PTO (パルストレイン出力)を使用して、軸の動作を制御します。

- MC_Power (Page 310)は、モーションコントロール軸を有効化/無効化します。
- MC_Reset (Page 313)は、すべてのモーションコントロールエラーをリセットします。認識可能なすべてのモーションコントロールエラーの認識ができます。
- MC_Home (Page 314)は、軸制御プログラムと軸機械位置決めシステムの関係確立します。
- MC_Halt (Page 317)は、すべてのモーションプロセスをキャンセルし、軸モーションを停止します。停止位置は定義されません。
- MC_MoveAbsolute (Page 319)は、絶対位置へのモーションを開始します。ジョブはターゲット位置に達すると終了します。
- MC_MoveRelative (Page 321)は、開始位置に対する相対位置決めモーションを開始します。
- MC_MoveVelocity (Page 323)は、軸を指定された速度で移動させます。
- MC_MoveJog (Page 326) は、テストおよびスタートアップの目的のために、ジョグモードを実行します。
- MC_CommandTable (Page 328)は、一連の軸コマンドを移動シーケンスとして実行します。
- MC_ChangeDynamic (Page 330)は、軸のダイナミクス設定を変更します。
- MC_WriteParam (Page 332)は、軸の機能を変更するために、ユーザープログラムからパラメータの選択番号を書き込みます。
- MC_ReadParam (Page 334)は、軸入力で定義された軸の現在の位置、速度などを示すパラメータの選択番号を読み出します。

CPU ファームウェアレベル

V4.1 ファームウェアの S7-1200 CPU では、各モーション命令の V5.0 バージョンを選択します。

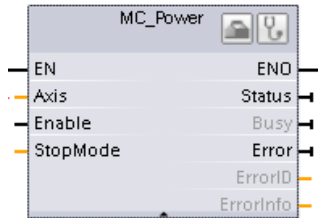
V4.0 以前のファームウェアの S7-1200 CPU では、各モーション命令の該当する V4.0、V3.0、V2.0、または V1.0 バージョンを選択します。

10.7.2 MC_Power (軸の有効化/無効化)命令

注記

軸がエラーのためにスイッチオフされると、エラーが除去され、確認応答された後、軸は再び自動的に有効化されます。これを行うには、このプロセス中にイネーブル入力パラメータが値 TRUE を保持している必要があります。

表 10- 27 MC_Power 命令

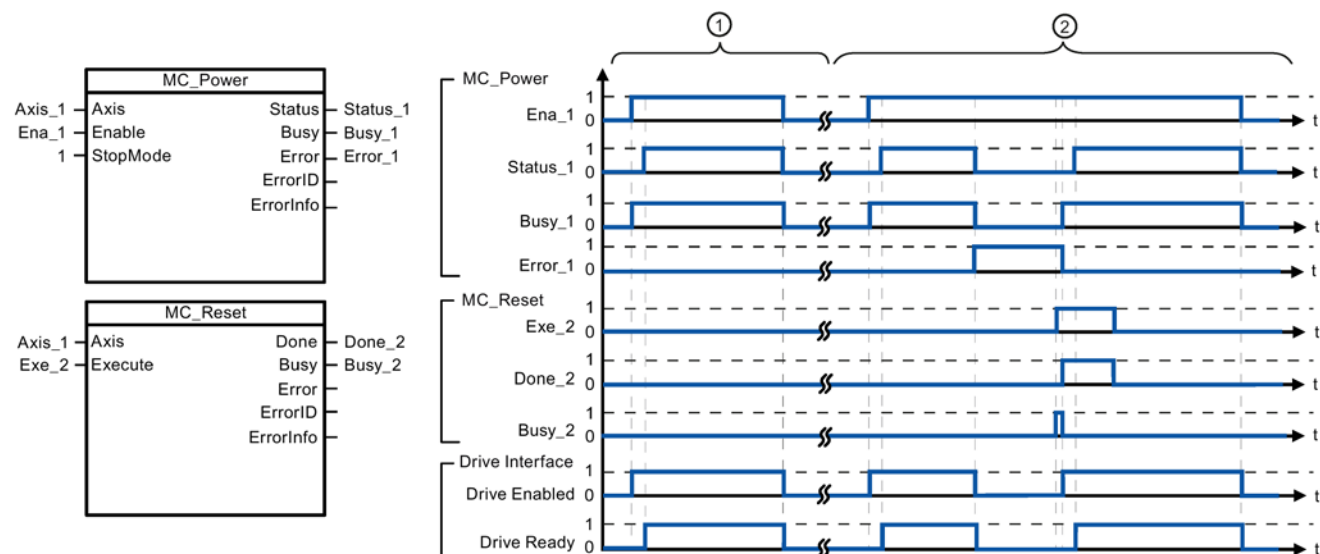
LAD / FBD	SCL	説明
<div><p>"MC_Power_DB"</p></div>	<pre>"MC_Power_DB"(Axis:=_multi_fb_in_, Enable:=_bool_in_, StopMode:=_int_in_, Status=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_Power モーションコントロール命令は、軸を有効化/無効化します。軸を有効化/無効化するには、その前に、以下の条件を確認します。</p> <ul style="list-style-type: none">• 該当のテクノロジーオブジェクトが正しく構成されていること。• 保留中のイネーブル禁止エラーが存在しないこと。 <p>MC_Power の実行は、モーションコントロールタスクによって中止できません。軸を無効化すると(入力パラメータ「Enable」= FALSE)、関連テクノロジーオブジェクトのすべてのモーションコントロールタスクが中止されます。</p>

¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_Power_DB」がインスタンス DB の名前です。

表 10- 28 MC_Power 命令のパラメータ

パラメータとタイプ		データタイプ	説明
Axis	IN	TO_Axis_1	軸テクノロジーオブジェクト
Enable	IN	Bool	<ul style="list-style-type: none"> FALSE (既定値): すべてのアクティブタスクがパラメータ設定された「StopMode」に従って中止され、軸が停止されます。 TRUE: モーションコントロールが軸を有効化しようとします。
StopMode	IN	Int	<ul style="list-style-type: none"> 0: 緊急停止: 軸を無効化する要求が保留中の場合、軸は、設定された緊急減速で制動されます。軸は、停止した後、無効化されます。 1: 即時停止: 軸を無効化する要求が保留中の場合、この軸は減速なしで無効化されます。パルス出力は直ちに停止されます。 2: ジャーク制御による緊急停止: 軸を無効化する要求が保留中の場合、この軸は、設定された緊急停止減速で制動されます。ジャーク制御を有効化すると、設定されたジャークが考慮されます。軸は、停止した後、無効化されます。
Status	OUT	Bool	<p>軸イネーブルのステータス:</p> <ul style="list-style-type: none"> FALSE: 軸が無効化されています: <ul style="list-style-type: none"> 軸はモーションコントロールタスクを実行せず、新しいタスクを受け入れません(例外: MC_Reset タスク)。 軸は原点復帰されません。 無効化時、軸が停止するまで、ステータスは FALSE に変わりません。 TRUE: 軸が有効化されます。 <ul style="list-style-type: none"> 軸は、モーションコントロールタスク実行の準備を完了しています。 軸有効化時、信号「ドライブが準備完了」が保留中になるまで、ステータスは TRUE に変わりません。軸構成で「ドライブが準備完了」ドライブインターフェースが構成されなかった場合、ステータスは直ちに TRUE に変わります。
Busy	OUT	Bool	<p>FALSE: MC_Power が動作中ではありません。</p> <p>TRUE: MC_Power が動作中です。</p>
Error	OUT	Bool	<p>FALSE: エラーは発生していません。</p> <p>TRUE: モーションコントロール命令「MC_Power」または関連テクノロジーオブジェクトで、エラーが発生しました。エラーの原因は、パラメータ「ErrorID」および「ErrorInfo」を参照してください。</p>
ErrorID	OUT	Word	パラメータ「Error」のエラーID
ErrorInfo	OUT	Word	パラメータ「ErrorID」のエラー情報 ID



- ① 軸が有効化された後、再び無効化されます。ドライブが「ドライブが準備完了」を CPU に返信した後、「Statud_1」経由で成功イネーブルを読み出すことができます。
- ② 軸イネーブルの後に、エラーが発生し、軸が無効化されました。エラーが除去され、「MC_Reset」によって確認応答されています。この後、軸が再び有効化されています。

構成されたドライブインターフェースで軸を有効化するには、以下の手順に従います。

1. 上記の必要条件をチェックします。
2. 入力パラメータ「StopMode」を必要な値で初期化します。入力パラメータ「Enable」を TRUE にセットします。

「ドライブが有効」用のイネーブル出力が、ドライブへの電力を有効にするために、TRUE に変わります。CPU が、ドライブの「ドライブが準備完了」信号を待機します。

「ドライブが準備完了」信号が、CPU の設定された準備完了入力で入手可能になったとき、軸は有効になります。出力パラメータ「Status」およびテクノロジーオブジェクトタグ<軸名>.StatusBits.Enable は、値 TRUE を示します。

構成されたドライブインターフェースなしで軸を有効化するには、以下の手順に従います。


1. 上記の必要条件をチェックします。
2. 入力パラメータ「StopMode」を必要な値で初期化します。入力パラメータ「Enable」を TRUE にセットします。軸が有効化されます。出力パラメータ「Status」およびテクノロジーオブジェクトタグ<軸名>.StatusBits.Enable は、値 TRUE を示します。

軸を無効化するには、以下の手順に従います。

1. 軸を停止します。
軸が停止しているときは、テクノロジーオブジェクトタグ<軸名>.StatusBits.StandStill で識別できます。
2. 軸が停止した後、入力パラメータ「Enable」を FALSE にセットします。
3. 出力パラメータ「Busy」および「Status」とテクノロジーオブジェクトタグ<軸名>.StatusBits.Enable が値 FALSE を示す場合、軸の無効化が完了しています。

10.7.3 MC_Reset (軸エラーリセット)命令

表 10- 29 MC_Reset 命令

LAD / FBD	SCL	説明
	<pre>"MC_Reset_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Restart:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_Reset 命令を使用して、「軸停止による動作エラー」および「構成エラー」を確認します。確認が必要なエラーについては、「ErrorID および ErrorInfo のリスト」の「対策」を参照してください。</p> <p>MC_Reset 命令を使用する前に、確認を必要とする保留中の構成エラーの原因を除去する必要があります(たとえば、「軸」テクノロジーオブジェクトの無効な加速値を有効な値に変更することによって)。</p> <p>V3.0 以降では、Restart コマンドを使用して、RUN 動作モードで軸構成をワークメモリにダウンロードすることができます。</p>

¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_Reset_DB」がインスタンス DB の名前です。

MC_Reset タスクは、他のすべてのモーションコントロールタスクによって中止できません。この新しい MC_Reset タスクは、他のすべてのアクティブモーションコントロールタスクを中止しません。

表 10- 30 MC_Reset 命令のパラメータ

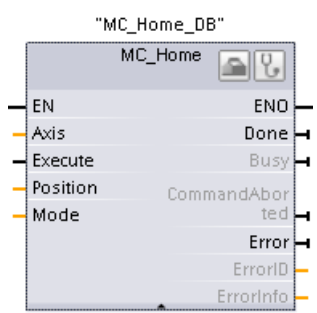
パラメータとタイプ	データタイプ	説明
Axis	IN	TO_Axis_1
Execute	IN	Bool
Restart	IN	Bool
Done	OUT	Bool
Busy	OUT	Bool
Error	OUT	Bool
ErrorID	OUTP	Word
ErrorInfo	OUT	Word

MC_Reset でエラーを確認するには、以下の手順に従います。

1. 上記の必要条件をチェックします。
2. 「Execute」入力パラメータでの立ち上がりエッジによってエラーの確認を開始します。
3. Done = TRUE でかつテクノロジーオブジェクトタグ<軸名>.StatusBits.Error = FALSE のとき、エラーが確認されました。.

10.7.4 MC_Home (軸の原点復帰)命令

表 10- 31 MC_Home 命令

LAD / FBD	SCL	説明
	<pre>"MC_Home_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Mode:=_int_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_Home 命令を使用して、軸座標を実際の物理的ドライブ位置に一致させます。原点復帰は、軸の絶対位置決めが必要です。</p> <p>MC_Home 命令を使用するには、最初に軸を有効化する必要があります。</p>

- ¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。
- ² この SCL の例では、「MC_Home_DB」がインスタンス DB の名前です。

以下のタイプの原点復帰が使用できます。

- 絶対直接原点復帰(Mode = 0): 現在の軸位置の値として、パラメータ「位置(Position)」の値が設定されます。
- 相対直接原点復帰(Mode = 1): 現在の軸位置が、パラメータ「位置(Position)」の値によってオフセットされます。
- パッシブ原点復帰(Mode = 2): パッシブ原点復帰中は、MC_Home 命令は原点復帰モーションを全く実行しません。このステップで必要な移動モーションは、ユーザーが他のモーションコントロール命令を使用して実行する必要があります。原点スイッチが検出されると、軸は原点復帰されます。
- アクティブ原点復帰(Mode = 3): 原点復帰手順は自動的に実行されます。

表 10- 32 MC_Home 命令のパラメータ

パラメータとタイプ		データタイプ	説明
Axis	IN	TO_Axis_PTO	軸テクノロジーオブジェクト
Execute	IN	Bool	立ち上がりエッジによるタスクの開始
Position	IN	Real	<ul style="list-style-type: none"> モード= 0、2、および 3 (原点復帰動作完了後の軸の絶対位置) モード= 1 (現在の軸位置の補正值) 限界値: $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Mode	IN	Int	原点復帰モード <ul style="list-style-type: none"> 0: 絶対直接原点復帰 新しい軸位置は、パラメータ「Position」の位置の値です。 1: 相対直接原点復帰 新しい軸位置は、現在の軸位置 + パラメータ「Position」の位置の値です。 2: パッシブ原点復帰 軸構成に従った原点復帰。原点復帰の後、新しい軸位置として、パラメータ「Position」の値が設定されます。 3: アクティブ原点復帰 軸構成に従った原点アプローチ。原点復帰の後、新しい軸位置として、パラメータ「Position」の値が設定されます。
Done	OUT	Bool	TRUE = タスクが完了
Busy	OUT	Bool	TRUE = このタスクが実行中です。
CommandAborted	OUT	Bool	TRUE = 実行中にタスクが別のタスクによって中止されました。
Error	OUT	Bool	TRUE = このタスクの実行中にエラーが発生しました。エラーの原因は、パラメータ「ErrorID」および「ErrorInfo」を参照してください。
ErrorID	OUT	Word	パラメータ「Error」のエラーID
ErrorInfo	OUT	Word	パラメータ「ErrorID」のエラー情報 ID

注記

軸の原点復帰は、以下の状況の下では無効になります。

- MC_Power 命令による軸の無効化
- 自動制御と手動制御の間の切り替え
- アクティブ原点復帰の開始時(原点復帰動作の正常終了後、軸原点復帰が再び使用できません。)
- CPU の電源オフ/オン後
- CPU の再起動後(RUN~STOP または STOP~RUN)

軸を原点復帰するには、以下の手順に従います。

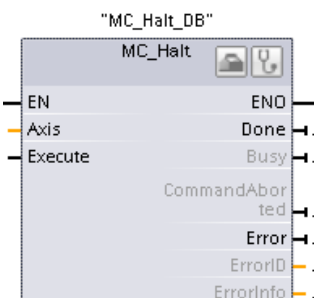
1. 上記の必要条件をチェックします。
2. 必要な入力パラメータに初期値を設定し、入力パラメータ「Execute」での立ち上がりエッジによって原点復帰動作を開始します。
3. 出力パラメータ「Done」およびテクノロジーオブジェクトタグ<軸名>.StatusBits.HomingDone が値 TRUE を示す場合、原点復帰は完了しています。

表 10- 33 応答の無効化

モード	説明	
0 または 1	MC_Home タスクは、他のすべてのモーションコントロールタスクによって中止できません。この新しい MC_Home タスクは、他のすべてのアクティブモーションコントロールタスクを中止しません。位置関連モーションタスクは、新しい原点復帰位置(Position 入力パラメータの値)に従った原点復帰の後に再開されます。	
2	MC_Home タスクは、以下のモーションコントロールタスクによって中止できます。 MC_Home タスク Mode = 2、3: この新しい MC_Home タスクは、次のアクティブモーションコントロールタスクを中止します。 MC_Home タスク Mode = 2: 位置関連モーションタスクは、新しい原点復帰位置(Position 入力パラメータの値)に従った原点復帰の後に再開されます。	
3	MC_Home タスクは、以下のモーションコントロールタスクによって中止できます。 <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	この新しい MC_Home タスクは、以下のアクティブモーションコントロールタスクを中止します。 <ul style="list-style-type: none"> • MC_Home Mode = 2、3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

10.7.5 MC_Halt (軸の一時停止)命令

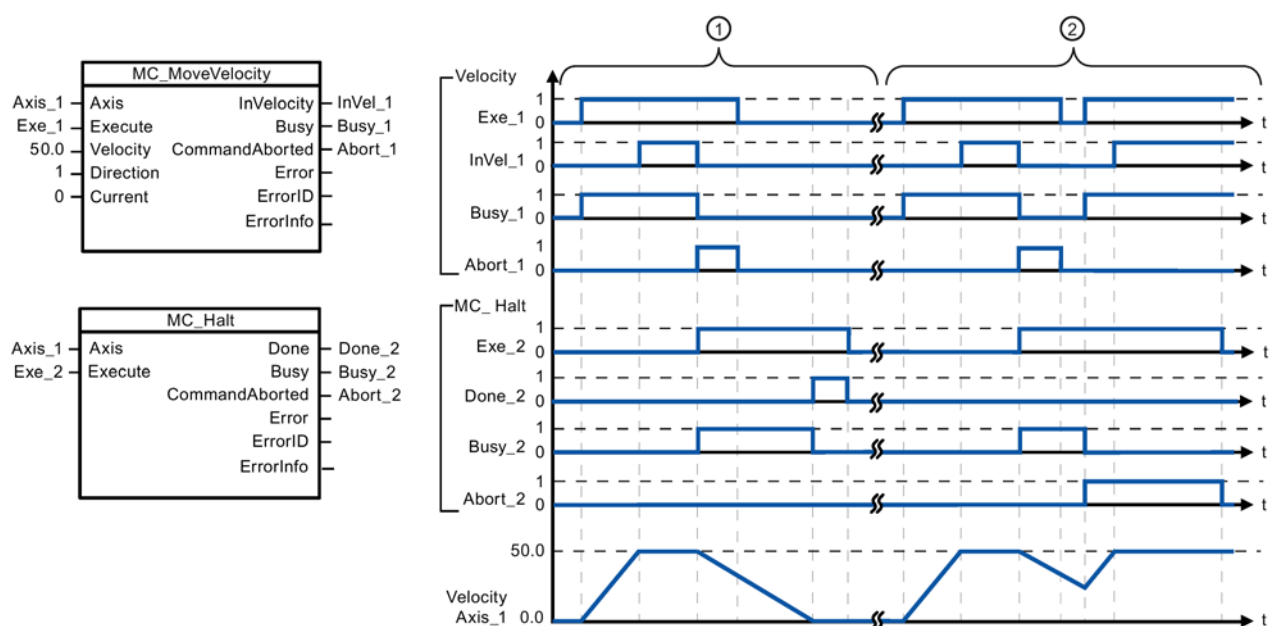
表 10- 34 MC_Halt 命令

LAD / FBD	SCL	説明
	<pre>"MC_Halt_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_Halt 命令を使用して、すべてのモーションを停止し、軸を停止します。軸の停止位置は定義されていません。</p> <p>MC_Halt 命令を使用するには、最初に軸を有効化する必要があります。</p>

- ¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。
- ² この SCL の例では、「MC_Halt_DB」がインスタンス DB の名前です。

表 10- 35 MC_Halt 命令のパラメータ

パラメータとタイプ	データタイプ	説明
Axis	IN	TO_Axis_1
Execute	IN	Bool
Done	OUT	Bool
Busy	OUT	Bool
CommandAborted	OUT	Bool
Error	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word



[ダイナミクス|全般]構成ウィンドウで、値: 加速 = 10.0 および減速 = 5.0 が設定されました。

- ① 軸が、MC_Halt タスクによって、停止するまで制動されます。軸の停止は、「Done_2」によって通知されます。
- ② MC_Halt タスクが軸を制動中に、このタスクが別のモーションタスクによって中止されています。この中止は、「Abort_2」によって通知されます。

応答の無効化

MC_Halt タスクは、以下のモーションコントロールタスクによって中止できます。

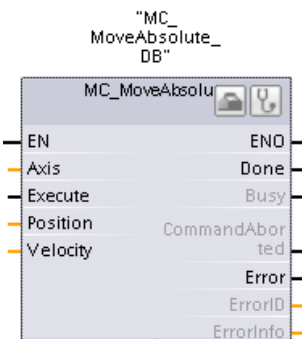
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

この新しい MC_Halt タスクは、以下のアクティブモーションコントロールタスクを中止します。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.6 MC_MoveAbsolute (絶対値位置決め)命令

表 10- 36 MC_MoveAbsolute 命令

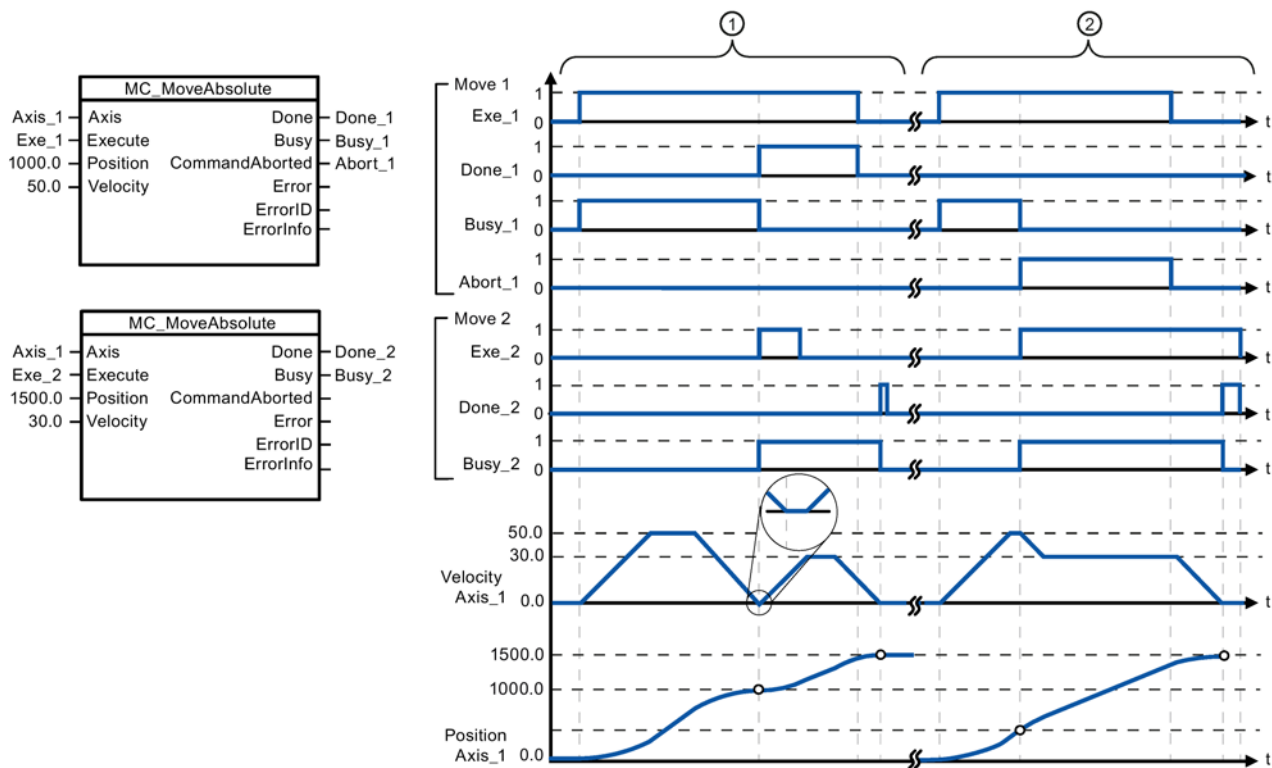
LAD / FBD	SCL	説明
	<pre>"MC_MoveAbsolute_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Velocity:=_real_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_MoveAbsolute 命令を使用して、絶対位置までの軸の位置決めモーションを開始します。</p> <p>MC_MoveAbsolute 命令を使用するには、最初に軸を有効化した後、軸を原点復帰する必要があります。</p>

¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_MoveAbsolute_DB」がインスタンス DB の名前です。

表 10- 37 MC_MoveAbsolute 命令のパラメータ

パラメータとタイプ	データタイプ	説明
Axis	IN	TO_Axis_1 軸テクノロジーオブジェクト
Execute	IN	Bool 立ち上がりエッジによるタスクの開始(既定値: False)
Position	IN	Real 絶対ターゲット位置(既定値: 0.0) 限界値: $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	IN	Real 軸の速度(既定値: 10.0) 設定された加速および減速、アプローチするターゲット位置のために、この速度に常に達するわけではありません。 限界値: 開始/停止速度 $\leq \text{Velocity} \leq$ 最大速度
Done	OUT	Bool TRUE = 絶対ターゲット位置に達しました
Busy	OUT	Bool TRUE = このタスクが実行中です。
CommandAborted	OUT	Bool TRUE = 実行中にタスクが別のタスクによって中止されました。
Error	OUT	Bool TRUE = このタスクの実行中にエラーが発生しました。エラーの原因は、パラメータ「ErrorID」および「ErrorInfo」を参照してください。
ErrorID	OUT	Word パラメータ「Error」のエラーID (既定値: 0000)
ErrorInfo	OUT	Word パラメータ「ErrorID」のエラー情報 ID (既定値: 0000)



[ダイナミクス|全般]構成ウィンドウで、値: 加速 = 10.0 および減速 = 10.0 が設定されました。

- ① 軸が、MC_MoveAbsolute タスクによって絶対位置 1000.0 まで移動します。軸がターゲット位置に達すると、これは「Done_1」によって通知されます。「Done_1」= TRUE のとき、ターゲット位置が 1500.0 の別の MC_MoveAbsolute タスクが開始されます。応答時間(たとえば、ユーザープログラムのサイクルタイムなど)のために、軸は少しの間停止します(拡大された部分を参照)。軸が新しいターゲット位置に達すると、これは「Done_2」によって通知されます。
- ② 1 つのアクティブな MC_MoveAbsolute タスクが、別の MC_MoveAbsolute タスクによって中止されます。この中止は、「Abort_1」によって通知されます。この後、軸は、新しい速度で新しいターゲット位置 1500.0 まで移動します。新しいターゲット位置に達すると、これは「Done_2」によって通知されます。

応答の無効化

MC_MoveAbsolute タスクは、以下のモーションコントロールタスクによって中止できます。

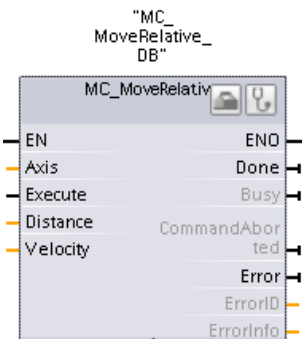
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

この新しい MC_MoveAbsolute タスクは、以下のアクティブモーションコントロールタスクを中止します。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.7 MC_MoveRelative (相対値位置決め)命令

表 10- 38 MC_MoveRelative 命令

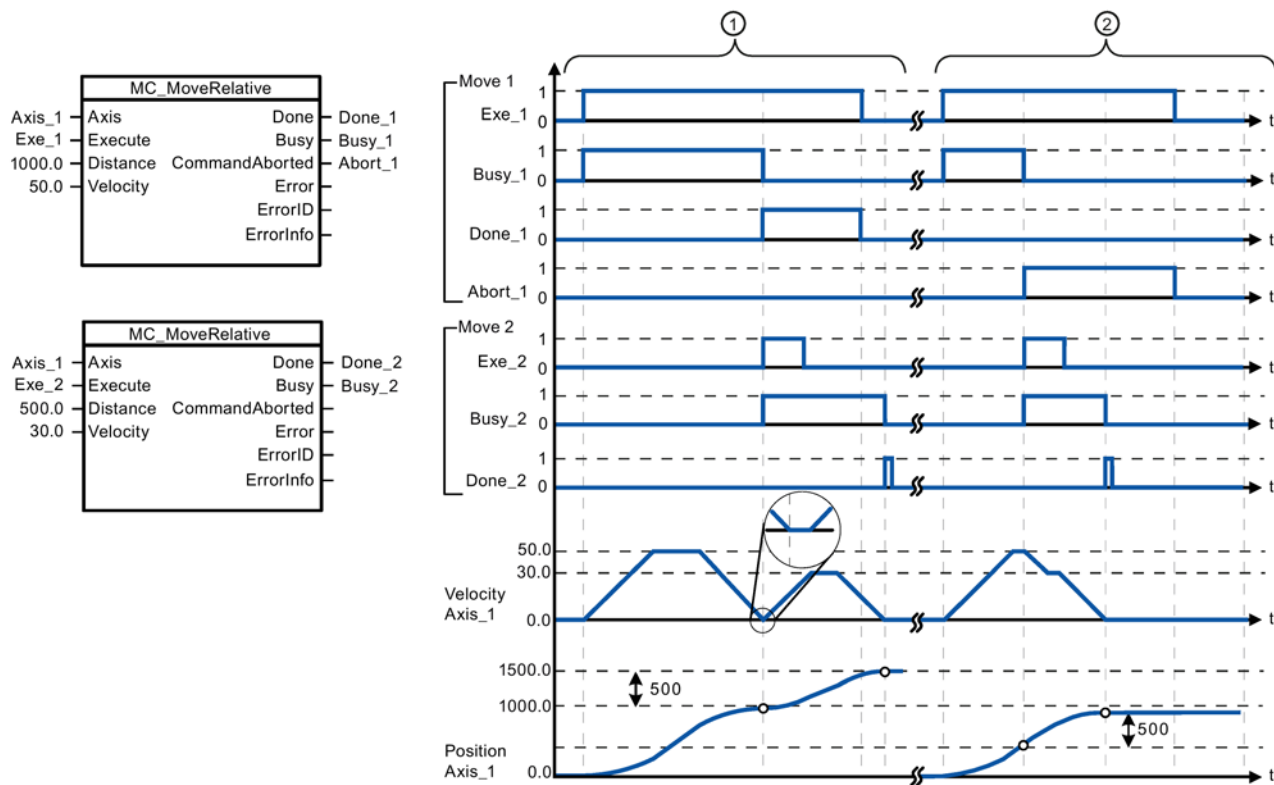
LAD / FBD	SCL	説明
	<pre>"MC_MoveRelative_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Distance:=_real_in_, Velocity:=_real_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_MoveRelative を使用して、開始位置に対する相対位置決めモーションを開始します。</p> <p>MC_MoveRelative 命令を使用するには、最初に軸を有効化する必要があります。</p>

¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_MoveRelative_DB」がインスタンス DB の名前です。

表 10- 39 MC_MoveRelative 命令のパラメータ

パラメータとタイプ	データタイプ	説明
Axis	IN	TO_Axis_1 軸テクノロジーオブジェクト
Execute	IN	Bool 立ち上がりエッジによるタスクの開始(既定値: FALSE)
Distance	IN	Real 位置決め動作の移動距離(既定値: 0.0) 限界値: $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
Velocity	IN	Real 軸の速度(既定値: 10.0) 設定された加速および減速、移動する距離のために、この速度に常に達するわけではありません。 限界値: 開始/停止速度 $\leq \text{Velocity} \leq$ 最大速度
Done	OUT	Bool TRUE = ターゲット位置に達しました
Busy	OUT	Bool TRUE = このタスクが実行中です。
CommandAborted	OUT	Bool TRUE = 実行中にタスクが別のタスクによって中止されました。
Error	OUT	Bool TRUE = このタスクの実行中にエラーが発生しました。エラーの原因は、パラメータ「ErrorID」および「ErrorInfo」を参照してください。
ErrorID	OUT	Word パラメータ「エラー(Error)」のエラー-ID (既定値: 0000)
ErrorInfo	OUT	Word パラメータ「ErrorID」のエラー情報 ID (既定値: 0000)



[ダイナミクス|全般]構成ウィンドウで、値: 加速 = 10.0 および減速 = 10.0 が設定されました。

- ① 軸が、MC_MoveRelative タスクによって距離(「Distance」) 1000.0 だけ移動します。軸がターゲット位置に達すると、これは「Done_1」によって通知されます。「Done_1」= TRUE のとき、移動距離が 500.0 の別の MC_MoveRelative タスクが開始されます。応答時間(たとえば、ユーザープログラムのサイクルタイムなど)のために、軸は少しの間停止します(拡大された部分を参照)。軸が新しいターゲット位置に達すると、これは「Done_2」によって通知されます。
- ② 1 つのアクティブな MC_MoveRelative タスクが、別の MC_MoveRelative タスクによって中止されます。この中止は、「Abort_1」によって通知されます。この後、軸は、新しい速度で新しい距離(「Distance」) 500.0 だけ移動します。新しいターゲット位置に達すると、これは「Done_2」によって通知されます。

応答の無効化

MC_MoveRelative タスクは、以下のモーションコントロールタスクによって中止できます。

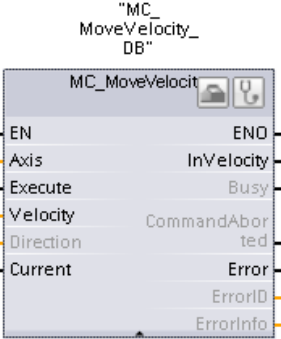
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

この新しい MC_MoveRelative タスクは、以下のアクティブモーションコントロールタスクを中止します。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.8 MC_MoveVelocity (速度制御)命令

表 10- 40 MC_MoveVelocity 命令

LAD / FBD	SCL	説明
	<pre>"MC_MoveVelocity_DB"(Axis:=_multi_fb_in_, Execute:=_bool_in_, Velocity:=_real_in_, Direction:=_int_in_, Current:=_bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_MoveVelocity 命令を使用して、軸を指定された速度で常に移動します。</p> <p>MC_MoveVelocity 命令を使用するには、最初に軸を有効化する必要があります。</p>

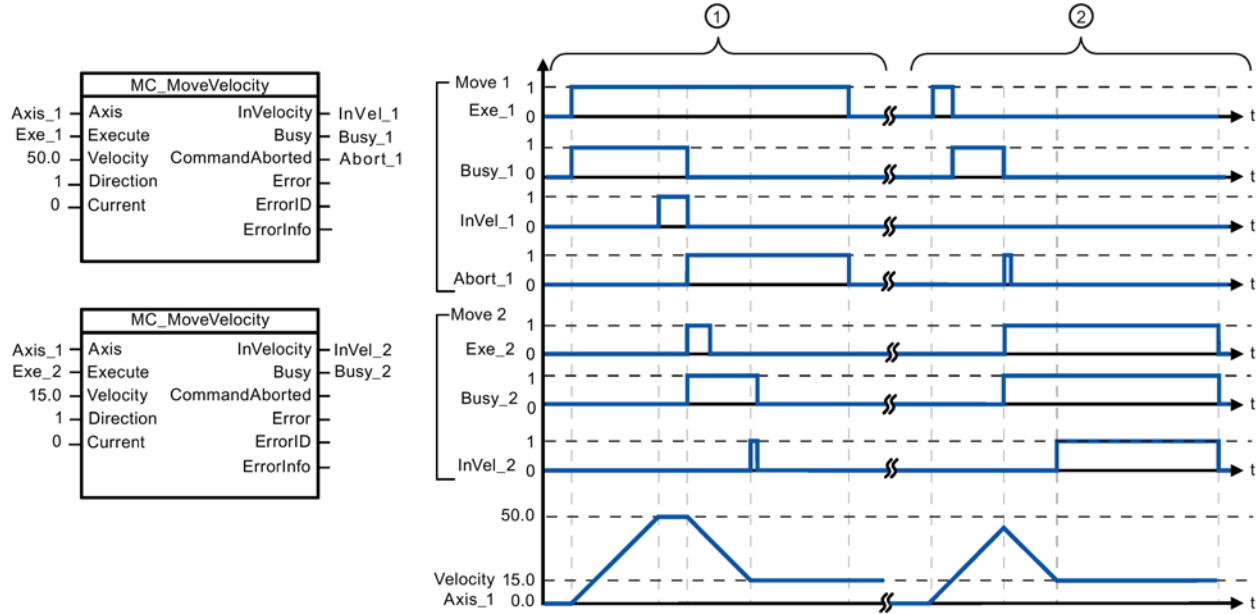
¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_MoveVelocity_DB」がインスタンス DB の名前です。

表 10- 41 MC_MoveVelocity 命令のパラメータ

パラメータとタイプ	データタイプ	説明
Axis	IN	TO_Axis_1
Execute	IN	Bool
Velocity	IN	Real
Direction	IN	Int
Current	IN	Bool
InVelocity	OUT	Bool

パラメータとタイプ	データタイプ	説明
Busy	OUT	Bool
CommandAborted	OUT	Bool
Error	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word



[ダイナミクス|全般]構成ウィンドウで、値: 加速 = 10.0 および減速 = 10.0 が設定されました。

- ① 1つのアクティブな MC_MoveVelocity タスクが、そのターゲット速度に達したことを、「InVel_1」によって通知します。この後、このタスクが別の MC_MoveVelocity タスクによって中止されます。この中止は、「Abort_1」によって通知されます。新しいターゲット速度 15.0 に達すると、これは「InVel_2」によって通知されます。この後、軸はこの新しい一定の速度で移動し続けます。
- ② 1つのアクティブな MC_MoveVelocity タスクが、そのターゲット速度に達する前に、別の MC_MoveVelocity タスクによって中止されます。この中止は、「Abort_1」によって通知されます。新しいターゲット速度 15.0 に達すると、これは「InVel_2」によって通知されます。この後、軸はこの新しい一定の速度で移動し続けます。

応答の無効化

MC_MoveVelocity タスクは、以下のモーションコントロールタスクによって中止できます。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

この新しい MC_MoveVelocity タスクは、以下のアクティブモーションコントロールタスクを中止します。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

注記**速度としてゼロが設定された(Velocity = 0.0)場合の動作**

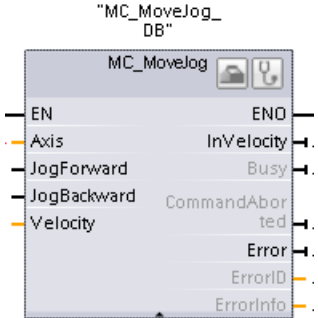
「Velocity」 = 0.0 (MC_Halt タスクなどの場合のように)の MC_MoveVelocity タスクはアクティブモーションタスクを中止し、軸を設定された減速で停止します。軸が停止すると、出力パラメータ「InVelocity」は、少なくとも 1 プログラムサイクルの間、TRUE を示します。

「Busy」は、減速動作中は値 TRUE を示しますが、「InVelocity」と一緒に FALSE に変わります。パラメータ「Execute」 = TRUE がセットされると、「InVelocity」および「Busy」はラッチされます。

MC_MoveVelocity タスクが開始されると、テクノロジーオブジェクトのステータスビット「SpeedCommand」がセットされます。軸停止時には、ステータスビット「ConstantVelocity」がセットされます。両方のビットは、新しいモーションタスクが開始されるとき、新しい状況に合わせて調整されます。

10.7.9 MC_MoveJog (ジョグモード)命令

表 10- 42 MC_MoveJog 命令

LAD / FBD	SCL	説明
	<pre>"MC_MoveJog_DB" (Axis:=_multi_fb_in_, JogForward:=_bool_in_, JogBackward:=_bool_in_, Velocity:=_real_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>MC_MoveJog 命令を使用して、軸を常に指定された速度でジョグモードで移動します。この命令は、通常、テストおよびコミッショニングのために使用されます。</p> <p>MC_MoveJog 命令を使用するには、最初に軸を有効化する必要があります。</p>

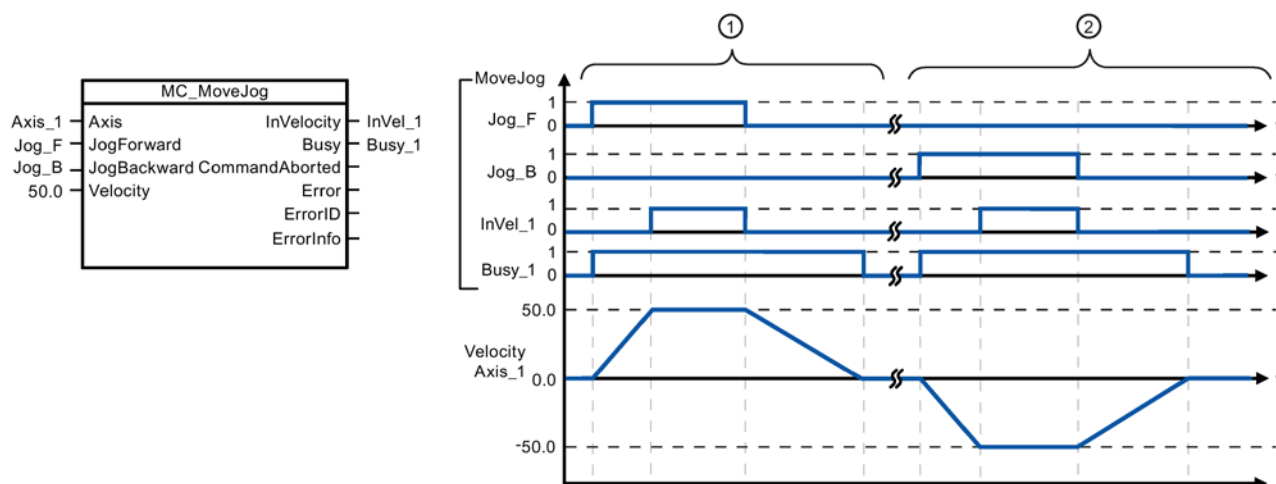
¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_MoveJog_DB」がインスタンス DB の名前です。

表 10- 43 MC_MoveJog 命令のパラメータ

パラメータとタイプ	データタイプ	説明
Axis	IN	TO_Axis_1 軸テクノロジーオブジェクト
JogForward ¹	IN	このパラメータが TRUE であるかぎり、軸はパラメータ「Velocity」で指定された速度で正の方向に移動します。パラメータ「Velocity」の値の符号は無視されます。(デフォルト値: FALSE)
JogBackward ¹	IN	このパラメータが TRUE であるかぎり、軸はパラメータ「Velocity」で指定された速度で負の方向に移動します。パラメータ「Velocity」の値の符号は無視されます。(デフォルト値: FALSE)
Velocity	IN	ジョグモード用の事前に設定された速度(既定値: 10.0) 限界値: 開始/停止速度 ≤ Velocity ≤ 最大速度
InVelocity	OUT	TRUE = パラメータ「Velocity」で指定された速度に達しました。
Busy	OUT	TRUE = このタスクが実行中です。
CommandAborted	OUT	TRUE = 実行中にタスクが別のタスクによって中止されました。
Error	OUT	TRUE = このタスクの実行中にエラーが発生しました。エラーの原因は、パラメータ「ErrorID」および「ErrorInfo」を参照してください。
ErrorID	OUT	パラメータ「エラー(Error)」のエラーID (既定値: 0000)
ErrorInfo	OUT	パラメータ「ErrorID」のエラー情報 ID (既定値: 0000)

¹ JogForward パラメータと JogBackward パラメータの両方が同時に TRUE の場合、軸は、設定された減速で停止します。エラーは、パラメータ「Error」、「ErrorID」、および「ErrorInfo」に示されます。



[ダイナミクス|全般]構成ウィンドウで、値: 加速 = 10.0 および減速 = 5.0 が設定されました。

- ① 軸は、「Jog_F」によって、ジョグモードで正の方向に移動します。ターゲット速度 50.0 に達すると、これは「InVel_1」によって通知されます。Jog_F がリセットされた後、軸は再び停止するまで制動されます。
- ② 軸は、「Jog_B」によって、ジョグモードで負の方向に移動します。ターゲット速度 50.0 に達すると、これは「InVel_1」によって通知されます。Jog_B がリセットされた後、軸は再び停止するまで制動されます。

応答の無効化

MC_MoveJog タスクは、以下のモーションコントロールタスクによって中止できます。

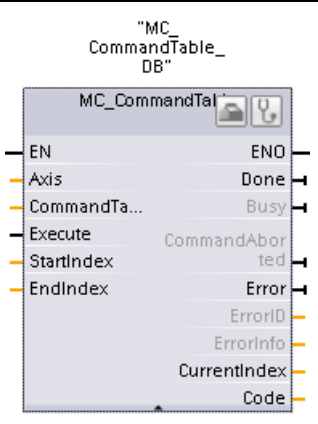
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

この新しい MC_MoveJog タスクは、以下のアクティブモーションコントロールタスクを中止します。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.10 MC_CommandTable (複数の軸コマンドを移動シーケンスとして実行)命令

表 10- 44 MC_CommandTable 命令

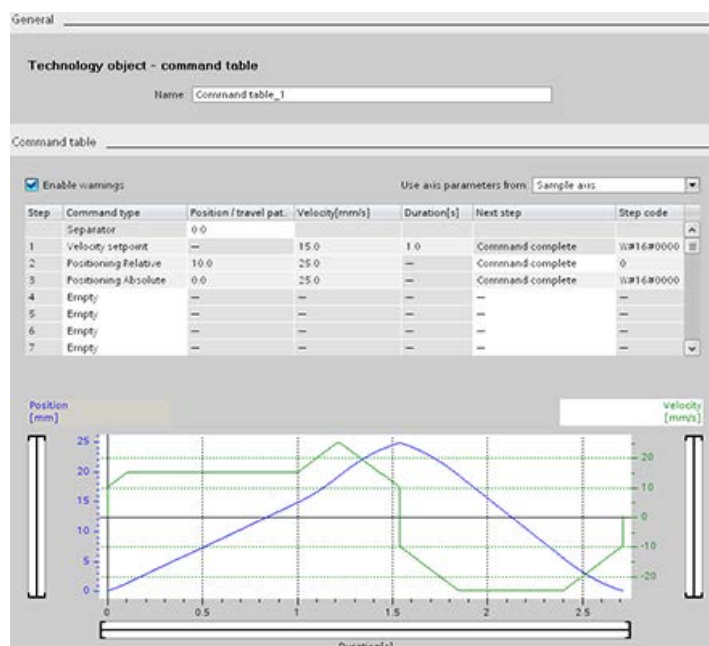
LAD / FBD	SCL	説明
	<pre> "MC_CommandTable_DB" (Axis:=_multi_fb_in_, CommandTable:=_multi_fb_in_, Execute:=_bool_in_, StartIndex:=_uint_in_, EndIndex:=_uint_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_, CurrentIndex=>_uint_out_, Code=>_word_out_); </pre>	<p>1つの移動シーケンスとして結合できるモータ制御軸の一連の個々のモーションを実行します。個々のモーションは、パルス列出力用テクノロジーオブジェクトコマンドテーブル (TO_CommandTable_PTO)に設定されます。</p>

- ¹ この命令を挿入すると、STEP 7は自動的に DB を作成します。
- ² この SCL の例では、「MC_CommandTable_DB」がインスタンス DB の名前です。

表 10- 45 MC_CommandTable 命令のパラメータ

パラメータとタイプ	データタイプ	初期値	説明
Axis	IN	TO_Axis_1	-
Table	IN	TO_CommandTable_1	-
Execute	IN	Bool	FALSE
StartIndex	IN	Int	1
EndIndex	IN	Int	32
Done	OUT	Bool	FALSE
Busy	OUT	Bool	FALSE
CommandAborted	OUT	Bool	FALSE
Error	OUT	Bool	FALSE
ErrorID	OUT	Word	16#0000
ErrorInfo	OUT	Word	16#0000
Step	OUT	Int	0
Code	OUT	Word	16#0000

[コマンドテーブル]構成ウィンドウで必要な移動シーケンスを作成し、その結果をトレンドダイアグラムのグラフィックビューと比較してチェックします。



コマンドテーブルの処理で使用するコマンドタイプを選択できます。最大 32 のジョブを入力できます。コマンドは順々に処理されます。

表 10- 46 MC_CommandTable コマンドタイプ

コマンドタイプ	説明
空	空タイプは、任意のコマンドを追加するプレースホルダとして使用されます。空エンタリは、コマンドテーブルを処理するとき、無視されます。
一時停止	軸を一時停止します。注記: このコマンドは、「速度セットポイント」コマンドの後にのみ実行されます。
相対位置決め	距離に基づいて軸を位置決めします。このコマンドは、指定された距離および速度によって軸を移動します。
絶対位置決め	位置に基づいて軸を位置決めします。このコマンドは、指定された速度を使用して、指定された位置まで軸を移動します。
速度セットポイント	指定された速度で軸を移動します。
待機	指定された時間間隔が経過するまで待機します。「待機」は、有効な移動モーションを停止しません。
セパレータ	選択した行の上に 1 つの「セパレータ」行を追加します。セパレータ行によって、単一のコマンドテーブル内に複数のプロファイルを定義できます。

MC_CommandTable 実行の前提条件:

- テクノロジーオブジェクト TO_Axis_PTO V2.0 を正しく構成する必要があります。
- テクノロジーオブジェクト TO_CommandTable_PTO を正しく構成する必要があります。
- 軸を解放する必要があります。

応答の無効化

MC_CommandTable タスクは、以下のモーションコントロールタスクによって中止できます。

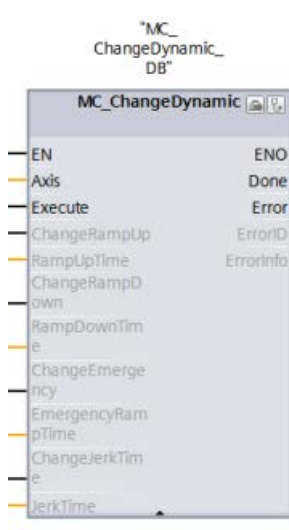
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

この新しい MC_CommandTable タスクは、以下のアクティブモーションコントロールタスクを中止します。

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable
- 最初の「相対位置決め」、「絶対位置決め」、「速度セットポイント」、または「一時停止」コマンドの起動による現在のモーションコントロールジョブ

10.7.11 MC_ChangeDynamic (軸の動的設定の変更)命令

表 10- 47 MC_ChangeDynamic 命令

LAD / FBD	SCL	説明
	<pre>"MC_ChangeDynamic_DB"(Execute:=_bool_in_, ChangeRampUp:=_bool_in_, RampUpTime:=_real_in_, ChangeRampDown:=_bool_in_, RampDownTime:=_real_in_, ChangeEmergency:=_bool_in_, EmergencyRampTime:=_real_in_, ChangeJerkTime:=_bool_in_, JerkTime:=_real_in_, Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>モーションコントロール軸の動的な設定を変更します。</p> <ul style="list-style-type: none">• ランプアップ時間(加速)値の変更• ランプダウン時間(減速)値の変更• 緊急停止ランプダウン時間(緊急停止減速)値の変更• 平滑化時間(ジャーク)値の変更

¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。
² この SCL の例では、「MC_ChangeDynamic_DB」がインスタンス DB の名前です。

表 10- 48 MC_ChangeDynamic 命令のパラメータ

パラメータとタイプ		データタイプ	説明
Axis	IN	TO_Axis_1	軸テクノロジーオブジェクト
Execute	IN	Bool	立ち上がりエッジによるコマンドの開始。既定値: FALSE
ChangeRampUp	IN	Bool	TRUE = 入力パラメータ「RampUpTime」に従って、ランプアップ時間を変更します。既定値: FALSE
RampUpTime	IN	Real	停止から設定された最大速度までジャーク制限なしで加速する時間(秒単位)。既定値: 5.00 この変更は、タグ<軸名>. Config.DynamicDefaults.Acceleration に影響します。この変更の有効性は、このタグの説明部に表示されます。
ChangeRampDown	IN	Bool	TRUE = 入力パラメータ「RampDownTime」に従って、ランプダウン時間を変更します。既定値: FALSE
RampDownTime	IN	Real	設定された最大速度から停止までジャーク制限なしで軸を減速する時間(秒単位)。既定値: 5.00 この変更は、タグ<軸名>. Config.DynamicDefaults.Deceleration に影響します。この変更の有効性は、このタグの説明部に表示されます。
ChangeEmergency	IN	Bool	TRUE = 入力パラメータ「EmergencyRampTime」に従って、緊急停止ランプダウン時間を変更します。既定値: FALSE
EmergencyRampTime	IN	Real	設定された最大速度から停止まで緊急停止モードで、ジャーク制限なしで軸を減速する時間(秒単位)。既定値: 2.00 この変更は、タグ<軸名>. Config.DynamicDefaults.EmergencyDeceleration に影響します。この変更の有効性は、このタグの説明部に表示されます。
ChangeJerkTime	IN	Bool	TRUE = 入力パラメータ「JerkTime」に従って、平滑化時間を変更します。既定値: FALSE
JerkTime	IN	Real	軸加速および減速ランプで使用される平滑化時間(秒単位)。既定値: 0.25 この変更は、タグ<軸名>. Config.DynamicDefaults.Jerk に影響します。この変更の有効性は、このタグの説明部に表示されます。
Done	OUT	Bool	TRUE = 変更された値がテクノロジーデータブロックに書き込まれました。変更が有効になると、タグの説明部が表示されます。既定値: FALSE
Error	OUT	Bool	TRUE = このコマンドの実行中にエラーが発生しました。エラーの原因は、パラメータ「ErrorID」および「ErrorInfo」を参照してください。既定値: FALSE
ErrorID	OUT	Word	エラーID。既定値: 16#0000
ErrorInfo	IN	Word	エラー情報。既定値: 16#0000

MC_ChangeDynamic 実行の前提条件:

- テクノロジーオブジェクト TO_Axis_PTO V2.0 を正しく構成する必要があります。
- 軸を解放する必要があります。

応答の無効化

MC_ChangeDynamic コマンドは、他のすべてのモーションコントロールコマンドによって中止できません。

新しい MC_ChangeDynamic コマンドは、すべてのアクティブモーションコントロールジョブを中止しません。

注記

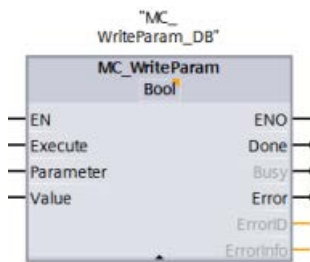
入力パラメータ「RampUpTime」、「RampDownTime」、「EmergencyRampTime」、および「RoundingOffTime」には、結果としての軸パラメータ「加速」、「遅延」、「緊急停止遅延」、および「ジャーク」を許可された限界値外にする値を指定できます。

MC_ChangeDynamic パラメータを、軸テクノロジーオブジェクトの動的な設定の限界値内に必ず保持するようにしてください。

10.7.12 MC_WriteParam (テクノロジーオブジェクトのパラメータを書き込み)命令

MC_WriteParam 命令を使用して、ユーザープログラムから、軸の機能を変更するパラメータの選択番号を書き込みます。

表 10- 49 MC_WriteParam 命令

LAD / FBD	SCL	説明
	<pre>"MC_WriteParam_DB"(Parameter:=_variant_in_, Value:=_variant_in_, Execute:=_bool_in_, Done:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	MC_WriteParam 命令を使用して、パブリックパラメータ(たとえば、加速およびユーザーDB の値など) への書き込みを行います。

- 1

この命令を挿入すると、STEP 7 は自動的に DB を作成します。
- 2

この SCL の例では、「MC_WriteParam_DB」がインスタンス DB の名前です。
- パブリックパラメータへの書き込みを行うことができます。「MotionStatus」および「StatusBits」へ書き込むことはできません。下の表に、有効なパラメータを記載します。

書き込み可能なパラメータの名前	書き込み可能なパラメータの名前
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter.PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration
Sensor[1].ActiveHoming.SideInput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel

書き込み可能なパラメータの名前	書き込み可能なパラメータの名前
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.SideInput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

表 10- 50 MC_WriteParam 命令のパラメータ

パラメータとタイプ	データタイプ	説明
PARAMNAME	IN	Variant
VALUE	IN	Variant
EXECUTE	IN	Bool
DONE	OUT	Bool
BUSY	OUT	Bool
ERROR	OUT	REAL
ERRORID	OUT	Word
ERRORINFO	OUT	Word

表 10- 51 ERRORID および ERRORINFO の条件コード

ERRORID (W#16#...)	ERRORINFO (W#16#...)	説明
0	0	Axis TO-DB パラメータの変更が成功しました
8410 _[1]	0028 _[1]	無効なパラメータ(長さが不正の Axis TO-DB パラメータ)の設定
8410 _[1]	0029 _[1]	無効なパラメータ (Axis TO-DB パラメータなし)の設定
8410 _[1]	002B _[1]	無効なパラメータ(呼び出し専用の Axis TO-DB パラメータ)の設定
8410 _[1]	002C _[1]	有効なパラメータの設定ですが、軸が無効化されていません
Config Error _[2]	Config Error _[2]	有効なパラメータ(パブリックな呼び出し専用 Axis TO-DB パラメータ)の設定ですが、範囲外です
Config Error _[3]	Config Error _[3]	有効なパラメータ(パブリックな Axis TO-DB パラメータ)の設定ですが、範囲外です

[1] MC_WriteParam のエラー

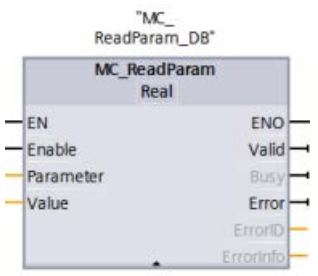
[2] MC_Power のエラー

[3] MC_Power および MC_MoveXXX か、MC_CommandTable のエラー

10.7.13 MC_ReadParam (テクノロジーオブジェクトのパラメータを読み取り)命令

MC_ReadParam 命令を使用して、軸入力で定義された軸の現在の位置、速度などを示すパラメータの選択番号を読み出します。

表 10- 52 MC_ReadParam 命令

LAD / FBD	SCL	説明
	<pre>"MC_ReadParam_DB" (Enable:=_bool_in_, Parameter:=_variant_in_, Value:=_variant_in_out_, Valid:=_bool_out_, Busy:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>MC_ReadParam 命令を使用して、サイクル制御点に関わりなく、単一のステータス値を読み出します。</p>

¹ この命令を挿入すると、STEP 7 は自動的に DB を作成します。

² この SCL の例では、「MC_ReadParam_DB」がインスタンス DB の名前です。

MC_ReadParam 命令は、イネーブル動作で機能します。入力「Enable」が TRUE であるかぎり、この命令は指定された「パラメータ」を「値」格納場所へ読み出します。

「MotionStatus」「Position」値が、各サイクル制御点(CCP)で、現在の HSC 値に基づいて更新されます。

「MotionStatus」「Velocity」値は、現在のセグメントの終わりのコマンド速度です(10 ミリ秒ごとに更新されます)。MC_ReadParam は、この値も読み出すことができます。

エラーが発生すると、この命令は、入力「Enable」での新しい立ち上がりエッジによってのみリセットできるエラー状態に切り替わります。

表 10- 53 MC_ReadParam 命令のパラメータ

パラメータとタイプ	データタイプ	説明
ENABLE	IN	Bool
PARAMETER	IN	Variant
VALID	OUT	Bool
BUSY	OUT	Bool
ERROR	OUT	Real
ERRORID	OUT	Word
ERRORINFO	OUT	Word
VALUE	INOUT	Variant

表 10- 54 ERRORID および ERRORINFO の条件コード

ERRORID (W#16#...)	ERRORINFO (W#16#...)	説明
0	0	パラメータの読み出しが成功
8410	0028	無効なパラメータ(不正な長さ)
8410	0029	無効なパラメータ(TO-DB なし)
8410	0030	無効なパラメータ(読み出しできません)
8411	0032	無効なパラメータ(不正な値)

TO パラメータ

軸「MotionStatus」は 4 つの値から構成されます。これらの値の変更をモニタする必要がある場合、プログラムが実行中でも読み出すことができます。

変数名	データタイプ	MC_ReadParam によって読み出し可能
MotionStatus:	Structure	なし
• Position	REAL	あり
• Velocity	REAL	あり
• Distance	REAL	あり
• TargetPosition	REAL	あり

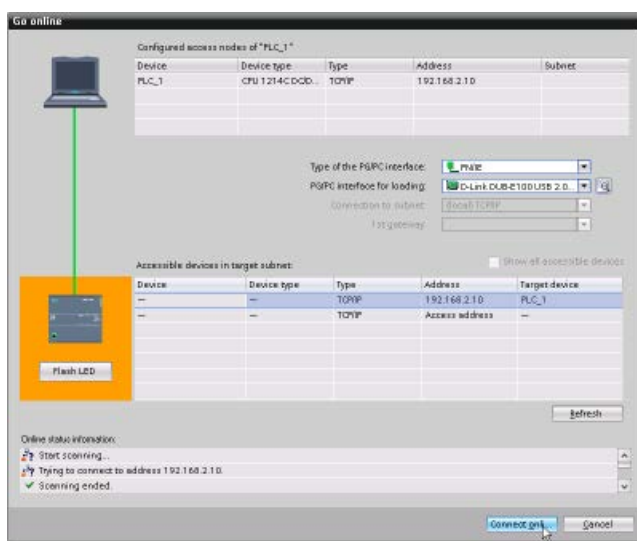
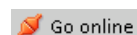
カンタン オンライン操作

11.1 オンラインで CPU に接続

プログラムおよびプロジェクトエンジニアリングデータのロードと、以下に示すような作業を行うために、プログラミングデバイスと CPU の間のオンライン接続を確立する必要があります。

- ユーザープログラムのテスト
- CPU の動作モードの表示と変更 (Page 338)
- CPU の日付と時刻の表示と設定 (Page 348)
- モジュール情報の表示
- オフラインプログラムブロックとオンラインプログラムブロックの比較と同期化 (Page 347)
- プログラムブロックのアップロードとダウンロード
- 診断と診断バッファの表示 (Page 348)
- 値のモニタおよび変更によってユーザープログラムをテストするためのウォッチテーブルの使用 (Page 340)
- CPU で値を強制するためのフォーステーブルの使用 (Page 342)

構成された 1 つの CPU へのオンライン接続を確立するために、プロジェクトナビゲーションツリーでその CPU をクリックし、プロジェクトビューで[オンライン接続]ボタンをクリックします。



初めてこの CPU とオンライン接続する場合は、PG/PC インターフェースで検出された CPU へのオンライン接続を確立する前に、[オンライン接続]ダイアログで、PG/PC インターフェースのタイプと特定の PG/PC インターフェースを選択する必要があります。

これで、ユーザーのプログラミングデバイスがこの CPU に接続されました。オレンジ色のフレームがオンライン接続を示します。この後、プロジェクトツリーのオンライン診断と、オンラインツールタスクカードを使用できます。

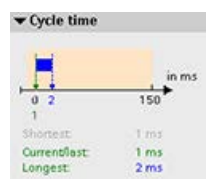
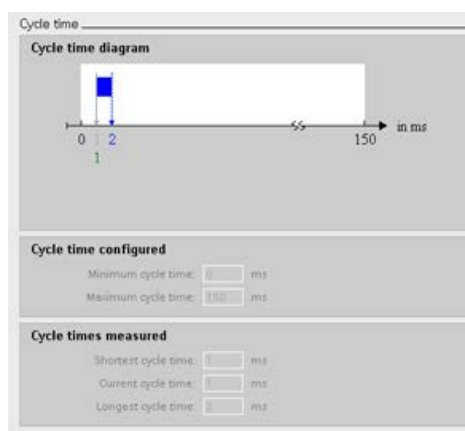
11.2 オンラインの CPU との相互作用

プロジェクトビューの[オンラインツール]タスクカードには、オンライン CPU の動作モードを示すオペレータパネルが表示されます。オペレータパネルで、オンライン CPU の動作モードも変更できます。動作モード(STOP または RUN)を変更するには、オペレータパネルのボタンを使用します。オペレータパネルには、メモリのリセットのための MRES ボタンもあります。

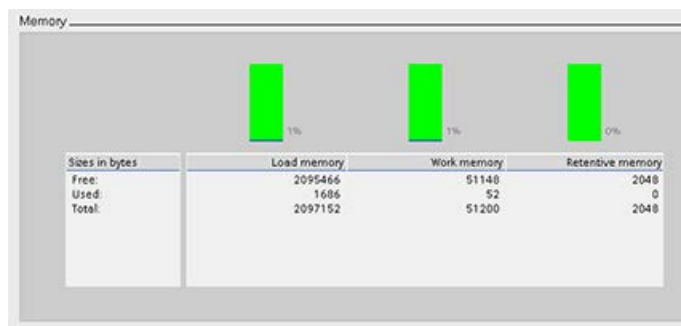


RUN/STOP インジケータの色は、CPU の現在の動作モードを示します。黄色は STOP モード、緑色は RUN モードを示します。

オペレータパネルを使用するには、STEP 7 と CPU の間のオンライン接続を確立する必要があります。デバイス構成で CPU を選択するか、オンライン CPU のコードブロックを表示した後、[オンラインツール]タスクカードからオペレータパネルを表示できます。



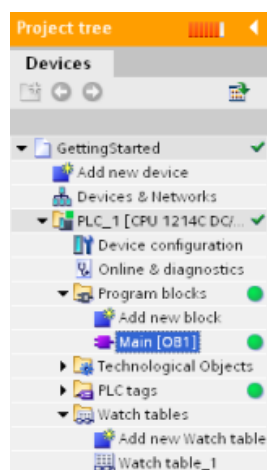
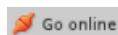
オンライン CPU のサイクルタイムをモニタできます。



CPU のメモリ使用率も表示できます。

11.3 オンラインで CPU の値をモニタ

タグをモニタするには、CPU へのオンライン接続が必要です。ツールバーの[オンライン接続]ボタンをクリックするだけです。



CPU へ接続すると、STEP 7 はワークエリアのヘッダーをオレンジ色に変えます。

プロジェクトツリーに、オフラインプロジェクトとオンライン CPU の比較が表示されます。緑色のボールは、CPU とプロジェクトが同期化されていて、両方が同一の構成およびユーザープログラムを持つことを示します。

タグテーブルにはタグが表示されます。ウォッチテーブルにもタグと直接アドレスを表示できます。

GettingStarted_1 ▸ PLC_1 ▸ Watch tables ▸ Watch table_1					
	Name	Address	Display format	Monitor value	Modify value
1	*On	%I 0.0	Bool		
2	*Off	%I 0.1	Bool		
3	*Run	%Q 0.0	Bool		

ユーザープログラムの実行をモニタし、タグの値を表示するには、ツールバーの[すべてモニタ]ボタンをクリックします。

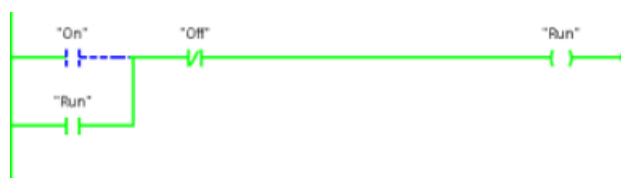
GettingStarted_1 ▸ PLC_1 ▸ Watch tables ▸ Watch table_1					
	Name	Address	Display format	Monitor value	Modify value
1	*On	%I 0.0	Bool	<input type="checkbox"/> FALSE	
2	*Off	%I 0.1	Bool	<input type="checkbox"/> FALSE	
3	*Run	%Q 0.0	Bool	<input type="checkbox"/> FALSE	

[モニタ値]フィールドには、各タグの値が表示されます。

11.4 ユーザープログラムのステータスの表示が簡単

LAD および FBD プログラミングエディタで、最大 50 のタグのステータスをモニタできます。エディタバーを使用して、LAD エディタを表示します。エディタバーによって、エディタを開いたり、閉じたりすることなしに、開いているエディタ間の表示を切り替えることができます。

プログラミングエディタのツールバーで、[モニタオン/オフ]ボタンをクリックして、ユーザープログラムのステータスを表示します。



プログラミングエディタのネットワークには、パワーフローが緑色で表示されます。命令またはパラメータを右クリックして、命令用の値を変更することもできます。

11.5 ウォッチテーブルを使用して CPU をモニタリング

ウォッチテーブルを使用して、CPU がユーザープログラムを実行している間に、データポイントをモニタまたは変更することができます。データポイントとしては、入力(I)、出力(Q)、M メモリ、DB、または周辺機器入力(たとえば、「On:P」または「I 3.4:P」)などがあります。モニタファンクションは、Q メモリから書き込まれた最後の値だけを表示でき、物理的出力の現在値を読み出さないため、物理的出力(Q0.0:P など)を正確にモニタすることはできません。

モニタファンクションは、プログラムシーケンスを変更しません。CPU 内のプログラムシーケンス情報とプログラムのデータを表示します。[値の変更]ファンクションを使用して、ユーザープログラムの実行をテストすることもできます。

	Name	Address	Display format	Monitor value	Monitor with trigger	Modify with trigger	Modify value
1	"Start"	%I0.0	Bool		Permanent	Permanent	
2	"Stop"	%I0.1	Bool		Permanent	Permanent	
3	"Running"	%M0.0	Bool		Permanent	Permanent	

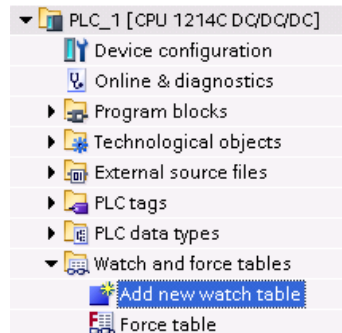
注記

高速カウンタ(HSC)、パルス幅変調(PWM)、パルストレイン出力(PTO)デバイスで使用するデジタル I/O ポイントは、デバイスコンフィグレーション中に割り付けられます。デジタル I/O ポイントアドレスがこれらのデバイスに割り当てられると、割り当てられた I/O ポイントアドレスの値は、ウォッチテーブルの「強制」ファンクションによって変更できません。

ウォッチテーブルでは、個々のタグの値をモニタまたは変更することができます。以下のオプションの 1 つを選択します。

- スキャンサイクルの開始または終了時
- CPU が STOP モードに切り替わるとき

- 「常時」(STOP から RUN への移行後に値はリセットされません)



ウォッチテーブルを作成するには

1. [新しいウォッチテーブルの作成]をダブルクリックして、新しいウォッチテーブルを開きます。
2. タグ名を入力し、ウォッチテーブルにタグを追加します。

タグをモニタするには、CPU へのオンライン接続が必要です。タグの修正では以下のオプションが使用できます。

- [今すぐ修正]では、1つのスキャンサイクルで選択されたアドレスの値をただちに変更します。
- [トリガで修正]では、選択されたアドレスの値を変更します。
このファンクションでは、選択されたアドレスが実際に修正されたかどうかを示すフィードバックはありません。変更のフィードバックが必要な場合は、[今すぐ修正]ファンクションを使用します。
- CPU が STOP モードのときに、[周辺機器出力の有効化する]を使用して、周辺機器出力をオンにすることができます。この機能は、出力モジュールの配線のテストに有効です。

ウォッチテーブルの上部のボタンを使用して、各種のファンクションを選択できます。モニタ対象のタグ名を入力し、ドロップダウンから表示フォーマットを選択します。CPU にオンライン接続した状態で[モニタ]ボタンをクリックすると、[モニタ値]フィールドのデータポイントの実際値が表示されます。

11.6 フォーステーブルの使用

フォーステーブルは、入力または出力ポイントの値を周辺機器入力または周辺機器出力アドレス用の指定された値で上書きする「強制」ファンクションを提供します。CPU は、この強制された値を、ユーザープログラムの実行の前に入力プロセスイメージに適用し、出力をモジュールに書き込む前に出力プロセスイメージに適用します。

注記

強制値は、フォーステーブルでなく、CPU に保管されます。

入力(または「I」アドレス)または出力(または「Q」アドレス)は強制できません。それに対して、周辺機器入力または周辺機器出力は強制できます。フォーステーブルでは、アドレスに「:P」が自動的に付加されます(たとえば: "On":P または "Run":P など)。

	Name	Address	Display format	Monitor value	Force value	F
1	"On":P	%I0.0.P	Bool		TRUE	<input checked="" type="checkbox"/>
2	"Off":P	%I0.1.P	Bool			<input type="checkbox"/>
3	"Run":P	%Q0.1.P	Bool			<input type="checkbox"/>

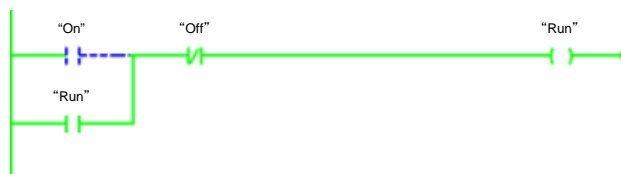
[強制値]セルに、強制する入力または出力の値を入力します。この後、[強制]列のチェックボックスを使用して、入力または出力の強制処理を有効化します。



[すべて強制]ボタンを使用して、フォーステーブルのタグの値を強制します。[強制処理を停止]ボタンをクリックして、タグの値をリセットします。

フォーステーブルでは、入力の強制された値のステータスをモニタできます。ただし、出力の強制された値はモニタできません。

プログラミングエディタで、強制された値のステータスをモニタすることもできます。



注記

フォーステーブルで入力または出力が強制されると、それらの強制アクションはプロジェクト構成の一部になります。STEP 7 を閉じて、強制されたエレメントは、クリアされるまでは、CPU プログラムで有効なままです。これらの強制されたエレメントをクリアするには、STEP 7 を使用してオンライン CPU と接続した後、フォーステーブルを使用して、それらのエレメントの強制ファンクションをオフにするか、停止する必要があります。

CPU では、フォーステーブルで物理的入力または出力アドレス(I_:P or Q_:P)を指定した後、強制ファンクションを開始することによって、入力および出力ポイント(複数可)を強制することができます。

プログラムでは、強制値で上書きされた物理入力を読み取ります。プログラムはこの強制値を使って処理を行います。プログラムが物理出力を書き込むと、出力値は強制値によって上書きされます。強制値は物理出力として表示され、処理に使用されます。

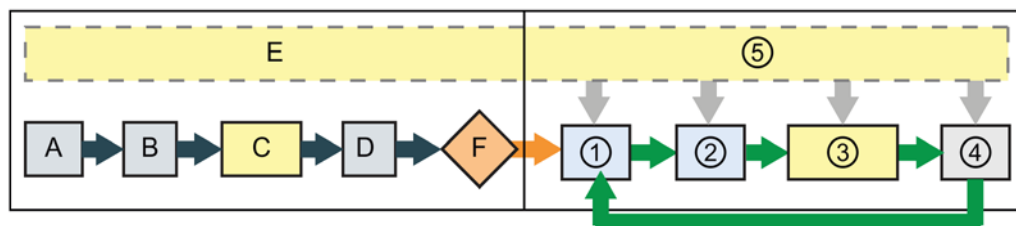
フォーステーブルで入力または出力が強制されると、これらの強制アクションはユーザープログラムの一部になります。プログラミングソフトウェアを閉じた後も、選択された強制ファンクションは動作中の CPU プログラムで有効のままになります。無効になるのは、プログラミングソフトウェアでオンライン接続して消去され、強制ファンクションが停止した場合です。強制ポイントを持ったプログラムをメモリカードから別の CPU にロードすると、プログラムで選択されたポイントが引き続き強制されます。

CPU が書き込み保護されたメモリカードからユーザープログラムを実行している場合、ウォッチテーブルから入出力の強制を開始または変更することはできません。書き込み保護されたユーザープログラムの値を上書きすることができないためです。書き込み保護された値を強制しようとする、エラーが発生します。ユーザープログラムの転送にメモリカードを使用する場合、そのメモリカード上にあるすべての強制要素は CPU に転送されます。

注記

HSC、PWM、PTO に割り付けられたデジタル I/O ポイントは強制できません

高速カウンタ(HSC)、パルス幅変調(PWM)、パルストレイン出力(PTO)デバイスで使用されるデジタル I/O ポイントは、デバイスコンフィグレーション中に割り付けられます。デジタル I/O ポイントアドレスがこれらのデバイスに割り当てられると、割り当てられた I/O ポイントアドレスの値は、フォーステーブルの強制ファンクションによって変更できません。



STARTUP

- A Iメモリ領域をクリアしても、強制ファンクションの影響は受けません。
- B 出力値の初期化は、強制ファンクションの影響を受けません。
- C スタートアップ OB の実行中に、CPU は、ユーザープログラムが物理入力にアクセスすると、強制値を適用します。
- D キューへの割り込みイベントの保存は、影響を受けません。
- E 出力への書き込み有効化は、影響を受けません。

RUN

- ① Qメモリを物理出力に書き込む間、出力が更新されると CPU は強制値を適用します。
- ② 物理入力を読み取るとき、CPU は、入力を Iメモリにコピーする直前に強制値を適用します。
- ③ ユーザープログラムの実行中に(プログラムサイクル OB)、CPU は、ユーザープログラムが物理入力にアクセスするか、または物理出力を書き込むと、強制値を適用します。
- ④ 通信要求とセルフテスト診断の処理は、強制ファンクションの影響を受けません。
- ⑤ スキャンサイクルのどの部分での割り込み処理であっても、影響は受けません。

11.7 DB のオンライン値を取得して開始値のリセット

オンライン CPU でモニタされている現在値(グローバル DB の開始値になります)を取得できます。

- CPU へオンライン接続する必要があります。
- CPU は RUN モードである必要があります。
- STEP 7 で、該当 DB を開いておく必要があります。



[モニタされた値のスナップショットの表示]ボタンを使用して、DB の選択されたタグの現在値を取得します。この後、これらの値を、DB の[開始値]列にコピーできます。

1. DB エディタで、[すべてのタグのモニタ]ボタンをクリックします。[モニタ値]列に、現在のデータ値が表示されます。
2. [モニター値のスナップショット]ボタンをクリックして、現在値を[スナップショット]列に表示します。
3. [すべてのモニタ]ボタンをクリックして、CPU のデータのモニタを停止します。
4. タグの[スナップショット]列に値をコピーします。
 - コピーする値を選択します。
 - 選択した値を右クリックして、コンテキストメニューを表示します。
 - [コピー]コマンドを選択します。
5. コピーされた値をタグの対応する[開始値]列に貼り付けます。(セルを右クリックした後、コンテキストメニューから[貼り付け]を選択します。)
6. コピーされた値を DB の新しい開始値として設定するために、プロジェクトを保存します。
7. DB をコンパイルし、CPU にダウンロードします。CPU が RUN モードに移行した後、DB は新しい開始値を使用します。

注記

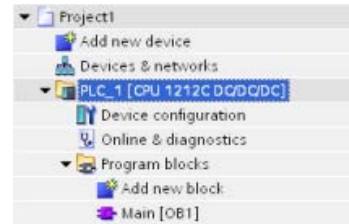
[値のモニタ]列に表示される値は、常に、CPU からコピーされます。STEP 7 は、すべての値が CPU の同一のスキャンサイクルから生じたものであるかどうかをチェックしません。

11.8 プロジェクトエレメントのアップロード

オンライン CPU から、またはプログラミングデバイスに接続されたメモリカードから、プログラムブロックをコピーすることもできます。

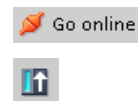
コピーされたプログラムブロックのためにオフラインプロジェクトを準備します。

1. オンライン CPU と一致する 1 つの CPU デバイスを追加します。
2. CPU ノードを展開してプログラムブロックフォルダを表示します。

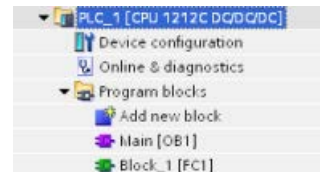


オンライン CPU からオフラインプロジェクトにプログラムブロックをアップロードするには、以下の手順に従います。

1. オフラインプロジェクトの[プログラムブロック]フォルダをクリックします。
2. [オンライン接続]ボタンをクリックします。
3. [アップロード]ボタンをクリックします。
4. [アップロード]ダイアログでユーザーの決定を確認します (Page 337)。

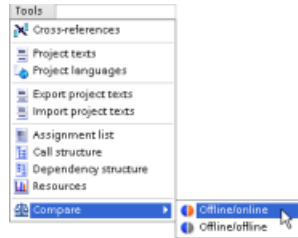


アップロードが完了すると、STEP 7 が、プロジェクトのすべてのアップロードされたプログラムブロックを表示します。



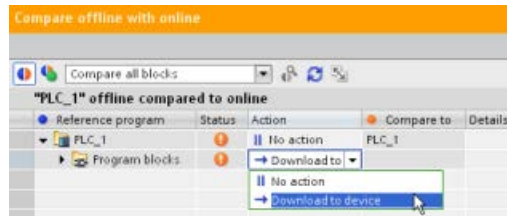
11.9 CPU のオフラインとオンライン比較

オンライン CPU のコードブロックをユーザープロジェクトのコードブロックと比較できます。ユーザープロジェクトのコードブロックがオンライン CPU のコードブロックと一致しない場合、[比較]エディタで、ユーザープロジェクトのコードブロックを CPU にダウンロードするか、オンライン CPU に存在しないプロジェクトのブロックを削除することによって、ユーザープロジェクトをオンライン CPU と同期化することができます。



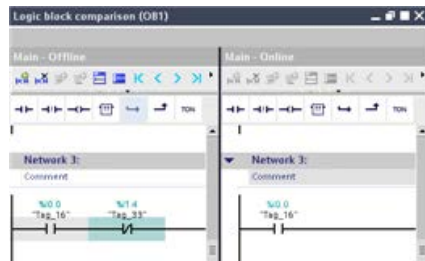
ユーザープロジェクトの CPU を選択します。

[オフライン/オンラインの比較]コマンドを使用して、[比較]エディタを開きます。(このコマンドには、[ツール]メニューを使用するか、ユーザープロジェクトの CPU を右クリックしてアクセスします。)



オブジェクトの[操作]列をクリックして、オブジェクトの削除、アクションなし、またはデバイスへのオブジェクトのダウンロードを選択します。

[操作の実行]ボタンをクリックして、コードブロックをロードします。



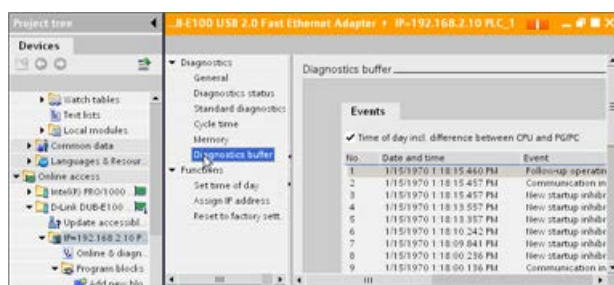
[比較先]列のオブジェクトを右クリックし、[詳細比較の開始]ボタンを選択して、コードブロックを両側に表示します。

詳細比較では、オンライン CPU のコードブロックとユーザープロジェクトの CPU のコードブロック間の相違が強調表示されます。

11.10 診断イベントの表示

CPU は、診断イベント(CPU 動作モードの移行、CPU またはモジュールによって検出されたエラーなど)ごとに 1 つのエントリを格納する診断バッファを提供します。診断バッファにアクセスするには、オンラインになっている必要があります。

各エントリには、イベントの発生日時、イベントのカテゴリ、イベントの説明が含まれています。エントリは、直近のイベントを一番上に、発生した日時順に表示されます。



CPU に電源が接続されている間、このログには最大過去 50 のイベントが表示されます。ログの容量に空きがなくなると、最も古いものが消去され、新しいものが追加されます。

電源が切断されても、過去 10 のイベントが保持されます。

11.11 IP アドレスと時刻の設定

オンライン CPU の IP アドレスと時刻を設定することができます。オンライン CPU のプロジェクトツリーから「オンライン診断」にアクセスした後、IP アドレスの表示または変更を行うことができます。オンライン CPU の時刻および日付パラメータを表示または設定することもできます。



注記

この機能は、MAC アドレスだけしか存在しないか(IP アドレスがまだ割り当てられていません)、工場出荷時設定にリセットされた CPU の場合のみ使用できます。

11.12 工場出荷時設定にリセット

S7-1200 は、以下の条件を満たす場合、元の工場出荷時設定にリセットすることができます。

- CPU がオンライン接続されていること。
- CPU が STOP モードであること。

注記

CPU が RUN モードのときにリセット操作を開始する場合、確認プロンプトを確認した後に CPU を STOP モードに切り替えることができます。

手順

CPU をその工場出荷時設定にリセットするには、以下の手順に従います。

1. CPU のオンライン&診断ビューを開きます。
2. [ファンクション]フォルダから、[出荷時設定へのリセット]を選択します。
3. IP アドレスを保持する場合、[IP アドレスの保持]チェックボックスを選択し、IP アドレスを削除する場合、[IP アドレスの削除]チェックボックスを選択します。
4. [リセット]ボタンをクリックします。
5. 確認プロンプトを[OK]で確認します。

結果

必要に応じて、モジュールが STOP モードに切り替わり、工場出荷時設定にリセットされます。CPU が以下のアクションを実行します。

メモ리카ードが CPU に挿入されている場合	メモ리카ードが CPU に挿入されていない場合
<ul style="list-style-type: none">• 診断バッファをクリアします• 時刻をリセットします• メモ리카ードによってワークメモリを復元します• すべてのオペランド領域に設定された初期値をセットします。• すべてのパラメータにそれぞれの設定値をセットします。• ユーザーが行った選択に基づいて、IP アドレスを保持または削除します。(MAC アドレスは固定で、決して変更されません。)¹• 存在する場合、制御データレコードを削除します	<ul style="list-style-type: none">• 診断バッファをクリアします• 時刻をリセットします• ワークメモリおよび内部ロードメモリをクリアします• すべてのオペランド領域に設定された初期値をセットします。• すべてのパラメータにそれぞれの設定値をセットします。• ユーザーが行った選択に基づいて、IP アドレスを保持または削除します。(MAC アドレスは固定で、決して変更されません。)¹• 存在する場合、制御データレコードを削除します

¹ [IP アドレスの保持]を選択した場合は、CPU が、IP アドレス、サブネットマスク、およびルーターアドレスの値として、ユーザーのハードウェアコンフィグレーションの設定値をセットします。これは、これらの値がユーザープログラムまたは別のツールによって変更されていない場合にかぎります。変更されている場合、CPU は変更された値を復元します。

11.13 ファームウェアの更新

オンライン STEP 7 および診断ツールから、接続された CPU のファームウェアを更新できます。

ファームウェア更新を実行するには、以下の手順に従います。

1. 接続された CPU のオンライン診断ビューを開きます。
2. [ファンクション]フォルダから[ファームウェアの更新]を選択します。
3. [参照]ボタンをクリックし、ファームウェア更新ファイルが格納された場所までナビゲートします。これは、S7-1200 ファームウェア更新ファイルを保存した場所、例えばサービス & サポート Web サイトからダウンロードし、保存したハードドライブの場所になります。(http://support.industry.siemens.com/cs/jp/ja/ps/13683/dl)
4. ユーザーのモジュールと互換性があるファイルを選択します。選択されたファイルと互換性があるモジュールが表に表示されます。
5. [更新の実行]ボタンをクリックします。それ以降、表示される複数のダイアログの指示に従い、必要に応じて、ユーザーの CPU の動作モードを変更します。

STEP 7 がファームウェア更新をロードする間、進捗ダイアログが表示されます。ファームウェア更新が終了すると、新しいファームウェアでモジュールを開始するかどうかを問うプロンプトが表示されます。

注記

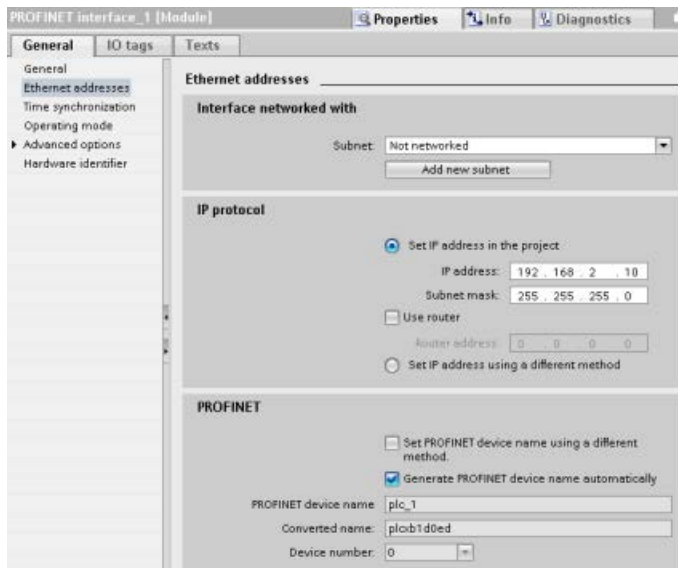
新しいファームウェアでモジュールを開始することを選択しないと、ユーザーがモジュールをリセット(たとえば、電源オフ/オンにすることによって)するまで、直前のファームウェアが有効なままです。新しいファームウェアは、ユーザーがモジュールをリセットした後にのみ、有効になります。

ファームウェア更新は、以下の追加方法の 1 つを使用して実行することもできます。

- メモリカードの使用 (Page 63)
- Web サーバー「モジュール情報」標準 Web ページの使用 (Page 256)

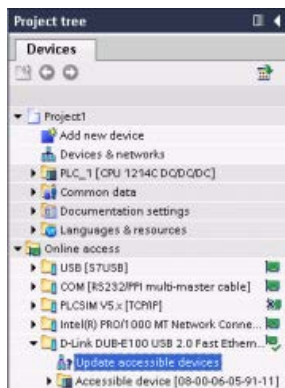
11.14 オンライン CPU に IP アドレスのダウンロード

IP アドレスを割り当てるには、以下のタスクを実行する必要があります。



- CPU 用の IP アドレスを設定します (Page 87)。
- 設定を保存し、CPU にダウンロードします。

CPU 用の IP アドレスおよびサブネットマスクは、プログラミングデバイスの IP アドレスおよびサブネットマスクと互換性がある必要があります。ユーザーの CPU 用の IP アドレスおよびサブネットマスクについては、御社関連のネットワーク専門家にお問い合わせください。



CPU が以前に設定済みでない場合は、「オンラインアクセス」を使用して、IP アドレスを設定することもできます。

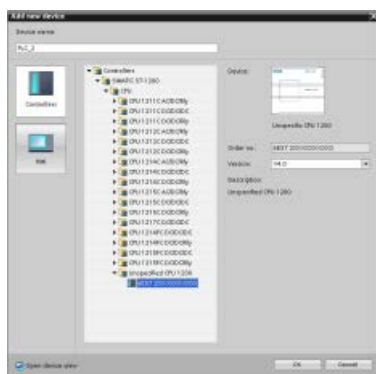
デバイス構成と一緒にダウンロードした IP アドレスは、PLC の電源オフ/オンでは失われません。

IP アドレス付きのデバイス構成をダウンロードした後、その IP アドレスは[オンラインアクセス]フォルダに表示されます。

11.15 「未指定 CPU」を使用したハードウェアコンフィグレーションのアップロード

プログラミングデバイスに接続できる物理的 CPU がある場合、ハードウェアコンフィグレーションをアップロードすることは簡単です。

最初にその CPU をプログラミングデバイスに接続し、新しいプロジェクトの作成の必要があります。

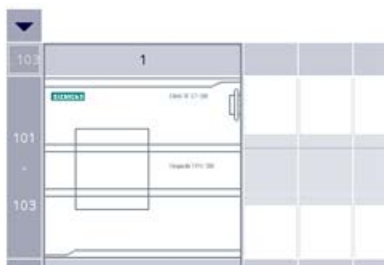


デバイス構成(プロジェクトビューまたはポータルビュー)で、新しいデバイスを追加しますが、特定の CPU を選択する代わりに、「未指定の CPU」を選択します。STEP 7 が未指定の CPU を作成します。



未指定の CPU を作成した後、オンライン CPU のハードウェアコンフィグレーションをアップロードできます。

- プログラミングエディタで、[オンライン]メニューから[ハードウェア検出]を選択します。
- デバイスコンフィグレーションエディタで、接続デバイスの構成検出用のオプションを選択します。



The device is not specified.
→ Please use the [hardware catalog](#) to specify the CPU.
→ or [detect](#) the configuration of the connected device.

オンラインダイアログからその CPU を選択した後、STEP 7 がその CPU からハードウェアコンフィグレーション(すべてのモジュール(SM、SB、または CM)を含めて)をアップロードします。IP アドレスはアップロードされません。「デバイス構成」に移行して、手動で IP アドレスを設定する必要があります。

11.16 RUN モードでダウンロード

CPU は、「RUN モードでのダウンロード」をサポートしています。この機能は、プログラムによって制御されているプロセスに対する影響を最小にして、ユーザープログラムの少量の変更を行うことを可能にするための機能です。ただし、この機能を実行すると、大量のプログラム変更も可能になり、その場合は破壊的影響または危険な影響さえ生じる場合があります。



警告

RUN モードでのダウンロードによるリスク

RUN モードの CPU へ変更をダウンロードすると、この変更は直ちにプロセス動作に影響します。RUN モードでプログラムを変更すると、予期しないシステム動作を生じ、その結果、要員の死または深刻な傷害や設備への破壊的影響を招く場合があります。

システム動作に対する RUN モードでの変更の影響を理解している資格のある要員だけが、RUN モードでのダウンロードを実行する必要があります。

「RUN モードでのダウンロード」機能を使用して、プログラムの変更を行い、ユーザーの CPU に、その CPU を STOP モードに切り替えないで、その変更をダウンロードすることができます。

- シャットダウンすることなく、現在のプロセスに対してそれほど重要でない変更(たとえば、パラメータ値の変更など)を行うことができます。
- この機能(たとえば、ノーマルオープン型スイッチまたはノーマルクローズ型スイッチ用論理の反転など)を使用して、より迅速にプログラムをデバッグすることができます。

以下のプログラムブロックおよびタグの変更を行い、それらを RUN モードでダウンロードできます。

- ファンクション(FC)、ファンクションブロック(FB)、およびタグテーブルの作成、上書き、および削除。
- データブロック(DB)、およびファンクションブロック(FB)用インスタンスデータブロックの作成、削除、および上書き。DB 構造体を追加し、それらを RUN モードでダウンロードできます。CPU は、ユーザーの設定に基づいて、既存のブロックタグの値は保持し、新しいデータブロックタグを初期値で初期化するか、またはすべてのデータブロックタグを初期値で初期化することができます。Web サーバー DB (コントロールまたはフラグメント)は RUN モードでダウンロードできません。
- オーガニゼーションブロック(OB)の上書き; ただし、OB の作成または削除はできません。

RUN モードで、最大 20 のブロックを 1 度にダウンロードできます。20 を超えるブロックをダウンロードする場合は、CPU を STOP モードに切り替える必要があります。

実際のプロセス(プログラムのデバッグ中に使用する可能性があるシミュレーションプロセスではなく)の変更をダウンロードする場合は、ダウンロードする前に、マシンおよびマシンオペレータに対する安全面での可能な影響を徹底的に検討しておくことが必須です。

注記

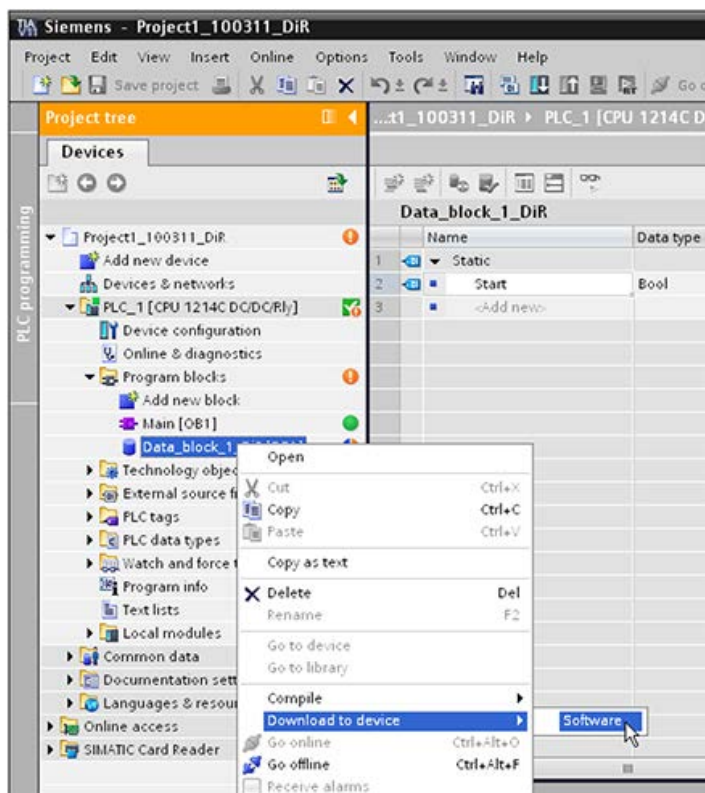
CPU が RUN モードで、プログラム変更が行われると、STEP 7 は、常に、最初に RUN でのダウンロードを試みます。ユーザーがこれを望まない場合は、ユーザーが CPU を STOP に切り替える必要があります。

実行された変更が「RUN でのダウンロード」でサポートされていない場合は、STEP 7 が、ユーザーに対して、CPU を STOP に移行する必要がある旨のプロンプトを表示します。

11.16.1 RUNモードでプログラムの変更

RUN モードでプログラムを変更するには、最初に、CPU およびプログラムが前提条件を満たすことを確認した後、以下の手順に従う必要があります。

1. RUN モードでユーザープログラムをダウンロードするには、以下の方法の 1 つを選択します。
 - [オンライン]メニューから、[デバイスへのダウンロード]コマンドを選択します。
 - ツールバーの[デバイスへのダウンロード]ボタンをクリックします。
 - 「プロジェクトツリー」で、[プログラムブロック]を右クリックし、[デバイスへのダウンロード|ソフトウェア]コマンドを選択します。



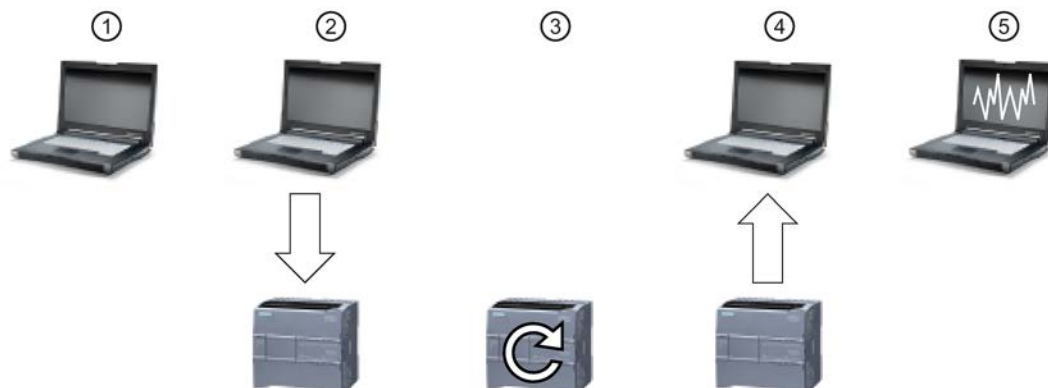
プログラムが正常にコンパイルされると、STEP 7 が、CPU へのプログラムのダウンロードを開始します。

2. STEP 7 が、ユーザープログラムのロードか、操作のキャンセルの選択を求めるプロンプトを表示するときは、[ロード]をクリックして、プログラムを CPU へダウンロードします。

11.17 トリガ条件を使用した CPU データのトレースとレコーディング

STEP 7 は、トレースおよびロジックアナライザ機能を提供します。この場合、レコーディングされたトレースデータをユーザーのプログラミングデバイスにアップロードし、STEP 7 ツールを使用して、ユーザーデータの分析、管理、およびグラフィック表示を行うことができます。トレースの作成および管理を行うには、STEP 7 プロジェクトツリーの[トレース]フォルダを使用します。

次の図に、トレース機能の各種のステップを示します。



- ① STEP 7 のトレースエディタで、トレースを構成します。記録するデータ値、記録持続時間、記録頻度、およびトリガ条件を設定できます。
- ② トレース構成を STEP 7 から PLC へ転送します。
- ③ PLC がプログラムを実行し、トリガ条件が発生したとき、トレースデータの記録を開始します。
- ④ 記録された値を PLC から STEP 7 へ転送します。
- ⑤ STEP 7 のツールを使用して、データの分析、グラフィック表示、および保存を行います。

トレースの最大サイズは、1 トレースあたり 512 Kbyte です。

アクセス例

トレースのプログラミング方法、構成のダウンロード方法、トレースデータのアップロード、およびロジックアナライザでのデータの表示に関する詳細は、STEP 7 情報システムを参照してください。詳細な例は、情報システムの「オンライン診断ファンクションの使用|トレースおよびロジックアナライザファンクションの使用」章に記載されています。

さらに、オンラインマニュアル『Industry Automation SINAMICS/SIMATIC トレースおよびロジックアナライザファンクションの使用』

(<http://support.automation.siemens.com/WW/view/en/64897128>)を参照ください。

カンタン IO-Link

12.1 IO-Link テクノロジーの概要

IO-Link は、PROFIBUS ユーザーオーガニゼーション(PNO)によって定義されたセンサおよびアクチュエータ用の革新的な通信テクノロジーです。IO-Link は、IEC 61131-9 に準拠した国際標準です。それは、センサおよびアクチュエータ(スレーブ)とコントローラ(マスタ)間のポイントツーポイント接続に基づいています。このため、それはバスシステムではなく、従来のポイントツーポイント接続のアップグレードです。

接続されたセンサ/アクチュエータによって、サイクリック動作データに加えて、拡張パラメータおよび診断データが伝送されます。データ伝送では、標準のセンサテクノロジーで使用するものと同じ 3 線式の接続ケーブルが使用されます。

12.2 IO-Link システムのコンポーネント

1 つの IO-Link システムは、IO-Link デバイス(通常、センサ、アクチュエータ、またはそれらの組み合わせ)、1 つの標準 3 線式センサ/アクチュエータケーブル、および 1 つの IO-Link マスタから構成されます。マスタは、任意の設計および保護等級のデバイスであって構いません。

1 つの IO-Link マスタは、1 つまたは複数のポートを持つことができます。SM 1278 4xIO リンクマスタには 4 つのポートがあります。それぞれのポートには、1 つの IO-Link デバイスまたは 1 つの標準センサ/アクチュエータを接続できます。IO-Link はポイントツーポイント通信システムです。

12.3 電源投入後

電源投入時、IO-Link デバイスは常に SIO モード(標準 I/O モード)です。マスタのポートは、さまざまな構成に設定できます。詳細は、『S7-1200 プログラマブルコントローラシステム マニュアル』の IO-Link の章を参照してください。

ポートを SIO モードに設定すると、マスタのこのポートは通常のデジタル入力と同様の動作を行います。このポートを IO-Link モード(通信モード)に設定すると、マスタが、接続された IO-Link デバイスの検出を試みます。このプロセスは、wake up と呼ばれます。

wake up 中、マスタは 1 つの定義された信号を送信し、スレーブデバイスの応答を待ちます。最初、マスタは、可能な限り最も高いボーレートでこれを試みます。これが成功しない場合、マスタは次の低いボーレートで試みます。マスタは、それぞれのボーレートで 3 回ずつデバイスアドレッシングを試みます。スレーブデバイスは、常に、定義された 1 つのボーレートだけをサポートしています。マスタが応答を受信すると(すなわち、スレーブデバイスが wake up された場合)、両方が通信を開始します。両方は、最初に、通信パラメータを交換した後、プロセスデータの周期的な交換を開始します。

動作中にスレーブデバイスが取り外されると、マスタは通信中止を検出し、それをフィールドバス特性でコントローラに報告した後、再度周期的にデバイスの wake up を試みます。別の wake up が成功した後、通信パラメータが再び読み出され、(該当する場合)検証された後、サイクリック通信チャンネルが再び開始されます。

12.4 IO-Link プロトコル

IO-Link システムは、3 つのタイプのデータを交換できます。

- サイクリックプロセスデータ(プロセスデータ入力、出力) → サイクリックデータ
- デバイスパラメータ(要求データオブジェクトに関する) → 非サイクリックデータ
- イベント → 非サイクリックデータ

IO-Link デバイスは、IO-Link マスタによって送信が要求された後にのみ、データを送信します。マスタの IDLE フレームの後およびマスタが明示的にデバイスパラメータデータおよびイベントを要求した後に、プロセスデータは送信されます。

12.5 フィールドバスでの構成

IO-Link マスタはフィールドバス上に通常のフィールドバスノードとして表示され、適切なデバイス記述によって関連ネットワーク構成に統合されます。これらのファイルは、IO-Link マスタの通信プロパティおよび他のプロパティ(ポート数など)を記述します。これらのファイルは、接続されている IO-Link デバイスは示しません。

ただし、IO-Link デバイス記述が(IODD)が、IO-Link デバイスまでのシステムアーキテクチャの完全な描写のために定義されています。IODD と IO-Link 構成ツール S7 PCT を利用して、どの IO-Link デバイスが IO-Link マスタのどのポートに接続されるかを設定できます。

詳細な構成情報は、S7 PCT ヘルプシステムと『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。

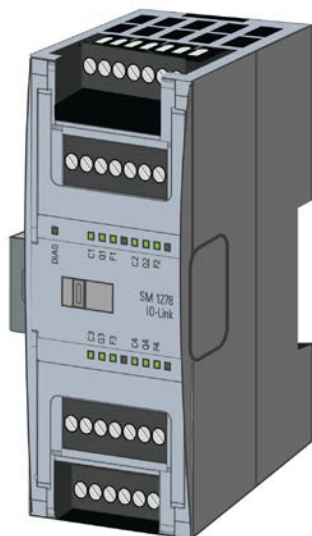
12.6 IO-Link と STEP 7 プログラム

IO-Link マスタは、ユーザーの STEP 7 S7-1200 コントローラプログラムで、IOL_CALL ファンクションブロック(FB)を使用して、IO-Link デバイスとの非サイクリック通信をプログラミングします。IOL_CALL FB は、ユーザープログラムが使用する IO-Link マスタと、マスタがデータ交換で使用するポートを指示します。

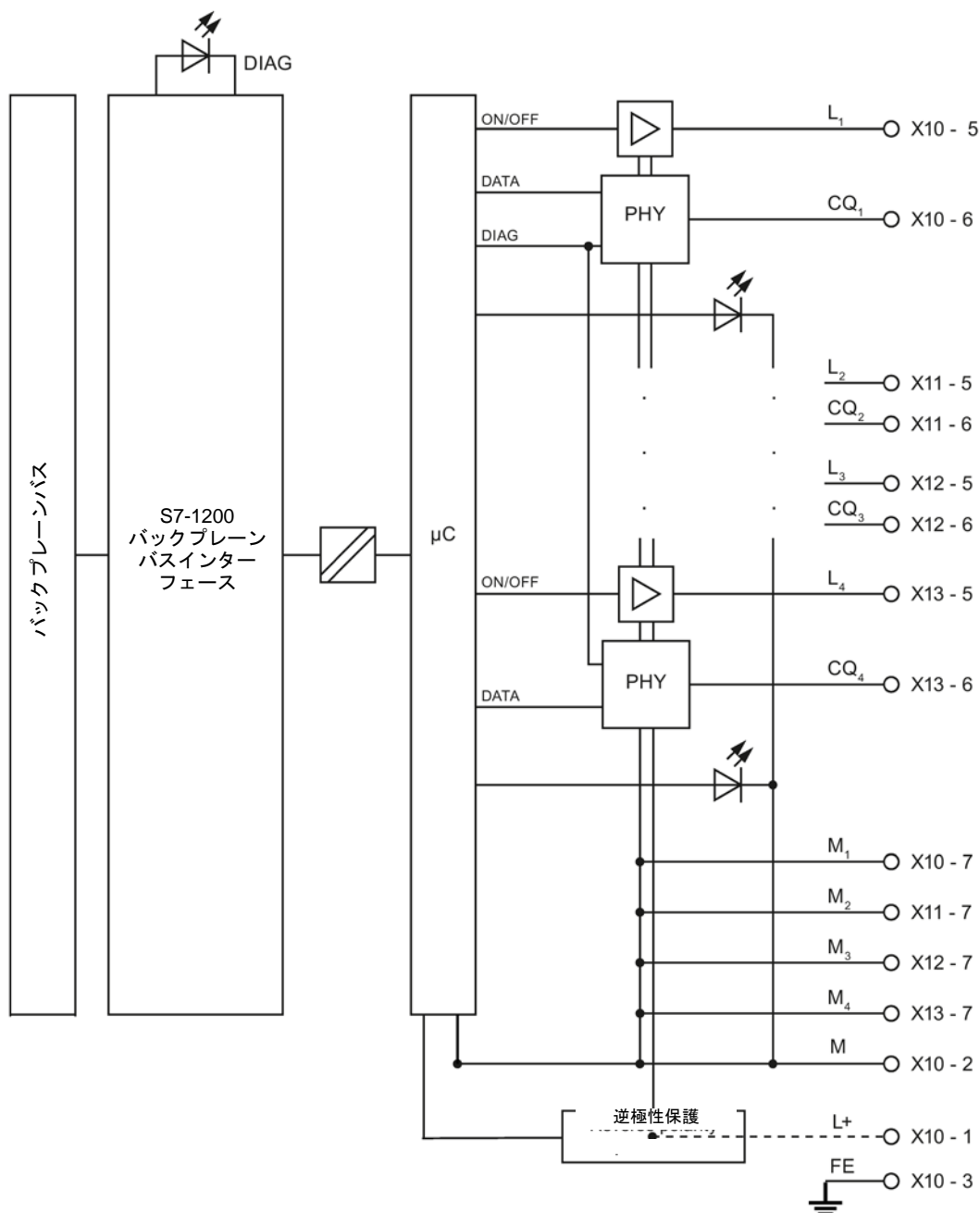
IOL_CALL FB の操作の詳細は、Siemens Industry オンラインサポート Web サイト (<http://support.automation.siemens.com>) にアクセスしてください。その Web サイトの検索ボックスに「IO-Link」を入力して、IO-Link 製品とそれらの使用に関する情報にアクセスします。

12.7 SM 1278 4xIO-Link マスタ

SM 1278 4xIO-Link マスタは、シグナルモジュールと通信モジュールの両方として動作する 4 つのポートを備えたモジュールです。それぞれのポートは、IO-Link モード、単一の 24 V DC デジタル入力、または単一の 24 V DC デジタル出力として動作できます。最大 4 つの IO-Link デバイス(3 線式接続)または最大 4 つの標準アクチュエータ/エンコーダに接続できます。

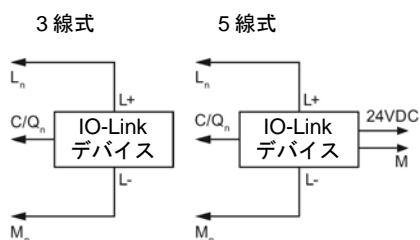


SM 1278 4xIO-Link マスタのブロックダイアグラム

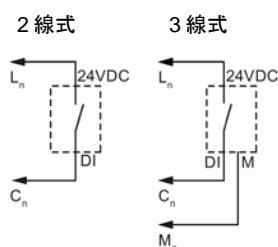


接続例

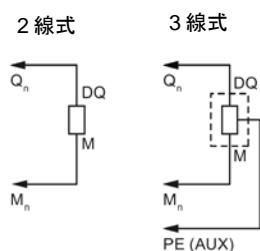
次の図は、IO-Link 動作モード用の構成を示します(3 線式および 5 線式)、ここで、 n = ポート番号です。



次の図は、DI 動作モード用の構成を示します(2 線式および 3 線式)、ここで、 n = ポート番号です。



次の図は、DQ 動作モード用の構成を示します(2 線式および 3 線式)、ここで、 n = ポート番号です。



SM 1278 4xIO-Link マスタの使用および構成に関する詳細情報

SM 1278 4xIO-Link マスタに関する詳細情報(ダイアグラム、接続、パラメータ割り当て、診断アラームなどを含めて)は、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。

技術仕様

A

A.1 一般技術仕様

規格への準拠

S7-1200 オートメーションシステムの設計は、以下の規格およびテスト仕様に準拠しています。S7-1200 オートメーションシステムのテスト基準は、これらの規格およびテスト仕様に基づいています。

すべての S7-1200 モデルがこれらの規格に従って認証できるわけではないこと、また、認証ステータスは通知なしで変更できることに注意してください。製品に表示されている定格を参照することによって該当する認証を識別するのは、ユーザーの責任です。部品番号による正確な承認の最新リストに関して詳細情報が必要な場合は、貴社のシーメンス担当者にご相談ください。

CE 承認



S7-1200 オートメーションシステムは、以下に記載する EC 指令に基づく要件および安全に関連する目的を満たしており、欧州共同体官報に記載されているプログラマブルコントローラの欧州整合規格(EN)に適合しています。

- EC 指令 2006/95/EC(低電圧指令)『特定の電圧限度内で使用するよう設計された電気機器』
 - EN 61131-2:2007 プログラマブルコントローラ - 機器要件およびテスト
- EC 指令 2004/108/EC (EMC 指令)『電磁両立性』
 - 排出基準
EN 61000-6-4:2007+A1:2011: 産業環境
 - 電磁波耐性基準
EN 61000-6-2:2005: 産業環境
- EC 指令 94/9/EC (ATEX)『爆発性雰囲気での使用を目的とした機器および保護システム』
 - EN 60079-15:2010: 保護のタイプ'n'

CE 適合宣言は、以下の住所でファイルに収録し所轄官庁に開示可能です。

Siemens AG
Sector Industry
IIA AS FA DH AMB
Postfach 1963
D-92209 Amberg
Germany

CULUS 認可



Underwriters Laboratories Inc. は以下に適合:

- Underwriters Laboratories Inc. UL 508 記載(工業用制御装置)
- Canadian Standards Association: CSA C22.2 Number 142 (プロセス制御機器)

注記

SIMATIC S7-1200 シリーズは CSA 規格に適合しています。

cULus ロゴは、S7-1200 が Underwriters Laboratories (UL)によって、UL 508 規格および CSA 22.2 No. 142 規格に適合していることが、試験および認証されたことを示しています。

FM 承認



Factory Mutual Research (FM)

承認規格クラス番号 3600 および 3611

以下での使用が承認済み:

クラス I、ディビジョン 2、ガスグループ A、B、C、D、温度クラス T3C Ta = 60 °C

クラス I、ゾーン 2、IIC、温度クラス T3 Ta = 60 °C

Canadian クラス I、ゾーン 2 での CEC 18-150 に従った設置

重要な例外: 同時に許可される入力または出力の数については、技術仕様を参照してください。Ta = 60 °C の場合、性能が下がるモデルがあります。



警告

コンポーネントを交換すると、クラス I、ディビジョン 2、およびゾーン 2 への適合性が損なわれる場合があります。

ユニットの修復は、資格のある Siemens Service Center だけが実行する必要があります。

IECEX 承認

EN 60079-0: 爆発性雰囲気 - 一般要件

EN60079-15: 爆発性雰囲気での使用を目的とした電気機器;

保護タイプ 'nA'

IECEX FMG14.0012X

Ex nA IIC Tx Gc

IECEX 定格情報は、FM 危険な場所情報を備えた製品上に表示される場合があります。

IECEX 定格が表示された製品だけが承認されます。部品番号による正確な承認の最新リストに関して詳細情報が必要な場合は、貴社のシーメンス担当者にご相談ください。

リレーモデルは、IECEX 承認には含まれません。

温度定格については、特定の製品表示を参照してください。

モジュールは、IEC 60079-15 の最小保護等級 IP54 を提供する適切な筐体の中に設置します。

ATEX 承認



ATEX 承認は、DC モデルのみに適用されます。ATEX 承認は、AC およびリレーモデルには適用されません。

EN 60079-0:2009: 爆発性雰囲気 - 一般要件

EN 60079-15:2010: 爆発性雰囲気での使用を目的とした電気機器;

保護タイプ'nA'

II 3 G Ex nA IIC T4 または T3 Gc

モジュールは、EN 60529 の最小保護等級 IP54 を提供する適切な筐体の中に設置するか、それと等価の保護等級を提供する場所に設置します。

接続されたケーブルおよび導線は、定格条件の下で測定された温度に合わせて、評価する必要があります。

設置では、過渡電圧が 119 V 未満に制限されていることを確認する必要があります。このセクションのサージ耐性を参照してください。

重要な例外: 同時に許可される入力または出力の数については、技術仕様を参照してください。Ta = 60 °C の場合、性能が下がるモデルがあります。

C-Tick 承認



S7-1200 オートメーションシステムは、AS/NZS CISPR16 (クラス A) の規格要件を満たしています。

Korea 認証



S7-1200 オートメーションシステムは、Korean 認証(KC マーク)の要件を満たしています。それはクラス A 機器として定義され、産業用用途のためのシステムで、住宅での使用は考慮されていません。

ユーラシア関税同盟の承認(ベルラーシ、カザフスタン、ロシア連邦)



EAC (ユーラシア適合性): 関税同盟の技術的規則(TR CU)に従った適合性の宣言

海事承認

S7-1200 製品は、特定の市場や用途に関する専門機関の承認を得るため定期的に提出されています。部品番号による正確な承認の最新リストに関して詳細情報が必要な場合は、貴社のシーメンス担当者にご相談ください。

船級協会:

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)
- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (日本海事協会)
- Korean Register of Shipping

産業環境

S7-1200 オートメーションシステムは、産業環境での使用向けに設計されています。

表 A-1 産業環境

用途	排出要件	電磁波耐性要件	耐ノイズ性要件
産業	EN 61000-6-4:2007+A1:2011	EN 61000-6-2:2005	EN 61000-6-2:2005

注記

S7-1200 オートメーションシステムは、産業領域での使用のためのシステムです。居住領域で使用すると、ラジオやTVの受信に影響を及ぼす場合があります。S7-1200 を居住領域で使用する場合は、その無線干渉放出が EN 55011 の限界値クラス B に適合することを確認する必要があります。

RF 干渉レベルクラス B を実現する適切な手段の例としては、以下があります。

- 接地された制御キャビネット内での S7-1200 の設置
- 給電線内でのノイズフィルタの使用

無線干渉放出が EN 55011 のクラス B に適合することを確認します。

個別の承認が必要です(最終的な設置では、居住領域での設置のためのすべての安全関連および EMC の要件を満たす必要があります)。

電磁環境両立性(EMC)

電磁環境両立性(EMC)は、電磁環境で意図したとおりに動作し、付近にある他の電気装置を妨害する可能性がある電磁妨害(EMI)の放射レベルなしで動作する、電気装置の能力です。

表 A-2 EN 61000-6-2 に従った電磁波耐性

電磁環境両立性 - EN 61000-6-2 による電磁波耐性	
EN 61000-4-2 静電気放電	すべての表面への 8 kV 空中放電 露出導電面への 6 kV 接触放電
EN 61000-4-3 放射無線周波数電磁界の耐性テスト	80~1000 MHz、10 V/m、1 kHz で 80% AM 1.4~2.0 GHz、3 V/m、1 kHz で 80% AM 2.0~2.7 GHz、1 V/m、1 kHz で 80% AM
EN 61000-4-4 ファストランジェントバースト	2 kV、5 kHz、AC および DC システム電源への結合ネット ワーク付き 2 kV、5 kHz、I/O への結合クランプ付き
EN 61000-4-5 サージ耐性	AC システム - 2 kV コモンモード、1 kV ディファレンシャル モード DC システム - 2 kV コモンモード、1 kV ディファレンシャル モード DC システムの場合、下のサージ耐性を参照
EN 61000-4-6 伝導妨害	150 kHz~80 MHz、10 V RMS、1kHz で 80% AM
EN 61000-4-11 電圧低下	AC システム 60 Hz の場合、1 サイクルで 0%、12 サイクルで 40%、お よび 30 サイクルで 70%

サージ耐性

落雷結合からサージを受ける配線システムは、外部保護を装備する必要があります。落雷タイプのサージからの保護を評価する 1 つの仕様が EN 61000-4-5 に存在し、この場合、動作限界値が EN 61000 6 2 によって確立されています。S7-1200 DC CPU およびシグナルモジュールは、この規格によって定義されたサージ電圧を受けたときに安全な動作を保持するために外部保護を必要とします。

必要なサージ耐性保護をサポートするいくつかのデバイスを下に記載します。これらのデバイスは、製造元の推奨事項に従って正しく設置された場合だけ、保護を提供します。他のベンダによって製造された同一または改善された仕様を備えたデバイスも使用できます。

表 A-3 サージ耐性保護をサポートするデバイス

サブシステム	保護デバイス
+24 V DC 電源	BLITZDUCTOR VT、BVT AVD 24、部品番号 918 422
産業用イーサネット	DEHNpatch DPA M CLE RJ45B 48、部品番号 929 121
RS-485	BLITZDUCTOR XT、Basic Unit BXT BAS、部品番号 920 300
	BLITZDUCTOR XT、Module BXT ML2 BD HFS 5、部品番号 920 271
RS-232	BLITZDUCTOR XT、Basic Unit BXT BAS、部品番号 920 300
	BLITZDUCTOR XT、Module BXT ML2 BE S 12、部品番号 920 222

A.1 一般技術仕様

サブシステム	保護デバイス
+24 V DC デジタル入力	DEHN, Inc., Type DCO SD2 E 24、部品番号 917 988
+24 V DC デジタル出力 およびセンサ電源	DEHN, Inc., Type DCO SD2 E 24、部品番号 917 988
アナログ I/O	DEHN, Inc., Type DCO SD2 E 12、部品番号 917 987
リレー出力	不要

表 A-4 EN 61000-6-4 による伝導性および放射妨害波

電磁環境両立性 - EN 61000-6-4 による伝導性放射および放射妨害波		
伝導性妨害波 EN 55011、クラス A、 グループ 1	0.15 MHz～0.5 MHz	<79dB (μV)準尖頭値; <66 dB (μV)平均
	0.5 MHz～5 MHz	<73dB (μV)準尖頭値; <60 dB (μV)平均
	5 MHz～30 MHz	<73dB (μV)準尖頭値; <60 dB (μV)平均
放射性妨害波 EN 55011、クラス A、 グループ 1	30 MHz～230 MHz	<40dB (μV/m)準尖頭値; 10m 離れて測定
	230 MHz～1 GHz	<47dB (μV/m)準尖頭値; 10m 離れて測定
	1 GHz～3 GHz	< 76dB (uV/m)準尖頭値、10m 離れて測定

環境条件

表 A-5 輸送および保管

環境条件 - 輸送および保管	
EN 60068-2-2、テスト Bb、乾燥熱および EN 60068-2-1、テスト Ab、低温	40 °C～+70 °C
EN 60068-2-30、テスト Db、湿気熱	25 °C～55 °C、95%湿度
EN 60068-2-14、テスト Na、温度衝撃	40 °C～+70 °C、滞留時間 3 時間、5 サイクル
EN 60068-2-32、自由落下	0.3 m、5 回、製品梱包
大気圧	1080～660h Pa (高度-1000～3500m に相当)

表 A-6 動作条件

環境条件 - 動作	
周囲温度範囲 (装置下 25 mm 流入空気)	20 °C～60 °C 横置き設置 20 °C～50 °C 縦置き設置 95%結露なし湿度 他の指定がないかぎり
大気圧	1080～795 hPa (高度-1000～2000m に相当)
不純物濃度	SO ₂ : < 0.5 ppm; H ₂ S: < 0.1 ppm; RH < 60%結露なし ISA-S71.04 深刻度レベル G1、G2、G3
EN 60068-2-14、テスト Nb、温度変化	5° C～55° C、3° C/分

環境条件 - 動作	
EN 60068-2-27 機械的衝撃	15 G、11 ms パルス、各 3 軸で 6 衝撃
EN 60068-2-6 正弦波振動	DIN レールマウント: 5~9 Hz から 3.5 mm、9~150 Hz から 1G パネルマウント: 5~9 Hz から 7.0 mm、9~150 Hz から 2G 各軸 10 スイープ、1 オクターブ/分

表 A-7 高電位絶縁テスト

高電位隔離テスト	
24 V DC / 5 V DC 定格回路	520 V DC (光絶縁境界のタイプテスト)
アースへの 115 V AC / 230 V AC 回路	1500 V AC
115 V AC / 230 V AC 回路~115 V AC / 230 V AC 回路	1500 V AC
115 V AC / 230 V AC 回路~24 V DC / 5 V DC 回路	1500 V AC (3000 V AC/4242 V DC タイプテスト)
イーサネットポート~24 V DC / 5 V DC 回路およびアース ¹	1500 V AC (タイプテストのみ)

¹ イーサネットポート絶縁は、危険な電圧に達する短い間のネットワーク障害中の危険を制限するように設計されています。それは、通常の AC 電源電圧絶縁の安全要件には適合しません。

保護クラス

- 保護クラス II、EN 61131-2 による(保護導体不要)

保護等級

- IP20 機械的保護、EN 60529
- 標準プローブでテストした、高電圧への指接触に対する保護。直径 12.5mm 未満のほこり、汚れ、水、異物に対する外部保護が必要です。

定格電圧

表 A-8 定格電圧

定格電圧	許容差
24 V DC	20.4 V DC~28.8 V DC
120/230 V AC	85 V AC~264 V AC、47~63 Hz

逆電圧の保護

逆電圧保護回路は、CPU、シグナルモジュール(SM)、およびシグナルボード(SB)用の+24 V DC 電源またはユーザー入力電源の端子ペアごとに提供されます。この回路を設置してもなお、反対の極性で異なる端子ペアを配線することによって、システムを損傷することが可能です。

S7-1200 システムの 24 V DC 電源入力ポートには、共通の論理回路によって複数の M 端子が接続され、相互接続されているものがあります。たとえば、データシートに「非絶縁」と記載されている場合に相互接続されている回路は次のとおりです。CPU の 24 V DC 電源、CPU のセンサ電源、SM のリレーコイルの電源入力、および非絶縁アナログ入力の電源。すべての非絶縁 M 端子は、同じ外部基準電圧に接続する必要があります。



警告

非絶縁 M 端子が複数の異なる基準電圧に接続されていると、予期しない電流が流れ、PLC および接続されている装置の損傷や予測できない動作が発生する原因になることがあります。

これらのガイドラインを遵守しなかった場合は、損傷や予測できない動作が発生する原因になり、死傷事故などの重大な人的傷害や物的損害が発生する可能性があります。

S7-1200 システムのすべての非絶縁 M 端子が、同一の電源の基準電圧に接続されていることを必ず確認してください。

DC 出力

短絡保護回路は、CPU、シグナルモジュール(SM)、およびシグナルボード(SB)の DC 出力では提供されません。

リレーの電気耐用年数

サンプロテストによって推定された通常の性能データを下に示します。実際の性能は、個々の用途に応じて異なる可能性があります。電気負荷に適合された外部保護回路は、接点の耐用年数を延ばします。N.C.接点は、通常、誘導およびランプ負荷条件で、N.O.接点の約 1/3 の耐用年数を持ちます。

外部保護回路は、接点の耐用年数を伸ばします。

表 A-9 通常の性能データ

アクチュエータの選択用データ				
持続的な熱電流		最大 2 A		
接点のキャパシティと耐用年数の切り替え				
	抵抗負荷の場合	電圧	Current	動作サイクルの数(通常)
		24 V DC	2.0 A	0.1 (100 万単位)
		24 V DC	1.0 A	0.2 (100 万単位)
		24 V DC	0.5 A	1.0 (100 万単位)
		48 V AC	1.5 A	1.5 (100 万単位)
		60 V AC	1.5 A	1.5 (100 万単位)
		120 V AC	2.0 A	1.0 (100 万単位)
		120 V AC	1.0 A	1.5 (100 万単位)
		120 V AC	0.5 A	2.0 (100 万単位)
		230 V AC	2.0 A	1.0 (100 万単位)
		230 V AC	1.0 A	1.5 (100 万単位)
		230 V AC	0.5 A	2.0 (100 万単位)
			誘導負荷の場合 (IEC 947-5-1 DC13/AC15 に準拠)	電圧
24 V DC	2.0 A			0.05 (100 万単位)
24 V DC	1.0 A			0.1 (100 万単位)
24 V DC	0.5 A			0.5 (100 万単位)
24 V AC	1.5 A			1.0 (100 万単位)
48 V AC	1.5 A			1.0 (100 万単位)
60 V AC	1.5 A			1.0 (100 万単位)
120 V AC	2.0 A			0.7 (100 万単位)
120 V AC	1.0 A			1.0 (100 万単位)
120 V AC	0.5 A			1.5 (100 万単位)
230 V AC	2.0 A			0.7 (100 万単位)
230 V AC	1.0 A			1.0 (100 万単位)
230 V AC	0.5 A			1.5 (100 万単位)
デジタル入力の有効化		可能		
周波数の切り替え				
	機械的	最大 10 Hz		
	抵抗負荷時	最大 1 Hz		

アクチュエータの選択用データ		
	誘導負荷時(IEC 947-5-1 DC13/AC15 に準拠)	最大 0.5 Hz
	ランプ負荷時	最大 1 Hz

内部 CPU メモリの保持

- 保持データおよびデータログデータの寿命: 10 年
- パワーダウン保持データ、書き込みサイクルに対する耐久性: 2 (100 万単位)サイクル
- データログデータ、1 エントリあたり最大 2 KB、書き込みサイクルに対する耐久性: 500 (百万単位)データログエントリ

注記

内部 CPU メモリに対するデータログの影響

それぞれのデータログ書き込みは、最小 2 KB のメモリを消費します。ユーザープログラムが少量のデータを頻繁に書き込む場合、それは、書き込みごとに、少なくとも 2 KB のメモリを消費しています。より良い改善方法は、小さいデータ項目を 1 つのデータブロック(DB)に蓄積し、そのデータブロックをより少ない頻度でデータログに書き込むことです。

ユーザープログラムが多くのデータログエントリを高い頻度で書き込む場合は、交換可能な SD メモリカードを使用することを検討してください。

A.2 CPU モジュール

S7-1200 で使用できるモジュールのより完全なリストは、『S7-1200 プログラマブルコントローラシステムマニュアル』またはカスタマサポート Web サイト (<http://www.siemens.com/tiaportal>) を参照してください。

表 A- 10 一般仕様

一般仕様		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
製品 番号	AC/DC/リレー	6ES7 211-1BE40-0XB0	6ES7 212-1BE40-0XB0	6ES7 214-1BG40-0XB0	6ES7 215-1BG40-0XB0	--
	DC/DC/リレー	6ES7 211-1HE40-0XB0	6ES7 212-1HE40-0XB0	6ES7 214-1HG40-0XB0	6ES7 215-1HG40-0XB0	--
	DC/DC/DC	6ES7 211-1AE40-0XB0	6ES7 212-1AE40-0XB0	6ES7 214-1AG40-0XB0	6ES7 215-1AG40-0XB0	6ES7 217-1AG40-0XB0
外形寸法 W x H x D (mm)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
重量	<ul style="list-style-type: none"> AC/DC/リレー DC/DC/リレー DC/DC/DC 	<ul style="list-style-type: none"> 420 グラム 380 グラム 370 グラム 	<ul style="list-style-type: none"> 425 グラム 385 グラム 370 グラム 	<ul style="list-style-type: none"> 475 グラム 435 グラム 415 グラム 	<ul style="list-style-type: none"> 585 グラム 550 グラム 520 グラム 	<ul style="list-style-type: none"> - - 530 グラム
電力 損失	<ul style="list-style-type: none"> AC/DC/リレー DC/DC/リレー DC/DC/DC 	<ul style="list-style-type: none"> 10 W 8 W 8 W 	<ul style="list-style-type: none"> 11 W 9 W 9 W 	<ul style="list-style-type: none"> 14 W 12 W 12 W 	<ul style="list-style-type: none"> 14 W 12 W 12 W 	<ul style="list-style-type: none"> - - 12 W
SM および CM バスで 使用可能な電流(5 V DC)		最大 750 mA	最大 1000 mA	最大 1600 mA	最大 1600 mA	最大 1600 mA
センサ電源で使用可能な 電流(24 V DC)		最大 300 mA	最大 300 mA	最大 400 mA	最大 400 mA	最大 400 mA
デジタル入力消費電流 (24 V DC)		4 mA/入力	4 mA/入力	4 mA/入力	4 mA/入力	4 mA/入力

表 A- 11 CPU の機能

CPU の機能		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
ユーザーメモリ						
• ワークメモリ		• 50 KB	• 75 KB	• 100 KB	• 125 KB	• 150 KB
• ロードメモリ		• 1MB	• 1MB	• 4 MB	• 4 MB	• 4 MB
• 保持型メモリ		• 10 KB	• 10 KB	• 10 KB	• 10 KB	• 10 KB
オンボードデジタル I/O 仕様 (Page 386)を参照。		6 入力 4 出力	8 入力 6 出力	14 入力 10 出力	14 入力 10 出力	14 入力 10 出力
オンボードアナログ I/O 仕様 (Page 397)を参照。		2 入力	2 入力	2 入力	2 入力 2 出力	2 入力 2 出力
プロセスイメージサイズ						
• 入力		• 1024 バイト	• 1024 バイト	• 1024 バイト	• 1024 バイト	• 1024 バイト
• 出力		• 1024 バイト	• 1024 バイト	• 1024 バイト	• 1024 バイト	• 1024 バイト
ビットメモリ(M)		4096 バイト	4096 バイト	8192 バイト	8192 バイト	8192 バイト
一時的な(ローカル)メモリ		• スタートアップおよびプログラムサイクル(関連 FB および FC を含めて)用の 16 KB • 他のすべての優先度レベルそれぞれ(FB および FC を含めて)のための 6 KB				
SM 拡張		なし	最大 2 SM	最大 8 SM	最大 8 SM	最大 8 SM
SB、CB、または BB 拡張		最大 1	最大 1	最大 1	最大 1	最大 1
CM 拡張		最大 3	最大 3	最大 3	最大 3	最大 3
高速 カウンタ	合計	任意の内蔵または SB 入力を使用するために、最大で 6 を構成				
	1 MHz	--	--	--	--	lb.2~lb.5 (差動入力)
	100/ ¹ 80 kHz	la.0~la.5	la.0~la.5	la.0~la.5	la.0~la.5	la.0~la.5
	30/ ¹ 20 kHz	--	la.6~la.7	la.6~la.5	la.6~la.5	la.6~la.1
パルス 出力 ²	合計	任意の内蔵または SB 入力を使用するために、最大で 4 を構成				
	1 MHz	--	--	--	--	Qa.0~Qa.3 (作動出力)
	100 kHz	Qa.0~Qa.3	Qa.0~Qa.3	Qa.0~Qa.3	Qa.0~Qa.3	Qa.4~Qb.1
	30 kHz	--	Qa.4~Qa.5	Qa.4~Qb.1	Qa.4~Qb.1	--
パルスキャッチ入力		6	8	14	14	14
遅延割り込み		1 ms 解像度で 合計 4	1 ms 解像度で 合計 4	1 ms 解像度で 合計 4	1 ms 解像度で 合計 4	1 ms 解像度で 合計 4
周期割り込み		1 ms 解像度で 合計 4	1 ms 解像度で 合計 4	1 ms 解像度で 合計 4	1 ms 解像度で 合計 4	1 ms 解像度で 合計 4
エッジ割り込み		6 の立ち上がり と 6 の立ち下がり	8 の立ち上がり と 8 の立ち下がり	12 の立ち上がり と 12 の立ち下がり	12 の立ち上がり と 12 の立ち下がり	12 の立ち上がり と 12 の立ち下がり
オプションの SB による		10 の立ち上がり と 10 の立ち下がり	12 の立ち上がり と 12 の立ち下がり	16 の立ち上がり と 16 の立ち下がり	16 の立ち上がり と 16 の立ち下がり	16 の立ち上がり と 16 の立ち下がり

CPU の機能	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
リアルタイムクロック <ul style="list-style-type: none"> 精度 保持時間(保守不要な超コンデンサ) 	<ul style="list-style-type: none"> +/- 60 秒/月 40 °C で通常 20 日間/最短で 12 日間 	<ul style="list-style-type: none"> +/- 60 秒/月 40 °C で通常 20 日間/最短で 12 日間 	<ul style="list-style-type: none"> +/- 60 秒/月 40 °C で通常 20 日間/最短で 12 日間 	<ul style="list-style-type: none"> +/- 60 秒/月 40 °C で通常 20 日間/最短で 12 日間 	<ul style="list-style-type: none"> +/- 60 秒/月 40 °C で通常 20 日間/最短で 12 日間
実行速度 <ul style="list-style-type: none"> ブール演算 ワードの移動 実数演算 	<ul style="list-style-type: none"> 0.08 μs / 命令 1.7 μs / 命令 2.3 μs / 命令 	<ul style="list-style-type: none"> 0.08 μs / 命令 1.7 μs / 命令 2.3 μs / 命令 	<ul style="list-style-type: none"> 0.08 μs / 命令 1.7 μs / 命令 2.3 μs / 命令 	<ul style="list-style-type: none"> 0.08 μs / 命令 1.7 μs / 命令 2.3 μs / 命令 	<ul style="list-style-type: none"> 0.08 μs / 命令 1.7 μs / 命令 2.3 μs / 命令

¹ HSC が直交位相動作モード用に構成されている場合、より低い速度が適用されます。

² リレー出力付きの CPU モデルの場合、デジタルシグナルボード(SB)を設置して、パルス出力を使用する必要があります。

表 A- 12 通信

技術データ	CPU 1211C、CPU 1212C、CPU 1214C	CPU 1215C、CPU 1217C
通信 <ul style="list-style-type: none"> データ速度 絶縁(PLC ロジックへの外部信号) ケーブルタイプ 	1 イーサネットポート <ul style="list-style-type: none"> 10/100 Mb/秒 絶縁済み変圧器、1500 V DC CAT5e シールドあり 	2 イーサネットポート <ul style="list-style-type: none"> 10/100 Mb/秒 絶縁済み変圧器、1500 V DC CAT5e シールドあり
デバイス	<ul style="list-style-type: none"> 4 HMI 1 PG 	<ul style="list-style-type: none"> 4 HMI 1 PG
イーサネット接続 ¹	8 (アクティブまたはパッシブ)	8 (アクティブまたはパッシブ)
CPU-to-CPU S7 接続 (GET/PUT)	<ul style="list-style-type: none"> 8 (クライアント) 3 (サーバー) 	<ul style="list-style-type: none"> 8 (クライアント) 3 (サーバー)

¹ オープンユーザーコミュニケーション接続(アクティブまたはパッシブ): TSEND_C、TRCV_C、TCON、TDISCON、TSEND、および TRCV。

表 A-13 CPU 1214C AC/DC/リレーの配線図

CPU 1214C AC/DC/リレー	
	<p>① 24 V DC センサ電源出力。ノイズ耐性を高めるために、センサ電源を使用しない場合でも、「M」をシャーシアースに接続します。</p>
	<p>② シンク入力の場合、「-」を「M」に接続します(図を参照)。ソース入力の場合、「+」を「M」に接続します。</p>
	<p>注記 1: X11 コネクタは金でなければなりません。『S7-1200 プログラマブルコントローラシステムマニュアル』の「付録 C」の「製品番号用予備部品」を参照してください。</p>
	<p>注記 2: L1 端子か、N (L2)端子のどちらかを、最大 240 V AC 電源に接続することができます。N 端子は L2 と見なすことができ、接地する必要はありません。L1 および N (L2)端子の場合、極性形成は不要です。</p>
	<p>注記 3: CPU のイーサネットポートについては、『S7-1200 プログラマブルコントローラシステムマニュアル』の「デバイス構成」を参照してください。</p>

表 A-14 CPU 1214C DC/DC/DC の配線図

CPU 1214C DC/DC/DC	
	<p>① 24 V DC センサ電源出力。ノイズ耐性を高めるために、センサ電源を使用しない場合でも、「M」をシャーシアースに接続します。</p>
	<p>② シンク入力の場合、「-」を「M」に接続します(図を参照)。ソース入力の場合、「+」を「M」に接続します。</p>
	<p>注記 1: X11 コネクタは金でなければなりません。『S7-1200 プログラマブルコントローラシステムマニュアル』の「付録 C」の「製品番号用予備部品」を参照してください。</p>
	<p>注記 2: CPU のイーサネットポートについては、『S7-1200 プログラマブルコントローラシステムマニュアル』の「デバイス構成」を参照してください。</p>

A.3 デジタル I/O モジュール

S7-1200 で使用できるモジュールのより完全なリストは、『S7-1200 プログラマブルコントローラシステムマニュアル』またはカスタマサポート Web サイト (<http://www.siemens.com/tiaportal>) を参照してください。

A.3.1 SB 1221、SB 1222、SB 1223 デジタル入出力(DI、DQ、DI/DQ)

表 A- 15 SB 1221 デジタル入力(DI)および SB 1222 デジタル出力(DQ)モジュール

全般		SB 1221 4 DI (200 kHz)	SB 1222 4 DQ (200 kHz)
製品番号		<ul style="list-style-type: none"> 24 V DC: 6ES7 221-3BD30-0XB0 5 V DC: 6ES7 221-3AD30-0XB0 	<ul style="list-style-type: none"> 24 V DC: 6ES7 222-1BD30-0XB0 5 V DC: 6ES7 222-1AD30-0XB0
外形寸法 W x H x D (mm)		38 x 62 x 21	38 x 62 x 21
重量		35 グラム	35 グラム
電力損失		<ul style="list-style-type: none"> 24 V DC: 1.5 W 5 V DC: 1.0 W 	0.5 W
消費電流	SM バス	40 mA	35 mA
	24 V DC	<ul style="list-style-type: none"> 24 V DC: 7 mA / 入力 + 20 mA 5 V DC: 15 mA / 入力 + 15 mA 	15 mA
入力/出力		4 入力(ソース)	4 出力(ソリッドステート - MOSFET)

Table A- 16 SB 1223 デジタル入出力(DI / DQ)組み合わせモジュール

全般		SB 1223 DI / DQ (200 kHz)	SB 1223 2 DI / 2 DQ
製品番号		<ul style="list-style-type: none"> 24 V DC: 6ES7 223-3BD30-0XB0 5 V DC: 6ES7 223-3AD30-0XB0 	24 V DC: 6ES7 223-0BD30-0XB0
外形寸法 W x H x D (mm)		38 x 62 x 21	38 x 62 x 21
重量		35 グラム	40 グラム
電力損失		<ul style="list-style-type: none"> 24 V DC: 1.0 W 5 V DC: 0.5 W 	24 V DC: 1.0 W
消費電流	SM バス	35 mA	50 mA
	24 V DC	<ul style="list-style-type: none"> 24 V DC: 7 mA / 入力 + 20 mA 5 V DC: 15 mA / 入力 + 15 mA 	4 mA/入力
入力/出力		2 入力(ソース) 2 出力(ソリッドステート - MOSFET)	2 入力(IEC タイプ 1 シンク) 2 出力(ソリッドステート - MOSFET)

注記

高速(200 kHz) SB は、「ソース」入力を利用します。標準 SB (20 kHz)は、「シンク」入力を利用します。デジタル入力および出力の仕様(Page 386)を参照してください。

高速(200 kHz)出力(SB 1222 および SB 1223)は、ソースか、シンクのどちらかになることができます。ソース出力の場合、「Load」を「-」に接続します(図を参照)。シンク出力の場合、「Load」を「+」に接続します。シンク構成とソース構成は両方とも同一の回路によってサポートされているため、ソース負荷の有効な状態はシンク負荷の有効な状態と逆です。ソース出力が正のロジックを示す(負荷に電流が生じると、Q ビットおよび LED がオンになります)のに対して、シンク出力は負のロジックを示します(負荷に電流が生じると、Q ビットおよび LED はオフになります)。ユーザープログラムなしのモジュールを差し込み接続すると、このモジュールの既定値は 0V で、シンク負荷がオンになります。

表 A- 17 デジタル SB の配線図

SB 1221 入力モジュール	SB 1222 出力モジュール	SB 1223 入出力モジュール
<p>SB 1221 DI 4 (200 kHz)</p>	<p>SB 1222 DQ 4 (200 kHz)</p>	<p>SB 1223 DI 2 / DQ 2 (200 kHz)</p>
<p>① ソース入力のみをサポートしています。</p>	<p>① ソース出力の場合、「Load」を「-」に接続します(図を参照)。シンク出力の場合、「Load」を「+」に接続します。シンク構成とソース構成は両方とも同一の回路によってサポートされているため、ソース負荷の有効な状態はシンク負荷の有効な状態と逆です。ソース出力が正のロジックを示す(負荷に電流が生じると、Q ビットおよび LED がオンになります)のに対して、シンク出力は負のロジックを示します(負荷に電流が生じると、Q ビットおよび LED はオフになります)。ユーザープログラムなしのモジュールを差し込み接続すると、このモジュールの既定値は 0 V で、シンク負荷がオンになります。</p>	<p>① ソース入力のみをサポートしています。</p> <p>② ソース出力の場合、「Load」を「-」に接続します(図を参照)。シンク出力の場合、「Load」を「+」に接続します。シンク構成とソース構成は両方とも同一の回路によってサポートされているため、ソース負荷の有効な状態はシンク負荷の有効な状態と逆です。ソース出力が正のロジックを示す(負荷に電流が生じると、Q ビットおよび LED がオンになります)のに対して、シンク出力は負のロジックを示します(負荷に電流が生じると、Q ビットおよび LED はオフになります)。ユーザープログラムなしのモジュールを差し込み接続すると、このモジュールの既定値は 0 V で、シンク負荷がオンになります。</p>

注記

高速(200 kHz) SB (SB 1221 および SB 1223)は、シンク入力だけをサポートしています。標準 SB 1223 は、ソース入力だけをサポートしています。

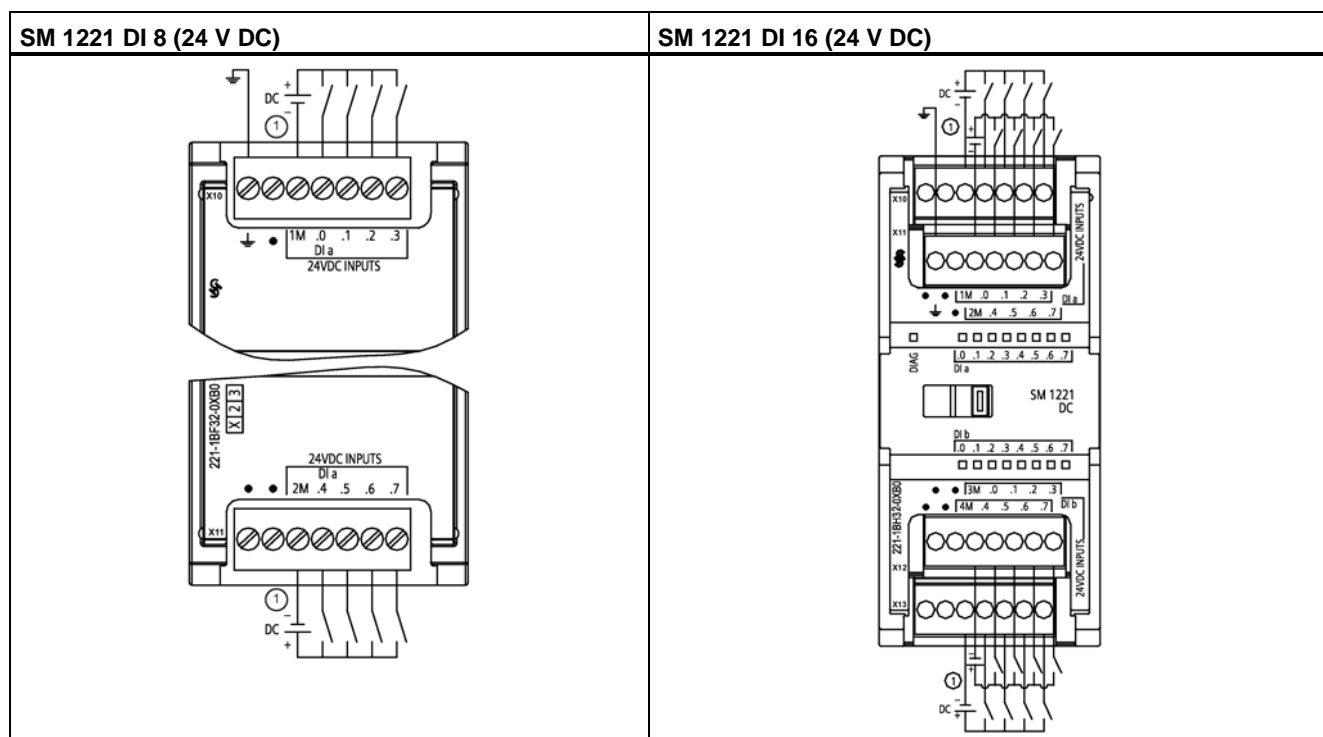
高速(200 kHz)出力(SB 1222 および SB 1223)は、ソースか、シンクのどちらかになることができます。ソース出力の場合、「Load」を「-」に接続します(図を参照)。シンク出力の場合、「Load」を「+」に接続します。シンク構成とソース構成は両方とも同一の回路によってサポートされているため、ソース負荷の有効な状態はシンク負荷の有効な状態と逆です。ソース出力が正のロジックを示す(負荷に電流が生じると、Q ビットおよび LED がオンになります)のに対して、シンク出力は負のロジックを示します(負荷に電流が生じると、Q ビットおよび LED はオフになります)。ユーザープログラムなしのモジュールを差し込み接続すると、このモジュールの既定値は 0 V で、シンク負荷がオンになります。

A.3.2 SM 1221 デジタル入力(DI)

表 A- 18 SM 1221 デジタル入力(DI)

技術データ		SM 1221 DI 8 24 V DC	SM 1221 DI 16 24 V DC
製品番号		6ES7 221-1BF32-0XB0	6ES7 221-1BH32-0XB0
入力数(DI) 仕様 (Page 386)を参照。		8	16
外形寸法 W x H x D (mm)		45 x 100 x 75	45 x 100 x 75
重量		170 グラム	210 グラム
電力損失		1.5 W	2.5 W
消費電流	SM バス	105 mA	130 mA
	24 V DC	4 mA/入力	4 mA/入力

表 A- 19 SM 1221 デジタル入力(DI)モジュールの配線図



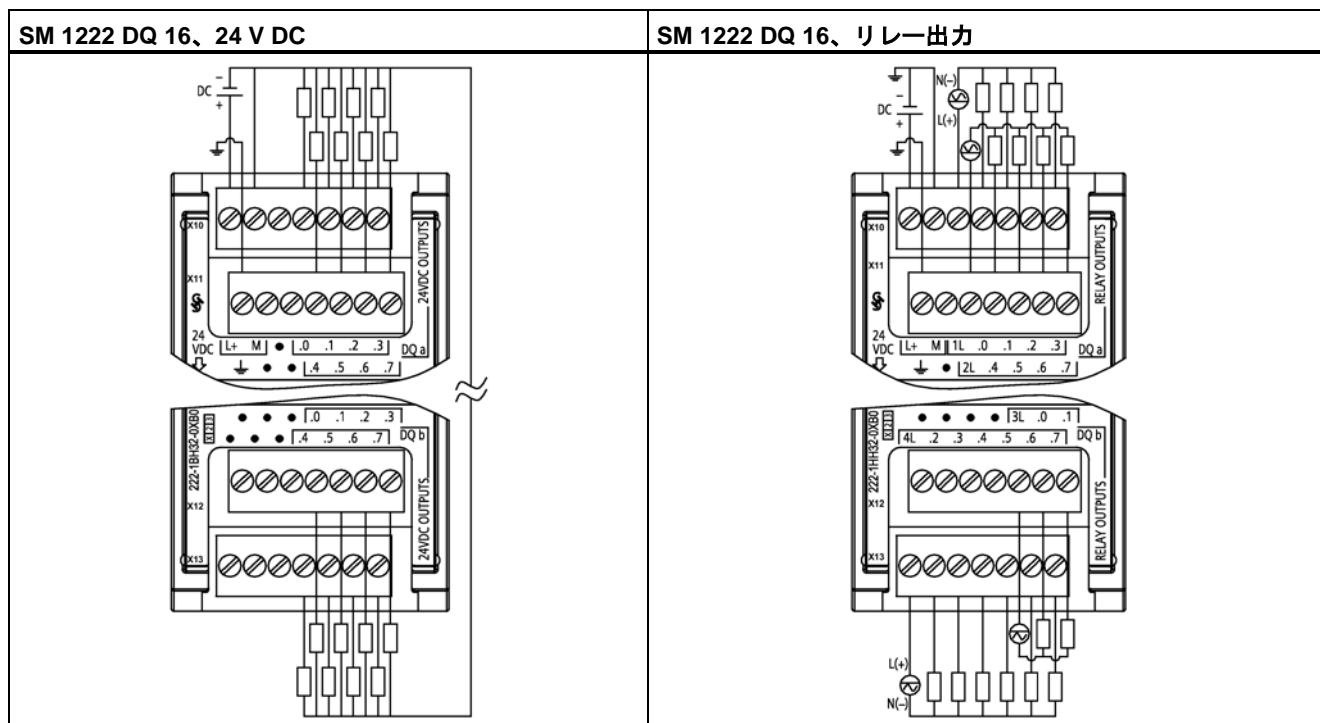
① シンク入力の場合、「-」を「M」に接続します(図を参照)。ソース入力の場合、「+」を「M」に接続します。

A.3.3 SM 1222 デジタル出力(DQ)

表 A- 20 SM 1222 デジタル出力(DQ)

技術データ		SM 1222 DQ (リレー)	SM 1222 DQ (24 V DC)
製品番号		<ul style="list-style-type: none"> DQ 8: 6ES7 222-1HF32-0XB0 DQ 8: 切り替え: 6ES7 222-1XF32-0XB0 DQ 16: 6ES7 222-1HH32-0XB0 	<ul style="list-style-type: none"> DQ 8: 6ES7 222-1BF32-0XB0 DQ 16: 6ES7 222-1BH32-0XB0
出力数(DQ) 仕様 (Page 386)を参照。		<ul style="list-style-type: none"> 8 (DQ 8 および DQ 8 切り替え) 16 (DQ 16) 	<ul style="list-style-type: none"> 8 (DQ 8) 16 (DQ 16)
外形寸法 W x H x D (mm)		<ul style="list-style-type: none"> DQ 8 および DQ 16: 45 x 100 x 75 DQ 8 切り替え: 70 x 100 x 75 	45 x 100 x 75
重量		<ul style="list-style-type: none"> DQ 8: 190 グラム DQ 8 切り替え: 310 グラム DQ 16: 260 グラム 	<ul style="list-style-type: none"> DQ 8: 180 グラム DQ 16: 220 グラム
電力損失		<ul style="list-style-type: none"> DQ 8: 4.5 W DQ 8 切り替え: 5 W DQ 16: 8.5 W 	<ul style="list-style-type: none"> DQ 8: 1.5 W DQ 16: 2.5 W
消費電流	SM バス	<ul style="list-style-type: none"> DQ 8: 120 mA DQ 8 切り替え: 140 mA DQ 16: 135 mA 	<ul style="list-style-type: none"> DQ 8: 120 mA DQ 16: 140 mA
	24 V DC	<ul style="list-style-type: none"> DQ 8 および DQ 16: 11 mA / 使用するリレーコイル DQ 8 切り替え: 16.7 mA / 使用するリレーコイル 	<ul style="list-style-type: none"> DQ 8: -- DQ 16: --

表 A-21 SM 1222 デジタル出力(DQ)モジュールの配線図

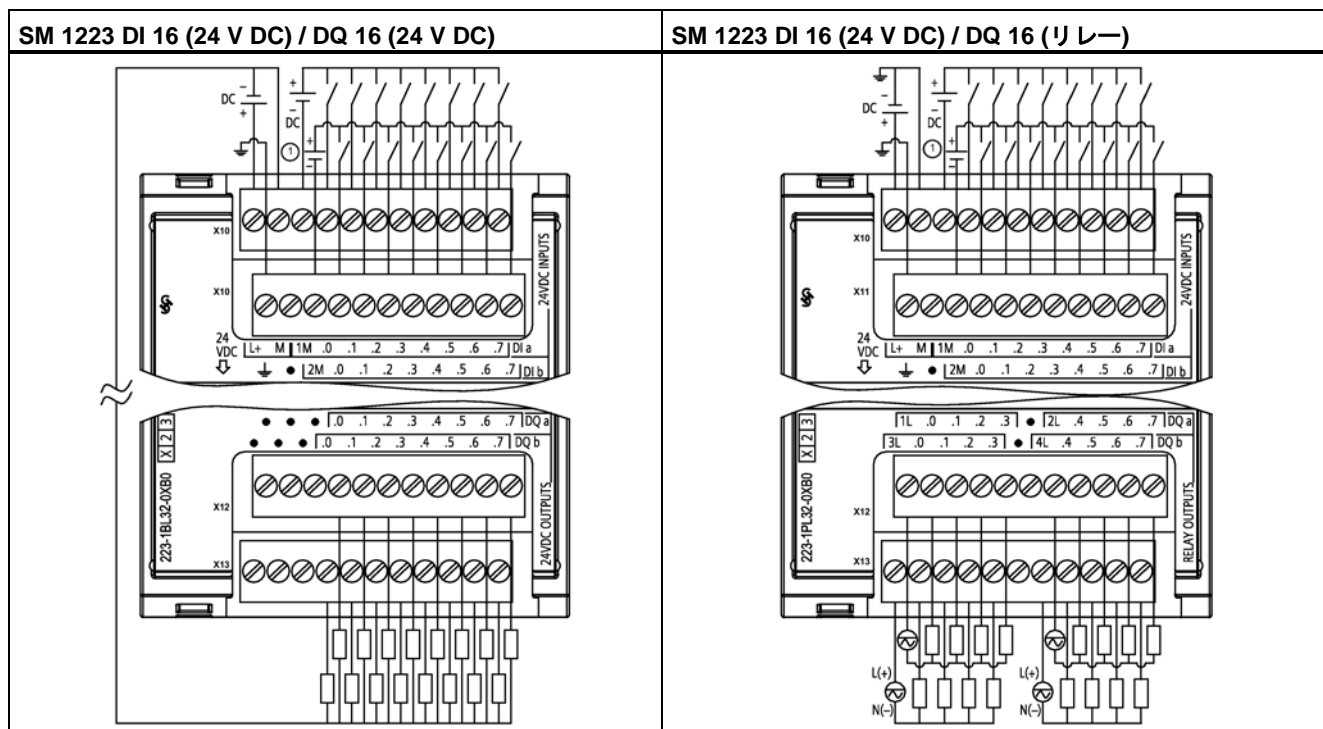


A.3.4 SM 1223 V DC デジタル入出力(DI / DQ)

表 A-22 SM 1223 デジタル入出力(DI / DQ)組み合わせモジュール

技術データ		SM 1223 DI (24 V DC) / DQ (リレー)	SM 1223 DI (24 V DC) / DQ (24 V DC)
製品番号		DI 8 / DQ 8: 6ES7 223-1PH32-0XB0 DI 16 / DQ 16: 6ES7 223-1PL32-0XB0	DI 8 / DQ 8: 6ES7 223-1BH32-0XB0 DI 8 / DQ 8: 6ES7 223-1BL32-0XB0
入出力の数(DI / DQ) 仕様 (Page 386)を参照。		<ul style="list-style-type: none"> 入力: 8 または 16 (24 V DC) 出力: 8 または 16 (リレー) 	<ul style="list-style-type: none"> 入力: 8 または 16 (24 V DC) 出力: 8 または 16 (24 V DC)
外形寸法 W x H x D (mm)		<ul style="list-style-type: none"> DI 8 / DQ 8: 45 x 100 x 75 DI 16 / DQ 16: 70 x 100 x 75 	<ul style="list-style-type: none"> DI 8 / DQ 8: 45 x 100 x 75 DI 16 / DQ 16: 70 x 100 x 75
重量		<ul style="list-style-type: none"> DI 8 / DQ 8: 230 グラム DI 16 / DQ 16: 350 グラム 	<ul style="list-style-type: none"> DI 8 / DQ 8: 210 グラム DI 16 / DQ 16: 310 グラム
電力損失		<ul style="list-style-type: none"> DI 8 / DQ 8: 5.5 W DI 16 / DQ 16: 10 W 	<ul style="list-style-type: none"> DI 8 / DQ 8: 2.5 W DI 16 / DQ 16: 4.5 W
消費電流	SM バス	<ul style="list-style-type: none"> DI 8 / DQ 8: 145 mA DI 16 / DQ 16: 180 mA 	<ul style="list-style-type: none"> DI 8 / DQ 8: 145 mA DI 16 / DQ 16: 185 mA
	24 V DC	4 mA / 使用する入力 11 mA / 使用するリレーコイル	4 mA / 入力

表 A-23 SM 1223 DI / DQ 組み合わせモジュールの配線図



① シンク入力の場合、「-」を「M」に接続します(図を参照)。ソース入力の場合、「+」を「M」に接続します。

A.3.5 SM 1223 120/230 V AC 入力 / リレー出力

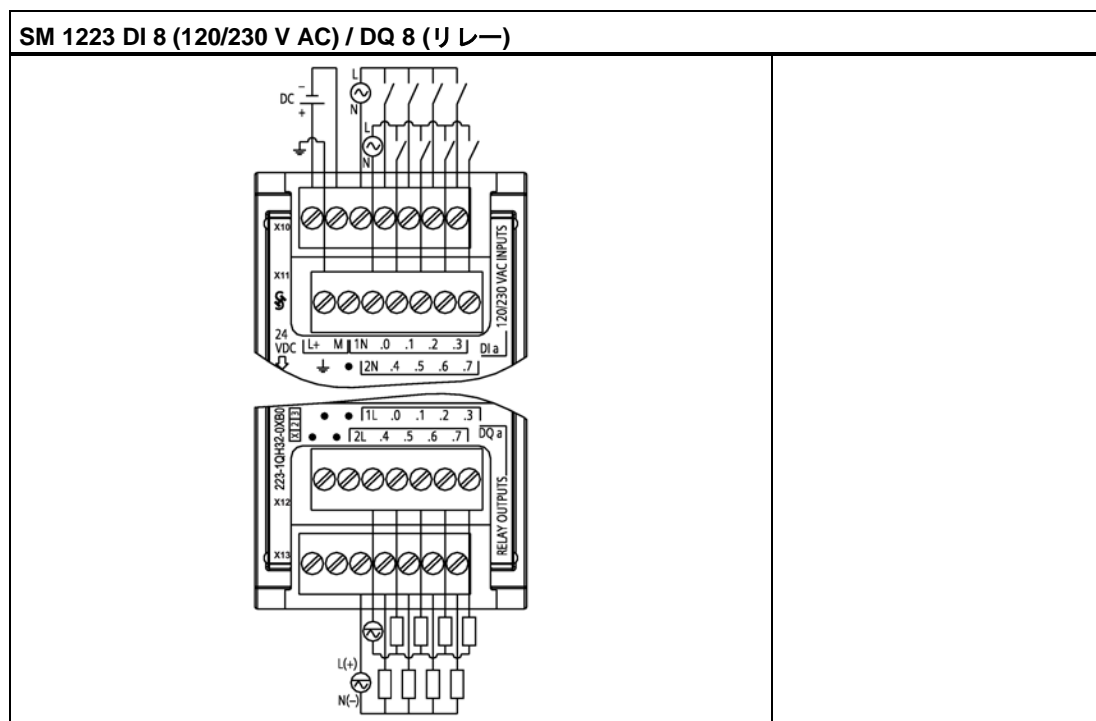
表 A- 24 SM 1223 V AC デジタル入出力(DI / DQ)組み合わせ

技術データ		SM 1223 DI (120/230 V AC) / DQ (リレー)
製品番号		DI 8 / DQ 8: 6ES7 223-1QH32-0XB0
入出力の数(DI / DQ)		入力: 8 (120/230 V AC) 120/230 V AC 入力の仕様(Page 388)を参照。 出力: 8 (リレー) デジタル出力の仕様(Page 389)を参照。
外形寸法 W x H x D (mm)		45 x 100 x 75
重量		190 グラム
電力損失		7.5 W
消費電流	SM バス	120 mA
	24 V DC	11 mA / 使用するリレーコイル

注記

SM 1223 DI 8 x 120/230 V AC、DQ 8 x リレーシグナルモジュール(6ES7 223-1QH32-0XB0)は、クラス 1、ディビジョン 2、ガスグループ A、B、C、D、温度クラス T4 Ta = 40 °C での使用のために承認されています。

表 A- 25 SM 1223 DI 8 (120/230 V AC) / DQ 8 (リレー)の配線図



A.4 デジタル入力および出力の仕様

A.4.1 24 V DC デジタル入力(DI)

表 A- 26 デジタル入力(DI)の仕様

技術データ	CPU、SM、および SB	高速 SB (200 kHz)
タイプ	<ul style="list-style-type: none"> CPU および SM: IEC タイプ 1 シンク (シンク/ソース) SB 1223: IEC タイプ 1 シンク (シンクのみ) 	SB 1221 200 kHz および SB 1223 200 kHz: ソース
定格電圧	24 V DC/ 4 mA、(公称)	24 V DC SB: 24 V DC/ 7 mA、(公称) 5 V DC SB: 5 V DC/ 15 mA、(公称)
常時許容電圧	30 V DC (最大)	24 V DC SB: 28.8 V DC 5 V DC SB: 6 V DC
サージ電圧	35 V DC/ 0.5 秒	24 V DC SB: 35 V DC/ 0.5 秒 5 V DC SB: 6 V
ロジック 1 信号(最小)	15 V DC/ 2.5 mA	24 V DC SB: L+- 10 V DC/ 2.9 mA 5 V DC SB: L+- 2.0 V DC/ 5.1 mA
ロジック 0 信号(最大)	5 V DC/ 1 mA	24 V DC SB: L+- 5 V DC/ 1.4 mA 5 V DC SB: L+- 1.0 V DC/ 2.2 mA
絶縁(フィールド側からロジック)	500 V AC/ 1 分間	500 V AC/ 1 分間
絶縁グループ	<ul style="list-style-type: none"> CPU: 1 SM 1221 DI 8: 2 SM 1221 DI 16: 4 SB 1223 DI 2: 1 SM 1223: 2 	<ul style="list-style-type: none"> SB 1221 DI 4: 1 SB 1223 DI 2: 1
フィルタ時間	0.2、0.4、0.8、1.6、3.2、6.4、12.8 ms (4 グループで選択可能)	0.2、0.4、0.8、1.6、3.2、6.4、12.8 ms (4 グループで選択可能)

技術データ	CPU、SM、および SB	高速 SB (200 kHz)
同時入力数	<ul style="list-style-type: none"> • SM 1221 および SM 1223 DI 8: 8 • SM 1221 および SM 1223 DI 16: 16 • SB 1223 DI 2: 2 • CPU 1211C: 6 (60 °C 横置きまたは 50 °C 縦置きで) • CPU 1212C: 4 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 8 (55 °C 横置きまたは 45 °C 縦置きで) • CPU 1214C、CPU 1215C: 7 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 14 (55 °C 横置きまたは 45 °C 縦置きで) • CPU 1217C: 60 °C 横置きまたは 50 °C 縦置きで、5 シンク/ソース入力(隣り合ったポイントなし)と、4 ディファレンシャル入力; 14 (55 °C 横置きまたは 45 °C 縦置きで) 	<ul style="list-style-type: none"> • SB 1221 DI 4: 4 • SB 1223 DI 2: 2
ケーブル長(メートル)	<ul style="list-style-type: none"> • 500 m シールドあり、300 m シールドなし • CPU: HSC 用のシールドあり 50 m 	50 m シールドありツイストペア

注記

周波数を 20 kHz より高い周波数に切り替えるとき、重要なことは、デジタル入力が方形波を受信するということです。入力に対する信号品質を改善するために、以下のオプションを検討してください。

- ケーブル長の最小化
- シンクだけのドライバからシンクとソースを備えたドライバへの変更
- より高い品質のケーブルへの変更
- 回路/コンポーネントの電圧を 24 V から 5 V へ低減(製品の低電圧動作が可能な場合)。『S7-1200 プログラマブルコントローラシステムマニュアル』の完全な仕様を参照してください。
- 入力への外部負荷の追加

A.4 デジタル入力および出力の仕様

表 A-27 HSC クロック入力速度(最大)

技術データ	単相	直角位相
CPU 1211C	100 kHz	80 kHz
CPU 1212C	100 kHz (la.0～la.5)および 30 kHz (la.6～la.7)	80 kHz (la.0～la.5)および 20 kHz (la.6～la.7)
CPU 1214C、CPU 1215C	100 KHz (la.0～la.5)および 30 KHz (la.6～la.5)	80 kHz (la.0～la.5)および 20 kHz (la.6～lb.5)
CPU 1217C	1 MHz (lb.2～lb.5) 100 kHz (la.0～la.5) 30 kHz (la.6～lb.1)	1 MHz (lb.2～lb.5) 80 kHz (la.0～la.5) 20 kHz (la.6～lb.1)
高速(200 kHz) SB	200 kHz	160 kHz
標準速度 SB	30 kHz	20 kHz

¹ ロジック 1 レベル = 15～26 V DC

A.4.2 120/230 V AC デジタル AC 入力

表 A-28 120/230 V AC デジタル入力

技術データ		SM
タイプ		IEC タイプ 1
定格電圧		120 V AC/ 6 mA、230 V AC/ 9 mA
常時許容電圧		264 V AC
サージ電圧		--
ロジック 1 信号(最小)		79 V AC/ 2.5 mA
ロジック 0 信号(最大)		20 V AC/ 1 mA
漏れ電流(最大)		1 mA
絶縁(フィールド側からロジック)		1500 V AC/ 1 分間
絶縁グループ ¹		4
入力遅延時間		<ul style="list-style-type: none"> 通常: 0.2～12.8 ミリ秒、ユーザーが選択可能 最大: --
2 線式近接センサ(Bero)の接続(最大)		1 mA
ケーブル長	シールドなし	300 m
	シールドあり	500 m
同時入力数		8

¹ 1 つのグループ内のチャンネルは同一位相であることが必要です。

A.4.3 デジタル出力(DQ)

表 A- 29 デジタル出力(DQ)の仕様

技術データ	リレー (CPU および SM)	24 V DC (CPU、SM、および SB)	200 kHz 24 V DC (SB)
タイプ	リレー、ドライ接点	ソリッドステート - MOSFET (ソース)	ソリッドステート - MOSFET (シンク/ ソース)
電圧範囲	5~30 V DC または 5~250 V AC	20.4~28.8 V DC	20.4~28.8 V DC ¹ 4.25~6.0 V DC ²
最大電流でロジック 1 信号	--	20 V DC (最小)	L+- 1.5 V ¹ L+- 0.7 V ²
10 kΩ 負荷付きのロジック 0 信号	--	CPU: 20 V DC (最小)、 0.1 V DC (最大) SB: 0.1 V DC (最大) SM DC: 0.1 V DC (最大)	1.0 V DC (最大) ¹ 0.2 V DC (最大) ²
電流(最大)	2.0 A	0.5 A	0.1 A
ランプ負荷	30 W DC/200 W AC	SB: 5 W	--
ON ステート抵抗	新規のとき最大 0.2 Ω	最大 0.6 Ω	最大 11 Ω ¹ または最大 7 Ω ²
オフ状態抵抗	--	--	最大 6 Ω ¹ または最大 0.2 Ω ²
ポイント当たりの漏れ電流	--	最大 10 μA	--
パルストレイン出力レート	CPU: N/A ³	CPU: 最大 100 kHz、最小 2 Hz ⁴ SB: 最大 20 kHz、最小 2 Hz ⁵	最大 200 kHz、最小 2 Hz
サージ電流	接点が閉じた状態で 7 A	CPU: 最大 100 ms で 8 A SB: 最大 100 ms で 5 A SM: 最大 100 ms で 8 A	0.11 A
過負荷保護	なし	なし	なし
絶縁(フィールド側から ロジック)	コイルから接点: 1500 V AC/ 1 分間 コイルからロジック: なし	500 V AC/ 1 分間	500 V AC/ 1 分間
絶縁グループ	<ul style="list-style-type: none"> CPU 1211C: 1 CPU 1212C: 2 CPU 1214C: 2 CPU 1215C: 2 SM DQ 8: 2 SM DQ 8 切り替え: 8 SM DQ 16: 4 	<ul style="list-style-type: none"> CPU: 1 SB: 1 SM (DQ 8): 1 SM (DQ 16): 1 	1 ⁵
絶縁抵抗	新規の場合 100 MΩ 最小	--	--
開いた接点間の絶縁	750 V AC/ 1 分間	--	--
同時出力数	CPU 1211C: 4 (60 °C 横置きまたは 50 °C 縦置きで)	--	--

A.4 デジタル入力および出力の仕様

技術データ	リレー (CPU および SM)	24 V DC (CPU、SM、および SB)	200 kHz 24 V DC (SB)
	CPU 1212C: 3 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 6 (55 °C 横置きまたは 45 °C 縦置きで)		--
	CPU 1214C: 5 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 10 (55 °C 横置きまたは 45 °C 縦置きで)		--
	CPU 1215C: 5 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 10 (55 °C 横置きまたは 45 °C 縦置きで)		--
	CPU 1217C: 60 °C 横置きまたは 50 °C 縦置きで、3 ソリッドステート - MOSFET (ソース)出力(隣り合ったポイントなし)と、4 ディファレンシャル出力 10 (55 °C 横置きまたは 45 °C 縦置きで)		--
	SM 1222 DQ8: 8	SM 1222 DQ8: 8	--
	SM1222 DQ 8 切り替え: 4 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで)、8 (55 °C 横置きまたは 45 °C 縦置きで)		
	SM 1223 DI 8/DQ 8 リレー: 8		
	SM 1222 DQ16: 8 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 16 (55 °C 横置きまたは 45 °C 縦置きで)	SM 1222 DQ16: 16	--
	SM 1223 DI 16/DQ 16 リレー: 8 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 16 (55 °C 横置きまたは 45 °C 縦置きで)	SM 1223 DI 8/DQ 8: 8	
	SM 1223 DI 8 x 120/230 V AC/DQ 8 リレー: 4、SM 1223 DI 16/DQ 16 リレー: 4 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 8 (55 °C 横置きまたは 45 °C 縦置きで)	SM 1223 DI 16/DQ 16: 16	
		SB 1223 DI 1 DQ 2、SM 1223 DI2 DQ2: 2	SB 1222 DQ 4: 2 (隣り合ったポイントなし、60 °C 横置きまたは 50 °C 縦置きで); 4 (55 °C 横置きまたは 45 °C 縦置きで)
			SB 1222 DQ 4 x 5 V DC: 4
			SB 1223 DI 2/DQ 2: 2
			SB 1223 DI 2/DQ 2: 2

技術データ	リレー (CPU および SM)	24 V DC (CPU、SM、および SB)	200 kHz 24 V DC (SB)
共通定格による電流	SM リレー: <ul style="list-style-type: none"> SM 1222 DQ 8 および DQ 16: 10 A SM 1222 DQ 8 切り替え: 2A SM 1223 DI 8/DQ 8:10 A SM 1223 DI 16/DQ 16: 8 A SM 1223 DI 8x120/230 V AC/DQ 8 リレー: 10 	SM 24 V DC <ul style="list-style-type: none"> SM 1222 DQ 16: 8 A SM 1223 DI 8/DQ 8: 4 A SM 1223 DI 16/DQ 16: 8 A 	--
誘導クランプ電圧	--	L+- 48 V、 1 W 損失	なし
最大リレースイッチング 周波数	1 Hz	--	--
切り替え遅延	最大 10 ms	CPU: <ul style="list-style-type: none"> Qa.0~Qa.3: 最大 1.0 μs、 オフからオン 最大 3.0 μs、オンからオフ Qa.4~Qb.1: 最大 50 μs、 オフからオン 最大 200 μs、オンからオフ SB: 最大 2 μ s、オフからオン; 最大 10 μ s、オンからオフ SM: 最大 50 μ s、オフからオン 最大 200 μ s、オンからオフ	1.5 μ s + 300 ns の立ち上がり ¹ 1.5 μ s + 300 ns の立ち下がり ¹ 200 ns + 300 ns の立ち上がり ² 200 ns + 300 ns の立ち下がり ²
機械寿命(負荷なし)	リレー: 10,000,000 回の開閉 サイクル	--	--
定格負荷での接点寿命	リレー: 100,000 回の開閉サ イクル	--	--
RUN から STOP の動作	最後の値または代替値 (デフォルト値 0)	最後の値または代替値 (デフォルト値 0)	最後の値または代替値 (デフォルト値 0)
ケーブル長(メートル)	500 m シールドあり、 150 m シールドなし	500 m シールドあり、 150 m シールドなし	50 m シールドありツイ ストペア

¹ 24 V DC 200 kHz SB² 5 V DC 200 kHz SB³ リレー出力付きの CPU モデルの場合、デジタルシグナルボード(SB)を設置して、パルス出力を使用する必要があります。⁴ 使用しているパルスレシーバおよびケーブルによっては、負荷抵抗を追加すると(定格電流の少なくとも 10%)、パルス信号品質およびノイズ耐性が改善する場合があります。⁵ SB 1223 200 kHz DI 2 / DQ 2: 入力に対する絶縁はありません。

A.5 アナログ I/O モジュール

S7-1200 で使用できるモジュールのより完全なリストは、『S7-1200 プログラマブルコントローラシステムマニュアル』またはカスタマサポート Web サイト (<http://www.siemens.com/tiaportal>) を参照してください。

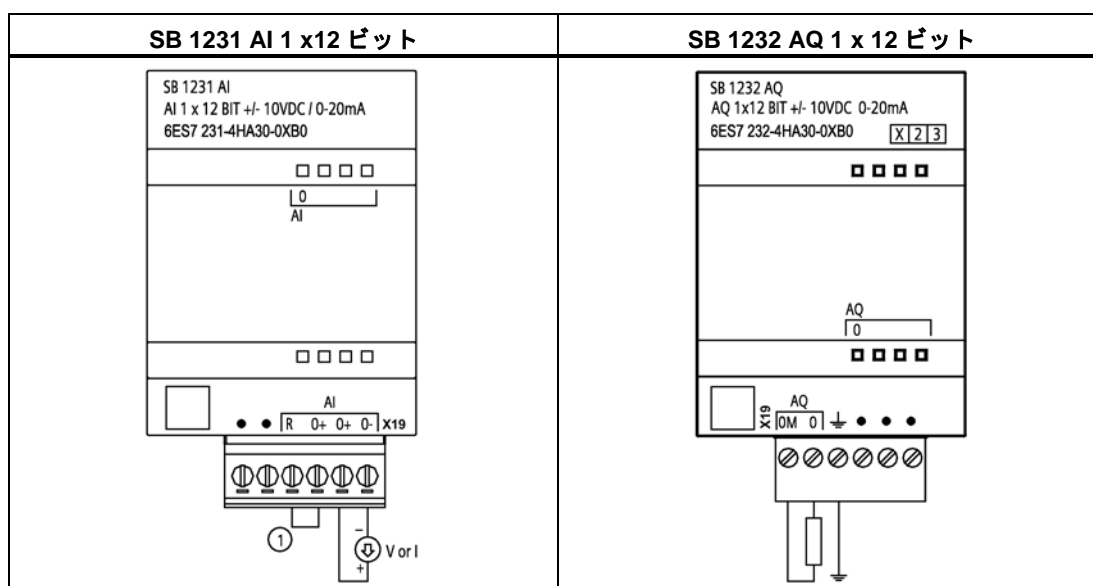
A.5.1 SB 1231 および SB 1232 アナログ入力(AI)および出力(AQ)

表 A- 30 一般仕様

技術データ	SB 1231 AI 1 x12 ビット ¹	SB 1232 AQ 1 x 12 ビット
製品番号	6ES7 231-4HA30-0XB0	6ES7 232-4HA30-0XB0
外形寸法 W x H x D (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
重量	35 グラム	40 グラム
電力損失	0.4 W	1.5 W
消費電流(SM バス)	55 mA	15 mA
消費電流(24 V DC)	なし	40 mA (負荷なし)
入出力の数	1	1
タイプ	電圧または電流(ディファレンシャル)	電圧または電流

¹ SB 1231 AI 1 x アナログ入力を使用するには、CPU のファームウェアが V2.0 以降であることが必要です。

表 A- 31 アナログ SB の配線図



① 電流用に「R」と「0+」を接続します。

A.5.2 SM 1231 アナログ入力(AI)

表 A- 32 SM 1231 アナログ入力(AI)

技術データ	SM 1231 AI 4 x 13 ビット	SM 1231 AI 8 x 13 ビット	SM 1231 AI 4 x 16 ビット
製品番号(MLFB)	6ES7 231-4HD32-0XB0	6ES7 231-4HF32-0XB0	6ES7 231-5ND32-0XB0
入力数	4 入力(AI)	8 入力(AI)	4 入力
タイプ	電圧または電流(ディファレンシャル)、グループ 2 で選択可能	電圧または電流(ディファレンシャル)、グループ 2 で選択可能	電圧または電流(ディファレンシャル)
外形寸法 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
重量	180 グラム	180 グラム	180 グラム
電力損失	1.5 W	1.5 W	1.8 W
消費電流(SM バス)	80 mA	90 mA	80 mA
消費電流(24 V DC)	45 mA	45 mA	65 mA

A.5.3 SM 1232 アナログ出力(AQ)

表 A- 33 SM 1232 アナログ出力(AQ)

技術データ	SM 1232 AQ 2 x 14 ビット	SM 1232 AQ 4 x 14 ビット
製品番号(MLFB)	6ES7 232-4HB32-0XB0	6ES7 232-4HD32-0XB0
出力の数とタイプ	2 出力(AQ)	4 出力(AQ)
外形寸法 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	180 グラム	180 グラム
電力損失	1.5 W	1.5 W
消費電流(SM バス)	80 mA	80 mA
消費電流(24 V DC)	45 mA (負荷なし)	45 mA (負荷なし)

A.5.4 SM 1234 アナログ入出力(AI/AQ)

表 A-34 SM 1234 アナログ入出力(AI / AQ)組み合わせ

技術データ	SM 1234 AI 4 x 13 ビット / AQ 2 x 14 ビット
製品番号(MLFB)	6ES7 234-4HE32-0XB0
入力数 タイプ	4 入力(AI) 電圧または電流(ディファレンシャル)、グループ 2 で選択可能
出力数 タイプ	2 出力(AQ) 電圧または電流(ディファレンシャル)
外形寸法 W x H x D (mm)	45 x 100 x 75
重量	220 グラム
電力損失	2.0 W
消費電流(SM バス)	80 mA
消費電流(24 V DC)	60 mA (負荷なし)

A.5.5 M 1231 (AI)、SM 1232 (AQ)、および SM 1234 (AI/AQ)の配線図

表 A-35 アナログ SM の配線図

SM 1231 AI 8 x 13 ビット	SM 1232 AQ 4 x 13 ビット	SM 1234 AI 4 x 13 ビット / AQ 2 x 14 ビット

注記

未使用の電圧入力チャンネルは、短絡する必要があります。

未使用の電流入力チャンネルは 0~20 mA の範囲に設定するか、断線エラー報告を無効化します。

電流モードに設定された入力は、モジュールに電力を供給し、適切な設定を行わないかぎり、ループ電流を伝導しません。

電流入力チャンネルは、外部電力がトランスミッタに供給されないかぎり、動作しません。

A.6 BB 1297 バッテリーボード

BB 1297 バッテリーボード

表 A- 36 一般仕様

技術データ	BB 1297 バッテリー
製品番号	6ES7 297-0AX30-0XA0
外形寸法 W x H x D (mm)	38 x 62 x 21
重量	28 グラム
リアルタイムクロックの保持時間	約 1 年
バッテリーのタイプ	CR1025 ¹
CPU の「保守」LED	バッテリーの交換が必要であることを指示します
ユーザープログラム	アプリケーション/システムが、バッテリーのステータスを評価できます

¹ BB 1297 の設置または BB のバッテリーの交換については、『S7-1200 プログラマブルコントローラシステムマニュアル』の第 2 章「設置」を参照してください。

BB 1297 バッテリーボードは、リアルタイムクロックの保持時間が 1 ヶ月を超えるような用途のために使用されます。以下に、BB 1297 バッテリーボードの機能を示します。

- PLC の電源オフ中にも時刻クロックをサポートしています。S7-1200 CPU は、BB 1297 バッテリーボードと連携して、最大 1 年間のアプリケーションの電源オフ期間中に時刻クロック保持をサポートします。
- 1 度に使用できるのは、BB 1297 バッテリーボードまたは他の SB の 1 つだけです。
- ホット差し込み接続/ホットスワッピングは許可されていません。CPU の電源がオフの間だけ、BB 1297 バッテリーボードの交換または差し込み接続を行うことができます。CPU の電源がオフで、BB 1297 が現在のバッテリーを交換するために取り外されている間は（ユーザーがバッテリーを交換中）、内部のスーパーコンデンサが時刻を保持します。
- バッテリーの交換が必要なときは、CPU の「Maint」LED が指示します。
- ユーザープログラムを使用して、バッテリーおよびバッテリーボードのステータスをモニタまたはチェックして、必要な場合、ユーザーメッセージを HMI または Web サーバーに表示することができます。

A.7 アナログ I/O の仕様

A.7.1 アナログ入力(CPU、SM、および SB)の仕様

表 A- 37 アナログ入力(AI)の仕様

技術データ	CPU	SB	SM
タイプ	電圧(シングルエンド)	電圧または電流(ディファレンシャル)	電圧または電流(ディファレンシャル)、グループ 2 で選択可能
範囲	0~10 V	±10 V、±5 V、±2.5、 0~20 mA、または 4 mA~20 mA	±10 V、±5 V、±2.5 V、 0~20 mA、または 4 mA~20 mA
解像度	10 ビット	11 ビット+符号ビット	12 ビット+符号ビット
フルスケールレンジ (データワード)	0~27,648	-27,648~27,648	-27,648~27,648
精度 (25 °C / -20~60 °C)	フルスケールの 3.0% / 3.5%	フルスケールの±0.3% / ±0.6%	フルスケールの±0.1% / ±0.2%
オーバーシュート/アンダー シュート範囲(データワード) (注 1 を参照)	電圧: 27,649~32,511	電圧: 32,511~27,649 / -27,649~-32,512	電圧: 32,511~27,649 / -27,649~-32,512
	電流: N/A	Current: 32,511~27,649 / 0~-4864	Current: 32,511~27,649 / 0~-4864
オーバーフロー/アンダーフ ロー(データワード) (注 1 を参照)	電圧: 32,512~32,767	電圧: 32,767~32,512 / -32,513~-32,768	電圧: 32,767~32,512 / -32,513~-32,768
	電流: N/A	Current: 32,767~32,512 / -4865~-32,768	Current: 32,767~32,512 / -4865~-32,768
最大耐電圧/電流	35 V DC (電圧)	±35 V / ±40 mA	±35 V / ±40 mA
平滑化 (注 2 を参照)	なし、弱、中、または強	なし、弱、中、または強	なし、弱、中、または強
ノイズ除去 (注 2 を参照)	10、50、または 60 Hz	400、60、50、または 10 Hz	400、60、50、または 10 Hz
測定原理	現在値の変換	現在値の変換	現在値の変換
コモンモード除去	なし	40 dB、DC~60 Hz	40 dB、DC~60 Hz
有効な信号範囲(信号+コモン モード電圧)	+12 V 未満で 0 V より大きい	+35 V 未満で 35 V より大きい	+12 V 未満で 12 V より大きい

技術データ	CPU	SB	SM
負荷インピーダンス	シングルエンド: 100 K Ω 以上	ディファレンシャル: 220 K Ω (電圧)、 250 Ω (電流) コモンモード: 55 K Ω (電圧)、 55 Ω (電流)	ディファレンシャル: 9 M Ω (電圧)、 250 Ω (電流) コモンモード: 4.5 M Ω (電圧)、 4.5 M Ω (電流)
絶縁(フィールド側からロジック)	なし	なし	なし
ケーブル長(メートル)	100 m、シールドありツイストペア	100 m、ツイストありおよびシールドあり	100 m、ツイストありおよびシールドあり
診断	オーバーフロー/アンダーフロー	オーバーフロー/アンダーフロー	オーバーフロー/アンダーフロー 24 V DC (低電圧)
注記 1: オーバーシュート/アンダーシュートおよびオーバーフロー/アンダーフローの範囲を識別するには、アナログ入力 の電圧および電流の測定範囲(Page 398)を参照してください。			
注記 2: 平滑化およびノイズ除去の値を識別するには、ステップ応答時間(Page 400)を参照してください。			

A.7.2 電圧および電流の入力(AI)測定範囲

表 A-38 電圧のアナログ入力表示(SB および SM)

システム		電圧測定範囲				
10 進数	16 進数	± 10 V	± 5 V	± 2.5 V	± 1.25 V	
32767	7FFF ¹	11.851 V	5.926 V	2.963 V	1.481 V	オーバーフロー
32512	7F00					
32511	7EFF	11.759 V	5.879 V	2.940 V	1.470 V	オーバーシュート範囲
27649	6C01					
27648	6C00	10 V	5 V	2.5 V	1.250 V	定格範囲
20736	5100	7.5 V	3.75 V	1.875 V	0.938 V	
1	1	361.7 μ V	180.8 μ V	90.4 μ V	45.2 μ V	
0	0	0 V	0 V	0 V	0 V	
-1	FFFF					
-20736	AF00	-7.5 V	-3.75 V	-1.875 V	-0.938 V	
-27648	9400	-10 V	-5 V	-2.5 V	-1.250 V	アンダーシュート範囲
-27649	93FF					
-32512	8100	-11.759 V	-5.879 V	-2.940 V	-1.470 V	アンダーフロー
-32513	80FF					
-32768	8000	-11.851 V	-5.926 V	-2.963 V	-1.481 V	

¹ 以下の理由の 1 つのために、7FFF が戻される場合があります。有効な値を入手できる前に(たとえば、電源投入後直ちに)、オーバーフロー(この表に記載)が発生するか、断線が検出された場合。

表 A- 39 電流のアナログ入力表示(SB および SM)

システム		電流測定範囲		
10 進数	16 進数	0 mA～20 mA	4 mA～20 mA	
32767	7FFF	23.70 mA	22.96 mA	オーバーフロー
32512	7F00			
32511	7EFF	23.52 mA	22.81 mA	オーバーシュート範囲
27649	6C01			
27648	6C00	20 mA	20 mA	公称範囲
20736	5100	15 mA	16 mA	
1	1	723.4 nA	4 mA + 578.7 nA	
0	0	0 mA	4 mA	
-1	FFFF			アンダーシュート範囲
-4864	ED00	-3.52 mA	1.185 mA	
-4865	ECFF			アンダーフロー
-32768	8000			

表 A- 40 電圧のアナログ入力表示(CPU 1215C および CPU 1217C)

システム		電圧測定範囲	
10 進数	16 進数	0～10 V	
32767	7FFF	11.851 V	オーバーフロー
32512	7F00		
32511	7EFF	11.759 V	オーバーシュート範囲
27649	6C01		
27648	6C00	10 V	定格範囲
20736	5100	7.5 V	
34	22	12 mV	
0	0	0 V	
負の値		負の値はサポートされていません。	

A.7.3 アナログ入力(AI)のステップ応答

次の表に、CPU、SB、および SM のアナログ入力(AI)のステップ応答時間を示します。

表 A- 41 アナログ入力のステップ応答(ミリ秒)

平滑化の選択(サンプル平均化)		積分時間の選択			
		400 Hz (2.5 ミリ秒)	60 Hz (16.6 ミリ秒)	50 Hz (20 ミリ秒)	10 Hz (100 ミリ秒)
なし(1 サイクル): 平均化なし	CPU	N/A	63	65	130
	SB	4.5	18.7	22.0	102
	SM	4	18	22	100
弱(4 サイクル): 4 サンプル	CPU	N/A	84	93	340
	SB	10.6	59.3	70.8	346
	SM	9	52	63	320
中(16 サイクル): 16 サンプル	CPU	N/A	221	258	1210
	SB	33.0	208	250	1240
	SM	32	203	241	1200
強(32 サイクル): 32 サンプル	CPU	N/A	424	499	2410
	SB	63.0	408	490	2440
	SM	61	400	483	2410
サンプルレート	CPU	N/A	4.17	5	25
	SB	0.156	1.042	1.250	6.250

A.7.4 アナログ入力のサンプル時間と更新時間

表 A- 42 SM および CPU のサンプル時間と更新時間

遮断周波数(積分時間)	サンプル時間	すべてのチャンネルの更新時間		
		4 チャンネル SM	8 チャンネル SM	CPU AI
400 Hz (2.5 ミリ秒)	0.625 ミリ秒 ¹	2.5 ミリ秒	10 ミリ秒	N/A ミリ秒
60 Hz (16.6 ミリ秒)	4.170 ミリ秒	4.17 ミリ秒	4.17 ミリ秒	4.17 ミリ秒
50 Hz (20 ミリ秒)	5.000 ミリ秒	5 ミリ秒	5 ミリ秒	5 ミリ秒
10 Hz (100 ミリ秒)	25.000 ミリ秒	25 ミリ秒	25 ミリ秒	25 ミリ秒

¹ 8 チャンネル SM のサンプルレートは 1.250 ミリ秒です。

表 A- 43 SB のサンプル時間と更新時間

遮断周波数(積分時間)	サンプル時間	SB 更新時間
400 Hz (2.5 ミリ秒)	0.156 ミリ秒	0.156 ミリ秒
60 Hz (16.6 ミリ秒)	1.042 ミリ秒	1.042 ミリ秒
50 Hz (20 ミリ秒)	1.250 ミリ秒	1.25 ミリ秒
10 Hz (100 ミリ秒)	6.250 ミリ秒	6.25 ミリ秒

A.7.5 アナログ出力の仕様

表 A- 44 アナログ出力(SB および SM)の仕様

技術データ	SB	SM
タイプ	電圧または電流	電圧または電流
範囲	±10 V、0~20 mA、または 4~20 mA	±10 V、0~20 mA、または 4~20 mA
解像度	電圧: 12 ビット 電流: 11 ビット	電圧: 14 ビット 電流: 13 ビット
フルスケールレンジ (データワード) (注 1 を参照)	電圧: -27,648~27,648 電流: 0~27,648	電圧: -27,648~27,648 電流: 0~27,648
精度 (25 °C / -20~60 °C)	フルスケールの±0.5% / ±1%	フルスケールの±0.3% / ±0.6%
安定時間 (新しい値の 95%)	電圧: 300 μS (R)、750 μS (1 uF) 電流: 600 μS (1 mH)、2 ms (10 mH)	電圧: 300 μS (R)、750 μS (1 uF) 電流: 600 μS (1 mH)、2 ms (10 mH)
負荷インピーダンス	電圧: ≥ 1000 Ω 電流: 600 Ω 以下	電圧: ≥ 1000 Ω 電流: 600 Ω 以下
RUN から STOP の動作	最後の値または代替値(デフォルト値 0)	最後の値または代替値(デフォルト値 0)
絶縁 (フィールド側からロジック)	なし	なし
ケーブル長(メートル)	100 m、ツイストありおよびシールドあり	100 m、ツイストありおよびシールドあり
診断	<ul style="list-style-type: none"> オーバーフロー/アンダーフロー 短絡接地(電圧モードのみ) 断線(電流モードのみ) 	<ul style="list-style-type: none"> オーバーフロー/アンダーフロー 短絡接地(電圧モードのみ) 断線(電流モードのみ) 24 V DC (低電圧)
注記 1: フルスケール範囲については、電圧および電流の出力範囲(Page 402)を参照してください。		

A.7.6 電圧および電流の出力(AQ)測定範囲

表 A- 45 電圧のアナログ出力表示(SB および SM)

システム		電圧出力範囲	
10 進数	16 進数	$\pm 10 \text{ V}$	
32767	7FFF	注 1 を参照	オーバーフロー
32512	7F00	注 1 を参照	
32511	7EFF	11.76 V	オーバーシュート範囲
27649	6C01		
27648	6C00	10 V	定格範囲
20736	5100	7.5 V	
1	1	361.7 μV	
0	0	0 V	
-1	FFFF	-361.7 μV	
-20736	AF00	-7.5 V	
-27648	9400	-10 V	アンダーシュート範囲
-27649	93FF		
-32512	8100	-11.76 V	アンダーフロー
-32513	80FF	注 1 を参照	
-32768	8000	注 1 を参照	

¹ オーバーフローまたはアンダーフロー条件では、アナログ出力は、STOP モードの代替値を取得します。

表 A- 46 電流のアナログ出力表示(SB および SM)

システム		電流出力範囲		
10 進数	16 進数	0 mA~20 mA	4 mA~20 mA	
32767	7FFF	注 1 を参照	注 1 を参照	オーバーフロー
32512	7F00	注 1 を参照	注 1 を参照	
32511	7EFF	23.52 mA	22.81 mA	オーバーシュート範囲
27649	6C01			
27648	6C00	20 mA	20 mA	定格範囲
20736	5100	15 mA	16 mA	
1	1	723.4 nA	4 mA + 578.7 nA	
0	0	0 mA	4 mA	
-1	FFFF		4 mA~578.7 nA	アンダーシュート範囲
-6912	E500		0 mA	
-6913	E4FF			不可能です。出力値は 0 mA までに制限されます。
-32512	8100			
-32513	80FF	注 1 を参照	注 1 を参照	アンダーフロー
-32768	8000	注 1 を参照	注 1 を参照	

¹ オーバーフローまたはアンダーフロー条件では、アナログ出力は、STOP モードの代替値を取得します。

表 A- 47 電流のアナログ出力表示(CPU 1215C および CPU 1217C)

システム		電流出力範囲	
10 進数	16 進数	0 mA～20 mA	
32767	7FFF	注 1 を参照	オーバーフロー
32512	7F00	注 1 を参照	
32511	7EFF	23.52 mA	オーバーシュート範囲
27649	6C01		
27648	6C00	20 mA	定格範囲
20736	5100	15 mA	
34	22	0.0247 mA	
0	0	0 mA	
負の値		負の値はサポートされていません。	

¹ オーバーフロー条件では、アナログ出力は、デバイス構成のプロパティ設定に従って動作します。[Reaction to CPU STOP]パラメータで、次のいずれかを選択します。[代替値の使用(Use substitute value)]または[最後の値の保持(Keep last value)]。

A.8 RTD および熱電対モジュール

熱電対(TC)モジュール(SB 1231 TC および SM 1231 TC)は、アナログ入力に接続された電圧の値を測定します。この値は、TC からの温度である場合と、電圧の場合があります。

- 電圧の場合、公称範囲のフルスケール値は 10 進数 27648 です。
- 温度の場合、温度に 10 を掛けた値が報告されます(たとえば、25.3 度は 10 進数 253 として報告されます)。

RTD モジュール(SB 1231 RTD および SM 1231 RTD)は、アナログ入力に接続された抵抗の値を測定します。この値は、温度である場合と、抵抗である場合があります。

- 抵抗の場合、公称範囲のフルスケール値は 10 進数 27648 です。
- 温度の場合、温度に 10 を掛けた値が報告されます(たとえば、25.3 度は 10 進数 253 として報告されます)。

RTD モジュールは、センサの抵抗器への 2 線式、3 線式、および 4 線式接続によって測定を行います。

注記

RTD および TC モジュールは、センサが接続されていない有効化されたチャンネルが存在する場合、32767 を報告します。開いた線の検出も有効化されている場合、そのモジュールは対応する赤の LED を点滅します。

4 線式接続が使用されている場合に、10 Ω RTD 範囲に関して、最も高い精度が実現されます。

2 線式モードでの接続線の抵抗はセンサの読み取りでエラーを生じさせるため、精度は保証されません。

注記

電力が印加された後、そのモジュールはアナログ→デジタルコンバータのために、内部較正を実行します。この時間の間、それぞれのチャンネルに関して、有効なデータが該当のチャンネルで使用可能になるまで、そのモジュールは値 32767 を報告します。ユーザープログラムでは、この初期化時間の考慮が必要になる場合があります。モジュール構成によってこの初期化時間の長さは異なる場合があるため、ユーザーの構成でのモジュールの動作を検証しておく必要があります。必要に応じて、ユーザープログラムに、モジュールの初期化時間を調整するロジックを含めることができます。

A.8.1 SB 1231 RTD および SB 1231 TC 仕様

注記

これらの TC および RTD SB を使用するには、CPU のファームウェアが V2.0 以降であることが必要です。

表 A- 48 一般仕様

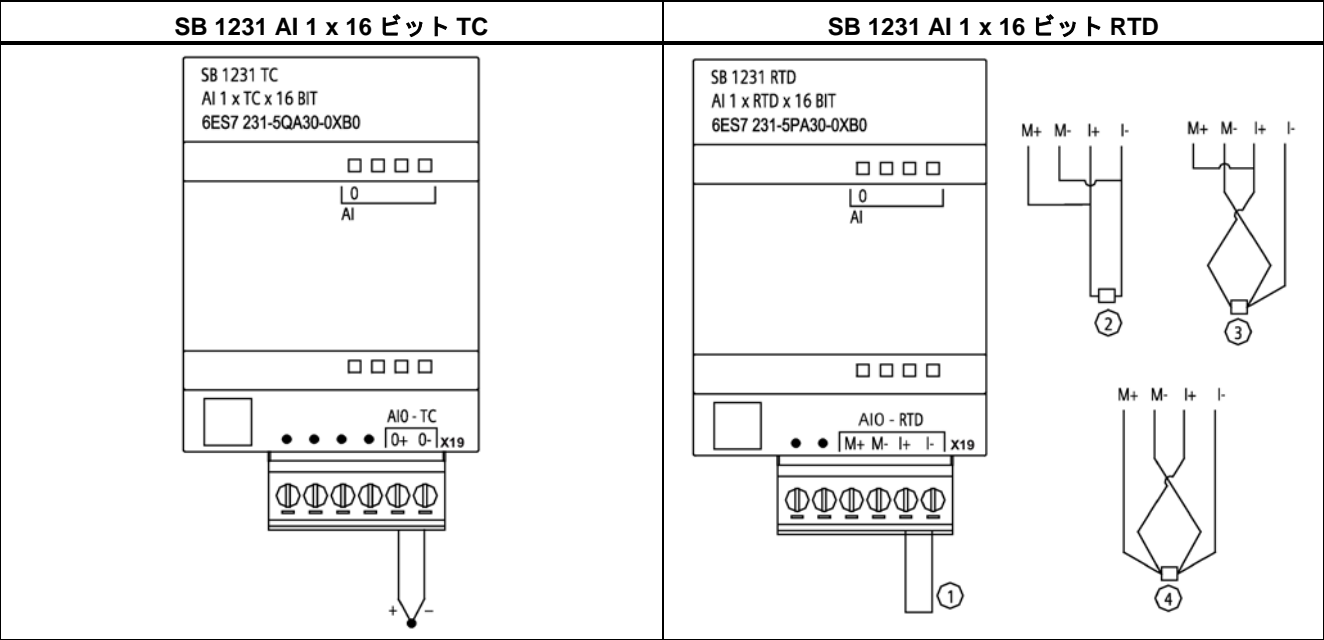
技術データ	SB 1231 AI 1 x 16 ビット TC	SB 1231 AI 1 x 16 ビット RTD
製品番号	6ES7 231-5QA30-0XB0	6ES7 231-5PA30-0XB0
外形寸法 W x H x D (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
重量	35 グラム	35 グラム
電力損失	0.5 W	0.7 W
消費電流(SM バス)	5 mA	5 mA
消費電流(24 V DC)	20 mA	25 mA
入力数(Page 410) タイプ	1 浮動 TC および mV	1 モジュール参照 RTD および Ω
診断	<ul style="list-style-type: none"> オーバーフロー/アンダーフロー^{1, 2} 断線³ 	<ul style="list-style-type: none"> オーバーフロー/アンダーフロー^{1, 2} 断線³

¹ オーバーフローおよびアンダーフロー診断アラーム情報は、モジュール構成でアラームが無効化されている場合でも、アナログデータ値に報告されます。

² RTD: 抵抗範囲では、アンダーフロー検出は絶対に有効化されません。

³ 断線アラームが無効化されていて、開いた配線条件がセンサ配線に存在する場合は、モジュールはランダムな値を報告できます。

表 A- 49 SB 1231 TC および RTD の配線図



- ① 未使用 RTD 入力のループバック
② 2 線式 RTD ③ 3 線式 RTD ④ 4 線式 RTD

A.8.2 SM 1231 RTD 仕様

表 A- 50 一般仕様

技術データ	SM 1231 AI 4 x RTD x 16 ビット	SM 1231 AI 8 x RTD x 16 ビット
製品番号	6ES7 231-5PD32-0XB0	6ES7 231-5PF32-0XB0
外形寸法 W x H x D (mm)	45 x 100 x 75	70 x 100 x 75
重量	220 グラム	270 グラム
電力損失	1.5 W	1.5 W
消費電流 (SM バス)	80 mA	90 mA
消費電流 ¹ (24 V DC)	40 mA	40 mA
入力数(Page 410) タイプ	4 モジュール参照 RTD および Ω	8 モジュール参照 RTD および Ω
診断	<ul style="list-style-type: none"> オーバーフロー/アンダーフロー^{2,3} 24 V DC 低電圧² 断線(電流モードのみ)⁴ 	<ul style="list-style-type: none"> オーバーフロー/アンダーフロー^{2,3} 24 V DC 低電圧² 断線(電流モードのみ)⁴

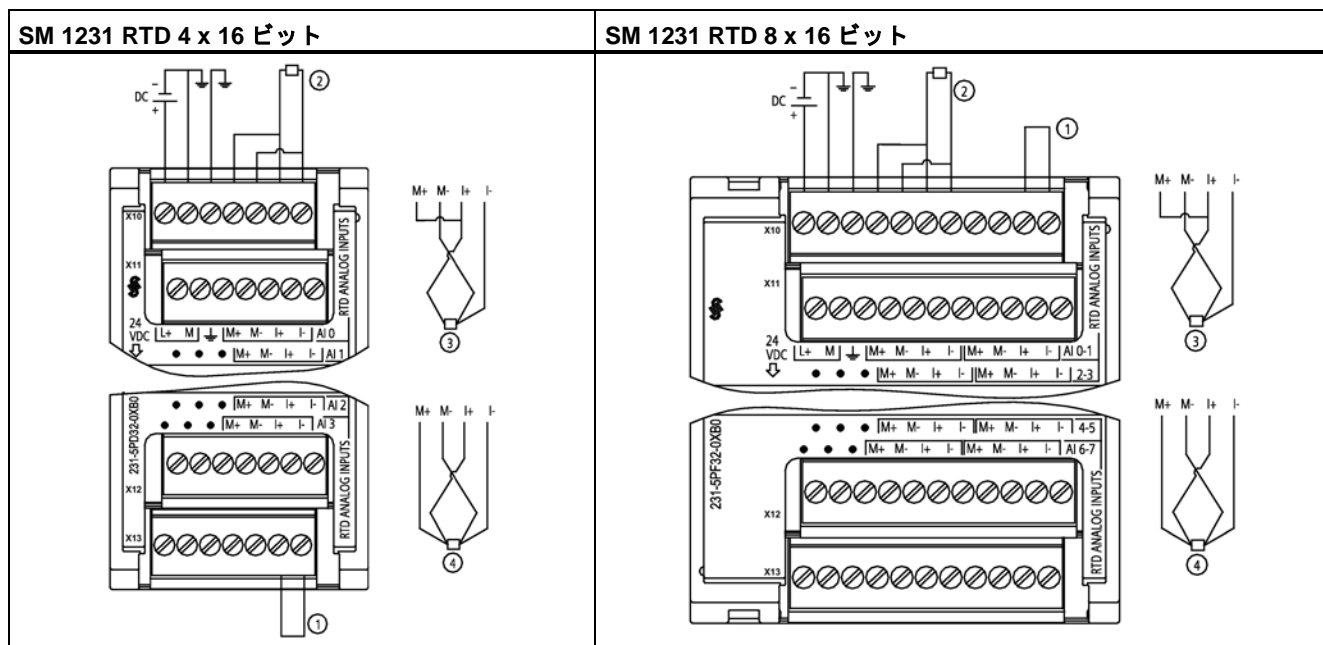
¹ 20.4~28.8 V DC (クラス 2、制限された電力、または CPU からのセンサ電力)

² オーバーフロー、アンダーフロー、および低電圧診断アラーム情報は、モジュール構成でアラームが無効化されている場合でも、アナログデータ値に報告されます。

³ 抵抗範囲では、アンダーフロー検出は絶対に有効化されません。

⁴ 断線アラームが無効化されていて、開いた配線条件がセンサ配線に存在する場合は、モジュールはランダムな値を報告できます。

表 A-51 RTD SM の配線図



① 未使用 RTD 入力のループバック

② 2 線式 RTD

③ 3 線式 RTD

④ 4 線式 RTD

注記: コネクタは金でなければなりません。『S7-1200 プログラマブルコントローラシステムマニュアル』の「付録 C」を参照してください。

A.8.3 SM 1231 TC 仕様

表 A- 52 一般仕様

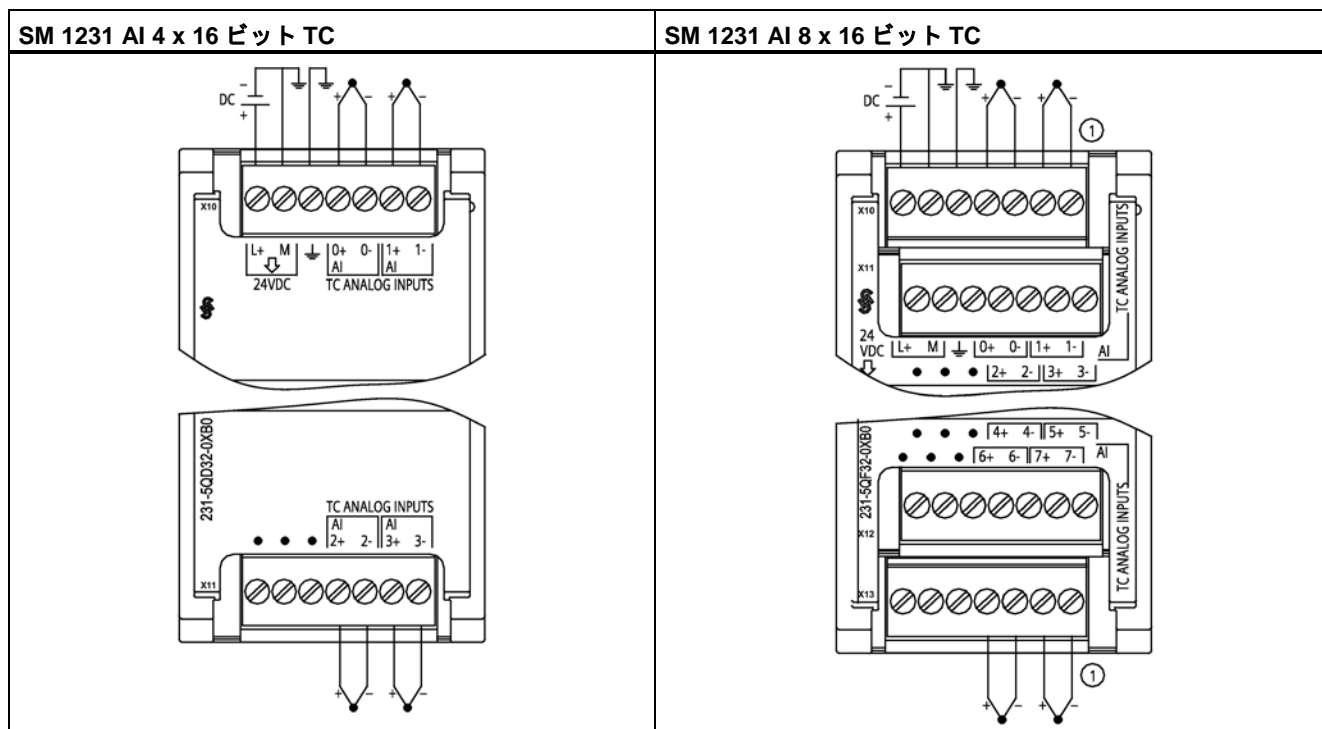
モデル	SM 1231 AI 4 x 16 ビット TC	SM 1231 AI 8 x 16 ビット TC
製品番号	6ES7 231-5QD32-0XB0	6ES7 231-5QF32-0XB0
外形寸法 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	180 グラム	xxx グラム
電力損失	1.5 W	1.5 W
消費電流 (SM バス)	80 mA	80 mA
消費電流 ¹ (24 V DC)	40 mA	40 mA
入力数(Page 410) タイプ	4 浮動 TC および mV	8 浮動 TC および mV
診断	<ul style="list-style-type: none"> • オーバーフロー/アンダーフロー² • 24 V DC 低電圧² • 断線(電流モードのみ)³ 	<ul style="list-style-type: none"> • オーバーフロー/アンダーフロー² • 24 V DC 低電圧² • 断線(電流モードのみ)³

¹ 20.4~28.8 V DC (クラス 2、制限された電力、または CPU からのセンサ電力)

² オーバーフロー、アンダーフロー、および低電圧診断アラーム情報は、モジュール構成でアラームが無効化されている場合でも、アナログデータ値に報告されます。

³ 断線アラームが無効化されていて、開いた配線条件がセンサ配線に存在する場合は、モジュールはランダムな値を報告できます。

表 A-53 TC SM の配線図



① SM 1231 AI 8 TC: 見やすくするために、TC 2、3、4、および 5 の接続は表示されていません。

A.8.4 RTD および TC (SM および SB) のアナログ入力仕様

表 A-54 RTD および TC モジュール(SM および SB) のアナログ入力

技術データ		RTD および熱電対(TC)
入力数		1 (SB)、4 または 8 (SM)
タイプ		<ul style="list-style-type: none"> RTD: モジュール参照 RTD および Ω TC: 浮動 TC および mV
範囲 <ul style="list-style-type: none"> 公称範囲(データワード) オーバーシュート/アンダーシュート範囲(データワード) オーバーフロー/アンダーフロー(データワード) 		RTD/TC タイプの表を参照: <ul style="list-style-type: none"> RTD (Page 413) TC (Page 412)
解像度	温度	0.1 °C / 0.1 °F
	抵抗/電圧	15 ビット+符号
最大耐電圧		± 35 V
ノイズ除去		選択されたフィルタ設定では 85 dB (10 Hz、50 Hz、60 Hz、または 400 Hz)
コモンモード除去		> 120 dB / 120 V AC

技術データ		RTD および熱電対(TC)
インピーダンス		$\geq 10 \text{ M}\Omega$
絶縁	フィールド側からロジック	500 V AC
	フィールド側から 24 V DC	SM RTD および SM TC: 500 V AC (SB RTD および SB TC の場合は適用されません)
	24 V DC からロジック	SM RTD および SM TC: 500 V AC (SB RTD および SB TC の場合は適用されません)
チャンネルからチャンネルの絶縁		<ul style="list-style-type: none"> SM RTD: なし (SB RTD の場合は適用されません) SM TC: 120 V AC (SB TC の場合は適用されません)
精度 (25 °C / -20~60 °C)		RTD/TC タイプの表を参照: <ul style="list-style-type: none"> RTD (Page 413) TC (Page 412)
繰り返し性		$\pm 0.05\% \text{ FS}$
最大センサ損失		<ul style="list-style-type: none"> RTD: 0.5 mW TC: 適用なし
測定原理		積分
モジュール更新時間		RTD/TC フィルタ選択の表を参照: <ul style="list-style-type: none"> RTD (Page 415) TC (Page 413)
冷接点エラー		<ul style="list-style-type: none"> RTD: 適用なし TC: $\pm 1.5 \text{ }^{\circ}\text{C}$
ケーブル長(メートル)		センサまで最大 100 m
ワイヤ抵抗		<ul style="list-style-type: none"> RTD: 20 Ω、10 Ω RTD の場合、最大 2.7 Ω TC: 最大 100 Ω

A.8.5 熱電対タイプ

表 A-55 熱電対タイプ(範囲および精度)

タイプ	アンダー範囲 最小値 ¹	公称範囲 下限値	公称範囲 上限値	オーバー 範囲最大値 ²	公称範囲 ^{3,4} 精度 @ 25 °C	公称範囲 ^{3,4} 精度 20 °C~60 °C
J	-210.0 °C	-150.0 °C	1200.0 °C	1450.0 °C	±0.3 °C	±0.6 °C
K	-270.0 °C	-200.0 °C	1372.0 °C	1622.0 °C	±0.4 °C	±1.0 °C
T	-270.0 °C	-200.0 °C	400.0 °C	540.0 °C	±0.5 °C	±1.0 °C
E	-270.0 °C	-200.0 °C	1000.0 °C	1200.0 °C	±0.3 °C	±0.6 °C
R & S	-50.0 °C	100.0 °C	1768.0 °C	2019.0 °C	±1.0 °C	±2.5 °C
B	0.0 °C	200.0 °C	800.0 °C	--	±2.0 °C	±2.5 °C
	--	800.0 °C	1820.0 °C	1820.0 °C	±1.0 °C	±2.3 °C
N	-270.0 °C	-200.0 °C	1300.0 °C	1550.0 °C	±1.0 °C	±1.6 °C
C	0.0 °C	100.0 °C	2315.0 °C	2500.0 °C	±0.7 °C	±2.7 °C
TXK / XK(L)	-200.0 °C	-150.0 °C	800.0 °C	1050.0 °C	±0.6 °C	±1.2 °C
電圧	-32512	-27648 -80 mV	27648 80 mV	32511	±0.05%	±0.1%

¹ アンダー範囲最小値より小さい熱電対値は、-32768 として報告されます。

² オーバー範囲最大値より大きい熱電対値は、32767 として報告されます。

³ 内部の冷接点エラーは、すべての範囲について±1.5 °C です。このエラーは、この表のエラーに加えられます。モジュールは、この仕様を満たすために、少なくとも 30 分の準備時間を必要とします。

⁴ 4 チャンネル SM TC のみの場合: 放射無線周波数 970 MHz~990 MHz が存在する場合、精度が劣化する場合があります。

注記

熱電対チャンネル

熱電対シグナルモジュールの各チャンネルは、異なる熱電対タイプ(モジュールの構成中にソフトウェアで選択可能)として設定できます。

A.8.6 熱電対フィルタの選択と更新時間

熱電対を測定する場合は、100 ミリ秒の積分時間を使用することを推奨します。これより短い積分時間を使用すると、温度読み取りエラーの繰り返し性が高まります。

表 A- 56 熱電対フィルタの選択と更新時間

遮断周波数(Hz)	積分時間(ミリ秒)	更新時間(秒)		
		1 チャンネル SB	4 チャンネル SM	8 チャンネル SM
10	100	0.301	1.225	2.450
50	20	0.061	0.263	0.525
60	16.67	0.051	0.223	0.445
400 ¹	10	0.031	0.143	0.285

¹ 400 Hz の遮断が選択されたときに、モジュールの解像度と精度を維持するには、積分時間を 10 ミリ秒に設定する必要があります。この遮断を選択すると、100 Hz および 200 Hz のノイズも遮断されます。

A.8.7 RTD センサタイプ選択表

表 A- 57 RTD モジュールによってサポートされているさまざまなセンサの範囲と精度

温度計数	RTD タイプ	アンダー 範囲最小値 ¹	公称範囲 下限値	公称範囲 上限値	オーバー 範囲最大値 ²	公称範囲 精度 @ 25 °C	公称範囲 精度 20 °C ~60 °C
Pt 0.003850 ITS90 DIN EN 60751	Pt 100 climatic	-145.00 °C	-120.00 °C	145.00 °C	155.00 °C	±0.20 °C	±0.40 °C
	Pt 10	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±1.0 °C	±2.0 °C
	Pt 50	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±0.5 °C	±1.0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	± 0.5 °C	±1.0 °C
	Pt 200	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	± 0.5 °C	±1.0 °C
	Pt 500						
	Pt 1000						
Pt 0.003910	Pt 10	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±1.0 °C	±2.0 °C
	Pt 50	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±0.8 °C	±1.6 °C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						

A.8 RTD および熱電対モジュール

温度計数	RTD タイプ	アンダー 範囲最小値 ¹	公称範囲 下限値	公称範囲 上限値	オーバー 範囲最大値 ²	公称範囲 精度 @ 25 °C	公称範囲 精度 20 °C ~60 °C
LG-Ni 0.005000	LG-Ni 1000	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
Ni 0.006170	Ni 100	-105.0 °C	-60.0 °C	180.0 °C	212.4 °C	±0.5 °C	±1.0 °C
Cu 0.004270	Cu 10	-240.0 °C	-200.0 °C	260.0 °C	312.0 °C	±1.0 °C	±2.0 °C
Cu 0.004260	Cu 10	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±0.6 °C	±1.2 °C
	Cu 100						
Cu 0.004280	Cu 10	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±0.7 °C	±1.4 °C
	Cu 100						

¹ アンダー範囲最小値より小さい RTD 値は、-32768 として報告されます。

² オーバー範囲最大値より大きい RTD 値は、+32767 として報告されます。

表 A-58 抵抗

範囲	アンダー範囲 最小値	公称範囲 下限値	公称範囲 上限値	オーバー範囲 最大値 ¹	公称範囲精度 @ 25 °C	公称範囲精度 20 °C ~60 °C
150 Ω	N/A	0 (0 Ω)	27648 (150 Ω)	176.383 Ω	±0.05%	±0.1%
300 Ω	N/A	0 (0 Ω)	27648 (300 Ω)	352.767 Ω	±0.05%	±0.1%
600 Ω	N/A	0 (0 Ω)	27648 (600 Ω)	705.534 Ω	±0.05%	±0.1%

¹ オーバー範囲最大値より大きい抵抗値は、32767 として報告されます。

A.8.8 RTD フィルタの選択と更新時間

表 A- 59 フィルタの選択と更新時間

ノイズ遮断周波数 (Hz)	積分時間 (ミリ秒)	更新時間(秒)		
		1 チャンネル SB	4 チャンネル SM	8 チャンネル SM
10	100	4/2 線式: 0.301 3 線式: 0.601	4/2 線式: 1.222 3 線式: 2.445	4/2 線式: 2.445 3 線式: 4.845
50	20	4/2 線式: 0.061 3 線式: 0.121	4/2 線式: 0.262 3 線式: .505	4/2 線式: 0.525 3 線式: 1.015
60	16.67	4/2 線式: 0.051 3 線式: 0.101	4/2 線式: 0.222 3 線式: 0.424	4/2 線式: 0.445 3 線式: 0.845
400 ¹	10	4/2 線式: 0.031 3 線式: 0.061	4/2 線式: 0.142 3 線式: 0.264	4/2 線式: 0.285 3 線式: 0.525

¹ 400 Hz のフィルタが選択されたときに、モジュールの解像度と精度を維持するには、積分時間を 10 ミリ秒に設定する必要があります。この遮断を選択すると、100 Hz および 200 Hz のノイズも遮断されます。

注記

モジュールは、センサが接続されていない有効化されたチャンネルが存在する場合、32767 を報告します。開いた線の検出も有効化されている場合、そのモジュールは対応する赤の LED を点滅します。

4 線式接続が使用されている場合に、10 Ω RTD 範囲に関して、最も高い精度が実現されます。

2 線式モードでの接続線の抵抗はセンサの読み取りでエラーを生じさせるため、精度は保証されません。

A.9 通信インターフェース

S7-1200 で使用できるモジュールのより完全なリストは、『S7-1200 プログラマブルコントローラシステムマニュアル』またはカスタマサポート Web サイト (<http://www.siemens.com/tiaportal>) を参照してください。

A.9.1 PROFIBUS マスタ/スレーブ

A.9.1.1 CM 1242-5 PROFIBUS DP スレーブ

表 A-60 CM 1242 5 の技術仕様

技術仕様	
製品番号	6GK7 242 5DX30 0XE0
インターフェース	
PROFIBUS への接続	9 ピン D-sub メスコネクタ
ネットワークコンポーネント(光ネットワークコンポーネントなど)を接続したときの PROFIBUS インターフェースの最大消費電流	15 mA / 5 V (バス端子の場合のみ) [*]
許可されている周囲条件	
周囲温度	
<ul style="list-style-type: none"> 保管時 輸送時 縦置きでの設置での動作時(DIN レールは横置きです) 横置きでの設置での動作時(DIN レールは縦置きです) 	<ul style="list-style-type: none"> -40° C ~ 70° C -40° C ~ 70° C 0° C ~ 55° C 0° C ~ 45° C
動作時の 25 °C での最大相対湿度(結露なし)	95 %
保護等級	IP20
電源、電流消費、および電力損失	
電源のタイプ	DC
バックプレーンバスからの電源	5 V
消費電流(通常)	150 mA
有効電力損失(通常)	0.75 W
電氣的絶縁	710 V DC/ 1 分間
<ul style="list-style-type: none"> PROFIBUS インターフェースからアース PROFIBUS インターフェースから内部回路 	
外形寸法と重量	
<ul style="list-style-type: none"> Width Height Depth 	<ul style="list-style-type: none"> 30 mm 100 mm 75 mm

技術仕様	
重量	
<ul style="list-style-type: none"> • ネット重量 • 梱包材を含めた重量 	<ul style="list-style-type: none"> • 115 g • 152 g

*) VP (ピン 6)と DGND (ピン 5)の間に接続された外部コンシューマの電流負荷が、バス端子用の最大 15 mA (短絡耐性)を超えてはいけません。

A.9.1.2 CM 1242-5 の D-sub ソケットのピンアウト

PROFIBUS インターフェース

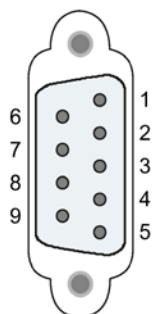


表 A- 61 D-sub ソケットのピンアウト

ピン	説明	ピン	説明
1	- 未使用 -	6	P5V2: +5V 電源
2	- 未使用 -	7	- 未使用 -
3	RxD/TxD-P: データライン B	8	RxD/TxD-N: データライン A
4	RTS	9	- 未使用 -
5	M5V2: データ基準電位(アース DGND)	ハウジング	アースコネクタ

A.9.1.3 CM 1243-5 PROFIBUS DP マスタ

表 A- 62 CM 1243 5 の技術仕様

技術仕様	
製品番号	6GK7 243 5DX30 0XE0
インターフェース	
PROFIBUS への接続	9 ピン D-sub メスコネクタ
ネットワークコンポーネント(光ネットワークコンポーネントなど)を接続したときの PROFIBUS インターフェースの最大消費電流	15 mA / 5 V (バス端子の場合のみ) *)
許可されている周囲条件	
周囲温度	
<ul style="list-style-type: none"> 保管時 輸送時 縦置きでの動作時(DIN レールは横置きです) 横置きでの動作時(DIN レールは縦置きです) 	<ul style="list-style-type: none"> -40° C～70° C -40° C～70° C 0° C～55° C 0° C～45° C
動作時の 25 °C での最大相対湿度(結露なし)	95 %
保護等級	IP20
電源、電流消費、および電力損失	
電源のタイプ	DC
電源/外部	24 V
<ul style="list-style-type: none"> 最小 最大 	<ul style="list-style-type: none"> 19.2 V 28.8 V
消費電流(通常)	
<ul style="list-style-type: none"> 24 V DC から S7-1200 バックプレーンバスから 	<ul style="list-style-type: none"> 100 mA 0 mA
有効電力損失(通常)	
<ul style="list-style-type: none"> 24 V DC から S7-1200 バックプレーンバスから 	<ul style="list-style-type: none"> 2.4 W 0 W
電源 24 V DC / 外部	
<ul style="list-style-type: none"> 最小ケーブル断面積 最大ケーブル断面積 ネジ端子の締め付けトルク 	<ul style="list-style-type: none"> 最小:0.14 mm² (AWG 25) 最大:1.5 mm² (AWG 15) 0.45 Nm (4 lb-in)
電氣的絶縁	710 V DC/ 1 分間
<ul style="list-style-type: none"> PROFIBUS インターフェースからアース PROFIBUS インターフェースから内部回路 	

技術仕様	
外形寸法と重量	
<ul style="list-style-type: none">• Width• Height• Depth	<ul style="list-style-type: none">• 30 mm• 100 mm• 75 mm
重量	
<ul style="list-style-type: none">• ネット重量• 梱包材を含めた重量	<ul style="list-style-type: none">• 134 g• 171 g

*) VP (ピン 6)と DGND (ピン 5)の間に接続された外部コンシューマの電流負荷が、バス端子用の最大 15 mA (短絡耐性)を超えてはいけません。

注記

CM 1243- (PROFIBUS マスタモジュール)は、CPU の 24 V DC センサ電源から電力を受ける必要があります。

A.9.1.4 PROFIBUS マスタ(CM 1243-5)には CPU からの 24 V DC 電源が必要

注記

CM 1243-5 (PROFIBUS マスタモジュール)は、CPU の 24 V DC センサ電源から電力を受ける必要があります。

A.9.1.5 CM 1243-5 の D-sub ソケットのピンアウト

PROFIBUS インターフェース

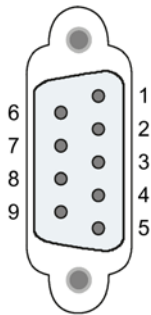


表 A- 63 D-sub ソケットのピンアウト

ピン	説明	ピン	説明
1	- 未使用 -	6	VP: バス端子抵抗専用の電源+5 V です; 外部デバイス用ではありません
2	- 未使用 -	7	- 未使用 -
3	RxD/TxD-P: データライン B	8	RxD/TxD-N: データライン A
4	CNTR P: RTS	9	- 未使用 -
5	DGND: データ信号および VP 用のアース	ハウジング	アースコネクタ

PROFIBUS ケーブル

注記

PROFIBUS ケーブルのシールドへの接触

PROFIBUS ケーブルのシールドに接触する必要があります。

これを行うには、PROFIBUS ケーブルの端の絶縁部分をはぎ取り、シールドを機能的アースに接続する必要があります。

A.9.2 GPRS CP

注記

CP 1242-7 は、海事用途用には承認されていません

CP 1242-7 は、海事承認を所有しません。

注記

これらのモジュールを使用するには、CPU のファームウェアが V2.0 以降であることが必要です。

A.9.2.1 CP 1242-7 GPRS

表 A- 64 CP 1242-7 GPRS V2 の技術仕様

技術仕様	
製品番号	6GK7 242 7KX3 0XE0
無線インターフェース	
アンテナコネクタ	SMA ソケット
公称インピーダンス	50Ω
無線接続	
最大送信能力	<ul style="list-style-type: none"> • GSM 850、クラス 4: +33 dBm ±2dBm • GSM 900、クラス 4: +33 dBm ±2dBm • GSM 1800、クラス 1: +30 dBm ±2dBm • GSM 1900、クラス 1: +30 dBm ±2dBm
GPRS	マルチスロットクラス 10 デバイスクラス B コード化スキーム 1...4 (GMSK)
SMS	発信モード: MO サービス: ポイントツーポイント
許可されている周囲条件	
周囲温度 <ul style="list-style-type: none"> • 保管時 • 輸送時 • 縦置きでの動作時(DIN レールは横置きです) • 横置きでの動作時(DIN レールは縦置きです) 	<ul style="list-style-type: none"> • -40° C～70° C • -40° C～70° C • 0 °C～55 °C • 0 °C～45 °C
動作時の 25 °C での最大相対湿度(結露なし)	95 %
保護等級	IP20
電源、電流消費、および電力損失	
電源のタイプ	DC

技術仕様	
電源/外部	24 V
<ul style="list-style-type: none"> 最小 最大 	<ul style="list-style-type: none"> 19.2 V 28.8 V
消費電流(通常)	
<ul style="list-style-type: none"> 24 V DC から S7-1200 バックプレーンバスから 	<ul style="list-style-type: none"> 100 mA 0 mA
有効電力損失(通常)	
<ul style="list-style-type: none"> 24 V DC から S7-1200 バックプレーンバスから 	<ul style="list-style-type: none"> 2.4 W 0 W
24 V DC 電源	
<ul style="list-style-type: none"> 最小ケーブル断面積 最大ケーブル断面積 ネジ端子の締め付けトルク 	<ul style="list-style-type: none"> 最小:0.14 mm² (AWG 25) 最大:1.5 mm² (AWG 15) 0.45 Nm (4 lb-in)
電氣的絶縁	710 V DC/ 1 分間
電源ユニットから内部回路	
外形寸法と重量	
<ul style="list-style-type: none"> Width Height Depth 	<ul style="list-style-type: none"> 30 mm 100 mm 75 mm
重量	
<ul style="list-style-type: none"> ネット重量 梱包材を含めた重量 	<ul style="list-style-type: none"> 133 g 170 g

A.9.2.2 GSM/GPRS アンテナ ANT794-4MR

ANT794-4MR GSM/GPRS アンテナの技術仕様

ANT794-4MR	
製品番号	6NH9860 1AA00
モバイル無線ネットワーク	GSM/GPRS
周波数範囲	<ul style="list-style-type: none"> 824～960 MHz (GSM 850、900) 1710～1880 MHz (GSM 1800) 1900～2200 MHz (GSM / UMTS)
特性	全方向性
アンテナゲイン	0 dB
インピーダンス	50 Ω
定在波比(SWR)	< 2.0
最大電力	20 W
極性	リニア垂直
コネクタ	SMA
アンテナケーブルの長さ	5 m
外部材料	ハード PVC、紫外線抵抗性
保護等級	IP20
許可されている周囲条件 <ul style="list-style-type: none"> 動作温度 輸送/保管温度 相対湿度 	<ul style="list-style-type: none"> -40 °C～+70 °C -40 °C～+70 °C 100 %
外部材料	ハード PVC、紫外線抵抗性
構造	5 m の固定ケーブルと SMA オスコネクタを備えたアンテナ
外形寸法(D x H) (mm)	25x193
重量 <ul style="list-style-type: none"> アンテナ(ケーブルを含めて) 付属品 	<ul style="list-style-type: none"> 310 g 54 g
インストール	付属のブラケットを使用

A.9.2.3 平面アンテナ ANT794-3M

平面アンテナ ANT794-3M の技術仕様

ANT794-3M		
製品番号	6NH9870 1AA00	
モバイル無線ネットワーク	GSM 900	GSM 1800/1900
周波数範囲	890～960 MHz	1710～1990 MHz
定在波比(SWR)	≤ 2:1	≤ 1.5:1
リターンロス(Tx)	≈ 10 dB	≈ 14 dB
アンテナゲイン	0 dB	
インピーダンス	50 Ω	
最大電力	10 W	
アンテナケーブル	SMA オスコネクタ付きの HF ケーブル RG 174 (固定長)	
ケーブル長	1.2 m	
保護等級	IP64	
許可された温度範囲	-40° C～+75° C	
可燃性	UL 94 V2	
外部材料	ABS PolyIac PA-765、ライトグレー(RAL 7035)	
外形寸法 (W x L x H) (mm)	70.5 x 146.5 x 20.5	
重量	130 g	

A.9.3 Teleservice (TS)

以下のマニュアルには、TS Adapter IE Basic およびモジュール型 TS Adapter の技術仕様が記載されています。

- 産業用ソフトウェアエンジニアリングツール
モジュール型 TS Adapter
- 産業用ソフトウェアエンジニアリングツール
TS Adapter IE Basic

この製品および製品マニュアルに関する詳細は、TS Adapter 用の製品カタログ Web サイト (<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=TS%20Adapter%20IE%20basic&tab=>) を参照してください。

A.9.4 RS485、RS232、RS422 通信

A.9.4.1 CB 1241 RS485 仕様

注記

この CB を使用するには、CPU のファームウェアが V2.0 以降であることが必要です。

表 A- 65 一般仕様

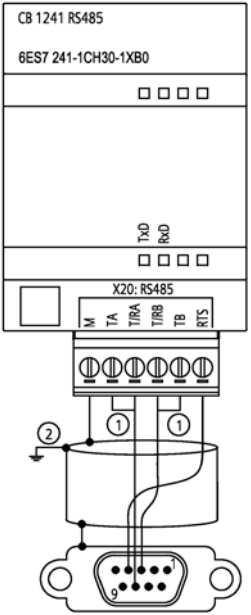
技術データ	CB 1241 RS485
製品番号	6ES7 241-1CH30-1XB0
外形寸法 W x H x D (mm)	38 x 62 x 21
重量	40 グラム

表 A- 66 トランスミッタおよびレシーバ

技術データ	CB 1241 RS485
タイプ	RS485 (2 線式半二重)
コモンモード電圧範囲	-7 V ~ +12 V、1 秒、3 VRMS 常時
送信側差動出力電圧	最小 2 V / $R_L = 100 \Omega$ 最小 1.5 V / $R_L = 54 \Omega$
終端およびバイアス	10K から B 上の +5 V へ、RS485 ピン 3 10K から A 上の GND へ、RS485 ピン 4
オプション終端	ピン TB をピン T/RB へ短絡、有効終端インピーダンスは 127 Ω 、RS485 ピン 3 へ接続 ピン TA をピン T/RA へ短絡、有効終端インピーダンスは 127 Ω 、RS485 ピン 4 へ接続
受信側入力インピーダンス	終端を含む 5.4K Ω 分
受信側閾値/感度	最小 +/- 0.2 V、60 mV ティピカルヒステリシス
絶縁 RS485 信号をシャーシアースへ RS485 信号を CPU ロジックコモンへ	500 V AC、1 分
ケーブル長、シールド済み	最長 1000 m
ボーレート	300 ボー、600 ボー、1.2 kbit、2.4 kbit、4.8 kbit、9.6 kbit (既定値)、9.2 kbit、38.4 kbit、57.6 kbit、76.8 kbit、115.2 kbit
パリティ	パリティなし(既定値)、偶数、奇数、Mark (パリティビットが常に 1 にセット)、空白(パリティビットが常に 0 にセット)
ストップビット数	1 (既定値)、2
フロー制御	サポートされていません
待機時間	0 ~ 65535 ミリ秒

表 A- 67 電源

技術データ	CB 1241 RS485
電力損失(損失)	1.5 W
最大消費電流(SM バス)	50 mA
最大消費電流(24 V DC)	80 mA

CB 1241 RS485 (6ES7 241-1CH30-1XB0)	
	
<p>① 図に示すように、「TA」と「TB」を接続してネットワークを終了します。(RS485 ネットワークの終端デバイスだけを終了します。)</p> <p>② シールド付きツイストペアケーブルを使用し、ケーブルシールドをアースに接続します。</p>	

RS485 ネットワークの 2 つの終端だけを終了します。2 つの終端デバイス間のデバイスは、終了されたり、バイアスされることはありません。トピック「RS485 ネットワークコネクタのバイアス処理と終了」を参照してください。

表 A- 68 CB 1241 RS485 (6ES7 241-1CH30-1XB0)のコネクタピンの位置

ピン	9 ピンコネクタ	X20
1	RS485 / ロジック GND	--
2	RS485 / 未使用	--
3	RS485 / TxD+	3 - T/RB
4	RS485 / RTS	1 - RTS
5	RS485 / ロジック GND	--
6	RS485 / 5 V 電力	--
7	RS485 / 未使用	--
8	RS485 / TxD-	4 - T/RA

ピン	9 ピンコネクタ	X20
9	RS485 / 未使用	--
SHELL		7 - M

A.9.4.2 CM 1241 RS422/485 仕様

CM 1241 RS422/485 の仕様

表 A- 69 一般仕様

技術データ	CM 1241 RS422/485
製品番号	6ES7 241-1CH32-0XB0
外形寸法 W x H x D (mm)	30 x 100 x 75
重量	155 グラム

表 A- 70 トランスミッタおよびレシーバ

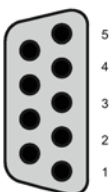
技術データ	CM 1241 RS422/485
タイプ	RS422 または RS485、9 ピン D-sub メスコネクタ
コモンモード電圧範囲	-7 V ~ +12 V、1 秒、3 VRMS 常時
送信側差動出力電圧	最小 2 V / $R_L = 100 \Omega$ 最小 1.5 V / $R_L = 54 \Omega$
終端およびバイアス	10K Ω から B 上の +5 V へ、PROFIBUS ピン 3 10K Ω から A 上の GND へ、PROFIBUS ピン 8 内部バイアスオプションが提供されるか、内部バイアスが存在しない場合もあります。すべての場合に外部終端が必要です。『S7-1200 プログラマブルコントローラシステムマニュアル』の「RS485 ネットワークコネクタのバイアス処理と終了」および「RS422 および RS485 の構成」を参照してください。
受信側入力インピーダンス	終端を含む 5.4K Ω 分
受信側閾値/感度	最小 +/- 0.2 V、60 mV ティピカルヒステリシス
絶縁 RS485 信号をシャーシアースへ RS485 信号を CPU ロジックコモンへ	500 V AC、1 分
ケーブル長、シールド済み	最長 1000 m (ボーレートによって異なります)
ボーレート	300 ボー、600 ボー、1.2 kbit、2.4 kbit、4.8 kbit、9.6 kbit (既定値)、9.2 kbit、38.4 kbit、57.6 kbit、76.8 kbit、115.2 kbit
パリティ	パリティなし(既定値)、偶数、奇数、Mark (パリティビットが常に 1 にセット)、空白(パリティビットが常に 0 にセット)
ストップビット数	1 (既定値)、2
フロー制御	RS422 モードの場合に、XON/XOFF がサポートされています
待機時間	0 ~ 65535 ミリ秒

A.9 通信インターフェース

表 A- 71 電源

技術データ	CM 1241 RS422/485
電力損失(損失)	1.1 W
+5 V DC から	220 mA

表 A- 72 RS485 または RS422 コネクタ(メスコネクタ)

ピン	説明	コネクタ (メス)	ピン	説明
1	ロジックまたは通信接地		6 PWR	+5 V、100 オーム直列抵抗付き:出力
2 TxD+ ¹	RS422 の場合に接続 RS485 では未使用: 出力		7	接続されていません
3 TxD+	信号 B (RxD/TxD+): 入力/出力		8 TXD-	信号 A (RxD/TxD-): 入力/出力
4 RTS ²	送信要求(TTL レベル): 出力		9 TXD- ¹	RS422 の場合に接続 RS485 では未使用: 出力
5 GND	ロジックまたは通信接地		SHELL	筐体接地

¹ ピン 2 および 9 は、RS422 用の信号送信のためにのみ使用されます。

² RTS は 1 つの TTL レベル信号で、この信号に基づいて、別の半二重デバイスを制御するために使用できます。それは送信するときだけ有効になり、その他の場合は無効です。

A.9.4.3 CM 1241 RS232 仕様

表 A- 73 一般仕様

技術データ	CM 1241 RS232
製品番号	6ES7 241-1AH32-0XB0
外形寸法(mm)	30 x 100 x 75
重量	150 グラム

表 A-74 トランスミッタおよびレシーバ

技術データ	CM 1241 RS232
タイプ	RS232 (全二重)
送信側出力電圧	最小+/- 5 V / $R_L = 3K \Omega$
送信側出力電圧	+/- 15 V DC (最大)
受信側入力インピーダンス	最小 3 K Ω
受信側閾値/感度	最小 0.8 V、最大 2.4 V 0.5 V (通常ヒステリシス)
受信側入力電圧	+/- 30 V DC (最大)
絶縁 RS 232 信号をシャーシアースへ RS 232 信号を CPU ロジックコモンへ	500 V AC、1 分
ケーブル長、シールド済み	最長 10 m
ボーレート	300 ボー、600 ボー、1.2 kbit、2.4 kbit、4.8 kbit、9.6 kbit (既定値)、 9.2 kbit、38.4 kbit、57.6 kbit、76.8 kbit、115.2 kbit
パリティ	パリティなし(既定値)、偶数、奇数、Mark (パリティビットが常に 1 にセット)、空白(パリティビットが常に 0 にセット)
ストップビット数	1 (既定値)、2
フロー制御	ハードウェア、ソフトウェア
待機時間	0~65535 ミリ秒

表 A-75 電源

技術データ	CM 1241 RS232
電力損失(損失)	1 W
+5 V DC から	200 mA

表 A-76 RS232 コネクタ(オスコネクタ)

ピン	説明	コネクタ (オスコネクタ)	ピン	説明
1 DCD	データキャリア検出: Input		6 DSR	データセットレディ: Input
2 RxD	DCE からの受信データ: Input		7 RTS	送信要求: 出力
3 TxD	DCE に送信されるデータ: 出力		8 CTS	送信可: Input
4 DTR	データ端末レディ: 出力		9 RI	リングインジケータ(使用しません)
5 GND	ロジックグラウンド		SHELL	筐体接地

A.10 テクノロジーモジュール

A.10.1 SM 1278 4xIO-Link マスタ SM

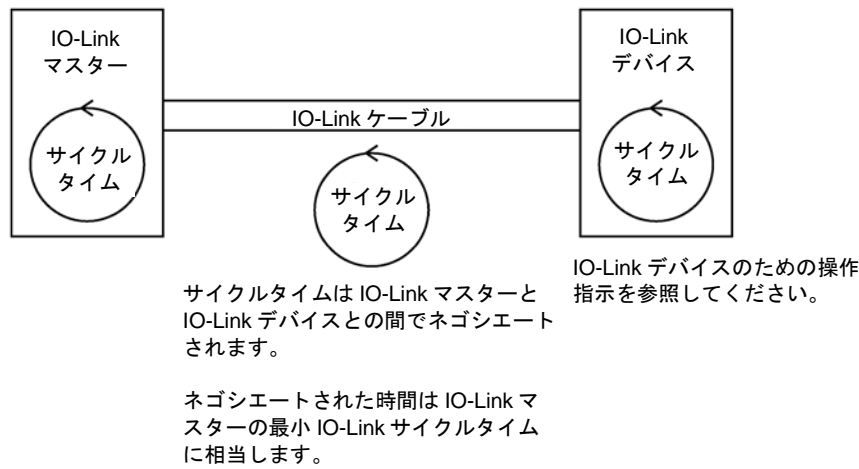
A.10.1.1 SM 1278 4xIO-Link マスタシグナルモジュール仕様

表 A- 77 一般仕様

技術データ		SM 1278 4xIO-Link マスタシグナルモジュール
製品番号		6ES7 278-4BD32-0XB0
外形寸法 W x H x D (mm)		45 x 100 x 75
重量		150 グラム
一般情報		
	I&M データ	あり; IM0~IM3
電源電圧		
	定格電圧(DC)	24 V DC
	有効な範囲の下限値(DC)	19.2 V; IO-Link を使用する場合、20.5 V (マスタ上の IO-Link デバイス用の電源電圧は少なくとも 20 V であることが必要です)
	有効な範囲の上限値(DC)	28.8 V DC
	極性反転保護	あり
入力電流		
	消費電流	65 mA; 負荷なしで
エンコーダ電源		
	出力数	4
	出力電流、定格値	200 mA
電力損失		
	通常の電力損失	1 W、ポート負荷を除いて
デジタル入出力		
	ケーブル長(メートル)	最大 20 m、シールドなし
SDLC		
	ケーブル長(メートル)	最大 20 m、シールドなし
IO リンク		
	ポートの数	4
	同時に制御できるポートの数	4
	IO-Link プロトコル 1.0	あり
	IO-Link プロトコル 1.1	あり
動作モード		
	IO リンク	あり
	DI	あり
	DQ	あり; 最大 100 mA
IO-Link デバイスの接続		
	ポートタイプ A	あり

技術データ		SM 1278 4xIO-Link マスタシグナルモジュール
	伝送速度	4.8 kBd (COM1)
		38.4 kBd (COM2)
		230.4 kBd (COM3)
	最小サイクルタイム	2 ミリ秒、動的、ユーザーデータ長に依存
	プロセスデータのサイズ、ポートあたり入力	32 バイト; 最大
	プロセスデータのサイズ、モジュールあたり入力	32 バイト
	プロセスデータのサイズ、ポートあたり出力	32 バイト; 最大
	プロセスデータのサイズ、モジュールあたり出力	32 バイト
	デバイスパラメータ用のメモリサイズ	2 KB
最大シールドなしケーブル長(m)		20 m
割り込み/診断/ステータス情報		
	ステータス表示	あり
割り込み		
	診断割り込み	あり; ポート診断は IO-Link モードでのみ使用できます
診断アラーム		
	診断	
	電源電圧のモニタ	あり
	短絡	あり
診断インジケータ LED		
	電源電圧のモニタ	あり; DIAG LED が赤で点滅
	チャンネルステータス表示	あり; チャンネルごとに、チャンネルステータス Qn (SIO モード)およびポートステータス Cn (IO-Link モード)用の 1 つの緑色の LED があります
	チャンネル診断用	あり; 赤の Fn LED
	モジュール診断用	あり; 緑色/赤の DIAG LED
電氣的絶縁		
	チャンネルの電氣的絶縁	
	チャンネル間	なし
	チャンネルとバックプレーンバス間	あり
許可されている電位差		
	異なる回路間	75 V DC/ 60 V AC(基本的絶縁)
絶縁		
	絶縁テスト用	707 V DC (タイプ: テスト)
周囲条件		
	動作温度	
	最小	-20 °C
	最大	60 °C
	横置き設置、最小値	-20 °C
	横置き設置、最大値	60 °C
	縦置き設置、最小値	-20 °C
	縦置き設置、最大値	50 °C

応答時間の概要



A.10.1.2 SM 1278 4xIO-Link マスタ SM 配線図

表 A- 78 SM 1278 IO-Link マスタの配線図

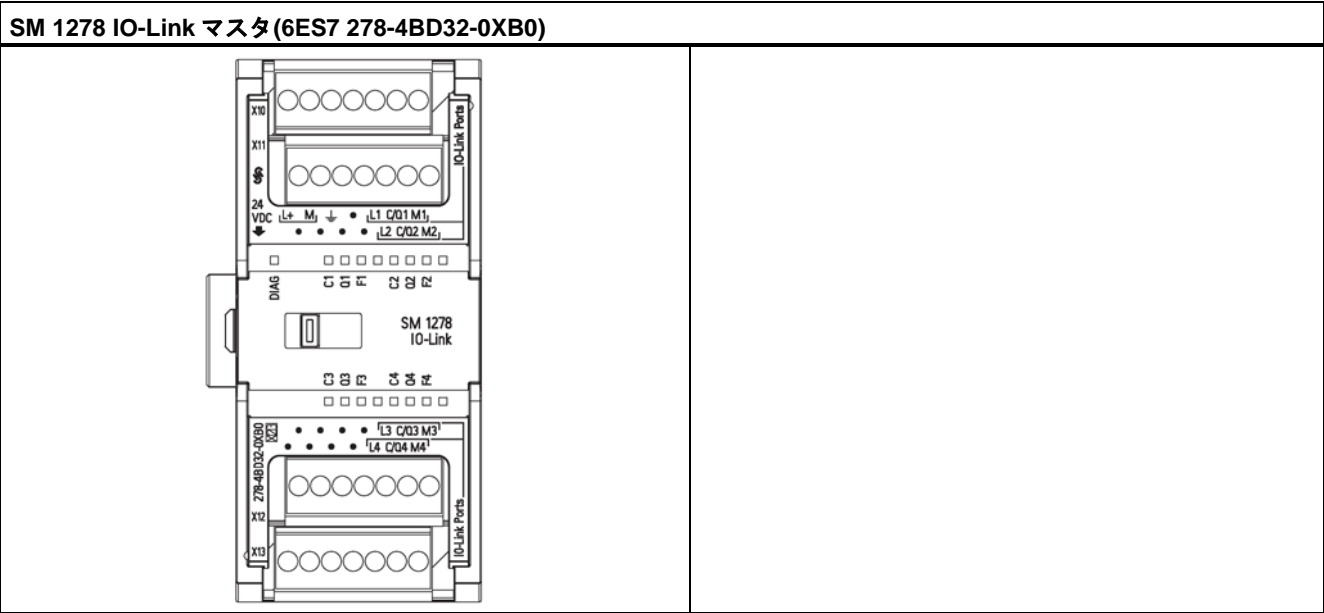


表 A- 79 SM 1278 IO-Link マスタ (6ES7 278-4BD32-0XB0) のコネクタ ピンの位置

ピン	X10	X11	X12	X13
1	L+ / 24 V DC	接続なし	接続なし	接続なし
2	M / 24 V DC	接続なし	接続なし	接続なし
3	機能的アース	接続なし	接続なし	接続なし

ピン	X10	X11	X12	X13
4	接続なし	接続なし	接続なし	接続なし
5	L ₁	L ₂	L ₃	L ₄
6	C/Q ₁	C/QL ₂	C/Q ₃	C/QL ₄
7	ML ₁	ML ₂	M ₃	ML ₄

A.11 コンパニオン製品

A.11.1 PM 1207 電源モジュール

PM 1207 は、SIMATIC S7-1200 用の電源モジュールです。このモジュールには、以下の特性があります。

- 入力 120/230 V AC、出力 24 V DC/ 2.5A
- 製品番号 6ESP 332-1SH71-4AA0

この製品と製品マニュアルに関する詳細は、PM 1207 用の製品カタログ Web サイト (<https://eb.automation.siemens.com/mall/en/de/Catalog/Product/6AG1332-1SH71-4AA0>) を参照してください。

A.11.2 CSM 1277 コンパクトスイッチモジュール

CSM1277 は、産業用イーサネットコンパクトスイッチモジュールです。このモジュールを使用して、S7-1200 のイーサネットインタフェースを拡張し、オペレータパネル、プログラミングデバイス、または他のコントローラとの同時通信を可能にすることができます。このモジュールには、以下の特性があります。

- 4 x 産業用イーサネットに接続するための RJ45 ソケット
- 3 外部 24 V DC 電源を一番上に接続するための端子ストリップ内の差し込みプラグ
- 産業用イーサネットポートの診断およびステータス表示用の複数の LED
- 製品番号 6GK7 277-1AA00-0AA0

この製品と製品マニュアルに関する詳細は、CSM 1277 用の製品カタログ Web サイト (<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277&tab=>) を参照してください。

A.11.3 CM CANopen モジュール

CM CANopen モジュールは、SIMATIC S7-1200 PLC と、CANopen を実行する任意のデバイス間のプラグインモジュールです。CM CANopen は、マスタまたはスレーブとして構成できます。2つの CM CANopen モジュールが存在します: CANopen モジュール(製品番号 021620-B)と、CANopen (耐久性が高い)モジュール(製品番号 021730-B)。

CANopen モジュールは、以下の機能を提供します。

- CPU ごとに 3 つのモジュールが接続可能
- 最大 16 の CANopen スレーブノードを接続
- 1 つのモジュールに対して、256 バイトの入力および 256 バイトの出力
- 3 つの LED により、モジュール、ネットワーク、および I/O ステータスに関する診断情報を提供
- CANopen ネットワーク構成の PLC への保管をサポート
- このモジュールは TIA ポータルのコンフィギュレーションスイートのハードウェアカタログに統合可能
- 統合済みの CANopen Configuration Studio または他の任意の外部 CANopen 構成ツールによる CANopen の構成
- CANopen 通信プロファイル CiA 301 rev. 4.2 および CiA 302 rev. 4.1 への適合
- カスタムプロトコル処理用の透過的な CAN 2.0A をサポート
- TIA ポータルでの各 PLC プログラミングで利用できる既製のファンクションブロック
- CM CANopen モジュールには以下が含まれます; サブネットワーク用のネジ端子付きの DSUB、CM CANopen Configuration Studio CD、および USB 構成ケーブル。

この製品と製品マニュアルに関する詳細は、CM CANopen 用の製品カタログ Web サイトを参照してください。

V3.0 CPU を V4.1 CPU に交換

B.1 V3.0 CPU を V4.1 CPU に交換

ユーザーの V3.0 CPU を V4.1 CPU に交換し (Page 79)、V3.0 CPU 向けにユーザーが設計した既存の STEP 7 プロジェクトを使用できます。ファームウェア更新によって V3.0 CPU を V4.1 CPU へアップグレードすることはできません。ハードウェアを交換する必要があります。V3.0 CPU を V4.1 CPU に交換するとき、さらに、ファームウェア更新 (Page 350) の有無のチェックし、存在する場合は、接続されたシグナルおよび通信モジュールへの適用が必要になる場合があります。

注記

V4.1 から V3.0 へのデバイス交換はできません

構成をダウンロードした後、V3.0 CPU を V4.1 CPU に交換することはできますが、V4.1 CPU を V3.0 CPU に交換することはできません。既存の STEP 7 V3.0 プロジェクトを表示するか、他の方法で使用する場合は、デバイス交換の前に、ユーザーの STEP 7 V3.0 プロジェクトのアーカイブを作成してください。

交換されたデバイス構成をまだダウンロードしていない場合は、デバイス交換を取り消すことができることに注意してください。ただし、ダウンロードした後は、V3.0 から V4.1 への交換を取り消すことはできません。

2 つの CPU バージョン間の構成上および機能上のいくつかの変更に注意しておく必要があります。

オーガニゼーションブロック

V4.1 では、OB 実行が割り込み可能であるか、割り込み不能であるかを (Page 60) を設定できます。従来の V3.0 CPU のプロジェクトの場合は、STEP 7 が既定ですべての OB を割り込み不能として設定します。

STEP 7 は、すべての OB 優先度 (Page 60) を V3.0 CPU STEP 7 プロジェクトのときの値にセットします。

この後、ユーザーが選択する場合は、割り込み可能性または優先度の設定を変更できます。

診断エラー割り込み OB 開始情報は、診断イベントが保留中でない場合は、全体としてサブモジュールの情報に基づいて決められます。

CPU パスワード保護

STEP 7 は、V4.1 CPU 用のパスワード保護レベル(Page 89)を、V3.0 CPU 用に設定された等価のパスワード保護レベルになるように設定し、V3.0 パスワードを V4.1 CPU 用の「フルアクセス(保護なし)」パスワードに割り当てます。

V3.0 保護レベル	V4.1 アクセスレベル
保護なし	フルアクセス(保護なし)
書き込み保護	読み取りアクセス
書き込み/読み取り保護	HMI アクセス

V4.1 アクセスレベル「アクセスなし(完全保護)」は、V3.0 には存在しなかったことに注意してください。

Web サーバー

ユーザーの V3.0 プロジェクトでユーザー定義 Web ページを使用する場合は、プロジェクトをアップグレードする前に、それらのページをユーザーのプロジェクトインストールフォルダのサブフォルダ[UserFiles\Webserver]に保管します。ユーザー定義ページをこの場所に保管すると、STEP 7 プロジェクトの保存でもユーザー定義 Web ページが保存されます。

V3.0 CPU を V4.1 CPU に交換すると、Web サーバーを有効化するためのユーザーの Web サーバープロジェクト設定および HTTPS 設定が、V3.0 のときの設定と同じになります。この後、Web サーバーを使用するために必要なユーザー、権限、パスワード(Page 255)、および言語を設定できます。追加権限を備えたユーザーを設定しないと、標準 Web ページ(Page 256)から表示できる事項に関してユーザーは制限を受けます。S7-1200 V4.1 CPU は、従来の事前設定された「admin」ユーザーおよびパスワードをサポートしていません。

S7-1200 V3.0 の Web サーバーデータログページでは、「ダウンロードおよびクリア」操作が提供されました。V4.1 の Web サーバーファイル参照ページ(Page 256) (このページからデータログにアクセスします)は、もう、この機能を提供していません。その代わりに、Web サーバーが、データログファイルのダウンロード、名前変更、および削除を行う機能を提供します。

転送カードの非互換性

V3.0 転送カード(Page 63)を使用して、V3.0 プログラムを V4.1 CPU に転送することはできません。V3.0 プロジェクトを STEP 7 で開き、デバイスを V4.1 CPU に変更した(Page 79)後に、STEP 7 プロジェクトを V4.1 CPU にダウンロードする必要があります。ユーザーのプロジェクトを V4.1 プロジェクトに変更した後は、それ以降のプログラム転送のために V4.1 転送カードを作成できます。

GET/PUT 通信

V3.0 では、既定で、GET/PUT 通信が有効化されました。V3.0 CPU を V4.1 CPU に交換する(Page 79)と、互換性情報セクションに、GET/PUT が有効化されていることを述べるメッセージが表示されます。

モーションコントロールのサポート

S7-1200 V4.1 CPU は、V1.0 および V2.0 モーションライブラリをサポートしていません。V1.0 または V2.0 モーションライブラリを備えた STEP 7 プロジェクトに対してデバイス交換を実行すると、デバイス交換では、コンパイル時に、V1.0 または V2.0 モーションライブラリの命令の代わりに、互換性のある V3.0 モーションコントロール命令(Page 309)が使用されます。

2つの異なるモーションコントロール命令バージョン(V3.0 および V5.0)を含む STEP 7 プロジェクトに対して V3.0 CPU から V4.1 CPU へのデバイス交換を実行すると、デバイス交換では、コンパイル時に、互換性のある V5.0 モーションコントロール命令(Page 309)が使用されます。

V3.0 CPU から V4.1 CPU へのデバイス交換中は、モーションコントロールテクノロジーオブジェクト(TO)バージョンは V3.0 から V5.0 へ自動的に切り替わりません。最新のバージョンにアップグレードする場合は、命令ツリーに移動して、下の表に示すように、ユーザーのプロジェクトに必要な S7-1200 モーションコントロールバージョンを選択する必要があります。

CPU バージョン	許可されているモーションコントロールバージョン
V4.1 (モーションコントロール V5.0)	V5.0、V4.0、または V3.0
V4.0 (モーションコントロール V4.0)	V4.0 または V3.0
V3.0 (モーションコントロール V3.0)	V3.0

TO の構造は、モーションコントロールバージョン V3.0 と V5.0 では異なります。すべての関連ブロックも同様に変更されています。ブロックインターフェース、ウォッチテーブル、およびトレースが、新しいモーションコントロール V5.0 の構造に合わせて更新されています。V3.0 CPU と V4.1 CPU のモーションコントロール軸パラメータの相違は、以下の 2 つの表を参照してください。

V3.0 CPU (モーションコントロール V3.0)	V4.1 CPU (モーションコントロール V5.0)
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal

V3.0 CPU (モーションコントロール V3.0)	V4.1 CPU (モーションコントロール V5.0)
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

名前が変更された「commandtable」パラメータだけが、コマンドの配列です。

V3.0	V4.1
Config.Command[]	Command[]

注記: 配列「Command[]」は、V3.0 のタイプ「TO_CmdTab_Config_Command」および V4.1 のタイプ「TO_Struct_Command」の UDT です。

命令の変更

以下の命令のパラメータまたは動作が変更されました。

- RDREC および WRREC (Page 148)
- CONV (Page 111)

HMI パネルでの通信

1 つまたは複数の HMI パネル(Page 22)がユーザーの S7-1200 V3.0 CPU に接続されていた場合、S7-1200 V4.1 CPU への通信は、ユーザーが使用する通信タイプと HMI パネルのファームウェアバージョンによって異なります。ユーザーのプロジェクトを再コンパイルして、CPU および HMI にダウンロードするか、HMI ファームウェアを更新してください(この 2 つの操作を両方とも実行しても構いません)。

プログラムブロックを再コンパイルするための必要条件

V3.0 CPU を V4.1 CPU に交換した後、V4.1 CPU にダウンロードする前に、すべてのプログラムブロックを再コンパイルする必要があります。さらに、ブロックのどれかがノウハウプロテクト(Page 91)されているか、コピー保護が PLC シリアル番号に結合されている (Page 92)場合は、ブロックのコンパイルおよびダウンロードを行う前に、その保護を解除する必要があります。(ただし、メモ리카ードに結合されたコピー保護を無効化する必要はありません。) コンパイルが成功したら、ノウハウプロテクトや PLC シリアル番号コピー保護を再設定することができます。ユーザーのプロジェクトに OEM (Original Equipment Manufacturer) が提供したブロックが存在し、それらのブロックの中にノウハウプロテクトされたブロックがある場合は、OEM に連絡をしていただいて、該当のブロックの V4.1 バージョンを提供することが必要になることに御注意ください。

一般的に、Siemens は、デバイス交換後に、STEP 7 でハードウェアコンフィグレーションおよびソフトウェアを再コンパイルして、ユーザーのプロジェクトのすべてのデバイスにダウンロードすることを推奨しています。プロジェクトのコンパイル時に検出されたすべてのエラーを訂正し、エラーがなくなるまで、再コンパイルしてください。この後、プロジェクトを V4.1 CPU へダウンロードできます。

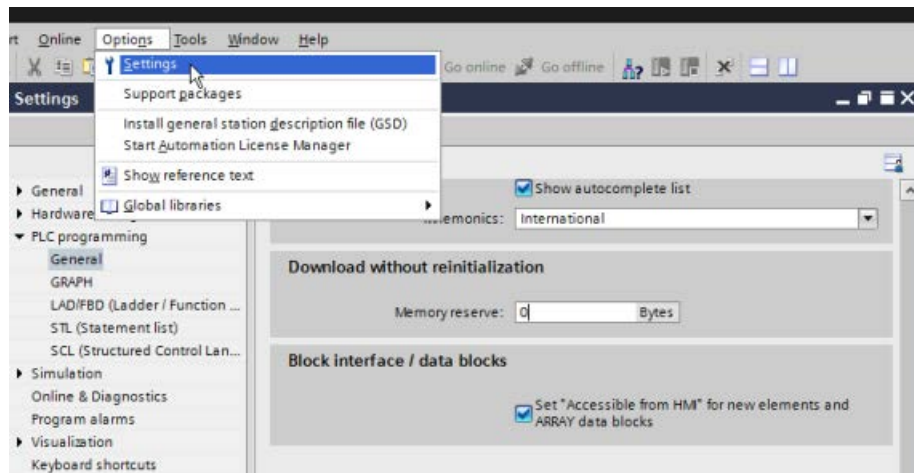
S7-1200 V3.0 プロジェクトが S7-1200 V4.1 CPU に適合しない場合があります

S7-1200 V4.0 では、再初期化せずにダウンロードをサポートするために、それぞれの DB に対して 100 バイトの予約領域が追加されました。

V4.1 CPU への V3.0 プロジェクトのダウンロードを試行する前に、DB の 100 バイトの予約領域を削除できます。

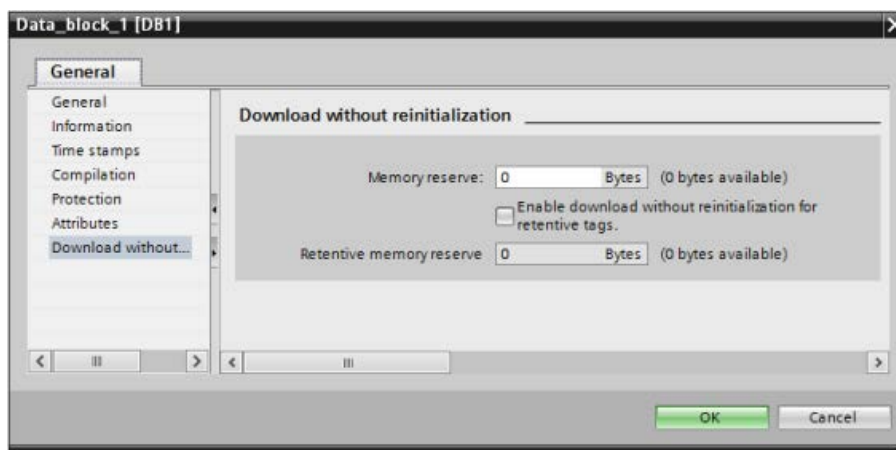
100 バイトの予約領域を削除するには、デバイス交換を実行する前に、以下の手順に従ってください。

1. TIA ポータルのメインメニューで、[オプション|設定]メニューコマンドを選択します。
2. ナビゲーションツリーで、[PLC プログラミング|一般ノード]を開きます。
3. [再初期化せずにダウンロード]領域で、メモリ予約を 0 バイトにセットします。



すでにデバイス交換を実行済みである場合は、それぞれのブロックから個々に 100 バイトの予約領域を削除する必要があります。

1. プロジェクトツリーで、[プログラムブロック]フォルダの 1 つのデータブロックを右クリックした後、ショートカットメニューから[プロパティ]を選択します。
2. [データブロックのプロパティ]ダイアログで、[再初期化せずにダウンロード]を選択します。
3. メモリ予約を 0 バイトにセットします。
4. ユーザーのプロジェクトのデータブロックごとにこの操作を繰り返します。



V4.1 の機能に関する完全な記述は、『S7-1200 プログラマブルコントローラシステムマニュアル』を参照してください。

索引

A

AS-i
AS-i スレーブの追加、166
AS-i マスタ CM 1243-2、165
AS-i マスタ CM1243-2 モジュールの追加、166
アドレス、168
ATEX 承認、365
AT タグオーバーレイ、70

C

CALCULATE (計算)、113
アナログのスケーリング、49
複雑な等式での使用、48
CANopen モジュール
021620-B、021630-B、434
CB 1241 RS485、426
CE 承認、363
CEIL (切り上げ)、112
CONV (変換)、111
CPU
AS-i アドレス、168
CPU 1211C、373
CPU 1212C、373
CPU 1214C、373
CPU 1215C、373
CPU 1217C、373
DB の値の取得、345
DB の開始値のリセット、345
HMI への通信の構成、141
HSC 構成、140
IP アドレス、87
OB の処理、97
PROFIBUS アドレス、163
PROFINET、87
RUN/STOP ボタン、34
アクセス保護、89
新しいデバイスの追加、78
アナログ入力のステップ応答時間、400
イーサネットポート、87
ウォッチテーブル、340
オペレータパネル、34、57、338
オンライン CPU からブロックをコピー、346
オンライン、348
オンライン接続、337

オンラインでのモニタ、339
概要、17
強制、342、343
工場出荷時設定へのリセット、349
サーマルゾーン、23、26
シグナルボード(SB)、21
診断バッファ、348
スタートアップ処理、82
スタートアップパラメータ、82
セキュリティレベル、89、89
デバイス構成、75
通信接続の数、146
通信のタイプ、143
通信ボード(CB)、21
動作モード、57
ネットワーク接続、142
ノウハウプロテクト、91
パスワード保護、89
パラメータの設定、82、86
比較チャート、18
プログラム実行、55
ブロックの比較と同期化、347
ブロックの呼び出し、103
未指定の CPU、77、352
モジュールの追加、80
CPU、メモリカード、またはパスワードへの結合、92
CPU プロパティ、ユーザー定義 Web ページ、261
CSM 1277 コンパクトスイッチモジュール、433
C-Tick 承認、365
CTRL_PWM 命令、123
cULus 承認、364

D

DB (データブロック)、100
DB の開始値のリセット、345
DC
出力、370
DeviceStates、130
DTL データタイプ、64

F

FAQ、5
FB (ファンクションブロック)、99
FBD (ファンクションブロックダイアグラム)、105
FC (ファンクション)、99

FLOOR、112
FM 承認、364
Freeport プロトコル、184

G

GET、170
GET (リモート CPU からデータを読み取る)
 接続の構成、157
Get LED status、129
GET_DIAG、130
Get_IM_Data、130
GPRS 経由の TeleService、175

H

HMI
 HMI 接続、51
 PROFINET 通信の構成、141
 画面、52
 ネットワーク接続、51
 はじめに、50、52
HMI 接続を作成する、51
HMI デバイス
 概要、22
 ネットワーク接続、142
HSC (高速カウンタ)
 構成、140、140
 操作、131、133
HSC クロック入力速度、388
HTML ページ、ユーザー定義、259
 開発、259
 更新、260
 ページ場所、261
HTTP 接続、Web サーバー、258

I

I メモリ
 LAD のモニタ、340
 ウォッチテーブル、339
 強制、342
 強制操作、343、343
 周辺機器入力アドレス(フォーステーブル)、342
 フォーステーブル、342
 モニタ、339
I/O
 LAD でのステータスのモニタ、340
 アドレス指定、68
 アナログ出力表示(電圧)、402
 アナログ出力表示(電流)、402

アナログ入力のステップ応答時間、400
アナログ入力表示(電圧)、398
アナログ入力表示(電流)、399
ウォッチテーブル、340
強制操作、343

IO-Link

構成、358
コンポーネント、357
データ、358
テクノロジーの概要、357
デバイスプロファイル、358
電源投入、357

IO-Link マスタ

イラストレーション、359
接続例、361
ブロックダイアグラム、360

IO-Link マスタシグナルモジュール、430

IP アドレス、87、87

オンライン CPU の構成、348

IP ルーター、87

ISO on TCP

アドホックモード、150

ISO on TCP プロトコル、149

ISO-on-TCP

接続 ID、150
接続の構成、156
パラメータ、154

K

Korea 認証の承認、365

L

LAD (ラダーロジック)

概要、104
ステータス、340、342
ステータスまたは値のモニタ、339
プログラミングエディタ、340
モニタ、340

LAD または FBD 命令への入力または出力の追加、33

LED (Get LED status)、129

LED インジケータ

LED 命令、129
通信インターフェース、186

M

MAC アドレス、87

Math、48、113

MC_ChangeDynamic (軸の動的な設定の変更)、330

MC_CommandTable、328
 MC_Halt (軸の一時停止)、317
 MC_Home (軸の原点復帰)、314
 MC_MoveAbsolute (軸の絶対位置決め)、319
 MC_MoveJog (ジョグモードでの軸の移動)、326
 MC_MoveRelative (軸の相対位置決め)、321
 MC_MoveVelocity (事前に定義された速度での軸の移動)、323
 MC_Power (軸の解放/ブロック)、310
 MC_ReadParam (テクノロジーオブジェクトのパラメータの読み出し)、334
 MC_Reset (エラーの確認)、313
 MC_WriteParam (テクノロジーオブジェクトのパラメータ)
 MODBUS
 バージョン、35
 ModuleStates、130
 MRES、オペレータパネル、34、57、338
 My Documentation Manager、5

N

NORM_X (正規化)、113

O

OB、(オーガニゼーションブロック)
 OPC、構成、180

P

PID

PID_3STEP (バルブ用同調機能付きの PID コントローラ)、203
 PID_3Step アルゴリズム、194
 PID_Compact (同調機能が統合された汎用 PID コントローラ)、197
 PID_Compact アルゴリズム、194
 PID_Temp (温度制御の処理が可能な汎用 PID コントローラ)、212
 概要、194
 コミッショニング、241

PLC

CPU の概要、17
 HSC 構成、140
 オンライン CPU からブロックをコピー、346
 強制、342
 強制操作、343
 ノウハウプロテクト、91
 タグ、42、46
 比較と同期化、347
 はじめに、41

ブロックの使用、58、95
 ブロックの呼び出し、103
 命令、46
 モジュールの追加、80
 モニタ、339

PLC システムの設計、58、95

PLC タグ

はじめに、42、46

PLC と HMI 間の

PLC 間の、142

ネットワーク接続を作成する、51

PM 1207 電源モジュール、433

PROFIBUS

CM 1243-5 (DP マスタ)モジュールの追加、162

DP スレーブの追加、162

CM 1242-5 (DP スレーブ)モジュール、159

CM 1243-5 (DP マスタ)モジュール、159

GET、170

PUT、170

S7 コネクション、171

アドレス、163

アドレス、設定、163

スレーブ、159

通信接続の数、146

ネットワーク接続、142

マスタ、159

PROFINET、141

GET、170

IP アドレス、87

PUT、170

S7 コネクション、171

アドホックモード、150

概要、149

接続 ID、150

通信接続の数、146

通信のタイプ、143

ネットワーク接続、142

ネットワークのテスト、88

PROFINET インターフェース

イーサネットアドレスのプロパティ、87

PROFINET RT、149

PTO (パルス列出力)、123

強制できません、343

PtP 通信、184

PUT、170

PUT (リモート CPU にデータを書き込む)

接続の構成、157

PWM

CTRL_PWM 命令、123

PWM (パルス幅変調)

強制できません、343

R

ROUND、112
RS232 および RS485 通信モジュール、186
RTD モジュールの概要、404
RUN/STOP ボタン、34
RUN モード、57、58
 オペレータパネル、34、57、338
 強制操作、343
 ツールバーのボタン、34
 プログラム実行、55
RUN モードでのダウンロード
 STEP 7 からの開始、354
 概要、353
RUN モードでのデバッグ、353
RUN モードでの編集、(RUN モードでのダウンロード)

S

S7-1200 機能の拡張、20
S7 通信
 接続の構成、157
SCALE_X (スケーリング)、113
SCL (構造化制御言語)
 CEIL (切り上げ)、112
 CONV (変換)、111
 DeviceStates、130
 FLOOR、112
 GET_DIAG、130
 Get_IM_Data、130
 LED ステータス、129
 MC_ChangeDynamic (軸の動的な設定の変更)、330
 MC_CommandTable、328
 MC_Halt (軸の一時停止)、317
 MC_Home (軸の原点復帰)、314
 MC_MoveAbsolute (軸の絶対位置決め)、319
 MC_MoveJog (ジョグモードでの軸の移動)、326
 MC_MoveRelative (軸の相対位置決め)、321
 MC_MoveVelocity (事前に定義された速度での軸の移動)、323
 MC_Power (軸の解放/ブロック)、310
 MC_ReadParam (テクノロジーオブジェクトのパラメータの読み出し)、334
 MC_Reset (エラーの確認)、313
 MC_WriteParam (テクノロジーオブジェクトのパラメータへの書き込み)、332
 ModuleStates、130
 NORM_X (正規化)、113
 PID の概要、194
 PID_3STEP (バルブ用同調機能付きの PID コントローラ)、203

PID_3Step アルゴリズム、194
PID_Compact (同調機能が統合された汎用 PID コントローラ)、197
PID_Compact アルゴリズム、194
PID_Temp (温度制御の処理が可能な汎用 PID コントローラ)、212
プログラミングエディタ、106
ROUND、112
SCALE_X (スケーリング)、113
Var セクション、106
概要、105
切り捨て、112
Siemens 技術サポート、6
SM および SB
 比較チャート、20
SMS、179
STARTUP モード
 強制操作、343
 プログラム実行、55
STEP 7
 AS-i、167
 LAD または FBD 命令への入力または出力の追加、33
 CPU の構成、82、86
 DB の値の取得、345
 DB の開始値のリセット、345
 HSC 構成、140
 PROFIBUS、163
 PROFINET、87
 RUN/STOP ボタン、34
 新しいデバイスの追加、78
 イーサネットポート、87
 エディタ間のドラッグ&ドロップ、36
 お気に入り、32
 オペレータパネル、34、57、338
 オンライン CPU からブロックをコピー、346
 拡張可能な入力または出力、33
 強制、342
 強制操作、343
 診断バッファ、348
 設定の変更、34
 デバイス構成、75
 ネットワーク接続、142
 パスワード保護、91
 比較と同期化、347
 プラグ接続が解除されたモジュール、39
 ポータルビューおよびプロジェクトビュー、31
 命令の挿入、32
 モジュールの構成、82、86
 モジュールの追加、80
 モニタ、339、340
 優先度クラス(OB)、59

STEP 7 Web ページ、5
 STEP 7 のプログラミング
 ユーザー定義 Web ページ、262
 STEP 7 用設定の変更、34
 STOP モード、57
 オペレータパネル、34、57、338
 強制操作、343
 ツールバーのボタン、34

T

TCON
 構成、156
 接続 ID、150
 接続パラメータ、154
 TCON_Param、154
 TCP
 アドホックモード、150
 接続 ID、150
 接続の構成、156、156
 パラメータ、154
 プロトコル、149
 TCP/IP 通信、141、149
 TIA ポータル
 CPU の構成、82、86
 PROFINET、87
 新しいデバイスの追加、78
 デバイス構成、75
 モジュールの構成、86
 TIA ポータル、ポータルビューおよびプロジェクト
 ビュー、31
 TRCV
 接続 ID、150
 TRCV (イーサネット経由のデータ受信(TCP))
 アドホックモード、150
 TRCV_C
 アドホックモード、150
 接続パラメータ、154
 TRCV_C (イーサネット経由のデータ受信(TCP))
 構成、156
 TRCV_C (イーサネット経由のデータ受信(TCP))
 接続 ID、150
 TRCV_C 命令、148
 TRUNC (切り捨て)、112
 TS Adapter、20
 TSAP (トランスポートサービスアクセスポイント)、
 158
 デバイスへの割り当て用命令、149
 TSEND
 接続 ID、150
 TSEND_C
 接続パラメータ、154

TSEND_C (イーサネット経由のデータ送信(TCP))
 構成、156
 接続 ID、150
 TSEND_C 命令、148
 TURCV
 接続パラメータ、154
 TURCV (イーサネット経由でデータを受信(UDP))
 構成、156
 TUSEND
 パラメータ、154
 TUSEND (イーサネット経由でデータを送信(UDP))
 構成、156

U

UDP
 接続の構成、156
 パラメータ、154
 UDP プロトコル、149
 USS プロトコルライブラリ、188

V

V3.0 CPU から V4.1 へのアップグレード、435
 V4.1 CPU 用 V3.0 CPU の交換、435

W

Web サーバー、255
 最大 HTTP 接続数、258
 制約事項、258
 Web ページ
 STEP 7 のサービス、サポート、およびドキュメン
 ト、5
 Web ページ、ユーザー定義、259
 WWW、262

あ

アクセス保護、CPU、89
 アクティブ/パッシブ接続、156
 アクティブ/パッシブ通信
 接続 ID、150
 パートナーの構成、156、172
 パラメータ、154
 新しい機能、27
 新しいデバイスを追加する
 既存のハードウェアを検出する、77
 未指定の CPU、77、352

新しいプロジェクト

- HMI 画面、52
- HMI 接続、51
- HMI デバイスの追加、50
- ネットワーク接続、51
- はじめに、41

アップロード

- オンライン CPU からブロックをコピー、346
- 検出する、352
- ユーザープログラム、346

アドホックモード、TCP および ISO on TCP、150

アドレス指定

- 一時的なメモリ、66
- グローバルメモリ、66
- 個々の入力(I)または出力(Q)、67
- データブロック、66
- ブールまたはビットの値、67
- プロセスイメージ、66
- メモリ領域、66

アナログ I/O

- 工学単位への変換、49
- 出力表示(電圧)、402
- 出力表示(電流)、402
- 入力のステップ応答時間、400
- 入力表示(電圧)、398
- 入力表示(電流)、399

アナログのスケールリング、49

アナログシグナルボード

- SB 1231、392
- SB 1232、392

アナログシグナルモジュール

- SM 1231、393
- SM 1231 RTD、407
- SM 1231 熱電対、409
- SM 1232、393
- SM 1234、394

アナログ入力

- CPU、SB、および SM の仕様、397
- RTD/TC SB および SM の仕様、410

い

イーサネット

- CSM 1277 コンパクトスイッチモジュール、433
- GET、170
- IP アドレス、87
- PUT、170
- アドホックモード、150
- 概要、149
- 接続 ID、150
- 通信、141
- 通信接続の数、146

通信のタイプ、143

ネットワーク接続、142

イーサネットプロトコル、149

マルチノード接続、171

イーサネット命令

TRCV_C、148

TSEND_C、148

位相調整、270

一時的なメモリ(L)、66

移動シーケンス(MC_CommandTable)、328

移動シーケンスとしての一連の軸コマンドの実行
(MC_CommandTable)、328

イベント、348

イベント実行、60

インスタンスデータブロック、66

インストール

ガイドライン、25

概要、25

空気の流れ、26

クリアランス、26

サーマルゾーン、23、26

シグナルモジュール(SM)、21

取り付け寸法、23

冷却、26

う

ウォッチテーブル、340

強制、126

モニタ、339

え

エディタ間のドラッグ&ドロップ、36

エディタを分割する

はじめに、42、46、46

遠隔制御、175

お

オーガニゼーションブロック

構成操作、98

作成、98

処理、97、97

ノウハウプロテクト、91

ファンクション、59

複数の周期、98

優先度クラス、59

呼び出し、59

[お気に入り]ツールバー、32

お問い合わせ情報、6、79

オプション処理(構成制御)、81
 オペレータパネル、22、34、57、338
 オンライン
 DB の値の取得、345
 DB の開始値のリセット、345
 IP アドレス、348
 RUN/STOP ボタン、34
 ウォッチテーブル、339、340
 オペレータパネル、34、57、338
 オンライン接続、337
 強制、342
 強制操作、343
 検出する、352
 サイクルタイムモニタ、338
 時刻、348
 ステータス、340
 ステータスまたは値のモニタ、339
 比較と同期化、347
 メモリ使用率のモニタ、338
 オンライン/オフライン CPU の比較と同期化、347
 オンライン CPU からブロックをコピー、346
 オンライン DB の値の取得、345
 オンライン診断ツール
 RUN モードでのダウンロード、353

か

海事承認、366
 ガイドライン
 設置、25
 カウンタ
 HSC (高速カウンタ)、131
 HSC 構成、140
 HSC 操作、133
 数量、19
 サイズ、19
 カウンタ命令、120
 拡張可能な命令、33
 カスタマサポート、6
 環境
 動作条件、368
 輸送および保管の条件、368

き

技術サポート、6
 技術仕様、363
 機能、新しい、27
 逆電圧の保護、370
 キューイング、60

強制、342
 I メモリ、342、343
 周辺機器入力、342、343
 スキャンサイクル、343
 入力および出力、343

く

空気の流れ、26
 クリアランス、空気の流れ、および冷却、26
 グローバルデータブロック、66、100
 グローバルメモリ、66
 グローバルライブラリ
 USS、188
 クロックメモリバイト、85

け

検出して、オンライン CPU をアップロードする、77
 検出する、352

こ

工場出荷時設定へのリセット、349
 構成制御(オプション処理)、81
 構造化プログラミング、95、95
 高速カウンタ
 HSC、131
 強制できません、343
 構成、140
 操作、133
 高電位絶縁テスト、369
 コードブロック、95
 CPU、メモリカード、またはパスワードへの結合、92
 DB (データブロック)、100
 FB (ファンクションブロック)、99
 FC (ファンクション)、99
 OB の数、19
 オーガニゼーションブロック(OB)、19、97
 カウンタ(数量およびメモリ要件)、19
 コードブロックの数、19
 コピー保護、92
 タイマ(数量およびメモリ要件)、19
 ネストレベル、19
 ノウハウプロテクト、91
 ブロックの呼び出し、103
 モニタ、19
 ユーザープログラムのサイズ、19
 割り込み、19

コピー保護

CPU、メモ리카ード、またはパスワードへの結合、92

コンパクトスイッチモジュール、CSM 1277、433

コンフィグレーション

AS-i、167

HSC (高速カウンタ)、140

IP アドレス、87

PID_Compact および PID_3Step 命令、224

PID_Temp 命令、227

PROFIBUS、163

PROFIBUS アドレス、163

PROFINET、87

検出する、77、352

産業用イーサネットポート、87

スタートアップパラメータ、82

ネットワーク接続、142

モジュールの追加、80

ユーザー定義 Web ページ、261

さ

サージ耐性、367

サービス & サポート、6

サーマルゾーン、23、26

サイクルタイムモニタ、338

最初のスキャンインジケータ、85

最大 Web サーバー接続数、258

サブネットマスク、87

サポート、6

産業環境

承認、366

し

シグナルボード(SB)

SB 1221 4 DI 24 V DC 200 kHz、377

SB 1221 4 DI 5 V DC 200 kHz、377

SB 1222 4 DQ 24 V DC 200 kHz、377

SB 1222 4 DQ 5 V DC 200 kHz、377

SB 1223 2 DI/2 DQ 24 V DC、377

SB 1223 DI/DQ 24 V DC 200 kHz、377

SB 1223 DI/DQ 5 V DC 200 kHz、377

SB 1231 AI 1 x 12 ビット、392

SB 1231 AI 1 x 16 ビット RTD、405

SB 1231 AI 1 x 16 ビット TC、405

SB 1231 RTD、405

SB 1231 TC、405

SB 1232 AQ 1 x 12 ビット、392

アナログ出力表示(電圧)、402

アナログ出力表示(電流)、402

アナログ入力のステップ応答時間、400

概要、21

デバイス構成、75

入力表示(電圧)、398

入力表示(電流)、399

モジュールの追加、80

シグナルボード(SM)

新しいデバイスの追加、78

シグナルモジュール(SM)

SM 1221 DI 16 24 V DC、380

SM 1221 DI 8 24 V DC、380

SM 1222 DQ 16 24 V DC、382

SM 1222 DQ 16 リレー、382

SM 1222 DQ 8 24 V DC、382

SM 1222 DQ 8 リレー、382

SM 1222 DQ 8 リレー切り替え、382

SM 1223 DI 1223 DI 120/230 V AC/DQ リレー、385

SM 1223 DI 16/DQ 16 リレー、383

SM 1223 DI 8/DQ 8/DQ リレー、383

SM 1223 DI 8/DQ 8、383、383

SM 1231 AI 4 x 13 ビット、393

SM 1231 AI 4 x 16 ビット、393

SM 1231 AI 4 x 16 ビット TC、409

SM 1231 AI 4 x RTD x 16 ビット、407

SM 1231 AI 8 x 13 ビット、393

SM 1231 AI 8 x 16 ビット TC、409

SM 1231 AI 8 x RTD x 16 ビット、407

SM 1232 AQ 2 x 14 ビット、393

SM 1232 AQ 4 x 14 ビット、393

SM 1234 AI 4 x 13 ビット/AQ 2 x 14 ビット、394

SM 1278 4xIO-Link マスタ、430

新しいデバイスの追加、78

アナログ出力表示(電圧)、402

アナログ出力表示(電流)、402

アナログ入力のステップ応答時間、400

アナログ入力表示(電圧)、398

アナログ入力表示(電流)、399

概要、21

デバイス構成、75

モジュールの追加、80

時刻

オンライン CPU の構成、348

システムメモリバイト、85

修正

プログラミングエディタのステータス、340

出力パラメータ、99

周波数、クロックビット、85

仕様

CB 1241 RS485、426

CM 1241 RS232、428

CM 1241 RS422/485、427

CPU 1211C、373
 CPU 1212C、373
 CPU 1214C、373
 CPU 1215C、373
 CPU 1217C、373
 SB 1221 4 DI 24 V DC 200 kHz、377
 SB 1221 4 DI 5 V DC 200 kHz、377
 SB 1222 4 DQ 24 V DC 200 kHz、377
 SB 1222 4 DQ 5 V DC 200 kHz、377
 SB 1223 2 DI/2 DQ 24 V DC、377
 SB 1223 DI/DQ 24 V DC 200 kHz、377
 SB 1223 DI/DQ 5 V DC 200 kHz、377
 SB 1231 AI 1 x 12 ビット、392
 SB 1231 AI 1 x 16 ビット RTD、405
 SB 1231 AI 1 x 16 ビット TC、405
 SB 1232 AQ 1 x 12 ビット、392
 SM 1221 DI 16 24 V DC、380
 SM 1221 DI 8 24 V DC、380
 SM 1222 DQ 16 24 V DC、382
 SM 1222 DQ 16 リレー、382
 SM 1222 DQ 8 24 V DC、382
 SM 1222 DQ 8 リレー、382
 SM 1222 DQ 8 リレー切り替え、382
 SM 1223 DI 120/230 V AC/DQ リレー、385
 SM 1223 DI 16/DQ 16 リレー、383
 SM 1223 DI 8/DQ 8、383、383
 SM 1223 DI 8/DQ 8 リレー、383
 SM 1231 AI 4 x 13 ビット、393
 SM 1231 AI 4 x 16 ビット、393
 SM 1231 AI 4 x 16 ビット TC、409
 SM 1231 AI 4 x RTD x 16 ビット、407
 SM 1231 AI 8 x 13 ビット、393
 SM 1231 AI 8 x 16 ビット TC、409
 SM 1231 AI 8 x RTD x 16 ビット、407
 SM 1232 AQ 2 x 14 ビット、393
 SM 1232 AQ 4 x 14 ビット、393
 SM 1234 AI 4 x 13 ビット/AQ 2 x 14 ビット、394
 SM 1278 4xIO-Link マスタ、430
 アナログ出力表示(電圧)、402
 アナログ出力表示(電流)、402
 アナログ入力表示(電圧)、398
 アナログ入力表示(電流)、399
 一般技術仕様、363
 環境条件、368
 産業用環境、366
 承認、363
 定格電圧、369
 電磁環境両立性(EMC)、367
 入力のステップ応答時間、400
 使用法を示すクロスリファレンス、127
 情報リソース、5

初期値

DB の開始値の取得とリセット、345

シリアル通信、184

診断

DeviceStates、130

GET_DIAG、130

Get_IM_Data、130

LED 命令、129

ModuleStates、130

ステータスインジケータ、85

バッファ、348

す

スキャンサイクル

強制操作、343、343

スタートアップパラメータ、82

ステータス

LED インジケータ(通信インターフェース)、186

LED 命令、129

スライス(タグ付きデータタイプの)、69

せ

整合性チェック、128

制約事項

Web サーバー、258

ユーザー定義 Web ページ、261

接続

HMI 接続、51

S7 コネクション、171

Web サーバー、258

イーサネットプロトコル、171

構成、154

接続 ID、150

接続数(PROFINET/PROFIBUS)、146

タイプ、マルチノード接続、171

通信のタイプ、143

ネットワーク接続、51

パートナー、156、172

セキュリティ

CPU、89

CPU、メモリカード、またはパスワードへの結合、92

アクセス保護、89

コードブロックのノウハウプロテクト、91

コピー保護、92

設定、34

接点

プログラミング、44

そ

相違点

ポイントツーポイント命令で、185
送信パラメータの設定、156、172

た

タイマ

サイズ、19
数量、19

タグ

オーバーレイ、70
強制操作、343
ステータスまたは値のモニタ、339
スライス、69
はじめに、42、46

タスクカード

列およびヘッダー、35
タスクカードの列およびヘッダー、35

つ

通信

AS-i アドレス、168
IP アドレス、87
PROFIBUS アドレス、163
PROFINET および PROFIBUS、143
TCON_Param、154
アクティブ/パッシブ、154、156、172
構成、154、156、172
接続 ID、150
接続数(PROFINET/PROFIBUS)、146
ネットワーク、141
ネットワーク接続、142
パラメータ、154

通信インターフェース

CB 1241 RS485、426
CM 1241 RS232、428
RS232 および RS485、186
モジュールの追加、80
モジュールの比較チャート、20

通信プロセッサ(CP)

概要、21
比較チャート、20
モジュールの追加、80

通信ボード(CB)

CB 1241 RS485、426
LED インジケータ、186
RS485、186
概要、21
比較チャート、20

モジュールの追加、80

通信モジュール(CM)

AS-i マスタ CM1243-2 モジュールの追加、166
CM 1243-5 (DP マスタ)モジュールの追加、162
CM 1241 RS232、428
CM 1241 RS422/RS485、427
LED インジケータ、186
RS232 および RS485、186
新しいデバイスの追加、78
概要、21
デバイス構成、75
比較チャート、20
モジュールの追加、80

通信モジュール(CM)、USS ライブラリ、188

て

定格電圧、369、369

テクノロジーモジュール、SM 1278 4xIO-Link マスタ、430

データ処理ブロック(DHB)、100

データタイプ、64

DTL、64

データブロック

値の取得、345、345
インスタンスデータブロック、66
オーガニゼーションブロック(OB)、97
開始値のリセット、345、345
グローバルデータブロック、66、100

データログ

データログの概要、124

テクノロジーオブジェクト

PID、195
モーションコントロール、272

デジタルシグナルボード

SB 1221、377
SB 1222、377
SB 1223、377

デジタルシグナルモジュール

SM 1221、380
SM 1222、382
SM 1223、383、385

デジタル出力

リレー、24 V DC CPU、SM、および SB の仕様、389

デジタル入力

SM の AC 仕様、388
CPU、SM、および SB の V DC 仕様、386

デバイス構成、75

AS-i、167
AS-i ポート、167
CPU の構成、82、86

PROFIBUS、163
 PROFINET、87
 新しいデバイスの追加、78
 イーサネットポート、87
 検出する、77、352
 デバイスタイプの変更、79
 ネットワーク接続、142
 プラグ接続が解除されたモジュール、39
 モジュールの構成、82、86
 モジュールの追加、80

デバイスの交換

V4.1 CPU 用 V3.0 CPU、435
 手順、79

デバイスの変更、79

デバイスを挿入する

未指定の CPU、77、352

電源モジュール

PM1207、433

電磁環境両立性、368

電磁環境両立性(EMC)、367

と

動作モード、34、34、57、338

動的結合、92

ドキュメント、5

トリガ処理

トレース、355

取り付け

ガイドライン、25

空気の流れ、26

クリアランス、26

サーマルゾーン、23、26

寸法、23

冷却、26

トレース機能、355

に

入力および出力

モニタ、339

認証

ATEX、365

CE、363

C-Tick、365

cULus、364

FM、364

Korea 認証、365

海事、366

ね

熱電対モジュールの概要、404

ネットワーク

ネットワーク接続、51

ネットワーク、LAD プログラミング、44

ネットワーク接続

HMI デバイス、51

デバイスの接続、142

ネットワーク通信、141

の

ノウハウプロテクト

パスワード保護、91

は

配線図

CB 1241 RS 485、426

CPU 1214C AC/DC/リレー、376

CPU 1214C DC/DC/DC、376

SB 1221 DI 4 200 kHz、379

SB 1223 DI 2/DQ 2 200 kHz、379

SB 1231 AI 1 x 12 ビット、392

SB 1231 AI 1 x 16 ビット RTD、406

SB 1231 AI 1 x 16 ビット TC、406

SB 1232 AQ 1 x 12 ビット、392

SM 1221 DI 16 24 V DC、381

SM 1221 DI 8 24 V DC、381

SM 1222 DQ 16 24 V DC、383

SM 1222 DQ 16 リレー、383

SM 1223 DI 16 V DC / DQ 16 リレー、384

SM 1223 DI 16 V DC/ DQ 16 24 V DC、384

SM 1223 DI 8 120/230 V AC/DQ 8 リレー、385

SM 1231 AI 8 x 13 ビット、394

SM 1231 RTD 4 x 16 ビット、408

SM 1231 RTD 8 x 16 ビット、408

SM 1232 AQ 4 x 13 ビット、394

SM 1234 AI 4 x 13 ビット/AQ 2 x 14 ビット、394

SM 1278 IO-Link マスタ、432

配線のガイドライン

空気の流れおよび冷却用のクリアランス、26

は

ハードウェアコンフィグレーション、75

AS-i、167

AS-i ポート、167

CPU の構成、82、86

- PROFIBUS、163
- PROFINET、87
- 新しいデバイスの追加、78
- イーサネットポート、87
- 検出する、77、352
- ネットワーク接続、142
- モジュールの構成、82、86
- モジュールの追加、80
- はじめに
 - CPU、41
 - HMI、50、52
 - HMI 接続、51
 - LAD プログラム、44、47
 - Math 命令、47
 - PLC タグ、42、46
 - 新しい PLC、41
 - アドレス指定、46
 - コードブロック、101
 - 接点、44
 - タグ、42、46
 - ネットワーク接続、51
 - プログラムブロック、101
 - プロジェクト、41
 - 分割エディタ、42、46、46
 - ボックス命令、47
 - 命令、46
- バスコネクタ、21
- パスワード保護
 - CPU、89
 - CPU、メモ리카ード、またはパスワードへの結合、92
 - CPU へのアクセス、89
 - コードブロック、91
 - コピー保護、92
- パッシブ/アクティブ通信
 - 接続 ID、150
 - パートナーの構成、156、172
 - パラメータ、154
- バッテリーボード、BB 1297、396
- パネル(HMI)、22
- パラメータ割り当て、99
- パラメータを設定する
 - CPU、82、86
 - PROFINET、87
 - イーサネットポート、87
 - モジュール、82、86
- パルス列出力(PTO)、123
- バルブ PID 同調、203
- ひ
 - 比較チャート
 - CPU モデル、18
 - HMI デバイス、22
 - モジュール、20
 - 比較命令、110
 - ビジュアルライゼーション、HMI デバイス、22
 - 日付と時刻 Long データタイプ、64
 - ビット論理、108
 - 標準 Web ページ、255、256
- ふ
 - ファームウェア更新
 - STEP 7 から、350
 - ファームウェアの更新
 - STEP 7 から、350
 - ファンクション(FC)、99
 - ノウハウプロテクト、91
 - ファンクションブロック(FB)
 - インスタンスデータブロック、99
 - 出力パラメータ、99
 - 初期値、99
 - ノウハウプロテクト、91
 - ブールまたはビットの値、67
 - フォーステーブル
 - 強制、342
 - 強制操作、343
 - 周辺機器入力のアドレス指定、342
 - プラグ接続が解除されたモジュール、39
 - フラグメント DB (ユーザー定義 Web ページ)
 - 生成、261
 - プログラミング
 - LAD または FBD 命令への入力または出力の追加、33
 - CPU、メモ리카ード、またはパスワードへの結合、92
 - FBD (ファンクションブロックダイアグラム)、105
 - LAD (ラダー)、104
 - PID_3STEP (バルブ用同調機能付きの PID コントローラ)、203
 - PID_3Step アルゴリズム、194
 - PID_Compact (同調機能が統合された汎用 PID コントローラ)、197
 - PID_Compact アルゴリズム、194
 - PID_Temp (温度制御の処理が可能な汎用 PID コントローラ)、212
 - PID の概要、194
 - SCL (構造化制御言語)、105、106
 - エディタ間のドラッグ&ドロップ、36
 - お気に入り、32

- 拡張可能な命令、33
- コードブロックの比較と同期化、347
- 構造化、95
- はじめに、46
- プラグ接続が解除されたモジュール、39
- 未指定の CPU、77、352
- 命令の挿入、32
- 優先度クラス、59
- リニア、95
- プログラミングエディタ
 - DB の値の取得、345
 - DB の開始値のリセット、345
 - ステータス、340
 - モニタ、340
- プログラム
 - CPU、メモ리카ード、またはパスワードへの結合、ブロック
 - 92
 - DB の値の取得、345
 - DB の開始値のリセット、345
 - Math 命令、47
 - オーガニゼーションブロック(OB)、97
 - オンライン CPU からブロックをコピー、346
 - サンプルネットワーク、44、47
 - はじめに、44
 - はじめに、47
 - パスワード保護、91
 - ブロックの呼び出し、103
 - 優先度クラス、59
- プログラムカード、63
- プログラム構造、95
- プログラム情報
 - 呼び出し構造体で、128
- プログラムの実行
 - 概要、55
 - ブロック構造、58
- プログラムのテスト、126
- プログラムのモニタ、126
- プログラムブロック
 - 作成、101
 - はじめに、41
- プロジェクト
 - CPU、メモ리카ード、またはパスワードへの結合、
 - 92
 - CPU へのアクセスの制限、89
 - HMI 画面、52
 - HMI 接続、51
 - HMI デバイスの追加、50
 - アクセス保護、89
 - コードブロックの保護、91
 - タグ、42、46
 - ネットワーク接続、51
 - はじめに、41
 - 比較と同期化、347
 - プログラム、46
- プロジェクトビュー、31、31
 - CPU パラメータの設定、82、86
 - PROFINET、87
 - 新しいデバイスの追加、78
 - イーサネットポートの構成、87
 - デバイス構成、75
 - モジュールの構成、82、86
- プロセスイメージ
 - 強制、342
 - 強制操作、343
 - ステータス、340、342
 - ステータスまたは値のモニタ、339
 - モニタ、340
- OB の数、19、60
- イベント、60
- オーガニゼーションブロック(OB)、19、58、59、60
- オンライン CPU からブロックをコピー、346
- カウンタ(数量およびメモリ要件)、19
- コードブロックの数、19
- スタートアップ OB、60
- 整合性チェック、128
- タイプ、58
- タイマ(数量およびメモリ要件)、19
- データブロック(DB)、58
- ネストレベル、19
- パスワード保護、91
- はじめに、101
- ファンクション(FC)、58
- ファンクションブロック(FB)、58
- 別のコードブロックの呼び出し、103
- モニタ、19
- ユーザープログラムのサイズ、19
- 割り込み、19、60
- ブロックのムーブ(MOVE_BLK)命令、110
- ブロック呼び出し
 - 基本、58
- プロトコル
 - freeport、184
 - ISO on TCP、149
 - Modbus、184
 - PROFINET RT、149
 - TCP、149
 - UDP、149
 - USS、184
 - 通信、184

へ

ベーシックパネル(HMI)、22

ほ

ポイントツーポイント通信、184

ポータルビュー、31

CPU の構成、82、86

PROFINET、87

新しいデバイスの追加、78

イーサネットポートの構成、87

モジュールの構成、82、86

ポート番号

通信パートナーへの割り当て、149

保護クラス、369

保護等級、369

保護レベル

CPU、89

CPU、メモ리카ード、またはパスワードへの結合、92

コードブロック、91

保持型メモリ、18、63

ボックス命令

はじめに、47

ホットライン、6

ポッドキャスト、5

ま

待ち時間、60

マニュアル、5

マルチノード接続

イーサネットプロトコル、171

接続タイプ、171

み

未指定の CPU、77、352、352

む

ムーブ命令、110

め

命令

CALCULATE、48

CALCULATE (計算)、113

CEIL (切り上げ)、112

CONV (変換)、111

CTRL_PWM)、123

DeviceStates、130

FLOOR、112

GET、170

GET_DIAG、130

Get_IM_Data、130

HSC (高速カウンタ)、131、133

LAD または FBD 命令への入力または出力の追加、33

LED ステータス、129

MC_ChangeDynamic (軸の動的な設定の変更)、330

MC_CommandTable、328

MC_Halt (軸の一時停止)、317

MC_Home (軸の原点復帰)、314

MC_MoveAbsolute (軸の絶対位置決め)、319

MC_MoveJog (ジョグモードでの軸の移動)、326

MC_MoveRelative (軸の相対位置決め)、321

MC_MoveVelocity (事前に定義された速度での軸の移動)、323

MC_Power (軸の解放/ブロック)、310

MC_ReadParam (テクノロジーオブジェクトのパラメータの読み出し)、334

MC_Reset (エラーの確認)、313

MC_WriteParam (テクノロジーオブジェクトのパラメータへの書き込み)、332

ModuleStates、130

NORM_X (正規化)、113

PID_Compact (同調機能が統合された汎用 PID コントローラ)、197

PID_Temp (温度制御の処理が可能な汎用 PID コントローラ)、212

PUT、170

ROUND、112

SCALE_X (スケーリング)、113

TRCV_C、148

TRUNC (切り捨て)、112

TSEND_C、148

WWW、262

アナログ値のスケーリング、49

移動、110

エディタ間のドラッグ&ドロップ、36

お気に入り、32

カウンタ、120

拡張可能な命令、33

強制操作、343

ステータス、340

ステータスまたは値のモニタ、339

挿入、32

ドラッグ&ドロップ、32

はじめに、46、47

- パラメータの追加、47
 - 比較、110
 - ビット論理、108
 - ブロックのムーブ(MOVE_BLK)、110
 - モーションコントロール、309
 - モニタ、340
 - 列およびヘッダー、35
 - 割り込み不可能なムーブ(UMOVE_BLK)命令、110
 - 命令のバージョン、35
 - 命令を挿入する
 - エディタ間のドラッグ&ドロップ、36
 - お気に入り、32
 - ドラッグ&ドロップ、32
 - 命令のバージョン、35
 - メインエントリ、382
 - メータへの書き込み、332
 - メモリ
 - 一時的なメモリ(L)、66
 - クロックメモリ、84
 - システムメモリ、84
 - 周辺機器入力アドレス(フォーステーブル)、342
 - 保持型メモリ、63
 - ロードメモリ、63
 - ワークメモリ、63
 - メモリカード
 - ロードメモリ、63
 - メモリ使用率のモニタ、オンライン、338
 - メモリ領域
 - 一時的なメモリ、66
 - グローバルメモリ、66
 - 即時アクセス、66
 - データブロック、66
 - ブールまたはビットの値のアドレス指定、67
 - プロセスイメージ、66
- も**
- モーションコントロール
 - MC_ChangeDynamic (軸の動的な設定の変更)、330
 - MC_CommandTable、328
 - MC_Halt (軸の一時停止)、317
 - MC_Home (軸の原点復帰)、314
 - MC_MoveAbsolute (軸の絶対位置決め)、319
 - MC_MoveJog (ジョグモードでの軸の移動)、326
 - MC_MoveRelative (軸の相対位置決め)、321
 - MC_MoveVelocity (事前に定義された速度での軸の移動)、323
 - MC_Power (軸の解放/ブロック)、310
 - MC_ReadParam (テクノロジーオブジェクトのパラメータの読み出し)、334
 - MC_Reset (エラーの確認)、313
 - MC_WriteParam (テクノロジーオブジェクトのパラ
位相調整、270
概要、265
原点復帰(アクティブ原点復帰のシーケンス)、308
原点復帰設定パラメータ、305
軸の原点復帰、304
軸の構成、273、282
ハードウェアおよびソフトウェアリミットスイッチ、
300
メータへの書き込み)、332
 - モーションコントロール命令、309
 - モジュール
 - サーマルゾーン、23、26
 - シグナルボード(SB)、21
 - シグナルモジュール(SM)、21
 - 通信プロセッサ(CP)、21
 - 通信ボード(CB)、21
 - 通信モジュール(CM)、21
 - パラメータの設定、82、86
 - 比較チャート、20
 - モジュールの交換、39
 - モニタリング
 - DB の値の取得、345
 - DB の開始値のリセット、345
 - LAD ステータス、340
 - LAD ステータスとウォッチテーブルの使用、339
 - LED 命令、129
 - 強制操作、343
 - フォーステーブル、342
- ゆ**
- ユーザーインターフェース
 - STEP 7 プロジェクトおよびポータルビュー、31
 - ユーザー定義 Web ページ、255、259
 - HTML エディタによる作成、259
 - WWW 命令による有効化、262
 - STEP 7 でのプログラミング、262
 - 更新、260
 - 構成、261
 - プログラムブロックの生成、261
 - ロードメモリの制約事項、261
 - ユーザー定義 Web ページ DB の作成、261
 - ユーザー定義 Web ページ DB の生成、261
 - ユーザー定義 Web ページの更新、260
 - ユーザー定義 Web ページの作成、259
 - ユーザー定義 Web ページの制御 DB
 - WWW 命令のパラメータ、262
 - ユーザープログラム
 - LAD または FBD 命令への入力または出力の追加、
33

CPU、メモリカード、またはパスワードへの結合、割り込み
92
エディタ間のドラッグ&ドロップ、36
オーガニゼーションブロック(OB)、97
お気に入り、32
オンライン CPU からブロックをコピー、346
拡張可能な命令、33
パスワード保護、91
命令の挿入、32
優先度
優先度クラス、59
処理の優先度、60

よ

呼び出し構造体、128

れ

例、PID
PID_3Step、構成設定、225
PID_Compact、構成設定、224
PID_Temp、構成設定、227
例、さまざまな
AT タグオーバーレイ、70
エディタ間のドラッグ&ドロップ、36
タグ付きデータタイプのスライス、70
トレースおよびロジックアナライザファンクション、
355
例、通信
AS-i スレーブアドレス指定、168
TSEND_C または TRCV_C 接続経由の CPU 通信、
153
PROFINET 通信プロトコル、149
遠隔制御、179
共通の送受信接続による CPU 通信、152
別々の送受信接続による CPU 通信、151
例、モーションコントロール
CPU 1211C、CPU 1212C、CPU 1214C、および
CPU 1215C のパルス出力速度の設定、269
CPU 1217C パルス出力速度の設定、268
MC 原点復帰の速度特性、308
テクノロジーオブジェクトモーションコマンドテー
ブルの設定、295
例、命令
CALCULATE、48
冷却、26

わ

ワークメモリ、18、63

概要、59
割り込み待ち時間、60
割り込み不可能なムーブ(UMOVE_BLK)命令、110

り

リニアプログラミング、95
リレーの電気耐用年数、371

る

ルーターの IP アドレス、87

ろ

ローカル/パートナー接続、156
ロードメモリ、18、63
ユーザー定義 Web ページ、261
ロジックアナライザ、355