

SIEMENS

SIMATIC Ident

RFID-Systeme Ident-Profil und Ident- Bausteine, Standardfunktion für Ident-Systeme

Funktionshandbuch

<u>Einleitung</u>	1
<u>Beschreibung</u>	2
<u>Bausteine parametrieren/programmieren</u>	3
<u>Fehlermeldungen</u>	4
<u>Anhang</u>	A
<u>Service & Support</u>	B

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 GEFAHR
bedeutet, dass Tod oder schwere Körpverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 WARNUNG
bedeutet, dass Tod oder schwere Körpverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 VORSICHT
bedeutet, dass eine leichte Körpverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Inhaltsverzeichnis

1	Einleitung.....	5
1.1	Security-Hinweise.....	7
2	Beschreibung	9
2.1	Anwendungsbereich und Merkmale.....	9
2.2	Bausteinfunktion	13
2.3	Abgrenzung zu anderen Programmbausteinen.....	14
3	Bausteine parametrieren/programmieren	17
3.1	Übersicht der Ident-Anweisungen.....	17
3.2	Parametrieren/programmieren	22
3.3	Allgemeiner Aufbau der Funktionsbausteine	25
3.4	Ident-Bausteine programmieren.....	28
3.4.1	Basis-Bausteine	28
3.4.1.1	Read.....	28
3.4.1.2	Read_MV.....	29
3.4.1.3	Reset_Reader.....	30
3.4.1.4	Set_MV_Program	31
3.4.1.5	Write	31
3.4.2	Status-Bausteine	33
3.4.2.1	Reader_Status	33
3.4.2.2	Tag_Status.....	37
3.4.3	Tag-Feld-Bausteine.....	40
3.4.3.1	Read_Tagfield	40
3.4.3.2	Write_Tagfield.....	41
3.4.4	Erweiterte Bausteine.....	43
3.4.4.1	AdvancedCmd	43
3.4.4.2	Config_Upload/-_Download.....	44
3.4.4.3	Inventory	47
3.4.4.4	Read_EPC_Mem.....	52
3.4.4.5	Read_TID	53
3.4.4.6	Read_UID.....	54
3.4.4.7	Set_Ant.....	55
3.4.4.8	Set_Param	56
3.4.4.9	Write_EPC_ID.....	59
3.4.4.10	Write_EPC_Mem	60
3.4.5	Reset-Bausteine.....	61
3.4.5.1	Reset_MOBY_D	61
3.4.5.2	Reset_MOBY_U	62
3.4.5.3	Reset_MV.....	63
3.4.5.4	Reset_RF200	64
3.4.5.5	Reset_RF300	65
3.4.5.6	Reset_Univ.....	66

3.5	Ident-Profil programmieren	67
3.5.1	Aufbau des Ident-Profiles	67
3.5.2	Datenstruktur des Ident-Profiles.....	71
3.5.3	Befehle des Ident-Profiles	73
3.5.3.1	Befehlsstruktur.....	74
3.5.3.2	Befehle.....	76
3.5.3.3	Erweiterte Befehle für Optische Lesesysteme (MV400/MV500)	95
3.5.3.4	Wirkung der Befehle.....	99
3.5.3.5	Befehlsbearbeitung	99
3.5.3.6	Neustart und Wiederanlauf parametrieren	100
3.5.3.7	Verkettung.....	101
3.5.3.8	Befehlswiederholung	103
3.6	Transponder-Adressierung.....	109
4	Fehlermeldungen	117
4.1	Aufbau des Ausgangsparameters "STATUS".....	117
4.2	STEP 7 - Fehlermeldungen.....	118
4.3	Fehler vom Kommunikationsmodul/Reader	118
4.4	Fehler vom optischen Lesegerät	127
4.5	Fehler vom Bus/Rückwandbus.....	129
4.6	Warnungen.....	130
A	Anhang	131
A.1	Interne Statusparameter	131
A.2	Zyklische Daten.....	133
A.2.1	Zyklisches Steuerwort	133
A.2.2	Zyklisches Statuswort.....	134
A.2.3	Verwendung des Quittier- und des Befehlszählers.....	135
A.3	Anschluss serieller Geräte via Freeport-Protokoll.....	140
A.3.1	Kompatible Ident-Geräte	140
A.3.2	Funktionen und Befehle	140
B	Service & Support	143

Einleitung

Zweck dieser Dokumentation

Als Schnittstelle zu den Kommunikationsdiensten stehen für Ihr Anwenderprogramm vorgefertigte Programmbausteine (FCs und FBs) zur Verfügung. Dieses Handbuch enthält Beschreibungen der Ident-Anweisungen (Ident-Profil und -Bausteine), mit deren Hilfe Sie verschiedene Ident-Systeme in Betrieb nehmen und parametrieren können.

Es richtet sich sowohl an Programmierer und Tester, als auch an Service- und Wartungstechniker.

Gültigkeitsbereich dieser Dokumentation

Diese Dokumentation ist gültig für das Ident-Profil bzw. die Ident-Bausteine mit der Anweisungsbibliothek V5.4.

Alternativ kann ab TIA Portal V14 SP1 die Parametrierung über das Technologieobjekt erfolgen.

ACHTUNG

Beschreibung für das Ident-Profil bzw. die Ident-Bausteine im TIA Portal ab V14 SP1

Die Parametrierung mit Hilfe des Ident-Profiles bzw. der Ident-Bausteine in Verbindung mit dem Technologieobjekt "SIMATIC Ident > TO_Ident" werden in der TIA Portal-Hilfe ab V14 SP1 beschrieben.

→ TIA Portal-Hilfe, Suchen nach: Technologieobjekt SIMATIC Ident

Einordnung in die Dokumentationslandschaft

Weitere Informationen zu den in diesem Handbuch genannten Bausteinen und Ident-Geräten, finden Sie im Internet

(<https://support.industry.siemens.com/cs/ww/de/ps/14970/man>) in den folgenden Handbüchern:

- FB 45 für MOBY U, MOBY D, RF200, RF300
- SIMATIC RF360R
- SIMATIC RF600
- Anschaltmodul ASM 456
- SIMATIC RF120C
- SIMATIC RF166C
- SIMATIC RF170C

- SIMATIC RF180C
- SIMATIC RF185C, RF186C, RF188C, RF186CI, RF188CI
- SIMATIC MV400
- SIMATIC MV500

Spezifikationen

Die in dem Handbuch enthaltenen Ident-Bausteine setzen auf dem Protokoll "Proxy Ident Function Block" auf. Die Spezifikation des "Proxy Ident Function Block" erhalten Sie über die PROFIBUS-Nutzerorganisation.

Marken

Folgende und eventuell weitere nicht mit dem Schutzrechtsvermerk ® gekennzeichnete Bezeichnungen sind eingetragene Marken der Siemens AG:

SIMATIC ®, SIMATIC RF ® und MOBY ®

Abkürzungen und Namenskonventionen

Innerhalb dieser Dokumentation werden folgende Begriffe/Abkürzungen synonym verwendet:

Transponder, Tag

Kommunikationsmodul (CM)

Mobiler Datenspeicher (MDS), Datenträger

Anschaltmodul (ASM)

1.1 Security-Hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen einen Bestandteil eines solchen Konzepts.

Die Kunden sind dafür verantwortlich, unbefugten Zugriff auf ihre Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Diese Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und nur wenn entsprechende Schutzmaßnahmen (z.B. Firewalls und/oder Netzwerksegmentierung) ergriffen wurden.

Weiterführende Informationen zu möglichen Schutzmaßnahmen im Bereich Industrial Security finden Sie unter:

<https://www.siemens.com/industrialsecurity>

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Produkt-Updates anzuwenden, sobald sie zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter:

<https://www.siemens.com/industrialsecurity>

Beschreibung

2.1 Anwendungsbereich und Merkmale

Die Ident-Anweisungen enthalten STEP 7-Funktionen für Identifikationssysteme. Die Anweisungen bestehen aus Ident-Bausteinen und dem Ident-Profil. Das Ident-Profil kann in den SIMATIC-Steuerungen S7-300, S7-400, S7-1200 und S7-1500 für diverse Kommunikationsmodule, RFID-Reader und optische Lesesysteme eingesetzt werden. Projektierbar ist es mit STEP 7 ab V5.5 und STEP 7 Basic / Professional ab V13. Die Ident-Bausteine setzen auf dem Ident-Profil auf und sind mit STEP 7 Basic / Professional ab V13 projektierbar.



Bild 2-1 Über die Ident-Anweisungen programmierbare Baugruppen: ASM 456, RF170C, RF180C, RF120C, RF166C, RF18xC/RF18xCI, RF61xR/RF68xR, MV400 und MV500

Projektierung und Programmierung der Baugruppen

Mithilfe von Automatisierungssystemen wie z. B. dem TIA Portal können Sie über die Baugruppenparameter parametrieren und mit Unterstützung des Technologieobjekts "SIMATIC Ident > TO_Ident" diese Baugruppen projektieren (ab TIA V14 SP1). Bei der Projektierung erstellen Sie ein Projekt, bestehend aus mehreren Baugruppen (z. B. Steuerung, Kommunikationsmodulen, Readern) und führen eine Basiskonfigurierung aller Projektkomponenten durch.

Mithilfe der Ident-Anweisungen (Ident-Profil/-Bausteine) können Sie die Steuerung, an der die Ident-Geräte angeschlossen sind, programmieren. Bei der Programmierung können Sie den einzelnen, bereits projektierten Baugruppen konkrete und automatisierte Aufgaben zuweisen.

Optionen zur Projektierung und Programmierung abhängig von der TIA Portal-Version

Ab der TIA Portal-Version V13.1 sind die Ident-Anweisungen in STEP 7 integriert und Sie können Ihre Ident-Geräte manuell projektieren und mit Hilfe der Ident-Anweisungen programmieren. Ab der TIA Portal-Version V14 SP1 ist das Technologieobjekt "TO_Ident" in STEP 7 enthalten, das Sie bei der Projektierung, Konfiguration und Diagnose unterstützt. Die Programmierung erfolgt weiterhin über die Ident-Anweisungen. Ab der TIA Portal-Version V16 Update 1 ist das Technologieobjekt "TO_Taglayout" in STEP 7 enthalten. Mithilfe dieses Technologieobjekt können Sie den Speicherbereich von Transpondern in bis zu 64 Adressbereiche - sogenannte Tag-Felder - einteilen und diese symbolisch adressieren.

Das Technologieobjekt "TO_Ident" erleichtert Ihnen die Projektierung und Konfiguration, was sich auch auf die Programmierung auswirkt.

Auswirkungen:

- Die Variable vom Datentyp "IID_HW_CONNECT" muss nicht mehr manuell angelegt werden.
- Die verschiedenen Reset-Bausteine werden durch den Baustein "Reset_Reader" ersetzt.

Eine ausführliche Beschreibung zur Projektierung, Konfiguration und Diagnose der Ident-Geräte, sowie zur Erstellung der Tag-Felder mithilfe der Technologieobjekte, sowie zu den Parametern der Ident-Geräte finden Sie in der TIA Portal-Hilfe.

→ TIA Portal-Hilfe, suchen nach: Technologieobjekte "SIMATIC Ident"

→ TIA Portal-Hilfe, suchen nach: Identifikationssysteme

Eine Beschreibung zum Anlegen eines Technologieobjekts "SIMATIC Ident" finden Sie auf den Seiten des Siemens Industry Online Supports (<https://support.industry.siemens.com/cs/ww/de/view/109769780>).

Eine ausführliche Beschreibung zur Programmierung mithilfe der Ident-Anweisungen finden Sie in diesem Handbuch oder in der TIA Portal-Hilfe.

→ TIA Portal-Hilfe, unter: PLC programmieren > Anweisungen > Optionspakete > SIMATIC Ident

Tabelle 2- 1 Versions- und Integrationsstände der Bausteine und Bibliotheken

TIA-Integration	Baustein-Version des Ident-Profils	Bibliotheks-version	Version des TIA Portals	Parametrierung via ...
Externe Anweisungsbibliothek	V1.19	n/a	alle	Bausteine ("HW_CONNECT"-Variable)
Integrierte Anweisungsbibliothek	V2.0	V2.0	V13 SP1	
	V2.1	V3.0	V13 SP1, Update3	
	V3.0	V4.0	V14 SP1	Technologieobjekt
	V4.1	V5.1	V15	
	V4.2	V5.2	V15.1	
	V4.3	V5.3	V16	
	V4.3	V5.4	V16, Update 1	
V4.3	V5.5	V17		

Projektierbare Konfigurationen

Das Ident-Profil und die Ident-Bausteine können gleichermaßen in unterschiedlichen Konfigurationen betrieben werden.

Tabelle 2- 2 Über die Ident-Anweisungen projektierbare Konfigurationen

Ident-Geräte/ -Systeme	Bussysteme		Steuerungen		
	PROFIBUS	PROFINET	S7-300/-400	S7-1200/-1500	SIMOTION
ASM 456	✓	--	✓	✓	✓
RF120C	--	--	--	✓ ¹⁾	--
RF170C	✓	✓	✓	✓	✓
RF180C	--	✓	✓	✓	✓
RF166C	✓	--	✓	✓	✓
RF185C/RF186C/ RF188C	--	✓	✓	✓	✓
RF186CI/RF188CI					
RF360R	--	✓	✓	✓	✓
RF610R/RF615R/ RF680R/RF685R	✓ ²⁾	✓	✓	✓	✓
MV400/MV500	✓ ²⁾	✓	✓	✓	✓

1) nur S7-1200

2) über Kommunikationsmodul

Das Mischen dieser Konfigurationen und das Anschließen unterschiedlicher Kommunikationsmodule ist ebenso möglich.

Unterschied zwischen Ident-Bausteinen und Ident-Profil

Das Ident-Profil ist ein einziger komplexer Baustein, der alle Befehle und Funktionen für RFID-Systeme und optische Lesesysteme enthält. Die Ident-Bausteine stellen eine vereinfachte Schnittstelle des Ident-Profiles dar. Jeder Ident-Baustein enthält einen einzelnen Befehl des Ident-Profiles.

Grundsätzlich sollten Sie sich am Anfang entscheiden welche Bausteine Sie verwenden möchten - Ident-Profil oder Ident-Bausteine. Sie können allerdings nur eine von beiden Varianten nutzen. Das Mischen von Ident-Profil und Ident-Bausteinen pro Kanal ist nicht möglich!

Die folgende Tabelle gibt Ihnen einen Überblick über die Unterschiede.

Tabelle 2- 3 Unterschiede zwischen Ident-Bausteine und Ident-Profil

Ident-Bausteine	Ident-Profil
<ul style="list-style-type: none"> • Pro Befehl ein Baustein • Einfache Programmierung • Systemspezifische Bausteine 	<ul style="list-style-type: none"> • Voller Funktionsumfang in einem Baustein • Komplexe Programmierung
Unterstützter Befehlsumfang: <ul style="list-style-type: none"> • Reader-Status • Inventory • Tag-Status • Read • Write • Set-Ant • Write-ID • Reset-Reader • ... 	Unterstützter Befehlsumfang: Alle im Reader implementierten Befehle wie z. B. <ul style="list-style-type: none"> • Inventory • Physical-Read • Get-Blacklist • Match-String-Funktionen (nur bei MV) • ...
Unterstützter Funktionsumfang: <ul style="list-style-type: none"> • AdvancedCMD für verkettete Befehlsstrukturen (das Aufsetzen der einzelnen Befehle einer Kette ist identisch mit dem Ident-Profil) 	Unterstützter Funktionsumfang: <ul style="list-style-type: none"> • Repeat • Verkettung

Ausführliche Informationen zu den Bausteinen finden Sie in den Kapiteln "Ident-Bausteine programmieren (Seite 28)" und "Ident-Profil programmieren (Seite 67)".

Empfehlung zur Auswahl der Bausteine

Wenn die Ident-Bausteine Ihre Funktionsanforderungen abdecken, nutzen Sie diese. Die Ident-Bausteine sind einfacher zu programmieren und können meist ohne weitere Dokumentation parametrisiert werden. Das Programm wird leichter lesbar und die Programmierung ist mit weniger Aufwand verbunden. Beachten Sie, dass die Verwendung vieler Ident-Bausteine ggf. zu längeren Speicherzeiten führen können. In diesem Fall ist die Verwendung des Ident-Profiles sinnvoller.

Das Ident-Profil ist ein komplexer Baustein. Wir empfehlen ausschließlich geschulten Anwendern oder bei speziellen Anforderungen das Ident-Profil zu verwenden.

2.2 Bausteinfunktion

Die Programmbausteine dienen als Kommunikationsschnittstelle zwischen einem Ident-Gerät (z. B. RF186C) und dem Anwenderprogramm. Die Bausteine unterstützen dabei folgende Funktionen:

- Konfiguration
- Befehlsbearbeitung
- Lesen und Schreiben von Daten
- Diagnose

Das Ident-Profil ist ein einziger komplexer Baustein, der alle Befehle und Funktionen für RFID-Systeme und Optische Lesesysteme enthält. Die Ident-Bausteine stellen eine vereinfachte Schnittstelle des Ident-Profiles dar. Jeder Ident-Baustein enthält einen einzelnen Befehl des Ident-Profiles.

Die Größe der Datenpuffer "IDENT_DATA" (bei den Ident-Bausteinen), "TXREF" und "RXREF" (bei dem Ident-Profil) können variabel sein. Die Parameter sind unter S7-300 / S7-400 als "Any"-Pointer und unter S7-1200 / S7-1500 als "Variant" definiert.

Tabelle 2- 4 Unterschied

S7-300/-400 ("Any"-Pointer)	S7-1200/-1500 ("Variant")
"IDENT_DATA", "TXREF", "RXREF": Es können unterschiedlich lange Arrays jeden Typs, mitgelieferte Status-UDTs und selbst definierte UDTs angelegt werden.	"IDENT_DATA", "TXREF", "RXREF": Es können nur Arrays vom Typ Byte angelegt werden. Die Länge ist variabel. Ausnahme: Bei den Ident-Bausteinen "Reader-Status" und "Tag-Status" können an "IDENT_DATA" auch die mitgelieferten Status-Datentypen angelegt werden.

2.3 Abgrenzung zu anderen Programmbausteinen

Von Programmbausteinen unterstützte Funktionen

Die folgende Tabelle zeigt eine Übersicht der von den Programmbausteinen unterstützten Funktionen.

Tabelle 2- 5 Unterstützte Funktionen der Programmbausteine

Programm- baustein	von Programmbaustein unterstützte Funktionen							
	Single- tag	Multi- tag	Normal- adressie- rung	File- hand- ler	PROFIBU S	PROFINET	MV	
Ident-Profil	✓	✓	✓	--	✓	✓	✓	Baustein auf Grund- lage der PNO- Spezifikation (Basiert auf Normpro- fil V1.19, jedoch ohne Filehandler)
Ident- Bausteine	✓	✓	✓	--	✓	✓	✓	Auf dem Ident-Profil basierende Bausteine
Normprofil V1.19	✓	✓	✓	✓	✓	✓	✓	Empfohlener Baustein für Fremdsteuerun- gen
FB 45	✓	--	✓	--	✓	✓	✓	Baustein für Single- tag-Applikationen
FB 55	✓	✓	✓	--	✓	✓	--	Baustein für Multitag- Applikationen
FC 56	✓	✓	--	✓	✓	--	--	Empfohlener Baustein für Filehandler
FC 44	✓	--	✓	--	✓	--	--	Nur für RF160C
Applikati- ons- bausteine RF160C	✓	--	✓	--	✓	--	--	Nur für RF160C
Applikati- ons- bausteine IO-Link	✓	--	✓	--	✓	✓	--	Nur für RF200 IO-Link

Kompatible Programmbausteine

Die folgende Tabelle zeigt die zu den Anschaltmodulen/Kommunikationsmodulen kompatiblen Programmbausteine.

Tabelle 2- 6 Kompatible Programmbausteine

Ident-Gerät/- Systeme	kompatible Programmbausteine in Verbindung mit ...		
	S7-300 / S7-400 und STEP 7 Classic V5.5	S7-300 / S7-400 und STEP 7 Basic/Professional	S7-1200 / S7-1500 und STEP 7 Basic/Professional
ASM 456	FB 45 FB 55 FC 56 Normprofil V1.19 Ident-Profil	FB 45 FB 55 FC 56 Ident-Profil Ident-Bausteine	Ident-Profil Ident-Bausteine
ASM 475	FB 45 FB 55	FB 45 FB 55	--
RF120C	--	--	Ident-Profil Ident-Bausteine
RF160C	FC 44 Applikationsbausteine für RF160C	FC 44 Applikationsbausteine für RF160C	Applikationsbausteine für RF160C
RF166C	FB 45 Normprofil V1.19 Ident-Profil	FB 45 Ident-Profil Ident-Bausteine	Ident-Profil Ident-Bausteine
RF170C	FB 45 FB 55 Ident-Profil	FB 45 FB 55 Ident-Profil Ident-Bausteine	Ident-Profil Ident-Bausteine
RF180C	FB 45 FB 55 FC 56 Normprofil V1.19 Ident-Profil	FB 45 FB 55 FC 56 Ident-Profil Ident-Bausteine	Ident-Profil Ident-Bausteine
RF185C/RF186C/ RF188C	FB 45 Normprofil V1.19	FB 45 Ident-Profil	Ident-Profil Ident-Bausteine
RF186CI/RF188CI	Ident-Profil	Ident-Bausteine	
RF360R	Ident-Profil	Ident-Profil Ident-Bausteine	Ident-Profil Ident-Bausteine
RF615R/RF680R/ RF685R	Ident-Profil	Ident-Profil Ident-Bausteine	Ident-Profil Ident-Bausteine
MV400/MV500	FB 79 Normprofil V1.19 Ident-Profil	FB 79 Ident-Profil Ident-Bausteine	Ident-Profil

Bausteine parametrieren/programmieren

Die Ident-Anweisungen mit dem Ident-Profil und den Ident-Bausteinen sind ab der Version V13 SP1 in STEP 7 integriert. Über die "Projektnavigation" können Sie für Ihre Steuerungen "Programmbausteine" anlegen. Sie finden die Ident-Anweisungen/-Bausteine in dem Register "Anweisungen" unter "Optionale Pakete > SIMATIC Ident".

3.1 Übersicht der Ident-Anweisungen

Die nachfolgende Tabelle gibt Ihnen einen Überblick über die aktuell vorhandenen Anweisungen/Bausteine.

Tabelle 3- 1 Übersicht der Ident-Anweisungen

Einordnung	Baustein-Name	Beschreibung	
Anweisungen/ Bausteine	Basis-Bausteine	Read	Mit Hilfe dieser Bausteine können Sie die Kommunikation mit den Ident-Systemen einfach programmieren. Die Basis-Bausteine beinhalten all die Bausteine, die sehr häufig zur Anwendung kommen.
		Write	
		Reset_Reader	
		Read_MV	
		Set_MV_Program	
	Status-Bausteine	Reader_Status	Mit Hilfe der Status-Bausteine erhalten Sie Informationen zum Reader bzw. zum Transponder.
		Tag_Status	
	Tag-Feld-Bausteine	Read_Tagfield	Mit Hilfe der Bausteine können Sie symbolisch auf vordefinierte Tagfelder zugreifen.
		Write_Tagfield	
	Erweiterte Bausteine	Config_Download	Mit Hilfe dieser Bausteine können Sie die Kommunikation mit den Ident-Systemen einfach programmieren. Die erweiterten Bausteine stellen weitere, seltener benötigte Funktionen für die Bedienung des Ident-Systems zur Verfügung.
		Config_Upload	
		Inventory	
		Read_EPC_Mem	
		Read_TID	
		Read_UID	
		Set_ANT_RF300	
		Set_Param	
		Write_EPC_ID	
		Write_EPC_Mem	
AdvancedCmd	Erweiterter Befehlsvorrat. Mit dem Baustein "AdvancedCmd" ist es möglich, auf weitere Befehle aus dem Ident-Befehlsvorrat zuzugreifen und verkettete Befehle auszuführen.		

3.1 Übersicht der Ident-Anweisungen

Einordnung		Baustein-Name	Beschreibung	
		Ident_Profile	Für Experten stehen diese Bausteine zur Verfügung, um komplexe Befehlsstrukturen optimal in den eigenen Programmablauf einzubinden. Zusätzlich ist es möglich Repeat-Kommandos und Verkettung zu nutzen.	
	Reset-Bausteine	Reset_MOBY_D	Mit Hilfe der Reset-Bausteine können Sie die Ident-Systeme initialisieren, wenn der Baustein "Reset_Reader" nicht von dem betreffenden Ident-System unterstützt wird.	
	Reset_MOBY_U			
	Reset_MV			
	Reset_RF200			
	Reset_RF300			
Reset_Univ				
PLC-Datentypen	Systemdatentypen	IID_HW_CONNECT	Datentyp für alle Bausteine zur physikalischen Adressierung von Kommunikationsmodul und Reader und zum Synchronisieren der Bausteine die pro Reader verwendet werden.	
		IID_CMD_STRUCT	Datentyp für das Ident-Profil, zum Parametrieren der Befehle.	
	Statusdatentypen	IID_READSTAT_81_RF2_3_U	Datentypen für das Ergebnis von "Reader_Status" mit dem jeweiligen Attribut. Die Datentypen helfen dabei, die vom Reader empfangenen Daten, komfortabel zu interpretieren und direkt weiterzubearbeiten ohne Datentyp-Konvertierungen vorzunehmen.	
		IID_READSTAT_84_MOBY_U		
		IID_READSTAT_86_RF300		
		IID_READSTAT_EF_RF300G2		
		IID_READSTAT_89_RF68xR		
		IID_CM_STATUS_A2		
		IID_TAG_STATUS_04_RF300		Datentypen für das Ergebnis von "Tag_Status" mit dem jeweiligen Attribut. Die Datentypen helfen dabei, die vom Reader empfangenen Daten, komfortabel zu interpretieren und direkt weiterzubearbeiten ohne Datentyp-Konvertierungen vorzunehmen.
		IID_TAG_STATUS_80_MOBY_U		
		IID_TAG_STATUS_82_RF300		
	IID_TAG_STATUS_83_SO			
	IID_IN_I_80			
	IID_IN_I_81	Datentypen für das Ergebnis von "Inventory" mit dem jeweiligen Attribut. Die Datentypen helfen dabei, die vom Reader empfangenen Daten, komfortabel zu interpretieren und direkt weiterzubearbeiten ohne Datentyp-Konvertierungen vorzunehmen.		
	IID_INVENT_00_MOBY_U			

Hinweis**Parallelbetrieb von Ident-Bausteinen und Ident-Profil ist nicht möglich**

Beachten Sie, dass der Reader nicht zeitgleich über die Ident-Bausteine und über das Ident-Profil betrieben werden kann.

Von Ident-Geräten unterstützte Bausteine/Befehle

An den verschiedenen Kommunikationsmodulen können verschiedene Reader, sowie optische Lesesysteme angeschlossen werden. Diese unterstützen abhängig von ihrer Konfiguration unterschiedliche Bausteine und Befehle. Die Kommunikationsmodule RF18xC/RF18xCI und RF166C, sowie die Reader RF360R (ab FW V2.0) verfügen über ein zusätzliches Konfigurationsmodul "CM-Konfiguration_x", mit dem bestimmte Befehle zu dem Gerät selbst gesendet werden können.

Die folgende Tabelle gibt Ihnen einen Überblick darüber, welche Ident-Geräte (Kommunikationsmodule und eigenständige Reader / optische Lesesysteme) welche Bausteine und Befehle unterstützen. Diese wurden in sogenannte "Baustein- und Befehlssammlungen" zusammengefasst. Diese Baustein- und Befehlssammlungen sind nachfolgend beschrieben.

Tabelle 3- 2 Unterstützte Baustein- und Befehlssammlungen

Ident-Gerät	Baustein- und Befehlssammlung
ASM456	RF200/RF300
	RF61xR/RF68xR
	MV400/500
RF120C	RF200/RF300
	MV400/500
RF170C	RF200/RF300
	MV400/500
	Freeport
RF180C	RF200/RF300
	MV400/500
RF166C	RF200/RF300
	RF300 fast protocol
	RF61xR/RF68xR
	MV400/500
	Freeport
	CM-Konfiguration
RF18xC/RF18xCI	RF200/RF300
	RF300 fast protocol
	MV400/500
	Freeport
	CM-Konfiguration
RF360R	RF200/RF300

Ident-Gerät	Baustein- und Befehlssammlung
	RF300 fast protocol
	CM-Konfiguration (ab FW V2.0)
RF61xR/RF68xR	RF61xR/RF68xR
MV400/MV500	MV400/500

Tabelle 3- 3 Unterstützte Bausteine/Befehle der jeweiligen Baustein- und Befehlssammlung

Ident-Bausteine	Befehle des Ident-Profiles
RF200/RF300	
<ul style="list-style-type: none"> • Read • Write • Reader_Status • Tag_Status • Read_Tagfield • Write_Tagfield • AdvancedCmd • Read_UID • Set_ANT_RF300 • Reset_Reader (Technologieobjekt "TO_Ident") • Reset_MOBY_D • Reset_MOBY_U • Reset_RF200 • Reset_RF300 • Reset_Univ 	<ul style="list-style-type: none"> • PHYSICAL_READ • PHYSICAL_WRITE • WRITE_CONFIG • READER-STATUS • TAG-STATUS • FORMAT • PUT
RF300 fast protocol	
<ul style="list-style-type: none"> • Read • Write • Reader_Status • Tag_Status • Read_Tagfield • Write_Tagfield • AdvancedCmd • Read_UID • Set_Ant_RF300 • Reset_Reader (Technologieobjekt "TO_Ident") • Reset_RF300 • Reset_Univ 	<ul style="list-style-type: none"> • PHYSICAL_READ • PHYSICAL_WRITE • WRITE_CONFIG • READER-STATUS • TAG-STATUS • FORMAT • PUT

Ident-Bausteine	Befehle des Ident-Profiles
RF61xR/RF68xR	
<ul style="list-style-type: none"> • Read • Write • Reader_Status • Read_Tagfield • Write_Tagfield • AdvancedCmd • Config_Upload • Config_Download • Inventory • Read_EPC_Mem • Read_TID • Set_Param • Write_EPC_ID • Write_EPC_Mem • Reset_Reader 	<ul style="list-style-type: none"> • PHYSICAL_READ • PHYSICAL_WRITE • WRITE_CONFIG • READER-STATUS • INVENTORY • WRITE-ID • KILL-TAG • LOCK-TAG-BANK • EDIT-BLACKLIST • GET-BLACKLIST • READ-CONFIG
Freeport	
<ul style="list-style-type: none"> • Read • Write • Reset_Reader (Technologieobjekt "TO_Ident") • Reset_Univ 	<ul style="list-style-type: none"> • PHYSICAL_READ • PHYSICAL_WRITE • WRITE_CONFIG
CM-Konfiguration	
<ul style="list-style-type: none"> • Reader_Status • Config_Upload • Config_Download • Reset_Reader 	<ul style="list-style-type: none"> • PHYSICAL_READ • PHYSICAL_WRITE • WRITE_CONFIG • READ-CONFIG
MV400/MV500	
<ul style="list-style-type: none"> • Read • Read_MV • Write • Set_MV_Program • Reset_Reader (Technologieobjekt "TO_Ident") • Reset_MV • Reset_Univ 	<ul style="list-style-type: none"> • PHYSICAL_READ • PHYSICAL_WRITE • WRITE_CONFIG

3.2 Parametrieren/programmieren

Sie können die Ident-Bausteine / das Ident-Profil entweder mit oder ohne Hilfe des Technologieobjekts "TO_Ident" parametrieren. Die Parametrierung mit Hilfe des Ident-Profiles bzw. der Ident-Bausteine in Verbindung mit dem Technologieobjekt "SIMATIC Ident > TO_Ident" werden in der TIA Portal-Hilfe ab V14 SP1 beschrieben. Wenn Sie die Bausteine ohne das Technologieobjekt parametrieren, dann benötigen Sie den Datentyp "IID_HW_CONNECT". Beachten Sie, dass dieser Datentyp ausschließlich in Bibliotheksversionen < V5.0 enthalten ist. Die Steuerungen SIMATIC S7-300/-400 sind nicht mit dem Technologieobjekt kompatibel.

Bevor Sie mit dem Parametrieren der Bausteine beginnen können, müssen Sie zuerst eine Variable des PLC-Datentyps "IID_HW_CONNECT" anlegen. Mit Hilfe des PLC-Datentyps "IID_HW_CONNECT" wird das Ident-System bzw. ein Kanal des Ident-Systems adressiert.

Adressierung der Ident-Geräte

Bei der Arbeit mit allen Anweisungen/Bausteinen, benötigen Sie den Datentyp "IID_HW_CONNECT" zur Reader-Adressierung. Die Befehlsparametrierung am Ident-Profil wird durch die Ident-Bausteine übernommen. Das Ident-Profil sowie der Baustein "AdvancedCMD" benötigen zusätzlich den Datentyp "IID_CMD_STRUCT" zur Parametrierung der einzelnen Befehle. Abhängig davon, ob Sie mit dem Ident-Profil oder den Ident-Bausteinen arbeiten, müssen Sie diese Datentypen wie in den folgenden Kapiteln beschrieben einbinden und parametrieren.

Parametrieren einer Variablen vom Datentyp "IID_HW_CONNECT"

Gehen Sie folgendermaßen vor, um eine Variable vom Datentyp "IID_HW_CONNECT" für einen Kanal zu parametrieren:

1. Legen sie eine neue Variable in einem Datenbaustein oder im statischen Bereich eines Funktionsbausteins an.

Reader_1								
	Name	Datentyp	Startwert	Remanenz	Erreichbar a...	Sichtbar i...	Einstellwert	Kommentar
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	connect	*IID_HW_CONNECT*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	HW_ID	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	only S7-1200/1500: HW identifier
4	CM_CHANNEL	Int	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	channel of communication module
5	LADDR	DWord	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	io address
6	Static	*IID_IN_SYNC*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Bild 3-1 Datenbaustein anlegen

2. Geben Sie die Adress-Daten des Ident-Gerätes bzw. des Kanals an.
 - HW_ID: Hardware-Kennung des Moduls (nur bei S7-1200 und S7-1500)
 - CM_CHANNEL: Kanal innerhalb eines Moduls
 - LADDR: E/A-Adresse des Moduls

Die Werte der Parameter "HW_ID" und "LADDR" können Sie in der Gerätekonfiguration in den Eigenschaften des Kommunikationsmoduls/Readers auslesen. Tragen Sie die ausgelesenen Parameterwerte in der Spalte "Startwert" der zugehörigen Parameter ein. Das Auslesen der Parameterwerte wird im Folgenden beschrieben.

Gehen Sie folgendermaßen vor, um die Parameterwerte "HW_ID" und "LADDR" für einen Kanal auszulesen:

1. Öffnen Sie die Netzsicht und doppelklicken Sie auf das Kommunikationsmodul.

Reaktion: TIA wechselt in die Gerätesicht und das Eigenschaftfenster des CMS wird geöffnet.

2. Markieren Sie in dem Register "Geräteübersicht" das entsprechende Modul.

Die in dem Register angezeigte E/A-Adresse, entspricht der "LADDR".

Beachten Sie, dass die Eingangs- und Ausgangsadresse den gleichen Wert haben müssen.

3. In dem Register "Eigenschaften > Systemkonstanten" finden Sie die HW-Kennung, welche der "HW_ID" entspricht.

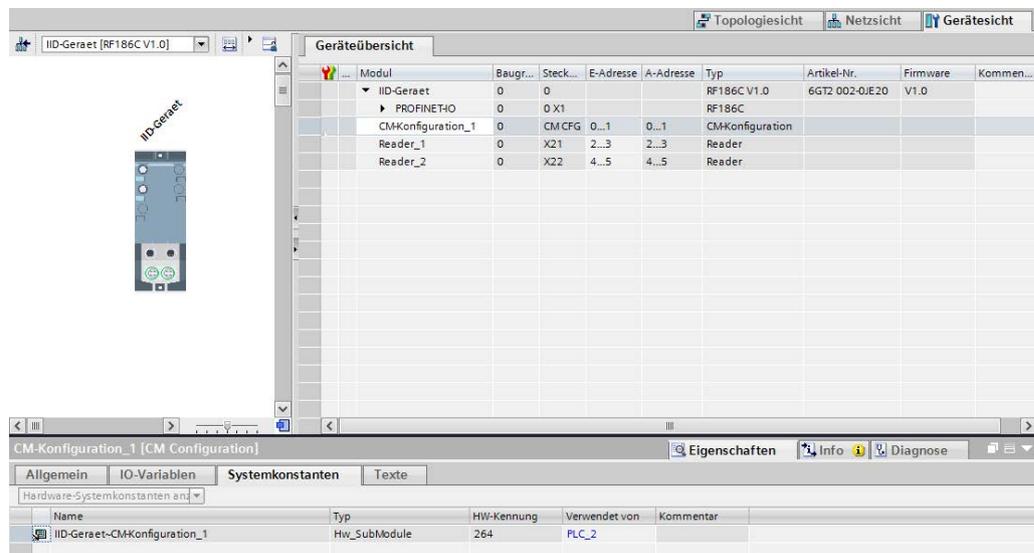


Bild 3-2 Der Parameter "HW-Kennung"

4. Übertragen Sie die Werte von "LADDR" und "HW_ID" in den PLC-Datentyp "IID_HW_CONNECT" des zu parametrierenden Readers.

Hinweis

User Mode einstellen

Beachten Sie, dass Sie in den Eigenschaften des Kommunikationsmoduls/Readers, dem Parameter "User Mode" den Wert "RFID-Normprofil" zuweisen und den passenden MOBY Mode auswählen.

Bei allen anderen Kommunikationsmodulen/Readern finden Sie die beiden Parameter direkt in den Eigenschaften der Baugruppe.

Der Datentyp "IID_HW_CONNECT" ist nun für einen Kanal angelegt. Für jeden weiteren Reader/Kanal wiederholen Sie diesen Vorgang. Wenn Sie einen anderen Kanal des Readers/CMs benutzen wollen, stellen Sie dies über den Parameter "CM_CHANNEL" ein. Die Parameter "HW_ID" und "LADDR" bleiben bei allen Kanälen/Readern/Antennen gleich. Bei einem Kommunikationsmodul RF166C oder RF18xC/RF18xCI ist jeder Reader einem eigenen Modul zugeordnet und hat eine eigene "HW_ID" und "LADDR". Beachten Sie, dass der Wert des Kanals immer "1" betragen muss.

Alle notwendigen Funktionsbausteine und Datentypen sind nun in Ihrem Projekt eingebunden und sie können nun mit dem Programmieren beginnen. Der Datentyp "IID_HW_CONNECT" ist ebenfalls angelegt und adressiert. Nun können Sie mit dem Programmieren der Bausteine beginnen.

Hinweis

"IID_CMD_STRUCT" projektieren

Wenn Sie mit dem Ident-Profil oder mit dem Baustein "AdvancedCmd" arbeiten, müssen Sie zusätzlich in dem bereits angelegten Datenbaustein ein weiteres Element mit dem Datentyp "IID_CMD_STRUCT" (Array [1...n]) anlegen.

Baustein anlegen

Bevor Daten mit einem Reader ausgetauscht werden können (z. B. Lesen/Schreiben von Daten) muss das betreffende Ident-Gerät einmalig zurückgesetzt werden. Bei der Verwendung der Ident-Bausteine können Sie dafür einen Reset-Baustein (z. B. "Reset-Reader") verwenden. Bei der Verwendung des Ident-Profils muss ein entsprechend konfigurierter "WRITE-CONFIG"-Befehl ausgeführt werden.

Voraussetzung

Der Datentyp "IID_HW_CONNECT" wurde parametriert.

Gehen Sie folgendermaßen vor, um einen Baustein einzubinden und den Aufruf zu parametrieren:

1. Öffnen Sie den von Ihnen erstellten Programmbaustein, durch einen Doppelklick im Register "Projektnavigation > Programmbausteine".
2. Ziehen Sie den gewünschten Baustein aus dem Anweisungsregister in den Programmbaustein.
3. Geben Sie am Eingangsparameter "HW_CONNECT" die Variable an, die Sie zuvor angelegt haben.

Der Baustein wird aufgerufen und mit dem entsprechenden Kanal verbunden.

Hinweis

Arbeit mit mehreren Kanälen/Lesestellen

Wenn Sie mit mehreren Kanälen/Lesestellen arbeiten, müssen sie darauf achten, dass für jeden Kanal / jede Lesestelle der Baustein mit einem eigenen Instanz-DB aufgerufen wird.

Hinweis

Arbeiten mit dem Ident-Profil oder mit dem Baustein "AdvancedCmd"

Wenn Sie mit dem Ident-Profil oder mit dem Baustein "AdvancedCmd" arbeiten, müssen Sie zusätzlich den Eingangsparameter "CMDREF" mit einer Variablen des Datentyps "IID_CMD_STRUCT" (Array [1...n]) verbinden.

3.3 Allgemeiner Aufbau der Funktionsbausteine

Aufbau der Bausteine anhand des Beispiel-Bausteins "FB"

Die nachfolgende Grafik zeigt einen Beispiel-Baustein mit Eingangs- und Ausgangsparametern wie sie bei allen Bausteinen in der gleichen Weise vorhanden sind.

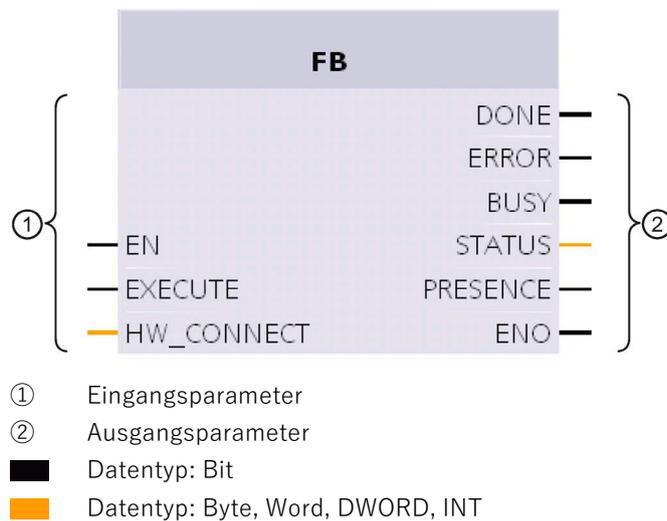


Bild 3-3 Beispiel-Baustein

Beschreibung der Parameter

Tabelle 3- 4 Beschreibung der Eingangs- und Ausgangsparametern

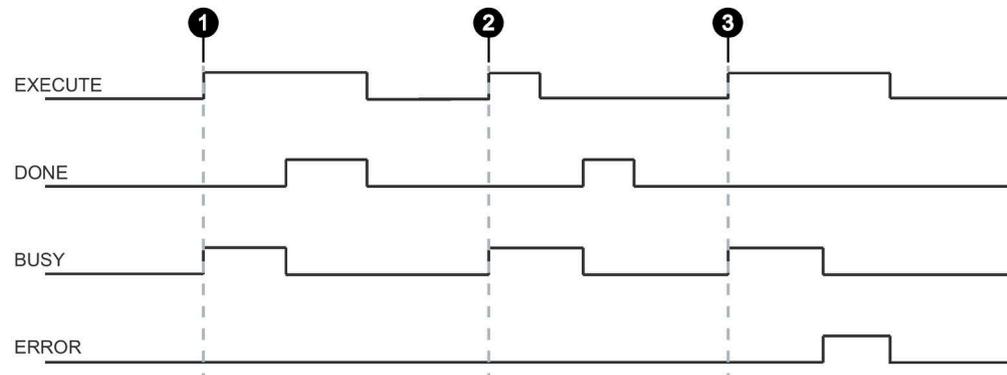
Parameter	Beschreibung
Eingangsparameter	
EN	Freigabeeingang
EXECUTE	An diesen Eingang muss eine positive Flanke angelegt werden, damit der Baustein den Befehl ausführt.
HW_CONNECT	Globaler Parameter vom Typ "IID_HW_CONNECT" um den Kanal/Reader zu adressieren und die Bausteine zu synchronisieren. Dieser Parameter muss einmal für jeden Kanal/Reader erstellt und adressiert werden. "HW_CONNECT" muss immer an die Bausteine übergeben werden, um den entsprechenden Kanal/Reader zu adressieren.
Ausgangsparameter	
DONE (BOOL)	Der Auftrag wurde abgearbeitet. Bei einem positiven Ergebnis wird dieser Parameter gesetzt.
ERROR (BOOL)	Der Auftrag wurde mit einem Fehler beendet. Der Fehlercode wird im Parameter "STATUS" angezeigt.
BUSY (BOOL)	Der Auftrag wird gerade ausgeführt.
STATUS (DWORD)	Anzeige der Fehlermeldung, wenn das "ERROR"-Bit gesetzt wurde.
PRESENCE (BOOL)	Über dieses Bit wird die Anwesenheit eines Transponders angezeigt. Der angezeigte Wert wird bei jedem Aufruf des Bausteins aktualisiert. Dieser Parameter kommt nicht bei Optischen Lesesystem spezifischen Bausteinen vor. Hinweis zu Freeport protocol: Wurde noch kein "Read" aufgesetzt, so wird durch das Anwesenheitsbit ("TP") mitgeteilt, dass neue Daten empfangen wurden und im Kommunikationsmodul zur Abholung durch einen "Read"-Befehl bereit stehen.
ENO	Freigabeausgang

Allgemeiner Ablauf beim Aufruf der Bausteine

Hinweis

Unterschiedliche Abläufe bei Ident-Profil und Normprofil V1.19

Beachten Sie, dass der Ablauf des Ident-Profiles nicht dem des Vorgänger Normprofils V1.19 entspricht. Das Normprofil V1.19 ist nicht kompatibel mit den Readern RF600.



- Fall ① Durch Setzen von EXECUTE (EXECUTE = 1) wird die Funktion/Anweisung gestartet. Wurde der Auftrag erfolgreich beendet (DONE = 1), müssen Sie EXECUTE wieder zurücksetzen. DONE wird zeitgleich zurückgesetzt.
- Fall ② EXECUTE wird nur für einen Zyklus gesetzt. Sobald BUSY gesetzt ist können Sie EXECUTE wieder zurücksetzen. Wurde der Auftrag erfolgreich beendet, wird DONE für einen Zyklus gesetzt.
- Fall ③ Handhabung wie Fall 1, jedoch mit Fehlerausgabe. Sobald ERROR gesetzt ist, steht in der Ausgabe STATUS der genaue Fehlercode. ERROR und STATUS behalten ihren Wert solange bei, wie EXECUTE gesetzt ist bzw. für einen Zyklus, wenn EXECUTE vor Beendigung des Bausteins zurückgesetzt wurde.

Bild 3-4 Allgemeiner Ablauf beim Aufruf der Bausteine

Funktionsweise der Bausteine

Sie können immer nur einen Befehl bzw. eine Befehlskette an ein Ident-Gerät je Kanal bzw. Lesestelle versenden. Sie können jedoch zwei oder mehrere Bausteine gleichzeitig starten. Die Bausteine werden dann in der Reihenfolge des Aufrufs abgearbeitet.

Dies gilt nicht bei den Reset-Bausteinen. Wird ein Reset-Befehl ausgeführt, wird der zu diesem Zeitpunkt aktive Befehl abgebrochen.

Beachten Sie, dass Ident-Bausteine immer komplett durchlaufen werden müssen. Wird die Bearbeitung abgebrochen, während ein Baustein abgearbeitet wird, dann müssen Sie nach dem Abbruch diesen Baustein erst beenden, bevor weitere Bausteine gestartet werden können.

3.4 Ident-Bausteine programmieren

3.4.1 Basis-Bausteine

3.4.1.1 Read

Der Baustein "Read" liest die Anwenderdaten vom Transponder aus und stellt diese im Puffer "IDENT_DATA" bereit. Die physikalische Adresse und die Länge der Daten, wird über die Parameter "ADDR_TAG" und "LEN_DATA" übergeben. Bei den Readern RF61xR/RF68xR liest der Baustein die Daten aus der Memory-Bank 3 (USER-Bereich) aus. Der eindeutige Zugriff auf einen bestimmten Transponder erfolgt über die optionalen Parameter "EPCID_UID" und "LEN_ID".

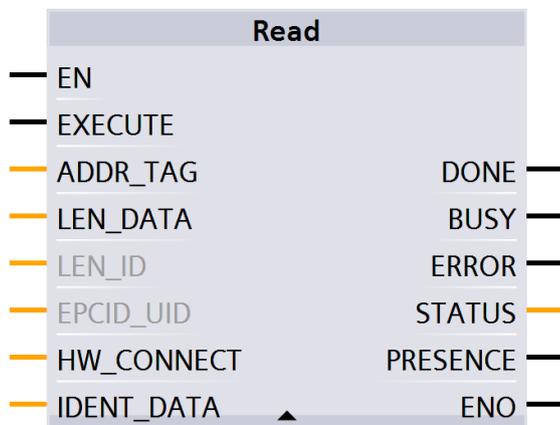


Bild 3-5 Baustein "Read"

Tabelle 3- 5 Erläuterung zum Baustein "Read"

Parameter	Datentyp	Default-Werte	Beschreibung
ADDR_TAG	DWord	DW#16#0	Physikalische Adresse auf dem Transponder ab der gelesen wird. Weitere Informationen zur Adressierung finden Sie im Kapitel "Transponder-Adressierung (Seite 109)". Bei MV: Ab Adresse "0" steht die Länge des gelesenen Codes (2 Byte). Ab Adresse "2" steht der gelesene Code. ¹⁾
LEN_DATA	Word	W#16#0	Länge der zu lesenden Daten
LEN_ID	Byte	B#16#0	Länge der EPC-ID/UID Default-Wert: 0x00 = unspezifizierter Singletag-Zugriff (RF200, RF300, RF61xR, RF68xR)

Parameter	Datentyp	Default-Werte	Beschreibung
EPCID_UID	Array[1...62] of Byte	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
IDENT_DATA	Any / Variant	0x00	Datenpuffer in dem die gelesenen Daten abgelegt werden. Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

¹⁾ Weitere Informationen zur Arbeit mit optischen Lesesystemen, finden Sie in den Betriebsanleitungen "SIMATIC MV420 / SIMATIC MV440" und "SIMATIC MV500".

3.4.1.2 Read_MV

Der Baustein "Read_MV" liest das Leseergebnis eines optischen Lesegeräts aus. Zum Auslesen der Parametrierung muss der "Read"-Baustein verwendet werden. Die Länge der zu lesenden Daten ermittelt der Baustein automatisch aus der Länge des angelegten Empfangspuffers. Die tatsächliche Länge des Leseergebnisses wird in dem Ausgangsparameter "LEN_DATA" ausgegeben. Die Daten werden im Datenpuffer "IDENT_DATA" abgelegt. Ist der Puffer zu klein, erscheint die Fehlermeldung "0xE7FE0400" und die zu erwartende Länge wird an "LEN_DATA" ausgegeben.

Um eine optimale Geschwindigkeit zu erreichen, empfehlen wir Ihnen, die Länge des Datentyps "IDENT_DATA" so anzupassen, dass diese möglichst nah der maximal erwarteten Länge des Leseergebnisses (2 Byte Codelänge + gelesener Code) entspricht.

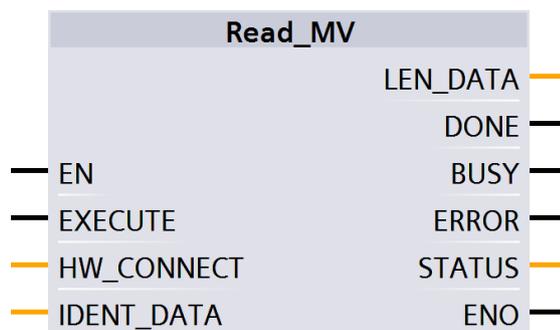


Bild 3-6 Baustein "Read_MV"

Tabelle 3- 6 Erläuterung zum Baustein "Read_MV"

Parameter	Datentyp	Default-Werte	Beschreibung
IDENT_DATA	Any / Variant	0x00	Leseergebnis In Byte 0 und 1 steht die Länge des gelesenen Codes. Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.
LEN_DATA	Word	W#16#0	Länge des Leseergebnis ≈ 2 Byte Codelänge + gelesener Code

3.4.1.3 Reset_Reader

Mit Hilfe des Bausteins "Reset_Reader" können Sie alle Reader-Typen der Siemens RFID-Systeme, sowie die optischen Lesegeräte zurücksetzen. Bei Verwendung des Technologieobjekts "SIMATIC Ident" können Sie den Baustein für alle Ident-Geräte verwenden, die an eine S7-1200/-1500 angeschlossen sind. Wenn Sie nicht mit dem Technologieobjekt arbeiten, dann können Sie den Baustein ausschließlich für die Geräte SIMATIC RF120C und SIMATIC RF61xR/RF68xR verwenden. Der Baustein "Reset_Reader" besitzt keine gerätespezifischen Parameter und wird über den Parameter "EXECUTE" ausgeführt.

Beschreibungen weiterer Reset-Bausteine für den Betrieb mit den Kommunikationsmodulen oder optischen Lesesystemen finden Sie im Kapitel "Reset-Bausteine (Seite 61)".

Mit dem Baustein "Reset_Reader" und den anderen Reset-Bausteinen können Sie jederzeit jeden laufenden Ident-Baustein unterbrechen. Diese Bausteine werden dann mit "DONE = true" und "ERROR = false" beendet.

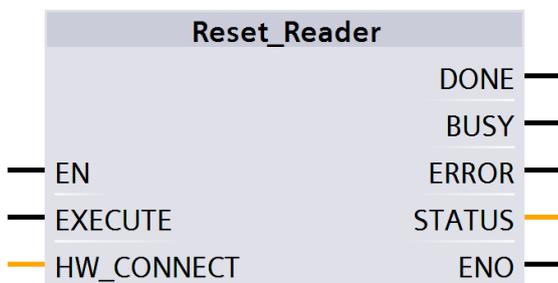


Bild 3-7 Baustein "Reset_Reader"

3.4.1.4 Set_MV_Program

Mit Hilfe des Bausteins "Set_MV_Program" können Sie das Programm in einer Kamera wechseln. Über den Parameter "PROGRAM" wird die gewünschte Programmnummer übergeben.

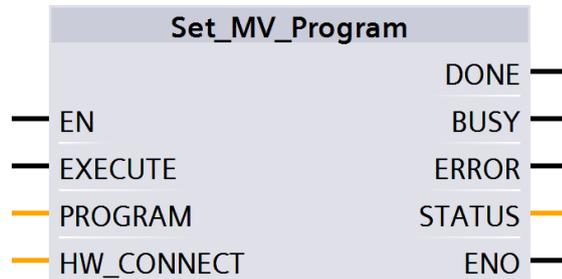


Bild 3-8 Baustein "Set_MV_Program"

Tabelle 3- 7 Erläuterung zum Baustein "Set_MV_Program"

Parameter	Datentyp	Default-Werte	Beschreibung
PROGRAM	Byte	B#16#1	Programmnummer Wertebereich: 0x01 ... 0x0F

3.4.1.5 Write

Der Baustein "Write" schreibt die Anwenderdaten aus dem Puffer "IDENT_DATA" auf den Transponder. Die physikalische Adresse und die Länge der Daten, wird über die Parameter "ADDR_TAG" und "LEN_DATA" übergeben. Bei den Readern RF61xR/RF68xR schreibt der Baustein die Daten in die Memory-Bank 3 (USER-Bereich). Der eindeutige Zugriff auf einen bestimmten Transponder erfolgt über die optionalen Parameter "EPCID_UID" und "LEN_ID".

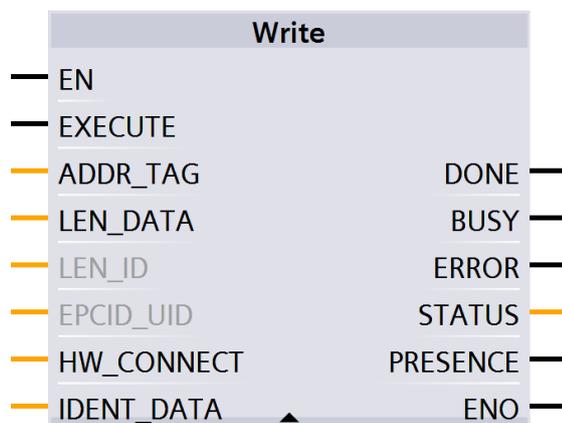


Bild 3-9 Baustein "Write"

Tabelle 3- 8 Erläuterung zum Baustein "Write"

Parameter	Datentyp	Default-Werte	Beschreibung
ADDR_TAG	DWord	DW#16#0	Physikalische Adresse auf dem Transponder ab der geschrieben wird. Weitere Informationen zur Adressierung finden Sie im Kapitel "Transponder-Adressierung (Seite 109)". Bei MV: Adresse ist immer 0. ¹⁾
LEN_DATA	Word	W#16#0	Länge der zu schreibenden Daten
LEN_ID	Byte	B#16#0	Länge der EPC-ID/UID Default-Wert: 0x00 = un spezifizierter Singletag-Zugriff (RF200, RF300, RF61xR, RF68xR)
EPCID_UID	Array[1...62] of Byte	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
IDENT_DATA	Any / Variant	0x00	Datenpuffer mit den zu schreiben den Daten. Bei MV: Erstes Byte codiert den entsprechenden MV-Befehl. ¹⁾ Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

¹⁾ Weitere Informationen zur Arbeit mit optischen Lesesystemen, finden Sie in den Betriebsanleitungen "SIMATIC MV420 / SIMATIC MV440" und "SIMATIC MV500".

3.4.2 Status-Bausteine

3.4.2.1 Reader_Status

Der Baustein "Reader_Status" liest Statusinformationen aus dem Reader bzw. Kommunikationsmodul (RF18xC/RF18xCI, RF166C sowie die Reader RF360R (ab FW V2.0) mit dem Modul "CM-Konfiguration_1") aus. Für die verschiedenen Reader-Familien gibt es unterschiedliche Status-Modi, die Sie über den Parameter "ATTRIBUTE" auswählen können.

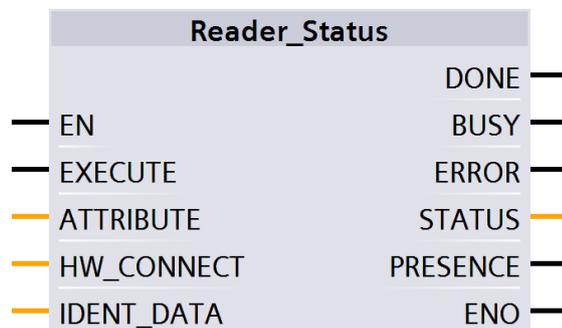


Bild 3-10 Baustein "Reader_Status"

Tabelle 3-9 Erläuterung zum Baustein "Reader_Status"

Parameter	Datentyp	Default-Werte	Beschreibung
ATTRIBUTE	Byte	B#16#81	Kennung der Status-Modi / Mögliche Eingaben: <ul style="list-style-type: none"> • RF200: 0x81 • RF300: 0x81, 0x86, 0xEF • RF61xR, RF68xR: 0x89 • MOBY U: 0x81, 0x84, 0x85 • MOBY D: 0x81
IDENT_DATA	Any / Variant	0x00	Ergebniswerte abhängig von ATTRIBUTE Hinweis: Sie können entweder einen "Array_of_Byte" oder eine Variable mit einem der Datentypen aus den nachfolgenden Tabellen anlegen.

Ergebnisse

Legen Sie den jeweils richtigen Datentyp, der dem ATTRIBUTE-Wert zugeordnet ist, am Eingang "IDENT_DATA" des Bausteins an, damit die Daten richtig interpretiert werden können.

Beachten Sie, dass die UDTs nur zusammen mit den Bausteinen "Reader_Status" oder "Tag_Status" verwendet werden können.

Tabelle 3- 10 ATTRIBUTE "0x81" (Datentyp "IID_READSTAT_81_RF2_3_U")

Name	Typ	Kommentar
status info	BYTE	Reader status mode
hardware	CHAR	Type of hardware
hardware_version	WORD	Version of hardware 0 = RF310R, RF340R, RF350R; 1 = RF380R, RF260R; 2 = RF310R (ISO), RF210R, RF220R; 3 = RF380R (ISO), RF240R; 4 = RF340R (ISO), RF350R (ISO), RF250R; 5 = RF310R (ISO); 8 = MV400/MV500; A = RF310R Gen2, RF290R; B = RF340R Gen2; C = RF350R Gen2; D = RF380R Gen2; G = RF360R; M = RF280R RS422; N = RF280R RS232
loader version	WORD	Version of loader
firmware	CHAR	Type of firmware
firmware version HB	BYTE	Version of firmware
firmware version LB	BYTE	Version of firmware
driver	CHAR	Type of driver
driver version	WORD	Version of driver
interface	BYTE	Type of interface 0x01 = RS422; 0x02 = RS232
baud	BYTE	Baudrate 0x01 = 19.2 kBd; 0x03 = 57.6 kBd; 0x05 = 115.2 kBd; 0x0D = 921 kBd
reserved1	BYTE	Reserved
reserved2	BYTE	Reserved
reserved3	BYTE	Reserved
distance limiting SLG	BYTE	Distance limiting of Reader
multitag SLG	BYTE	Multitag Reader
field ON control SLG	BYTE	Field ON control
field ON time SLG	BYTE	Field On time
sync SLG	BYTE	Synchronization with Reader
status_ant	BYTE	Status of antenne 0x01 = on; 0x02 = off
stand by	BYTE	Time of standby after command
MDS control	BYTE	Presence mode

Tabelle 3- 11 ATTRIBUTE "0x86" (Datentyp "IID_READSTAT_86_RF300")

Name	Typ	Kommentar
status info	BYTE	Reader status mode
FZP	BYTE	Error counter passive: distortion without communication
ABZ	BYTE	Dropout counter
CFZ	BYTE	Code error counter
SFZ	BYTE	Signature error counter
CRCFZ	BYTE	CRC-error counter
BSTAT	BYTE	Status of last command
ASMFZ	BYTE	Error counter for host interface (ASM)
reserved0	ARRAY [1..17]	Reserved

Tabelle 3- 12 ATTRIBUTE "0x6F" (Datentyp "IID_READSTAT_EF_RF300G2")

Name	Typ	Kommentar
status info	SINT	Reader status mode
hardware	SINT	Type of hardware
hardware_version	INT	Version of hardware 0 = RF310R, RF340R, RF350R; 1 = RF380R, RF260R; 2 = RF310R (ISO), RF210R, RF220R; 3 = RF380R (ISO), RF240R; 4 = RF340R (ISO), RF350R (ISO), RF250R; 5 = RF310R (ISO); A = RF310R Gen2, RF290R; B = RF340R Gen2; C = RF350R Gen2; G = RF360R; M = RF280R RS422; N = RF280R RS232
reserved	INT	Reserved
loader	SINT	Type of loader
loader version 1	SINT	Version of loader
loader version 2	SINT	Version of loader
loader version 3	SINT	Version of loader
loader version 4	SINT	Version of loader
firmware	SINT	Type of firmware
firmware version 1	SINT	Version of firmware
firmware version 2	SINT	Version of firmware
firmware version 3	SINT	Version of firmware
firmware version 4	SINT	Version of firmware
reserved1	DINT	Reserved
driver	SINT	Type of driver
driver version 1	SINT	Version of driver
driver version 2	SINT	Version of driver
reserved2	INT	Reserved

Tabelle 3- 13 ATTRIBUTE "0xA2" (Datentyp "IID_CM_STATUS_A2") kompatibel mit RF18xC/RF18xCI, RF166C und RF360R (ab FW V2.0)

Name	Typ	Kommentar
Status info	BYTE	CM device status
hardware version	BYTE	Version of hardware
firmware_version	ARRAY[1..4] of CHAR	Version of firmware
config ID	DWORD	Config ID as unix timestamp
serial_number	ARRAY[1..16] of CHAR	Serial number
device_type	BYTE	Type of device 0x00 = RF185C; 0x01 = RF186C; 0x02 = RF188C; 0x03 = RF186CI; 0x04 = RF188CI; 0x06 = RF166C; 0x07 = RF360R
upload request	BYTE	Upload request for configuration data
active_connections	BYTE	Active connections 0x00 = PLC; 0x01 = OPC UA; 0x02 = XML
io_mode	BYTE	Mode of in-/outputs 0x00 = Disabled; 0x01 = Single Input; 0x02 = Single Output; 0x03 = Support of IO link devices/hubs
low_voltage	BYTE	Low voltage warning 0x00 = OK; 0x01 = low voltage
reserved0	BYTE	Reserved
reserved1	BYTE	Reserved
number serial interfaces	BYTE	Number of serial interfaces supported by CM
serial_interface_data	ARRAY[1..4] of IID IN SERIAL INTERFACE	

3.4 Ident-Bausteine programmieren

Name	Typ	Kommentar
serial_interface_data[1]	IID IN SERIAL INTERFACE	
connection_status	BYTE	Status of communication between reader and CM 0x00 = down; 0x01 = up
reader_type	CHAR	Type of connected reader 0 = RF310R, RF340R, RF350R; 1 = RF380R, RF260R; 2 = RF310R (ISO), RF210R, RF220R; 3 = RF380R (ISO), RF240R; 4 = RF340R (ISO), RF350R (ISO), RF250R; 5 = RF310R (ISO); 8 = MV400/MV500; A = RF310R Gen2, RF290R; B = RF340R Gen2; C = RF350R Gen2; D = RF380R Gen2; M = RF280R RS422; N = RF280R RS232
hardware_error	BYTE	Hardware error
reserved	BYTE	Reserved
serial_interface_data[2]	IID IN SERIAL INTERFACE	
serial_interface_data[3]	IID IN SERIAL INTERFACE	
serial_interface_data[4]	IID IN SERIAL INTERFACE	

Tabelle 3- 14 ATTRIBUTE "0x89" (Datentyp "IID_READSTAT_89_RF68xR") kompatibel mit RF61xR/RF68xR

Name	Typ	Kommentar
status info	BYTE	Reader status mode (Subcommand)
hardware version	BYTE	Version of hardware
firmware version	ARRAY[1..4] of CHAR	Version of firmware
config ID	DWORD	Unix timestamp
inventory_status	WORD	0 = inventory not active; 1 = inventory active; 2 = presence mode active
sum of filtered tags	WORD	All filtered Tags
filtered smoothing	WORD	Filtered Tags through Smoothing
filtered blacklist	WORD	Filtered Tags through Blacklist
filtered data-filter	WORD	Filtered Tags through Data-Filter
filtered RSSI threshold	WORD	Filtered Tags through RSSI Threshold
filtered RSSI delta	WORD	Filtered Tags through RSSI Delta

Tabelle 3- 15 ATTRIBUTE "0x84" (Datentyp "IID_READSTAT_84_MOBY_U")

Name	Typ	Kommentar
status info	BYTE	Reader status mode
number MDS	BYTE	Range 1..24
UID	ARRAY [1..24] of DWord	

Werte des Parameters "ATTRIBUTE"

Weitere Details zu den einzelnen Status-Modi entnehmen Sie bitte den zu den Modi passenden Handbüchern "FB 45" und "FB 55".

Die hier angegebenen Kennungen der Status-Modi entsprechen folgenden Kennungen in den anderen Handbüchern:

0x81	≐	0x01
0x82	≐	0x02
0x83	≐	0x03
0x85	≐	0x05
0x87	≐	0x07
0x88	≐	0x08
0x90	≐	0x10
0x91	≐	0x11
0x92	≐	0x12
0xA0	≐	0x20
0xA1	≐	0x21

3.4.2.2

Tag_Status

Der Baustein "Tag_Status" liest die Statusinformationen des Transponders aus. Für die verschiedenen Transponder-Typen und Reader-Familien gibt es unterschiedliche Status-Modi, die Sie über den Parameter "ATTRIBUTE" auswählen können.

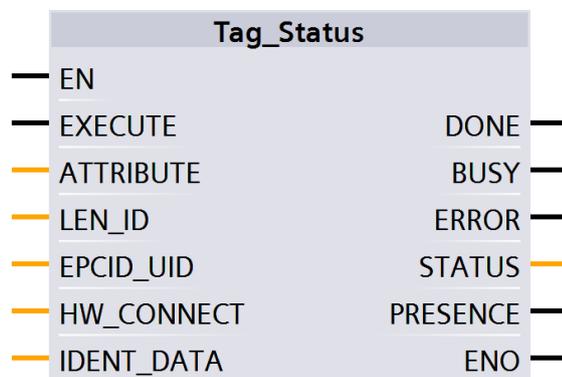


Bild 3-11 Baustein "Tag_Status"

Tabelle 3- 16 Erläuterung zum Baustein "Tag_Status"

Parameter	Datentyp	Default-Werte	Beschreibung
ATTRIBUTE	Byte	B#16#0	Kennung der Status-Modi / Mögliche Eingaben: <ul style="list-style-type: none"> RF200: 0x83 RF300 (RF300T): 0x04, 0x82 RF300 (ISO): 0x83 RF300 (Gen2): 0x83 MOBY D: 0x83 ¹⁾ MOBY U: 0x80
LEN_ID	Byte	B#16#0	Länge der EPC-ID / UID Default-Wert: 0x00 = un spezifizierter Singletag-Zugriff (RF61xR, RF68xR)
EPCID_UID	Array[1...62] of Byte	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
IDENT_DATA	Any / Variant	0x00	Ergebniswerte je nach Attributen Hinweis: Sie können entweder einen "Array_of_Byte" oder eine Variable mit einem der Datentypen aus den nachfolgenden Tabellen anlegen.

¹⁾ nur SLG D10S

Ergebnisse

Beachten Sie, dass die UDTs erst dann angezeigt werden, wenn die Bausteine "Reader_Status" oder "Tag_Status" verwendet werden.

Tabelle 3- 17 ATTRIBUTE "0x04" (Datentyp "IID_TAG_STATUS_04_RF300")

Name	Typ	Kommentar
reserved	BYTE	
status info	BYTE	Tag status mode
UID	ARRAY [1..8] of BYTE	
MDS_type	BYTE	Type of Tag 0x01 = Tag without FRAM; 0x02 = Tag with FRAM 8K; 0x03 = Tag with FRAM 32K; 0x04 = Tag with FRAM 64K; 0x05 = Tag with FRAM 128K; 0x06 = Tag with FRAM 256K

Name	Typ	Kommentar
Lock state	BYTE	Write Protection Status EEPROM
Reserved1	ARRAY [1..6] of BYTE	

Tabelle 3- 18 ATTRIBUTE "0x82" (Datentyp "IID_TAG_STATUS_82_RF300")

Name	Typ	Kommentar
reserved	BYTE	
status info	BYTE	Tag status mode
UID	ARRAY [1..8] of BYTE	
LFD	BYTE	Magnetic flux density: correlation between limit-value
FZP	BYTE	Error counter passive: distortion without communication
FZA	BYTE	Error counter active: distortion during communication
ANWZ	BYTE	Presence counter: measure value for presence time
reserved1	ARRAY [1..3] of BYTE	

Tabelle 3- 19 ATTRIBUTE "0x83" (Datentyp "IID_TAG_STATUS_83_ISO")

Name	Typ	Kommentar
reserved	BYTE	
status info	BYTE	Tag status mode
UID	ARRAY [1..8] of BYTE	
MDS_Type	BYTE	Type of Tag 0x00 = unknown; 0x03 = ISO: MDS D3xx, Infineon; 0x04 = ISO: MDS D4xx, Fujitsu; 0x05 = ISO: MDS D1xx, NXP; 0x06 = ISO: MDS D2xx, TI; 0x07 = ISO: MDS D261, STM; 0x08 = ISO: MDS D5xx, Fujitsu - 8k; 0x11 = RF300 - 0k; 0x12 = RF300 - 8k; 0x13 = RF300 - 32k; 0x14 = RF300 - 64k; 0x21 = MIFARE Classic - 1k, NXP; 0x22 = MIFARE Classic 1k, In- fineon; 0x23 = MIFARE Classic 4k, NXP; 0x41 (A) = P2P RF310R Gen2; 0x42 (B) = P2P RF340R Gen2; 0x43 (C) = P2P RF350R Gen2; 0x44 (D) = P2P RF380R Gen2
IC version	BYTE	Chip version
size HB	BYTE	Size of Memory (high Byte)
size LB	BYTE	Size of memory (low Byte)
lock state	BYTE	Write protection status EEPROM
block size	BYTE	Size of a block in addressable memory
number of block	BYTE	Number of blocks in addressable memory

Tabelle 3- 20 ATTRIBUTE "0x80" (Datentyp "IID_TAG_STATUS_80_MOBY_U")

Name	Typ	Kommentar
UID	ARRAY [1..4] of BYTE	Unique identifier (MDS-Number)
MDS type	BYTE	Type of Tag
sum subframe access 1	BYTE	Sum of subframe access Byte 1
sum subframe access 2	BYTE	Sum of subframe access Byte 2
sum subframe access 3	BYTE	Sum of subframe access Byte 3
sum subframe access 4	BYTE	Sum of subframe access Byte 4

Name	Typ	Kommentar
sum_searchmode_access_1	BYTE	Sum of search mode access Byte 1
sum_searchmode_access_2	BYTE	Sum of search mode access Byte 2
ST_date_Week	BYTE	Date of last sleep-time change (week of year)
ST_date_Year	BYTE	Date of last sleep-time change (year)
battery_left_1	BYTE	Battery power left (percent) Byte 1
battery_left_2	BYTE	Battery power left (percent) Byte 2
ST	BYTE	Actual sleep-time on MDS

Werte des Parameters "ATTRIBUTE"

Weitere Details zu den einzelnen Status-Modi entnehmen Sie bitte den zu den Modi passenden Handbüchern "FB 45" und "FB 55".

Die hier angegebenen Kennungen der Status-Modi entsprechen folgenden Kennungen in den anderen Handbüchern:

- 0x04 ≙ 0x01
- 0x82 ≙ 0x02
- 0x83 ≙ 0x03
- 0x84 ≙ 0x04
- 0x85 ≙ 0x05

3.4.3 Tag-Feld-Bausteine

Voraussetzung für die Verwendung der nachfolgenden Tag-Feld-Befehle ist, dass Tag-Felder mit Hilfe des Technologieobjekts "TO_TagLayout" projiziert wurden.

3.4.3.1 Read_Tagfield

Der Baustein "Read_Tagfield" liest Daten aus einem Tag-Feld des Transponders aus und stellt dieses formatiert am Parameter "IDENT_DATA" bereit. Am Parameter "TAGLAYOUT" wird ein Technologieobjekt "TO_TagLayout" übergeben und über den Parameter "TAGFIELD" wird das jeweilige Tag-Feld ausgewählt.

Das Technologieobjekt "TO_TagLayout" erzeugt einen Datenbaustein "<Name des TO>_Indexes", womit am Parameter "TAGFIELD" symbolisch ein Tag-Feld ausgewählt werden kann. Zusätzlich wird ein Datentyp "<Name des TO>_Datatypes" zur Verfügung gestellt, mit dem eine Variable erstellt werden kann, welche alle Tag-Felder mit den entsprechenden Datentypen enthält. Hiermit kann symbolisch ein Speicherbereich für das Tag-Feld am Parameter "IDENT_DATA" adressiert werden.

Die Variable am Parameter "IDENT_DATA" muss dem Datentyp des Tag-Feldes entsprechen. Ein Variable vom Typ "Array of Byte" ist immer erlaubt. Der eindeutige Zugriff auf einen bestimmten Transponder erfolgt über die optionalen Parameter "EPCID_UID" und "LEN_ID".

Weiterführende Informationen zu dem Technologieobjekt "TO_TagLayout" findet Sie unter "Technologieobjekt "TO_TagLayout"" in der TIA Portal-Hilfe.

Tabelle 3- 21 Erläuterung zum Baustein "Read_Tagfield"

Parameter	Deklaration	Datentyp	Default-Werte	Beschreibung
TAGFIELD	Input	INT	1	Auswahl eines Tag-Feldes Der Wert entspricht der Nummer des Tag-Feldes im Technologieobjekt "TO_TagLayout". Für die symbolische Adressierung kann ein Element des erzeugten Datenbausteins "<Name des TO>_Indexes" verwendet werden.
LEN_ID	Input	BYTE	B#16#0	Länge der EPC-ID/UID Default-Wert: 0x00 = unspezifizierter Singletag-Zugriff (RF200, RF300, RF61xR, RF68xR)
EPCID_UID	Input	ARRAY[1...62] OF BYTE	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird an Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird an Anfang des Puffers eingetragen ("LEN_ID = 8")
TAGLAYOUT	In/Out	TO_TAGLAYOUT	--	Technologieobjekt "TO_TagLayout"
HW_CONNECT	In/Out	TO_IDENT	--	Technologieobjekt "TO_Ident" des Ident-Geräts
IDENT_DATA	In/Out	VARIANT	0x00	Datenpuffer in dem die gelesenen Daten abgelegt werden. Variable vom Datentyp des Tag-Feldes oder "ARRAY OF BYTE".

3.4.3.2 Write_Tagfield

Der Baustein "Write_Tagfield" schreibt Daten aus dem Parameter "IDENT_DATA" in ein Tag-Feld des Transponders. Am Parameter "TAGLAYOUT" wird ein Technologieobjekt "TO_TagLayout" übergeben und über den Parameter "TAGFIELD" wird das jeweilige Tag-Feld ausgewählt.

Das Technologieobjekt "TO_TagLayout" erzeugt einen Datenbaustein "<Name des TO>_Indexes", womit am Parameter "TAGFIELD" symbolisch ein Tag-Feld ausgewählt werden kann. Zusätzlich wird ein Datentyp "<Name des TO>_Datatypes" zur Verfügung gestellt mit dem eine Variable erstellt werden kann, welche alle Tag-Felder mit den entsprechenden Datentypen enthält. Hiermit kann symbolisch ein Speicherbereich für das Tag-Feld am Parameter "IDENT_DATA" adressiert werden.

Die Variable am Parameter "IDENT_DATA" muss dem Datentyp des Tag-Feldes entsprechen. Ein Variable vom Typ "Array of Byte" ist immer erlaubt. Der eindeutige Zugriff auf einen bestimmten Transponder erfolgt über die optionalen Parameter "EPCID_UID" und "LEN_ID".

Weiterführende Informationen zu dem Technologieobjekt "TO_TagLayout" findet Sie unter "Technologieobjekt "TO_TagLayout"" in der TIA Portal-Hilfe.

Tabelle 3- 22 Erläuterung zum Baustein "Write_Tagfield"

Parameter	Deklaration	Datentyp	Default-Werte	Beschreibung
TAGFIELD	Input	INT	1	Auswahl eines Tag-Feldes Der Wert entspricht der Nummer des Tag-Feldes im Technologieobjekt "TO_TagLayout". Für die symbolische Adressierung kann ein Element des erzeugten Datenbausteins "<Name des TO>_Indexes" verwendet werden.
LEN_DATA	Input	WORD	W#16#0	Länge der zu schreibenden Daten
LEN_ID	Input	BYTE	B#16#0	Länge der EPC-ID/UID Default-Wert: 0x00 = unspezifizierter Singletag-Zugriff (RF200, RF300, RF61xR, RF68xR)
EPCID_UID	Input	ARRAY[1...62] OF BYTE	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird an Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird an Anfang des Puffers eingetragen ("LEN_ID = 8")
TAGLAYOUT	In/Out	TO_TAGLAYO UT	--	Technologieobjekt "TO_TagLayout"
HW_CONNEC T	In/Out	TO_IDENT	--	Technologieobjekt "TO_Ident" des Ident-Geräts
IDENT_DATA	In/Out	VARIANT	0x00	Datenpuffer mit den zu schreibenden Daten. Variable vom Datentyp des Tag-Feldes oder "ARRAY OF BYTE".

3.4.4 Erweiterte Bausteine

3.4.4.1 AdvancedCmd

Mit dem Baustein "AdvancedCmd" kann jeder Befehl ausgeführt werden, auch Befehle die durch die anderen Bausteine nicht dargestellt werden. Dieser allgemeine Aufbau kann für alle Befehle verwendet werden und ist ausschließlich für geschulte Anwender konzipiert.

Dieser Baustein ermöglicht es Ihnen optional eine Reihe von verketteten Befehlen mit nur einem einzigen Aufruf des Bausteins auszuführen. Dafür stellt der Baustein einen CMD-Puffer für 100 Befehle zur Verfügung. Alle verketteten Befehle müssen ab dem ersten Platz im Puffer eingetragen werden. Zusätzlich muss für jeden verketteten Befehl das "Chained-Bit" in der CMD-Struktur gesetzt werden. Ausschließlich bei dem letzten Befehl in der Kette wird das "Chained-Bit" nicht gesetzt. Weitere Informationen zum "Chained-Bit" finden Sie im Kapitel "Verkettung (Seite 101)".

Im Eingangsparameter "CMD" muss die komplette Befehlsstruktur angegeben werden. Die Struktur für den Parameter "CMD" müssen Sie in einem Datenbaustein anlegen.

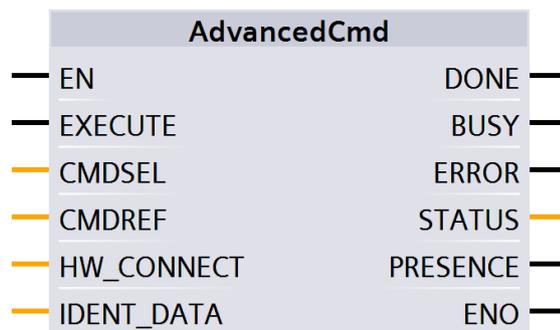


Bild 3-12 Baustein "AdvancedCmd"

Tabelle 3- 23 Erläuterung zum Baustein "AdvancedCmd"

Parameter	Datentyp	Default-Werte	Beschreibung
CMDSEL	Int	1	Auswahl des auszuführenden Befehls "CMDREF"; 1 ⇒ 1. Befehl, ... Der Wert des Parameters "CMDSEL" kann nie > 100 sein.
CMDREF	Any / Variant	--	Eine ausführliche Beschreibung des Parameters finden Sie in den Kapiteln: <ul style="list-style-type: none"> "Befehle des Ident-Profiles (Seite 73)" "Befehlsstruktur (Seite 74)"
IDENT_DATA	Any / Variant	0x00	Puffer für zu schreibende bzw. gelesene Daten. Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

3.4.4.2 Config_Upload/-_Download

Mithilfe der Bausteine "Config_Upload" und "Config_Download" können Sie über das Steuerungsprogramm die Konfiguration der an dieser Steuerung angeschlossenen Kommunikationsmodule RF166C sowie RF18xC/RF18xCI und der Reader RF360R (ab FW V2.0) sowie RF61xR/RF68xR auslesen ("Config_Upload") oder schreiben ("Config_Download").

Die Konfigurationsdaten sind nicht interpretierbare Daten. Speichern Sie die Daten in der Steuerung, um im Falle eines Gerätetauschs diese wieder auf den Reader zu schreiben. Bei gleicher Konfiguration mehrerer Reader müssen Sie die Konfigurationsdaten nur einmal pro Reader-Typ sichern. Eine einzelne Sicherung kann beim Download auf alle gleich konfigurierten Reader kopiert werden.

Die Bytes 6-9 (siehe nachfolgende Tabelle) beinhalten eine Konfigurations-ID (CONFIG_ID) mit einer eindeutigen Versionskennung. Mit der Konfigurations-ID können Sie überprüfen, ob bei einem "Config_Upload" die gelesenen Konfigurationsdaten mit den in der Steuerung gespeicherten Konfigurationsdaten übereinstimmen. Außerdem können Sie mit Hilfe des Geräte-Status die Seriennummer überprüfen, um festzustellen, ob das Gerät getauscht wurde.

Die Konfigurationsdaten sind wie folgt aufgebaut:

Tabelle 3- 24 Aufbau der Konfigurationsdaten

Byte	Name
0	Strukturkennung (2 Byte)
2	Längenangabe (4 Byte) Länge von Versionskennung und Parameterblock

Byte	Name
6	<p>Versionskennung (≙ CONFIG_ID; 4 Byte)</p> <p>Mit Hilfe der Kennung können Sie die Konfiguration eindeutig identifizieren. Es handelt sich dabei um einen Zeitstempel im Linux-Format.</p> <p>Der Zeitstempel gibt an, wie viele Sekunden seit dem 1. Januar 1979, 00:00 Uhr vergangen sind. Die Kennung wird beim Erzeugen einer Konfiguration vergeben.</p>
10 ... Ende "DATA"	Parameterblock

Der "Config_Upload/Config_Download" kann an jedem Kanal des RF61xR/RF68xR durchgeführt werden. Es werden immer dieselben Konfigurationsdaten übertragen. Die Konfigurationsdaten der Reader RF360R sowie der Kommunikationsmodule RF166C sowie RF18xC/RF18xCI können ausschließlich über das Modul "CM-Konfiguration_1" übertragen werden.

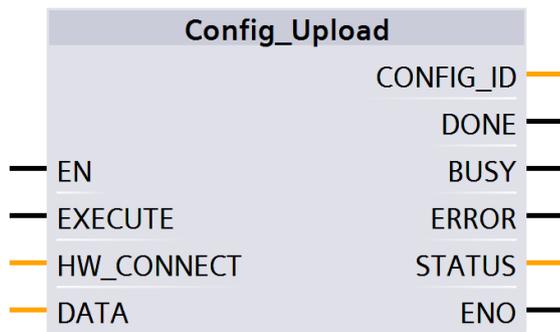


Bild 3-13 Baustein "Config_Upload"

Tabelle 3- 25 Erläuterung zum Baustein "Config_Upload"

Parameter	Datentyp	Default-Wert	Beschreibung
DATA	Any / Variant	--	Datenpuffer für Konfigurationsdaten. Die reale Länge der Daten hängt von der Komplexität der Projektierung und dem Firmware-Ausgabestand des Readers ab. Bei einer Standardprojektierung empfehlen wir eine Speichergröße von 4 KB. Verwenden Sie erweiterte Reader-Projektierungen (Filterungen) oder möchten Sie in Zukunft die Projektierung ändern ohne die Speichergröße "DATA" anpassen zu müssen, empfehlen wir Ihnen eine Speichergröße von 8-16 KB. Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.
CONFIG_ID	DWord	0x00	Versionskennung (4 Byte) Mit Hilfe der Kennung können Sie die Konfiguration eindeutig identifizieren. Es handelt sich dabei um einen Zeitstempel im Linux-Format. Der Zeitstempel gibt an, wie viele Sekunden seit dem 1. Januar 1979, 00:00 Uhr vergangen sind. Die Kennung wird beim Erzeugen einer Konfiguration vergeben.

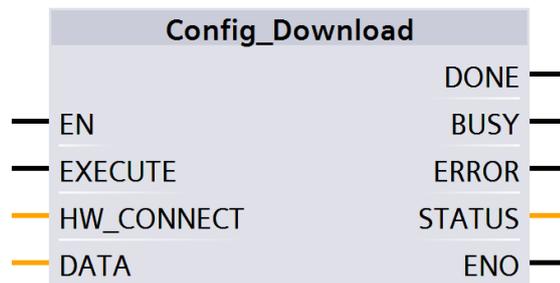


Bild 3-14 Baustein "Config_Download"

Tabelle 3- 26 Erläuterung zum Baustein "Config_Download"

Parameter	Datentyp	Default-Wert	Beschreibung
DATA	Any / Variant	--	<p>Datenpuffer für Konfigurationsdaten. Die reale Länge der Daten hängt von der Komplexität der Projektierung und dem Firmware-Ausgabestand des Readers ab. Bei einer Standardprojektierung empfehlen wir eine Speichergröße von 4 KB. Verwenden Sie erweiterte Reader-Projektierungen (Filterungen) oder möchten Sie in Zukunft die Projektierung ändern ohne die Speichergröße "DATA" anpassen zu müssen, empfehlen wir Ihnen eine Speichergröße von 8-16 KB.</p> <p>Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.</p>

3.4.4.3 Inventory

Der Baustein "Inventory" aktiviert die Ausführung von Inventories bei RF600-Readern (RF61xR/RF68xR). Bei einem einzelnen Inventory werden Erfassungszyklen für alle Antennen mit jeweils allen Polarisierungen abgefragt. Sind an einem Reader z. B. 2 Antennen mit jeweils 3 Polarisierungen angeschlossen, dann beinhaltet ein Inventory 6 Erfassungszyklen.

Beachten Sie, dass die Länge des Datenpuffers ("IDENT_DATA") mindestens der Länge der maximal zu erwartenden Daten entsprechen muss. Werden mehr Transponder erkannt und Daten ausgelesen, als in der zugewiesenen Pufferlänge von "IDENT_DATA" Platz haben, gehen die Daten dieser Transponder verloren. Dieses Verhalten wird durch den Fehler "0xE7FE0400" (Pufferüberlauf) angezeigt.

Mit Hilfe der Parameter "DURATION" und "DUR_UNIT" können Sie die Dauer der Inventories festlegen.

Es gibt vier verschiedene Modi, welche Sie über den Parameter "ATTRIBUTE" auswählen können.

- Beim Start wird eine bestimmte Dauer/Anzahl (Zeitdauer, Anzahl Inventories, Anzahl "Observed"-Events bzw. erkannte Transponder) angegeben. Dabei wird zwischen folgenden drei Optionen unterschieden:
 - Dauer
Inventories über eine festgelegte Zeitdauer durchführen
 - Anzahl Inventories
Eine festgelegte Anzahl an Inventories durchführen
 - Anzahl "Observed"-Events
Solange Inventories durchführen, bis eine festgelegte Anzahl Transponder gleichzeitig erkannt wurde.

Für diese Dauer/Anzahl werden dann Inventories vom Reader durchgeführt. Ist die angegebene Zeit/Anzahl erreicht, wird der Baustein beendet und er liefert alle erkannten Transponder in "IDENT_DATA" zurück. D. h. weitere Befehle können erst dann ausgeführt werden, wenn alle Inventories vollständig durchgeführt wurden. Die Einheit (Zeit oder Anzahl) wird über "DUR_UNIT" und der Wert (Zeitwert oder Anzahl) wird über "DURATION" angegeben. Dieser Modus kann über die Attribute "0x80" und "0x81" ausgeführt werden. Bei dem jeweiligen Attribut werden mehr oder weniger Daten über die erkannten Transponder geliefert.

- Mit den Attributen "0x86" ("Presence_Mode" starten) und "0x87" ("Presence_Mode" beenden) können Inventories dauerhaft ausgeführt werden. Die Anwesenheit eines Transponders kann dann immer über "PRESENCE" abgefragt werden, ohne dass der Baustein mit "EXECUTE" gestartet werden muss. Es werden keine Informationen über die erkannten Transponder bei Ausführung des Befehls zurückgeliefert!

Um Informationen über die erkannten Transponder zu erhalten, führen Sie einen der ersten beiden oben genannten Aufrufe durch (mit Zeit / Anzahl Inventories = 0).

Wenn dieser Modus aktiv ist, werden Transponder betreffende Befehle nicht sofort ausgeführt, sondern erst, wenn ein Transponder erfasst wird. Dadurch werden kürzere Reaktionszeiten ermöglicht, da der Befehl bereits ansteht, wenn der Transponder ins Antennenfeld kommt.

Der "Presence_Mode" ist sinnvoll im Rahmen der "Repeat Command"-Funktion.

An dem Ausgangsparameter "NUMBER_TAGS" wird die Anzahl der erkannten Transponder ausgegeben. Bei den Attributen "0x80" und "0x81" wird bei Abschluss des Lesevorgangs die Summe aller erkannten Transponder angezeigt. Bei dem Attribut "0x86" wird an dem Ausgangsparameter "NUMBER_TAGS" immer die Anzahl der aktuell erkannten Transponder angezeigt (max. 15), ohne dass der Baustein mit "EXECUTE" gestartet werden muss.

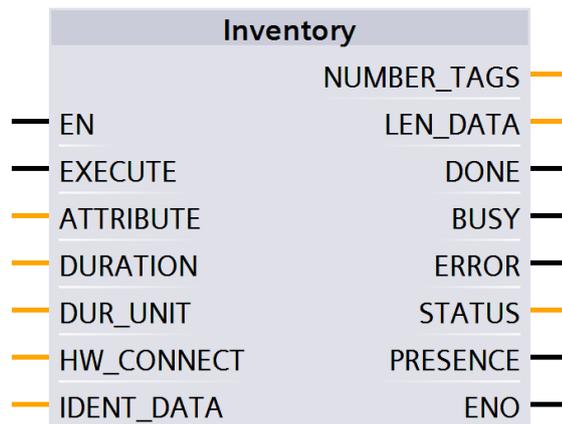


Bild 3-15 Baustein "Inventory"

Tabelle 3- 27 Erläuterung zum Baustein "Inventory"

Parameter	Datentyp	Default-Werte	Beschreibung
ATTRIBUTE	Byte	B#16#0	Auswahl des Status-Modus: <ul style="list-style-type: none"> • MOBY U: 0x00 ¹⁾ • RF600: 0x80, 0x81, 0x86, 0x87
DURATION	Word	W#16#0	Zeitdauer abhängig von "DUR_UNIT" Zeitdauer oder Anzahl Inventories oder Anzahl der "Observed"-Events Bsp.: <ul style="list-style-type: none"> • 0x00 ≙ kein Inventory • 0x01 ≙ ein Inventory
DUR_UNIT	Word	W#16#0	Einheit für "DURATION" <ul style="list-style-type: none"> • 0x00 ≙ Zeit [ms] • 0x01 ≙ Inventories • 0x02 ≙ Anzahl der Transponder die den Zustand "Observed" erreichen
IDENT_DATA	Any / Variant	0x00	Datenpuffer für Inventory-Daten Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.
NUMBER_TAGS	Int	0x00	Anzahl der Transponder im Antennenfeld
LEN_DATA	Word	W#16#0	Länge der gültigen Daten Während der Befehlsbearbeitung = 0

¹⁾ Ausschließlich im Multitag-Betrieb

Ergebnis-Anzeige

Die nachfolgend genannten Datentypen werden im TIA Portal bei einem neu erstellten Projekt nicht in der Auswahlliste der Datentypen angezeigt. Um diese Datentypen verwenden zu können, müssen Sie den Namen des Datentyps (z. B. "IID_IN_I_81") per Hand eintragen. Nach der ersten Verwendung wird der Datentyp automatisch im Projekt und auch in der Auswahlliste aufgeführt.

Ergebnisse für RF600

Die Anzahl der Elemente "TAG_DATA[x]" der Datentypen der ATTRIBUTE "0x80" und "0x81" ist abhängig von der Anzahl der zu erwartenden Transponder. Aus diesem Grund müssen Sie sich den Empfangspuffer selbst zusammenstellen. Beachten Sie bei der Erstellung des Empfangspuffer "IDENT_DATA"/Datentyps folgenden Aufbau:

- Das erste Element "NUM_MDS" ist immer vom Typ "WORD".
- Das darauffolgende Element "TAG_DATA" ist immer vom Typ "ARRAY". In dem "ARRAY" müssen Sie die Anzahl der zu erwartenden Transponder eintragen ("n").

Beachten Sie, dass die erstellten Datentypen bei den Steuerungen S7-1200/S7-1500 nicht direkt an dem Parameter "IDENT_DATA" angelegt werden können. Informationen zu diesem Thema finden Sie auf den Seiten des Siemens Industry Online Supports (<https://support.industry.siemens.com/cs/ww/de/view/109752738>).

Die nachfolgenden Tabellen zeigen beispielhaft den Aufbau des Empfangspuffer "IDENT_DATA"/Datentyp für die ATTRIBUTE "0x80" und "0x81". Da die ATTRIBUTE "0x86" und "0x87" keine Daten zurückliefern, wird deren Aufbau nicht weiter ausgeführt.

Tabelle 3- 28 ATTRIBUTE "0x80"

Name	Typ	Kommentar
NUM_MDS	WORD	Number of MDS
TAG_DATA	ARRAY[1..n] of IID_IN_I_80	Length of EPC ID
TAG_DATA[1]	IID_IN_I_80	
Reserved	BYTE	
ID_Len	BYTE	Length of EPC ID
EPC_ID	ARRAY[1..62] of BYTE	EPC-ID
tagPC	WORD	
TAG_DATA[2]	IID_IN_I_80	
...	...	
TAG_DATA[n]	IID_IN_I_80	

Tabelle 3- 29 ATTRIBUTE "0x81"

Name	Typ	Kommentar
NUM_MDS	WORD	Number of MDS
TAG_DATA	ARRAY[1..n] of IID_IN_I_81	
TAG_DATA[1]	IID_IN_1_81	
reserved	BYTE	
ID_LEN	BYTE	EPC length
EPC_ID	ARRAY[1..62] of BYTE	EPC-ID
tagPC	WORD	
RSSI	BYTE	RSSI value
MaxRSSI	BYTE	highest RSSI value
MinRSSI	BYTE	lowest RSSI value
channel	BYTE	channel; 1..15_ESTI; 1..53:FCC
antenna	BYTE	antenna; bit coded; Bit 0 = antenna 1; Bit 1 = antenna 2; Bit 2 = antenna 3; Bit 3 = antenna 4
polarization	BYTE	polarization of antenna; 0=undefined; 1=circular; 2=vertical linear; 3=horizontal
time	Time_OF_Day	S7 timestamp
power	BYTE	power in dBm
filterDataAvailable	BYTE	0=false; 1=true
Inventoried	WORD	1)
TAG_DATA[2]	IID_IN_1_81	
...	...	
TAG_DATA[n]	IID_IN_1_81	

1) Angabe, wie oft der Transponder über die Luftschnittstelle erkannt wurde, bis er in den Zustand "Observed" wechselt.

Ergebnisse für MOBY U

Tabelle 3- 30 ATTRIBUTE "0x00" (Datentyp "IID_INVENT_00_MOBY_U")

Name	Typ	Kommentar
number MDS	WORD	Number of MDS
UID length	WORD	length of UID
UID	ARRAY[1..12] of IID_IN_I_8BYTE	
UID[1]	IID_IN_I_8BYTE	
UID	ARRAY[1..8] of BYTE	
UID[1]	BYTE	
UID[2]	BYTE	
UID[3]	BYTE	
UID[4]	BYTE	
UID[5]	BYTE	
UID[6]	BYTE	

Name	Typ	Kommentar
UID[7]	BYTE	
UID[8]	BYTE	
UID[2]	"IID_IN_I_8BYTE"	
UID[3]	"IID_IN_I_8BYTE"	
UID[4]	"IID_IN_I_8BYTE"	
UID[5]	"IID_IN_I_8BYTE"	
UID[6]	"IID_IN_I_8BYTE"	
UID[7]	"IID_IN_I_8BYTE"	
UID[8]	"IID_IN_I_8BYTE"	
UID[9]	"IID_IN_I_8BYTE"	
UID[10]	"IID_IN_I_8BYTE"	
UID[11]	"IID_IN_I_8BYTE"	
UID[12]	"IID_IN_I_8BYTE"	

3.4.4.4 Read_EPC_Mem

Der Baustein "Read_EPC_Mem" liest Daten ab Adresse 4 aus dem EPC-Speicher des RF600-Transponders aus. Der Zugriff erfolgt auf Bank 1 ab Startadresse 4. Über den Parameter "LEN_DATA" wird die auszulesende Länge des EPC-Speichers angegeben.

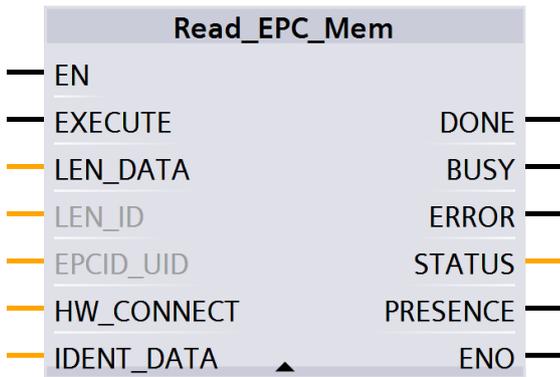


Bild 3-16 Baustein "Read_EPC_Mem"

Tabelle 3- 31 Erläuterung zum Baustein "Read_EPC_Mem"

Parameter	Datentyp	Default-Werte	Beschreibung
LEN_DATA	Word	W#16#0	Länge des auszulesenden EPC-Speichers (1 ... 62 Byte)
LEN_ID	Byte	B#16#0	Länge der EPC-ID / UID Default-Wert: 0x00 = unspezifizierter Singletag-Zugriff (RF61xR, RF68xR)
EPCID_UID	Array[1...62] of Byte	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
IDENT_DATA	Any / Variant	0x00	Datenpuffer in den die gelesenen EPC-Speicherdaten abgelegt werden. Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

3.4.4.5 Read_TID

Der Baustein "Read_TID" liest Daten aus dem TID-Speicherbereich (Tag Identification Memory Bank) des RF600-Transponders aus. Über den Parameter "LEN_DATA" wird die zu lesende Länge der TID angegeben. Die Länge der TID variiert abhängig vom Transponder und ist dem dazugehörigen Datenblatt zu entnehmen.

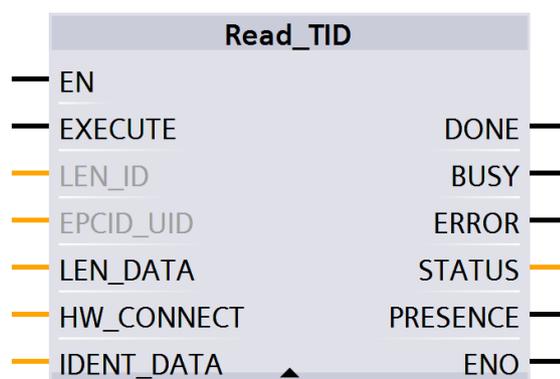


Bild 3-17 Baustein "Read_TID"

Tabelle 3- 32 Erläuterung zum Baustein "Read_TID"

Parameter	Datentyp	Default-Werte	Beschreibung
LEN_ID	Byte	B#16#0	Länge der EPC-ID / UID Default-Wert: 0x00 = un spezifizierter Singletag-Zugriff (RF61xR, RF68xR)
EPCID_UID	Array[1...62] of Byte	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
LEN_DATA	Word	W#16#4	Länge des auszulesenden TID-Speichers
IDENT_DATA	Any / Variant	0x00	Gelesene TID Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

3.4.4.6 Read_UID

Der Baustein "Read_UID" liest die UID eines HF-Transponders aus. Die UID hat immer eine feste Länge von 8 Byte.

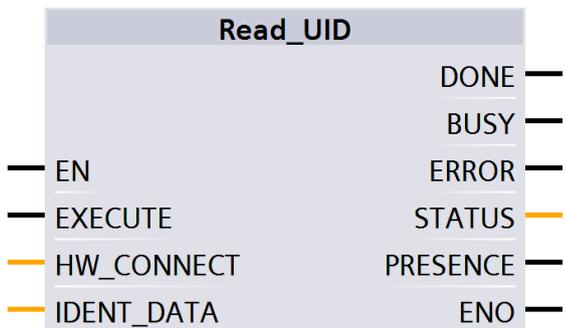


Bild 3-18 Baustein "Read_UID"

Tabelle 3- 33 Erläuterung zum Baustein "Read_UID"

Parameter	Datentyp	Default-Wert	Beschreibung
IDENT_DATA	Any / Variant	0x00	UID Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

3.4.4.7 Set_Ant

Mit Hilfe des Bausteins "Set_Ant" können Sie Antennen aus- oder einschalten. Der Baustein "Set_Ant_RF300" kann auch für RF200, MOBY D und MOBY U verwendet werden.

Set_Ant_RF300

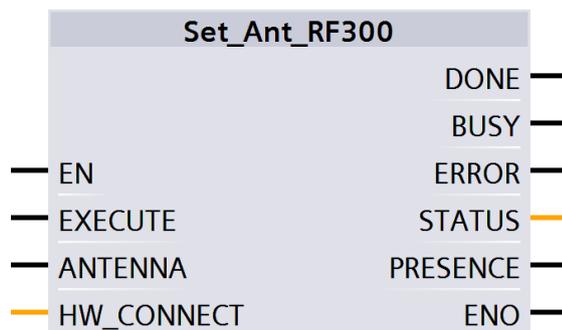


Bild 3-19 Baustein "Set_Ant_RF300"

Tabelle 3- 34 Erläuterung zum Baustein "Set_Ant_RF300"

Parameter	Datentyp	Default-Wert	Beschreibung
ANTENNA	Bool	False	False = Antenne ausschalten True = Antenne einschalten

3.4.4.8 Set_Param

Mit Hilfe des Bausteins "Set_Param" können Sie UHF-Parameter (z. B. die Antennenleistung) an einem RF61xR/RF68xR, sowie die Datum-/Uhrzeit-Parameter eines RF18xC/RF18xCI, RF166C oder RF360R (ab FW V2.0) während der Laufzeit ändern.

Hinweis

Einstellungen nur flüchtig gespeichert

Beachten Sie, dass die im Baustein "Set_Param" hinterlegten Parameter nur flüchtig gespeichert werden. Wird die Spannung der Baugruppe unterbrochen, gehen die hinterlegten Werte verloren und müssen erneut gesetzt werden.

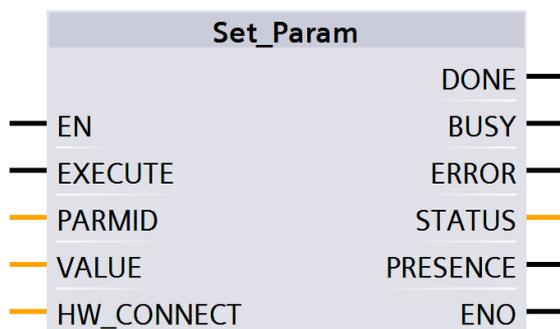


Bild 3-20 Baustein "Set_Param"

Tabelle 3- 35 Erläuterung zum Baustein "Set_Param"

Parameter	Datentyp	Default-Werte	Beschreibung
PARMID	DWORD	0x00	Parameterkennung
VALUE	DWORD	0x00	Parameterwert

Werte für "PARMID" und "VALUE"

Abhängig von dem verwendeten Ident-Gerät, können folgende Parameter mit Hilfe dieses Bausteins ausgelesen werden.

Tabelle 3- 36 Parameterwerte

PARMID (hex)	PARMID (ASCII)	Parameter	VALUE
0x41315057	A1PW	Antenne 01: Strahlungsleistung	Wertebereich: 0,5 ... 33 Schrittweite: 0,25 Strahlungsleistung der Antenne in [dBm]. Byte 1 und 2 sind unbelegt, Byte 3 stellt die Ganzzahl dar und Byte 4 die Nachkommastelle. Beispiel: Eine Strahlungsleistung von 10,25 dBm entspricht einem "VALUE" von "0x0A19".
0x41325057	A2PW	Antenne 02: Strahlungsleistung	
0x41335057	A3PW	Antenne 03: Strahlungsleistung	
0x41345057	A4PW	Antenne 04: Strahlungsleistung	
0x41315452	A1TR	Antenne 01: RSSI-Schwellwert	Wertebereich: 0 ... 255 Schwellwert für RSSI. Transponder mit niedrigeren Werten werden verworfen. Einheitsloser Wert, ohne direkten Bezug zur Strahlungsleistung.
0x41325452	A2TR	Antenne 02: RSSI-Schwellwert	
0x41335452	A3TR	Antenne 03: RSSI-Schwellwert	
0x41345452	A4TR	Antenne 04: RSSI-Schwellwert	
0x5331444C	S1DL	Lesestelle 01: RSSI Delta	Wertebereich: 0 ... 255 Differenz für RSSI-Werte. Transponder mit niedrigeren Werten bezogen auf den Transponder mit dem höchsten RSSI-Wert werden verworfen. Einheitsloser Wert, ohne direkten Bezug zur Strahlungsleistung.
0x5332444C	S2DL	Lesestelle 02: RSSI Delta	
0x5333444C	S3DL	Lesestelle 03: RSSI Delta	
0x5334444C	S4DL	Lesestelle 04: RSSI Delta	
0x4131504F	A1PO	Antenne 01: Polarisierung	Wertebereich: 0, 1, 2, 4 Polarisierung der Antenne (für intelligente Antennen, z. B. interne Antenne RF685R) <ul style="list-style-type: none"> • 0: default, undefiniert • 1: zirkular • 2: vertikal linear • 4: horizontal linear Eingabe ist bitkodiert. Kombinationen sind möglich (Werte addieren).
0x4132504F	A2PO	Antenne 02: Polarisierung	
0x4133504F	A3PO	Antenne 03: Polarisierung	
0x4134504F	A4PO	Antenne 04: Polarisierung	

PARMID (hex)	PARMID (ASCII)	Parameter	VALUE
0x52364353	R6CS	Modulationsschema	<p>Wertebereiche: 32, 33, 34, 35, 37, 65</p> <p>Modulationsschema der Lesestelle</p> <p>Festlegung welche Transponder-Typen erkannt werden (ISO 18000-63/-62).</p> <ul style="list-style-type: none"> • 32: Tx: 40 kbps / Rx: 40 kbps / FMO • 33: Tx: 40 kbps / Rx: 62,5 kbps / Miller4 • 34: Tx: 40 kbps / Rx: 75 kbps / Miller4 • 35: Tx: 80 kbps / Rx: 62,5 kbps / Miller4 • 37: Tx: 80 kbps / Rx: 400 kbps / FMO • 65: Tx: 40 kbps / Rx: 40 kbps / ISO 18000-62
0x57544348	WTCH	Datum und Uhrzeit	<p>Wertebereich: 01.01.2000 00:00 Uhr ... 19.01.2038 3:14 Uhr</p> <p>01.01.2000 01:00 Uhr ≙ 946684800</p> <p>Datum und Uhrzeit (UTC)</p> <p>Zeit in Sekunden seit 01.01.1970; Setzen der internen Reader-/CM-Uhr, wobei Datum und Uhrzeit gesetzt werden.</p>
0x57544F44	WTOD	Uhrzeit	<p>Wertebereich: 0:00 ... 23:59 Uhr</p> <p>S7-Uhrzeit (TOD, UTC)</p> <p>Millisekunden seit Mitternacht; Setzen der internen Reader-/CM-Uhr, wobei nur die Uhrzeit verändert wird und nicht das Datum.</p>
0x57444154	WDAT	Datum	<p>Wertebereich: 01.01.2000 ... 18.01.2038</p> <p>S7-Datum</p> <p>Tage seit 01.01.1990; Setzen der internen Reader-/CM-Uhr, wobei nur das Datum verändert wird und nicht die Uhrzeit.</p>

3.4.4.9 Write_EPC_ID

Der Baustein "Write_EPC_ID" überschreibt die EPC-ID des RF600-Transponders und passt die Länge der EPC-ID im Speicher des Transponders an. Über den Parameter "LEN_ID_NEW" wird die zu schreibende neue EPC-ID-Länge und über die Parameter "LEN_ID" und "EPCID_UID" wird die bisherige EPC-ID angegeben.

Beachten Sie, dass sich beim Ausführen des Bausteins nur ein Transponder im Antennenfeld befindet. Dadurch wird sichergestellt, dass die Identifizierung beim Schreiben der ID eindeutig ist. Befinden sich mehrere Transponder im Antennenfeld, wird eine negative Antwort zurückgegeben.

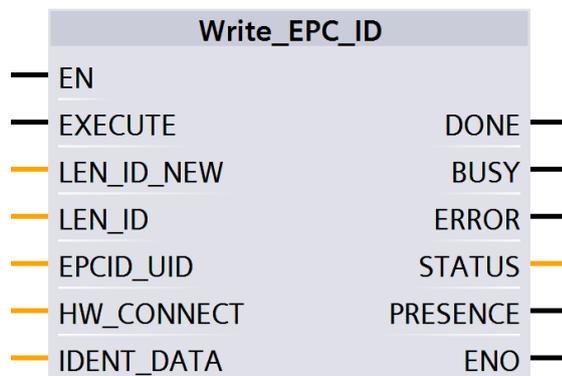


Bild 3-21 Baustein "Write_EPC_ID"

Tabelle 3- 37 Erläuterung zum Baustein "Write_EPC_ID"

Parameter	Datentyp	Default-Werte	Beschreibung
LEN_ID_NEW	Byte	W#16#0C	Länge der aktuellen EPC-ID
LEN_ID	Byte	B#16#0	Länge der bisherigen EPC-ID
EPCID_UID	Array[1...62] of Byte	0x00	Bisherige EPC-ID
IDENT_DATA	Any / Variant	0x00	Aktuelle EPC-ID Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

3.4.4.10 Write_EPC_Mem

Der Baustein "Write_EPC_Mem" überschreibt den EPC-Speicher ab Adresse 4 des RF600-Transponders. Über den Parameter "LEN_DATA" wird die zu überschreibende Länge des EPC-Speichers angegeben.

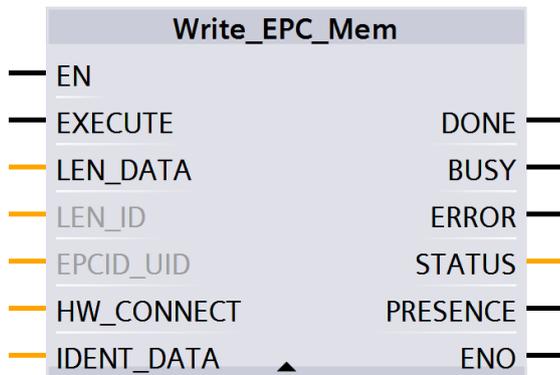


Bild 3-22 Baustein "Write_EPC_Mem"

Tabelle 3- 38 Erläuterung zum Baustein "Write_EPC_Mem"

Parameter	Datentyp	Default-Werte	Beschreibung
LEN_DATA	Word	W#16#0	Länge des zu überschreibenden EPC-Speichers (1 ... 62 Byte)
LEN_ID	Byte	B#16#0	Länge der EPC-ID / UID Default-Wert: 0x00 = un spezifizierter Singletag-Zugriff (RF61xR, RF68xR)
EPCID_UID	Array[1...62] of Byte	0x00	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
IDENT_DATA	Any / Variant	0x00	Datenpuffer mit den zu überschreibenden EPC-Speicherdaten. Hinweis: S7-1200/S7-1500: Bei Variant ist derzeit nur ein "Array_of_Byte" mit variabler Länge anlegbar. Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

3.4.5 Reset-Bausteine

Die in diesem Kapitel beschriebenen Reset-Bausteine benötigen Sie, wenn Sie die optischen Lesesysteme oder die Kommunikationsmodule ohne das Technologieobjekt "SIMATIC Ident" betreiben möchten. Um diese Bausteine auch mit RF120C verwenden zu können, müssen Sie in der Gerätekonfiguration die Einstellung "Parameter über FB / Optische Lesegeräte" parametrieren haben.

Bei Verwendung des Technologieobjekts können Sie die spezifischen Reset-Bausteine verwenden, um andere als im Technologieobjekt eingestellte Reader-Parametrierungen zur Laufzeit zu verändern.

Beachten Sie, dass bei den Parametern automatisch der Default-Wert verwendet wird, wenn Sie keinen Wert übergeben.

3.4.5.1 Reset_MOBY_D

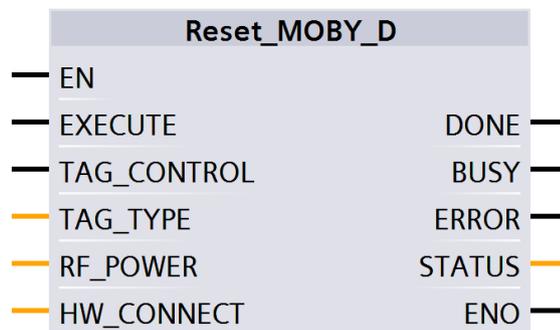


Bild 3-23 Baustein "Reset_MOBY_D"

Tabelle 3- 39 Erläuterung zum Baustein "Reset_MOBY_D"

Parameter	Datentyp	Default-Werte	Beschreibung
TAG_CONTROL	Bool	True	Anwesenheitskontrolle
TAG_TYPE	Byte	0x01	Transponder-Typ: • 0x01 = jeder ISO-Transponder
RF_POWER	Byte	0x00	Ausgangsleistung HF-Leistung von 0,5 W bis 10 W in 0,25 W-Schritten (Wertebereich: 0x02 ... 0x28)

3.4.5.2 Reset_MOBY_U

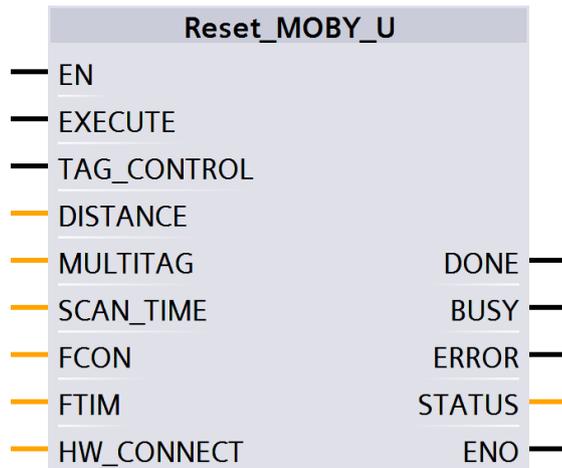


Bild 3-24 Baustein "Reset_MOBY_U"

Tabelle 3- 40 Erläuterung zum Baustein "Reset_MOBY_U"

Parameter	Datentyp	Default-Werte	Beschreibung
TAG_CONTROL	Bool	True	Anwesenheitskontrolle
DISTANCE	Byte	0x23	Reichweitenbegrenzung (Wertebereich: 0x02 ... 0x23 oder 0x82 ... 0xA3 für verminderte Sende- leistung)
MULTITAG	Byte	0x01	Maximale Anzahl der parallel im Antennenfeld bearbeitbaren Trans- ponder. (Wertebereich: 0x01 ... 0x12)
SCAN_TIME	Byte	0x00	Scanning_Time: Stand-by-Zeit des Transponders (Wertebereich: 0x00 ... 0xC8)
FCON	Byte	0x00	field_ON_control: Bero-Betriebsart (Wertebereich: 0x00 ... 0x03)
FTIM	Byte	0x00	field_ON_time: Zeit für Bero- Betriebsart (Wertebereich: 0x00 ... 0xFF)

3.4.5.3 Reset_MV

Um Kameras der optischen Lesesysteme zurückzusetzen, rufen Sie den Baustein auf und führen Sie den Parameter "EXECUTE" aus.

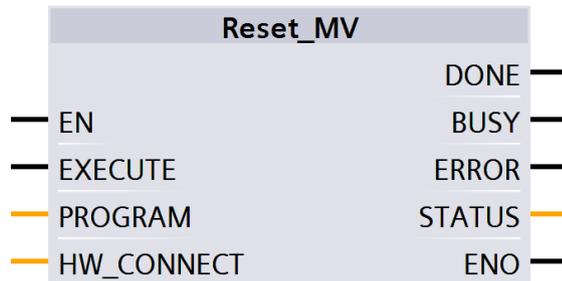


Bild 3-25 Baustein "Reset_MV"

Tabelle 3- 41 Erläuterung zum Baustein "Reset_MV"

Parameter	Datentyp	Default-Werte	Beschreibung
PROGRAM	Byte	0x00	Programmanwahl <ul style="list-style-type: none"> • B#16#0: Reset ohne Programmanwahl bzw. im Falle einer Diagnose wird der Fehlercode für "IN_OP = 0" am Ausgangsparameter "STATUS" angezeigt. • B#16#1 ... B#16#0F: Nummer des zu startenden Programms ⇒ Reset mit Programmanwahl (ab FW V5.1 der MV4x0)

3.4.5.4 Reset_RF200

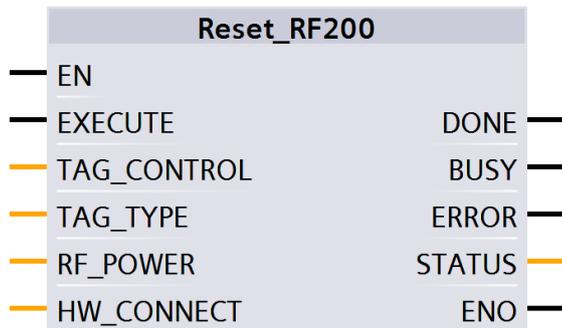


Bild 3-26 Baustein "Reset_RF200"

Tabelle 3- 42 Erläuterung zum Baustein "Reset_RF200"

Parameter	Datentyp	Default-Werte	Beschreibung
TAG_CONTROL	Byte	0x01	Anwesenheitskontrolle
TAG_TYPE	Byte	0x01	Transponder-Typ: <ul style="list-style-type: none"> • 0x01 = jeder ISO-Transponder • 0x03 = MDS D3xx - Optimierung
RF_POWER	Byte	0x04	Ausgangsleistung; nur für RF290R relevant HF-Leistung von 0,5 W bis 5 W in 0,25 W-Schritten (Wertebereich: 0x02 ... 0x14). Default-Wert 0x04 ≙ 1 W.

3.4.5.5 Reset_RF300

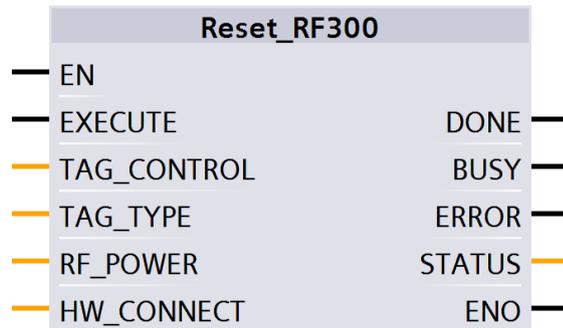


Bild 3-27 Baustein "Reset_RF300"

Tabelle 3- 43 Erläuterung zum Baustein "Reset_RF300"

Parameter	Datentyp	Default-Werte	Beschreibung
TAG_CONTROL	Byte	0x01	Anwesenheitskontrolle <ul style="list-style-type: none"> • 0x00 = Aus • 0x01 = Ein • 0x04 = Anwesenheit (Antenne ist aus. Antenne wird nur eingeschaltet, wenn ein Read- oder Write-Befehl geschickt wird.)
TAG_TYPE	Byte	0x00	Transponder-Typ: <ul style="list-style-type: none"> • 0x00 = RF300-Transponder • 0x01 = jeder ISO-Transponder
RF_POWER	Byte	0x00	Ausgangsleistung; nur für RF380R relevant HF-Leistung von 0,5 W bis 2 W in 0,25 W-Schritten (Wertebereich: 0x02 ... 0x08). Default-Wert: 0x00 \approx 1,25 W. Diese Einstellung wird bei dem Reader RF380R der 2. Generation (6GT2801-3BAx0) nicht benötigt, da die Leistungsgrenzen abhängig vom Reader-Transponder-Abstand automatisch optimiert werden.

3.4.5.6 Reset_Univ

Der Baustein "Reset_Univ" ist ein universeller Reset-Baustein, mit dem Identifikationssysteme zurückgesetzt werden können.

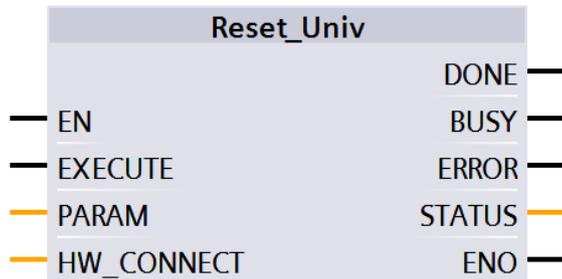


Bild 3-28 Baustein "Reset_Univ"

Tabelle 3- 44 Erläuterung zum Baustein "Reset_Univ"

Parameter	Datentyp	Default-Wert	Beschreibung
PARAM	Array [1...16] of Byte	0x00	Daten für Reset-Telegramm Die hier einzustellenden Daten können für Spezialeinstellungen bei Bedarf vom Support bereitgestellt werden.

Tabelle 3- 45 Aufbau des Parameters "PARAM"

Byte	1	2...5	6	7...8	9	10	11	12	13...14	15	16
Wert (RF200, RF300, MOBY U/D)	0x04	0x00	0x0A	0x00	scanning_time	param	option_1	distance_limiting	Anzahl der Transponder	field_on_control	field_on_time
Wert (Freeport)	0x04	0x00	0x0A	0x00	0x00	0x00	option_1	0x00	0x00	0x00	0x00

3.5 Ident-Profil programmieren

3.5.1 Aufbau des Ident-Profiles

Hinweis

Parallelbetrieb von Ident-Bausteinen und Ident-Profil ist nicht möglich

Beachten Sie, dass das CM bzw. der Reader nicht zeitgleich über die Ident-Bausteine und über das Ident-Profil betrieben werden kann.

Die in dem Kapitel "Ident-Bausteine programmieren (Seite 28)" beschriebenen Bausteine stellen eine vereinfachte Schnittstelle des Ident-Profiles dar. Sollten die Funktionalitäten der Bausteine für Ihre Anwendung nicht ausreichen, können Sie alternativ das Ident-Profil verwenden. Mithilfe des Ident-Profil können Sie komplexe Befehlsstrukturen programmieren und mit Befehlswiederholung arbeiten. Die nachfolgende Grafik zeigt das Ident-Profil, inklusive der damit umsetzbaren Befehle.

Hinweis

Ident-Profil für geschulte Anwender

Das Ident-Profil ist ein komplexer Baustein und beinhaltet alle Funktionalitäten der Ident-Bausteine. Das Ident-Profil wurde speziell für geschulte Baustein-Anwender konzipiert, die komplexe Funktionen mit Hilfe eines einzigen Bausteins projektieren wollen. Ungeschulten Anwendern empfehlen wir die Verwendung der Ident-Bausteine.

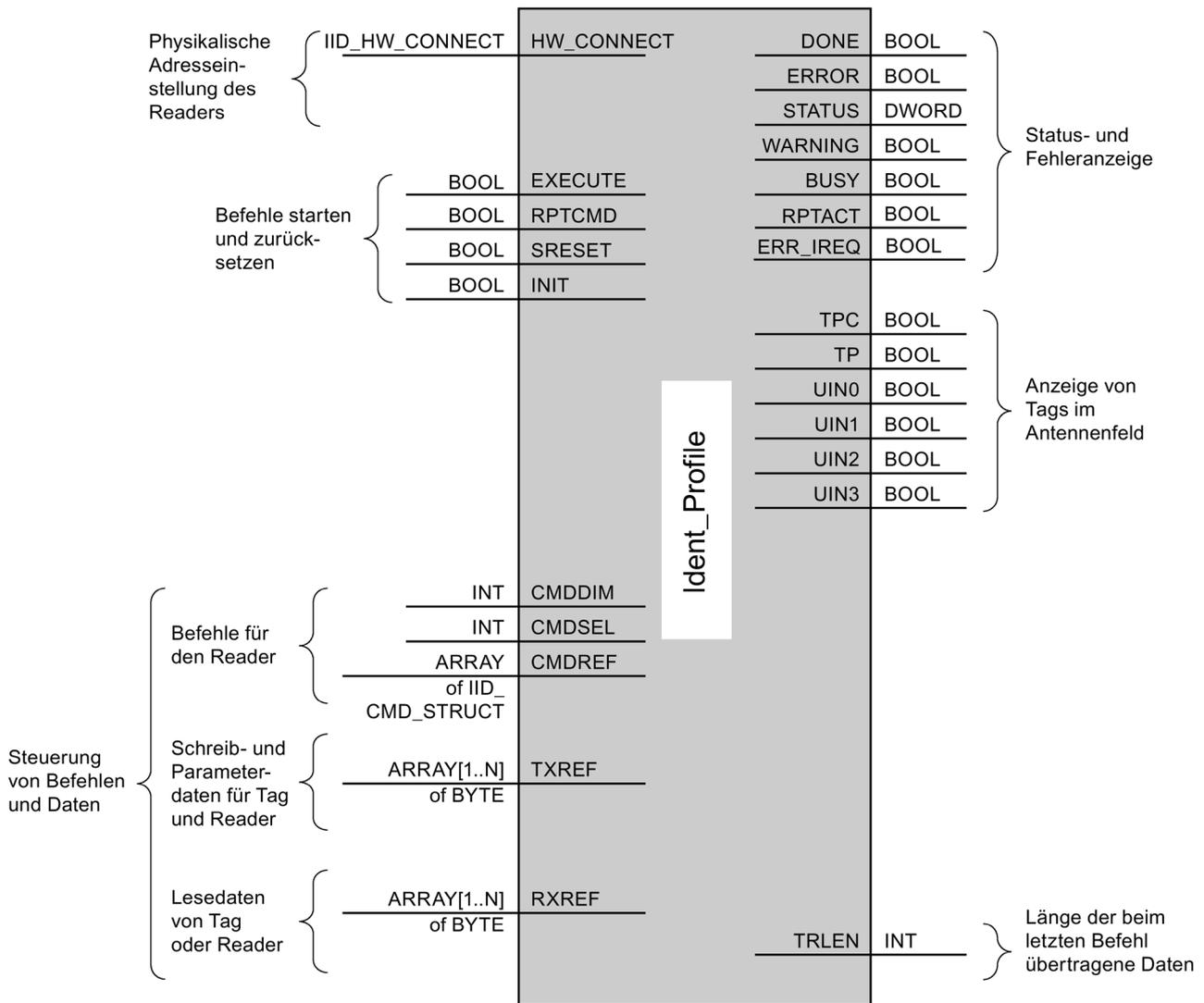


Bild 3-29 Die Input-Parameter des Ident-Profiles

Hinweis

Arbeit mit mehreren Kanälen/Lesestellen

Wenn Sie mit mehreren Kanälen/Lesestellen arbeiten, müssen sie darauf achten, dass für jeden Kanal / jede Lesestelle der Baustein mit einem eigenen Instanz-DB aufgerufen wird.

Schnittstellenbeschreibung

Tabelle 3- 46 Eingangsparameter

Eingangsparameter	Datentyp	Default-Wert	Bedeutung
HW_CONNECT	IID_HW_CONNECT	--	Globaler Parameter vom Typ "IID_HW_CONNECT" um den Kanal bzw. die Lesestelle zu adressieren und die Bausteine zu synchronisieren. Die Adressierung erfolgt wie in Kapitel "Parametrieren/programmieren (Seite 22)" beschrieben.
	TO_IDENT	--	Technologieobjekt "TO_Ident" des Ident-Geräts
EXECUTE	BOOL	FALSE	TRUE = Anstoß eines neuen Befehls Vor dem Starten müssen Sie den Befehl und die entsprechenden Parameter, in den mit "CMDREF" verknüpften Speicher, setzen.
RPTCMD	BOOL	FALSE	TRUE = Wiederholung des aktuell ausgeführten oder als nächstes auszuführenden Befehls durch den Reader / das Kommunikationsmodul
SRESET	BOOL	FALSE	TRUE = Abbruch des aktuell im Reader/Kommunikationsmodul verarbeiteten Befehls
INIT	BOOL	FALSE	TRUE = Reader/Kommunikationsmodul führt einen Neustart aus und wird neu parametrier Der erste Befehl, der an einen Reader / ein Kommunikationsmodul gesendet wird, muss immer zusammen mit einem "INIT" erfolgen.
CMDDIM	INT	10	Anzahl der Befehle im Parameter "CMDREF"
CMDSEL	INT	0	Auswahl des auszuführenden Befehls "CMDREF"; 1 ⇒ 1. Befehl, ... Der Wert des Parameters "CMDSEL" kann nie größer als der Wert des Parameters "CMDDIM" sein.
CMDREF	ARRAY[1...n] of IID_CMD_STRUCT	--	Befehlsfeld Das Feld kann bis zu 100 Befehle aufnehmen. Die Befehle sind komplexe Variablen des Typs "IID_CMD_STRUCT". Weitere Informationen zu "CMDREF" finden Sie im Kapitel "Befehle des Ident-Profiles (Seite 73)".
TXREF	ARRAY[1...n] of BYTE	--	Verweis auf globalen Speicherbereich für Sendedaten. Der Speicherbereich kann mit anderen Baustein-Instanzen geteilt werden. Der Wert "n" der einzelnen Bausteine ist variabel und kann bis zu 32 KB groß sein.
RXREF	ARRAY[1...n] of BYTE	--	Verweis auf globalen Speicherbereich für Empfangsdaten. Der Speicherbereich kann mit anderen Baustein-Instanzen geteilt werden. Der Wert "n" der einzelnen Bausteine ist variabel und kann bis zu 32 KB groß sein.

Tabelle 3- 47 Ausgangsparamater

Ausgangsparamater	Datentyp	Default-Wert	Bedeutung
DONE	BOOL	FALSE	TRUE = Befehl wurde fehlerfrei ausgeführt.
ERROR	BOOL	FALSE	TRUE = Fehler wurde erkannt. Der Fehler wird im Parameter "STATUS" ausgegeben. Das Bit wird mit dem Start eines neuen Befehls automatisch zurückgesetzt.
STATUS	DWORD	FALSE	Warnung- und Fehler Bei "ERROR = TRUE" bzw. "WARNING = TRUE" steht im Parameter "STATUS" die Fehler- bzw. Warninformation. Weitere Informationen finden Sie im Kapitel "Fehlermeldungen (Seite 117)".
WARNING	BOOL	FALSE	TRUE = Warnung wurde erkannt. Die Warnung wird im Parameter "STATUS" ausgegeben. Wenn der Parameter "ERROR" nicht zeitgleich gesetzt ist, wurden die Daten korrekt bearbeitet. Das Bit wird mit dem Start eines neuen Befehls automatisch zurückgesetzt.
BUSY	BOOL	FALSE	TRUE = Der Baustein bearbeitet einen Befehl. Andere Befehle, außer "INIT" und "SRESET", können nicht gestartet werden.
RPTACT	BOOL	FALSE	TRUE = "RPTCMD" ist aktiv Das Quittierungs-Bit zeigt an, dass der "Repeat-Mode" des CMs/Readers aktiv ist.
ERR_IREQ	BOOL	FALSE	TRUE = am Kommunikationsmodul oder Reader ist ein Fehler aufgetreten (z. B. beim Hochlauf des Readers, Verbindungsunterbrechung oder bei RF61xR, RF68xR bei Antennenfehlern oder Konfigurationsänderungen)
TPC	BOOL	FALSE	Transponder Presence Changed (nur bei aktivem "Presence_Mode") TRUE = Ein neuer Transponder kommt ins Antennenfeld des Readers oder ein Transponder hat das Antennenfeld verlassen. Der Parameter wird nach erfolgreicher Ausführung des nächsten "INVENTORY"- (0x80, 0x81, 0x87) oder "INIT"- Befehls auf "FALSE" gesetzt.
TP	BOOL	FALSE	Transponder Presence TRUE = ein Transponder befindet sich im Antennenfeld des Readers. Hinweis zu Freeport protocol: Wurde noch kein "Read" aufgesetzt, so wird durch das Anwesenheitsbit ("TP" bzw. "ANZ_MDS_present") mitgeteilt, dass neue Daten empfangen wurden und im Kommunikationsmodul zur Abholung durch einen "Read"- Befehl bereit stehen.

Ausgangsparameter	Datentyp	Default-Wert	Bedeutung
UIN0	BOOL	FALSE	Bei RFID-Readern wird die Anzahl der im Antennenfeld befindlichen Transponder angezeigt. UIN0 ... UIN3 sind als Binärwert zu interpretieren. Beispiel: UIN3 = 0; UIN2 = 0; UIN1 = 1; UIN0 = 1 ⇒ 3 Transponder Bei optischen Lesegeräten werden die verschiedenen Zustände des Lesegeräts angezeigt. <ul style="list-style-type: none"> • UIN0: Entspricht "IN_OP"-Bit des Lesegeräts • UIN1: Entspricht "RDY"-Bit des Lesegeräts • UIN2 + UIN3: Diese zwei Bits werden als unsigned Wert interpretiert (Bit 2 ist das niederwertigere Bit) welcher die Anzahl verfügbarer decodierter Codes darstellt. Wenn der Wert = 3 ist, dann sind drei oder mehr decodierte Codes verfügbar.
UIN1	BOOL	FALSE	
UIN2	BOOL	FALSE	
UIN3	BOOL	FALSE	
TRLEN	INT	0	Anzahl der Daten die nach erfolgreicher Ausführung des Befehls empfangen wurden.

3.5.2 Datenstruktur des Ident-Profiles

Bei jedem Aufruf des Ident-Profiles müssen Sie die Parameter ("HW_CONNECT", "CMDREF", "TXREF" und "RXREF"), wie im Kapitel "Aufbau des Ident-Profiles (Seite 67)" beschrieben, versorgen.

Der Aufruf des Ident-Profiles erfolgt immer über den Eingangsparameter "HW_CONNECT" und die "IN/OUT"-Parameter "CMDREF", "TXREF" und "RXREF". Alle drei Parameter müssen in einem Datenbaustein angelegt werden. Der Zusammenhang zwischen den drei "IN/OUT"-Parametern wird im Folgenden genauer beschrieben:

- CMDREF (Befehlspeicher):
 - Array[1...n] of IID_CMD_STRUCT
- TXREF (Sendepuffer):
 - Array[1...n] of Byte
 - Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.
- RXREF (Empfangspuffer):
 - Array[1...n] of Byte
 - Bei Any können zusätzlich auch andere Datentypen/UDTs angelegt werden.

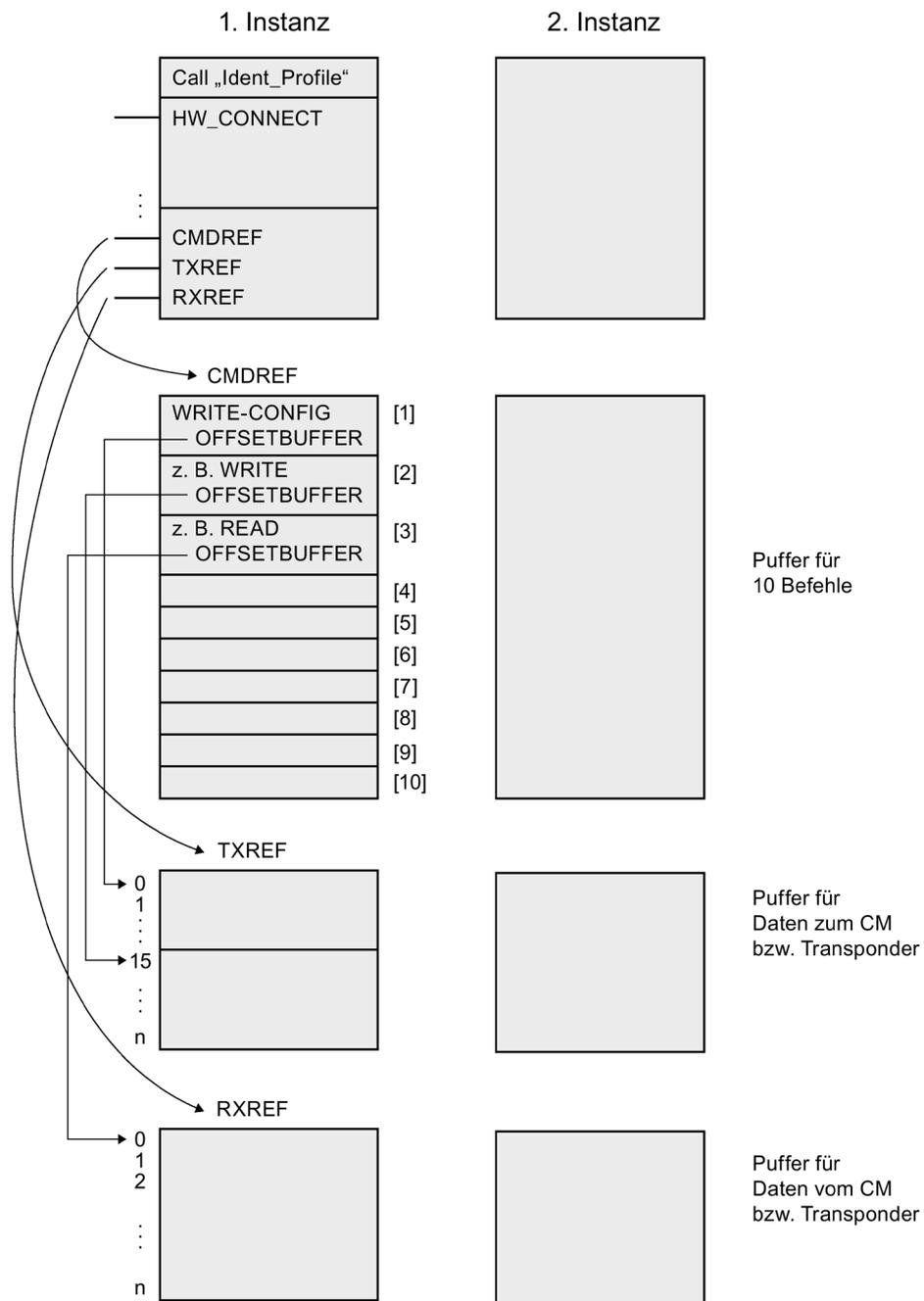


Bild 3-30 Datenstruktur-Beispiel des Ident-Profiles

Erläuterung zu dem Datenstruktur-Beispiel

• CMDREF[1]:

Befehl "WRITE-CONFIG", OFFSETBUFFER = 0

An der Stelle "CMDREF[1]" müssen Sie den Befehl "WRITE-CONFIG" parametrieren, damit der "INIT/Reset" korrekt ausgeführt wird.

• CMDREF[2]:

Befehl "WRITE", OFFSETBUFFER = 15

• CMDREF[3]:

Befehl "READ", OFFSETBUFFER = 0

Wird der Befehl "CMDREF[2]" ausgewählt, wird ein Schreibbefehl gestartet und die zu schreibenden Daten werden aus dem Parameter "TXREF" ab dem Byte 15 geholt. Wird der Befehl "CMDREF[3]" ausgewählt, werden die gelesenen Daten ab Byte 0 im Parameter "RXREF" abgelegt.

3.5.3 Befehle des Ident-Profiles

Die nachfolgende Tabelle enthält alle Befehle, die das Ident-Profil und der Baustein "AdvancedCMD" unterstützen.

Tabelle 3- 48 Befehle des Ident-Profil

Befehl	Befehlscode		Verwendete Parameter	Beschreibung
	HEX	ASCII		
PHYSICAL-READ	0x70	'p'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA, ADDR_TAG, MEM_BANK, PSWD	Liest Daten von einem Transponder / Optischen Lesesystem durch Angabe der physikalischen Anfangsadresse, der Länge und des Passworts aus.
PHYSICAL-WRITE	0x71	'q'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA, ADDR_TAG, MEM_BANK, PSWD	Schreibt Daten auf einen Transponder / Optischen Lesesystem durch Angabe der physikalischen Anfangsadresse, der Länge und des Passworts.
READER-STATUS	0x74	't'	OFFSETBUFFER, ATTRIBUTES	Liest den Status des Readers aus.
TAG-STATUS	0x73	's'	OFFSETBUFFER, EPCID_UID, LEN_ID, ATTRIBUTES	Liest den Status eines Transponders aus.
INVENTORY	0x69	'i'	OFFSETBUFFER, ATTRIBUTES, DURATION, DUR_UNIT	Fordert eine Liste aller gegenwärtig zugänglichen Transponder innerhalb der Antennenreichweite an.
FORMAT	0x66	'f'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA	Initialisiert den Transponder.
PUT	0x65	'e'	OFFSETBUFFER, EPCID_UID, LEN_ID, LEN_DATA	Überträgt weitere Befehle, die nicht im Normprofil-Standard festgelegt sind. Dazu ist für jeden Befehl eine entsprechende Datenstruktur im Sendedaten-Puffer definiert.

Befehl	Befehlscode		Verwendete Parameter	Beschreibung
	HEX	ASCII		
WRITE-ID	0x67	'g'	OFFSETBUFFER, EPCID_UID, LEN_ID, NEW-LEN_ID, PSWD	Schreibt eine neue EPC-ID in den Transponder.
KILL-TAG	0x6A	'j'	EPCID_UID, LEN_ID, PSWD	Der Transponder wird permanent deaktiviert.
LOCK-TAG-BANK	0x79	'y'	EPCID_UID, LEN_ID, PSWD, ACTION, MASK	Der entsprechende Speicherbereich des Transponders wird gemäß Vorgabe gesperrt.
EDIT-BLACKLIST	0x7A	'z'	EPCID_UID, LEN_ID, MODE	Die Black List wird bearbeitet. Es kann der derzeitige Transponder hinzugefügt, alle erkannten Transponder hinzugefügt, einzelne Transponder gelöscht oder alle Transponder gelöscht werden.
GET-BLACKLIST	0x6C	'l'	OFFSETBUFFER	Die gesamten TagIDs aus der Black List werden ausgelesen.
READ-CONFIG	0x61	'a'	--	Liest die Parameter aus dem Kommunikationsmodul/Reader aus.
WRITE-CONFIG	0x78	'x'	LEN_DATA, CONFIG	Sendet neue Parameter an das Kommunikationsmodul bzw. den Reader.

3.5.3.1 Befehlsstruktur

Bevor Sie einen Befehl mit "EXECUTE" bzw. "INIT" starten können, müssen Sie diesen definieren. Für die einfache Definition eines Befehls wurde mit Hilfe des Datentyps "IID_CMD_STRUCT" der Befehlsbuffer "CMDREF" erstellt. In dem Befehlsbuffer stehen Ihnen 100 Bereiche zur Verfügung, in denen Befehle parametrierbar sind. Über den Parameter "CMDSEL" wird festgelegt, welcher Befehl (1...n) mit "EXECUTE" gestartet wird.

Beachten Sie, dass das erste Element im Puffer immer für "INIT" reserviert ist. D. h. wenn "INIT" gesetzt wird, muss "CMDSEL" auf "1" stehen und das Element "1" im CMD-Puffer mit den entsprechenden Einstellungen gefüllt sein. Die folgende Tabelle enthält die Befehlsstruktur der Parameter. Nicht jeder Befehl nutzt alle Parameter.

Tabelle 3- 49 Befehlsstruktur der Parameter

Parameter	Datentyp	Default-Wert	Beschreibung
CMD	BYTE	B#16#0	Befehlscode (vergleiche Tabelle im Kapitel "Befehle des Ident-Profiles (Seite 73)")
ATTRIBUTES	BYTE	B#16#0	Subcommand-Bezeichner für einige Befehle (z. B. "READER-STATUS", "INVENTORY", usw.)
OFFSETBUFFER	INT	0	Relativer Offset innerhalb des Empfangsdaten-Puffers. Der Parameter gibt die Adresse innerhalb des Speicherbereichs an, an der das erste Byte der empfangenen Daten gespeichert werden muss oder das erste Byte der zu sendenden Daten erwartet wird. Alle folgenden Bytes müssen in aufsteigenden Adressen gespeichert werden.

Parameter	Datentyp	Default-Wert	Beschreibung
EPCID_UID	ARRAY[1..62] OF BYTE	B#16#0	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
LEN_DATA	WORD	W#16#0	Anzahl der zu lesenden bzw. zu schreibenden Bytes
ADDR_TAG	DWORD	DW#16#0	Physikalische Startadresse auf dem Transponder
CONFIG	BYTE	B#16#0	<ul style="list-style-type: none"> • 0x01 = Reset, keine Konfigurationsdaten • 0x02 = kein Reset, zu sendende Konfigurationsdaten • 0x03 = Reset, zu sendende Konfigurationsdaten • 0x80 = kein Reset, nur Einzelparameter
CHAINED	BOOL	FALSE	<ul style="list-style-type: none"> • 0x00 = nicht verkettet • 0x01 = verkettet <p>Alle verketteten Befehle müssen dieses Bit gesetzt haben, bis auf den letzten Befehl. Die Befehle werden in der Reihenfolge wie sie in der CMD-Struktur stehen abgearbeitet.</p>
EXT_UHF	STRUCT	--	Struktur für Zusatzparameter
LEN_ID	BYTE	B#16#0	Länge der gültigen Daten im Feld "EPCID_UID".
MEM_BANK	BYTE	B#16#3	Memory Bank auf dem Transponder <ul style="list-style-type: none"> • 0x00 = RESERVED • 0x01 = EPC • 0x02 = TID • 0x03 = USER
PSWD	DWORD	DW#16#0	Passwort für Transponder-Zugriff 0x00 = kein Passwort
EDIT_BLACKLIST_MODE	BYTE	B#16#0	Modus <ul style="list-style-type: none"> • 0x00 = TagID hinzufügen • 0x01 = Alle "Observed"-Transponder hinzufügen • 0x02 = TagID löschen • 0x03 = Alle löschen
INVENTORY_DURATION	WORD	W#16#0	Zeitdauer Zeitdauer oder Anzahl Inventories oder Anzahl der "Observed"-Events Bsp.: <ul style="list-style-type: none"> • 0x00 = kein Inventory • 0x01 = ein Inventory

Parameter	Datentyp	Default-Wert	Beschreibung
INVENTORY_DUR_UNIT	WORD	W#16#0	Einheit für "DURATION" <ul style="list-style-type: none"> • 0x00 = Zeit [ms] • 0x01 = Inventories • 0x02 = Anzahl der Transponder die den Zustand "Observed" erreichen
LOCK-TAG-BANK_ACTION	WORD	W#16#0	Lock-Action (siehe "EPC-Specification")
LOCK-TAG-BANK_MASK	WORD	W#16#0	Lock-Mask (siehe "EPC-Specification")

3.5.3.2 Befehle

Nachfolgend werden für die einzelnen Befehle alle relevanten Parameter und Parameterwerte genannt. Nicht genannte Parameter erhalten den im vorherigen Kapitel angegebenen Default-Wert.

Tabelle 3- 50 PHYSICAL-READ

Parameter	Wert / Beschreibung
CMD	0x70
OFFSETBUFFER	Offset im Empfangspuffer "RXREF"
LEN_DATA	Länge empfangene Daten
ADDR_TAG	Adresse auf dem Transponder
CHAINED	<ul style="list-style-type: none"> • True = verkettet • False = nicht verkettet
EPCID_UID	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
LEN_ID	Länge der EPC-ID / UID (2-62 Byte) Default-Wert: 0x00 ≙ unspezifizierter Singletag-Zugriff
MEM_BANK	Speicherbank <ul style="list-style-type: none"> • 0x00 ≙ Reserved • 0x01 ≙ EPC • 0x02 ≙ TID • 0x03 ≙ USER
PSWD	Passwort 0x00 ≙ kein Passwort
RXREF	Gelesene Daten

Tabelle 3- 51 PHYSICAL-WRITE

Parameter	Wert / Beschreibung
CMD	0x71
OFFSETBUFFER	Offset im Sendepuffer "TXREF"
LEN_DATA	Länge der zu schreibenden Daten
ADDR_TAG	Adresse auf dem Transponder
CHAINED	<ul style="list-style-type: none"> • True = verkettet • False = nicht verkettet
EPCID_UID	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
LEN_ID	Länge der EPC-ID / UID (2-62 Byte) Default-Wert: 0x00 ≙ unspezifizierter Singletag-Zugriff
MEM_BANK	Speicherbank <ul style="list-style-type: none"> • 0x00 ≙ Reserved • 0x01 ≙ EPC • 0x02 ≙ TID • 0x03 ≙ USER
PSWD	Passwort 0x00 ≙ kein Passwort
TXREF	Zu schreibende Daten

Tabelle 3- 52 READER-STATUS

Parameter	Wert / Beschreibung
CMD	0x74
OFFSETBUFFER	Offset im Empfangspuffer "RXREF"
ATTRIBUTES	Kennung der Status-Modi / Mögliche Eingaben: <ul style="list-style-type: none"> • RF200: 0x81 • RF300: 0x81, 0x86, 0xEF • RF61xR, RF68xR: 0x89 • MOBY U: 0x81, 0x84, 0x85 • MOBY D: 0x81
	<ul style="list-style-type: none"> • RF18xC/RF18xCI, RF166C, RF360R (ab FW 2.0): 0xA2
RXREF	Empfangene Statusdaten Die Datenstruktur der Status-Modi finden Sie im Kapitel "Reader_Status (Seite 33)".

Tabelle 3- 53 TAG-STATUS

Parameter	Wert / Beschreibung
CMD	0x73
OFFSETBUFFER	Offset im Empfangspuffer "RXREF"
ATTRIBUTES	Kennung der Status-Modi / Mögliche Eingaben: <ul style="list-style-type: none"> • RF200: 0x83 • RF300 (RF300T): 0x04, 0x82 • RF300 (ISO): 0x83 • RF300 Gen2: 0x83 • MOBY D: 0x83 ¹⁾ • MOBY U: 80
EPCID_UID	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
LEN_ID	Länge der EPC-ID / UID (2-62 Byte) Default-Wert: 0x00 ≙ un spezifizierter Singletag-Zugriff
RXREF	Empfangene Statusdaten Die Datenstruktur der Status-Modi finden Sie im Kapitel "Tag_Status (Seite 37)".

¹⁾ Nur SLG D10S

Tabelle 3- 54 INVENTORY

Parameter	Wert / Beschreibung
CMD	0x69
OFFSETBUFFER	Offset im Empfangspuffer "RXREF"
ATTRIBUTES	Kennung der Status-Modi / Mögliche Eingaben: <ul style="list-style-type: none"> • 0x80 ≙ Inventory mit kurzen Transponder-Informationen • 0x81 ≙ Inventory mit vielen Transponder-Informationen • 0x86 ≙ Presence-Mode an • 0x87 ≙ Presence-Mode aus
INVENTORY_DURATION	Nur bei 0x80 und 0x81: Zeitdauer Zeitdauer oder Anzahl Inventories oder Anzahl der "Observed"-Events Bsp.: <ul style="list-style-type: none"> • 0x00 ≙ kein Inventory • 0x01 ≙ ein Inventory

Parameter	Wert / Beschreibung
INVENTORY_DUR_UNIT	Nur bei 0x80 und 0x81: Einheit für "DURATION" <ul style="list-style-type: none"> • 0x00 ≙ Zeit [ms] • 0x01 ≙ Inventories • 0x02 ≙ Anzahl der Transponder die den Zustand "Observed" erreichen
RXREF	Empfangene Daten Die Datenstruktur der Status-Modi finden Sie im Kapitel "Inventory (Seite 47)".

Tabelle 3- 55 FORMAT (nicht bei RF200 und RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x66
OFFSETBUFFER	Offset im Sendepuffer "TXREF"
LEN_DATA	Länge der zu übertragenden Parameterdaten (15 Byte)
EPCID_UID	Puffer für bis zu 62 Byte EPC-ID, 8 Byte UID oder 4 Byte Handle-ID. <ul style="list-style-type: none"> • 2-62-Byte EPC-ID wird am Anfang des Puffers eingetragen (Länge wird durch "LEN_ID" beschrieben) • 8-Byte UID wird am Anfang des Puffers eingetragen ("LEN_ID = 8")
LEN_ID	Länge der EPC-ID / UID (2-62 Byte)
TXREF	Zu schreibende Parameterdaten

Tabelle 3- 56 Aufbau des Datenanhangs für den Befehl "FORMAT" bei Normaladressierung

Byte	1...8	9	10	11	12	13	14	15
Wert	0x00	0x06	0x03	0x00	INIT-Wert	0x00	MSB	LSB

Tabelle 3- 57 Erläuterung zum Aufbau des Datenanhangs für den Befehl "FORMAT"

Byte	Beschreibung
Byte 1...8	reserviert für Security-Code (muss mit "0" belegt sein, da SIMATIC RFID bisher keinen Code hat)
Byte 9	Länge der nachfolgenden Daten, hier 6
Byte 10	fest auf "0x03"
Byte 11	fest auf "0x00"
Byte 12	"INIT"-Wert: Mit diesem Wert wird der Datenbereich des Transponders beschrieben (Hex-Format).
Byte 13	fest auf "0x00"

Byte	Beschreibung
Byte 14	Speichergröße des Transponders (Endadresse + 1; High-Byte, Hex-Format)
Byte 15	Speichergröße des Transponders (Endadresse + 1; Low-Byte, Hex-Format)

Beim Initialisieren der Transponder mit den RF300-Readern der 2. Generation hat die Adressangabe folgende Bedeutung:

- Vollständiger "INIT": Adresse = 0 oder Adresse = Speichergröße des Transponders
Bedeutet, dass der komplette Adressbereich initialisiert wird.
- Teilweiser "INIT": Adresse ≠ 0
Bedeutet, dass die Initialisierung bis zur angegebenen Adresse erfolgt.

Tabelle 3- 58 Speichergrößen der Transponder

Transponder-Typ			Speichergröße	INIT-Dauer
2 KB	MOBY U	RAM ¹⁾	08 00	ca. 1 s
32 KB	MOBY U	RAM ¹⁾	80 00	ca. 1,5 s
44 Byte	MOBY D	I-Code 1	00 2C	ca. 0,4 s
112 Byte	MOBY D	ISO I-Code SLI	00 70	ca. 0,5 s
256 Byte	MOBY D	ISO Tag-it HF-I	01 00	ca. 1 s
992 Byte	MOBY D	ISO my-d	03 E0	ca. 3 s
2000 Byte	MOBY D	FRAM	07 D0	ca. 3 s
20 Byte	RF300	EEPROM	00 14	ca. 0,2 s
8 KB	RF300	FRAM ¹⁾	20 00	0,9 s
32 KB	RF300	FRAM ¹⁾	80 00	3,6 s
64 KB	RF300	FRAM ¹⁾	FF 00	7,2 s
752 Byte	MOBY E	EEPROM	02 EF	ca. 1,5 s

¹⁾ Der OTP-Speicher wird nicht mit diesem Befehl initialisiert.

Tabelle 3- 59 PUT (nicht bei RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x65
OFFSETBUFFER	Offset im Sendepuffer "TXREF"
LEN_DATA	Länge der zu übertragenden Parameterdaten
TXREF	Zu schreibende Parameterdaten

Tabelle 3- 60 Datenstruktur des PUT-Befehls

Put_SET_ANT		Schaltet die Antenne des Readers aus und ein.								
		<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>'N'</td> <td>'A'</td> <td>Mode</td> </tr> </table>	1	2	3	'N'	'A'	Mode		
1	2	3								
'N'	'A'	Mode								
Mode		RF200/RF300, MOBY U/D: <ul style="list-style-type: none"> • 0x01 ≙ Antenne ein • 0x02 ≙ Antenne aus 								
Length		3								
Put_END		Beendet die Kommunikation mit einem Transponder (nur bei MOBY U).								
		<table border="1"> <tr> <td>1</td> <td>2</td> <td>3 ... 10</td> <td>11</td> </tr> <tr> <td>'N'</td> <td>'K'</td> <td>UID</td> <td>Mode</td> </tr> </table>	1	2	3 ... 10	11	'N'	'K'	UID	Mode
1	2	3 ... 10	11							
'N'	'K'	UID	Mode							
UID		UID des Transponders								
Mode		<ul style="list-style-type: none"> • 0x00 ≙ Bearbeitung des Transponders beenden • 0x01 ≙ Bearbeitungspause des Transponders 								
Length		11								

Tabelle 3- 61 WRITE-ID (nur bei RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x67
OFFSETBUFFER	Offset im Sendepuffer "TXREF"
EPCID_UID	Bisherige EPC-ID
LEN_ID	Länge der bisherigen EPC-ID (2-62 Byte)
LEN_DATA	Länge der neuen EPC-ID
PSWD	Passwort 0x00 ≙ kein Passwort
TXREF	Neue EPC-ID

Tabelle 3- 62 KILL-TAG (nur bei RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x6A
EPCID_UID	EPC-ID
LEN_ID	Länge der EPC-ID / UID (2-62 Byte)
PSWD	Passwort muss ≠ 0x00 sein

Tabelle 3- 63 LOCK-TAG-BANK (nur bei RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x79
EPCID_UID	EPC-ID
LEN_ID	Länge der EPC-ID / UID (2-62 Byte)
PSWD	Passwort 0x00 ≙ kein Passwort
LOCK_TAG_BANK_ACTION	Siehe EPC-Norm
LOCK_TAG_BANK_MASK	Siehe EPC-Norm

Tabelle 3- 64 EDIT-BLACKLIST (nur bei RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x7A
EDIT_BLACKLIST_MODE	<ul style="list-style-type: none"> • 0x00 ≙ EPC-ID hinzufügen • 0x01 ≙ Alle "Observed"-Transponder hinzufügen • 0x02 ≙ EPC-ID löschen • 0x03 ≙ Alle löschen
EPCID_UID	EPC-ID 0x00 ≙ unspezifizierter Singletag-Zugriff ¹⁾
LEN_ID	Länge der EPC-ID (2-62 Byte) 0x00 ≙ unspezifizierter Singletag-Zugriff

¹⁾ Wenn bei "EDIT_BLACKLIST_MODE" = 0x00 oder 0x02 ausgewählt wurde, muss die EPC-ID inkl. der ID-Länge angegeben werden.

Tabelle 3- 65 GET-BLACKLIST (nur bei RF61xR/RF68xR)

Parameter	Wert / Beschreibung
CMD	0x6C
OFFSETBUFFER	Offset im Empfangspuffer "RXREF"
RXREF	Gelesene BlackList-IDs

Tabelle 3- 66 Ergebnis von GET-BLACKLIST

Name	Typ	Kommentar
NUM_IDS	INT	Number of MDS
TAG_DATA	IID_IN_I_80[n]	
TAG_DATA[1]	IID_IN_I_80	
Reserved	SINT	
ID_Len	SINT	Length of EPC ID
EPC_ID	SINT[1..62]	EPC-ID
TAG_DATA[2]	IID_IN_I_80	
...	...	
TAG_DATA[n]	IID_IN_I_80	

Tabelle 3- 67 READ-CONFIG

Parameter	Wert / Beschreibung
CMD	0x61
OFFSETBUFFER	Offset im Empfangspuffer "RXREF"
RXREF	Gelesene Reset-Parameter

Tabelle 3- 68 WRITE-CONFIG

Parameter	Wert / Beschreibung
CMD	0x78
OFFSETBUFFER	Offset im Sendepuffer "TXREF"
LEN_DATA	Länge der Parameterdaten
CONFIG	<ul style="list-style-type: none"> • 0x01 ≙ Kommunikations-Reset, keine Konfigurationsdaten • 0x02 ≙ kein Kommunikations-Reset, zu sendende Konfigurationsdaten • 0x03 ≙ Kommunikations-Reset, zu sendende Konfigurationsdaten • 0x80 ≙ kein Kommunikations-Reset, Einzelparameter
TXREF	Zu sendende Konfigurationsdaten

Aufbau des Konfigurationsdatenanhangs von "WRITE-CONFIG"

Für RF360R, RF61xR/RF68xR und RF18xC/RF18xCI sowie RF166C:

- Bei CONFIG = 0x01:
Reset_Reader; LEN_DATA = 0x00
- Bei CONFIG = 0x03:
Für den Baugruppentausch ist es möglich alle Konfigurationsdaten aus dem CM/Reader zu lesen und in der Steuerung zu speichern. Bei einem Baugruppentausch

können diese Daten von der Steuerung in den Reader geladen werden. Dazu werden die Befehle "WRITE-CONFIG" (0x03) für den Download zum CM/Reader und "READ-CONFIG" für den Upload aus dem Reader genutzt.

Byte	Name
0	Strukturkennung (2 Byte)
2	Längenangabe (4 Byte) Länge von Versionskennung und Parameterblock
6	Versionskennung (4 Byte) Mit Hilfe der Kennung können Sie die Konfiguration eindeutig identifizieren. Es handelt sich dabei um einen Zeitstempel im Linux-Format. Der Zeitstempel gibt an, wie viele Sekunden seit dem 1. Januar 1979, 00:00 Uhr vergangen sind. Die Kennung wird beim Erzeugen einer Konfiguration vergeben.
10 ... Ende "DATA"	Parameterblock

LEN_DATA = Größe der Konfigurationsdaten + 6 Byte

- Bei CONFIG = 0x80:

Mit Hilfe dieses Befehls können Sie UHF-Parameter (z. B. die Antennenleistung) an einem RF61xR/RF68xR, sowie die Datum-/Uhrzeit-Parameter eines RF360R, RF166C oder RF18xC/RF18xCI während der Laufzeit ändern.

Byte	Name
0 ... 3	Parameterkennung "PARMID" (4 Byte)
4 ... 7	Parameterwert "VALUE" (4 Byte)

Mögliche Werte für Parameterkennung und -wert finden Sie im Kapitel "Set_Param (Seite 56)".

LEN_DATA = 0x08

Für RF200, RF300, MOBY D/U und Freeport

Tabelle 3- 69 Für RF200, RF300, MOBY D/U und Freeport bei "CONFIG = 0x03"

Byte	1	2...5 ¹⁾	6 ²⁾	7...8	9	10	11	12	13...14	15	16
Wert (RF200, RF300, MOBY U/D)	0x04	0x00	0x0A 0x05	0x00	scan- ning_ time	param	opti- on_1	distance_ limiting	Anzahl der Trans- ponder	field_on_ control	field_on_ time
Wert (Free- port)	0x04	0x00	0x0A	0x00	0x00	0x00	opti- on_1	0x00	0x00	0x00	0x00

¹⁾ Bei dem Kommunikationsmodul RF180C ab V2.2 in Verbindung mit MOBY U sind Byte 4 mit Kalenderwoche und 5 mit Jahr vorbelegt.

²⁾ Bei den im Tabellentitel genannten Readern wird im Byte 6 der Wert "0x0A" (LEN_DATA = 0x10) verwendet. Bei der MOBY I-Migration in RF300-Reader der 2. Generation wird der Wert "0x05" (LEN_DATA = 0x0B) verwendet.

Nachfolgend werden die Parameterwerte der Bytes 9 ... 16 beschrieben.

Tabelle 3- 70 Parameterwerte der Bytes 9 ... 16

Byte	Wert	RFID-System	Bedeutung
Byte 9	scanning_ time	RF200, MOBY D	Bit 0...7: Reserviert (Mit Wert "0" zu belegen.)
		RF300	Bit 0...1: Reserviert (Mit Wert "0" zu belegen.) Bit 2...4: Modulationstiefe <ul style="list-style-type: none"> • 0: 0 % (Default) • 1: 7 % • 2: 10 % • 3: 15 % • 4: 20 % • 5: 25 % • 6: 30 % • 7: 100 % Bit 5...7: Eingangsdämpfung <ul style="list-style-type: none"> • 0: Default (in der Firmware hinterlegte Default-Dämpfung, abhängig vom Reader-Typ und Luftschnittstellenprotokoll) • 1: 0 dB • 2: 5 dB • 3: 10 dB • 4: 15 dB • 7: -5 dB

Byte	Wert	RFID-System	Bedeutung												
		MOBY U	<p>"scanning_time" beschreibt die Standby-Zeit für den Transponder. Erhält der Transponder vor Ablauf der "scanning_time" einen weiteren Befehl, so kann dieser sofort bearbeitet werden. Erhält der Transponder einen Befehl nach Ablauf der "scanning_time", so wird die Befehlsbearbeitung um die "sleep_time" des Transponder verzögert.</p> <p>Eine "scanning_time" sollte nur eingestellt werden, wenn</p> <ul style="list-style-type: none"> • der Transponder mit mehreren Befehlen bearbeitet wird und • der Bearbeitungsvorgang in einer minimalen Zeit abgeschlossen sein muss. <p>Beachten Sie, dass die "scanning_time" die Batterielebensdauer beeinflusst. Je größer die eingestellte "scanning_time", desto kürzer die Batterielebensdauer. Genauere Berechnungen finden Sie im MOBY U-Handbuch für Projektierung, Montage und Service.</p>												
			<table border="1"> <thead> <tr> <th>Wert</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>keine Standby-Zeit (Default)</td> </tr> <tr> <td>0x01</td> <td>7 ms Standby-Zeit</td> </tr> <tr> <td>0x02</td> <td>14 ms Standby-Zeit</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>0xC8</td> <td>1400 ms Standby-Zeit</td> </tr> </tbody> </table>	Wert	Beschreibung	0x00	keine Standby-Zeit (Default)	0x01	7 ms Standby-Zeit	0x02	14 ms Standby-Zeit	0xC8	1400 ms Standby-Zeit
Wert	Beschreibung														
0x00	keine Standby-Zeit (Default)														
0x01	7 ms Standby-Zeit														
0x02	14 ms Standby-Zeit														
...	...														
0xC8	1400 ms Standby-Zeit														

Byte	Wert	RFID-System	Bedeutung	
Byte 10	param	RF200, RF300, MOBY D/U	Einstellung der RFID-Betriebsart und Anwesenheitskontrolle	
			Bit	Beschreibung
			7 ... 5	<p>Dieser Parameter schaltet die Anwesenheitskontrolle am Reader ein oder aus.</p> <p>Mögliche Werte:</p> <ul style="list-style-type: none"> • 0: Betrieb ohne Anwesenheitskontrolle ¹⁾ Antenne ist permanent eingeschaltet. • 1: Betrieb mit Anwesenheitskontrolle Antenne ist permanent eingeschaltet. • 2: RF300 (2. Generation): Betrieb ohne Anwesenheitskontrolle und mit Semaphore-Verfahren (RF300, ISO 14443 [MIFARE Classic, MOBY E]) Antenne ist permanent eingeschaltet. • 3: RF300 (2. Generation): Betrieb mit Anwesenheitskontrolle und mit Semaphore-Verfahren (RF300, ISO 14443 [MIFARE Classic, MOBY E]) Antenne ist permanent eingeschaltet. • 4: Betrieb ohne Anwesenheitskontrolle Antenne ist ausgeschaltet. Die Antenne wird nur eingeschaltet, wenn einer der folgenden Befehle gesendet wird: Read, Write, Init, Tag-Status • 6: RF300 (2. Generation): Betrieb ohne Anwesenheitskontrolle und mit Semaphore-Verfahren (RF300, ISO 14443 [MIFARE Classic, MOBY E]) Antenne ist ausgeschaltet. Die Antenne wird nur eingeschaltet, wenn einer der folgenden Befehle gesendet wird: Read, Write, Init, Tag-Status <p>Informationen zum Semaphore-Verfahren können Sie der Produktinformation "Input-Parameter für das RF300-System" entnehmen.</p> <p>¹⁾ Hinweis zum LED-Verhalten bei MOBY I-Migration: In der Betriebsart "ohne Anwesenheitskontrolle" bleibt die LED des Readers so lange blau, bis ein erster Zugriff (Read/Write/Init) auf den Transponder erfolgt ist. Solange keine weiteren Zugriffe erfolgen oder Befehle ausgeführt werden, blinkt die LED grün. Sobald ein Befehl gesendet wird, leuchtet die LED grün oder orange, abhängig davon ob sich ein Transponder im Antennenfeld befindet.</p>

Byte	Wert	RFID-System	Bedeutung													
			4	<p>RF300, 2. Generation</p> <p>Dieser Parameter schaltet bei RF300, 2. Generation für die Transponder-Typen RF300 und ISO 14443 (MIFARE Classic, MOBY E) den ECC-Modus ein. Bei allen anderen Systemen muss der Wert "0" eingegeben werden.</p> <p>Mögliche Werte:</p> <ul style="list-style-type: none"> • 0: Der ECC-Modus ist ausgeschaltet. • 1: Der ECC-Modus ist eingeschaltet. <p>Im ECC-Modus kann der Reader Bit-Fehler auf dem Transponder mit hoher Wahrscheinlichkeit erkennen. Wenn möglich werden bei einem lesenden Zugriff korrigierte Daten zurückgeliefert (die Daten auf dem Transponder bleiben unverändert). Bei einem schreibenden Zugriff werden - wenn möglich - die betroffenen Daten auf dem Transponder korrigiert. Der ECC-Modus kann ausschließlich bei Transpondern verwendet werden, die zuvor bei gesetztem ECC-Bit vollständig initialisiert und dadurch ECC-formatiert wurden. Dazu wird der Transponder in Blöcke zu 16 Byte eingeteilt, wobei 14 Byte für Nutzdaten und 2 Byte für ECC-Informationen (CRC) reserviert werden. Dies führt zu einer Verringerung des nutzbaren Speichervolumens um ca. 1/8.</p> <p>Wurden Bit-Fehler erkannt und korrigiert, wird im Ausgangsparameter "STATUS" die Warnung "0xF0FE002" ausgegeben. Wenn der Bit-Fehler nicht korrigiert werden konnte, wird der Fehler "0xE1FE0700" ausgegeben.</p> <p>Hinweis: Beachten Sie, dass in diesem Modus ausschließlich die Transponder vom Typ RF300 und ISO 14443 (z. B. MOBY E) unterstützt werden.</p>												
Werte der Bits 3 ... 0																
			Werte der Bits 3 ... 0	<table border="1"> <thead> <tr> <th>Betriebsart</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Reserviert</td> </tr> <tr> <td>1</td> <td>MOBY I</td> </tr> <tr> <td>5</td> <td>MOBY U/D, RF200, RF300</td> </tr> <tr> <td>6</td> <td>MOBY U</td> </tr> <tr> <td>7</td> <td>MOBY D</td> </tr> </tbody> </table>	Betriebsart	Beschreibung	0	Reserviert	1	MOBY I	5	MOBY U/D, RF200, RF300	6	MOBY U	7	MOBY D
Betriebsart	Beschreibung															
0	Reserviert															
1	MOBY I															
5	MOBY U/D, RF200, RF300															
6	MOBY U															
7	MOBY D															
			0	Reserviert												
			1	MOBY I												
			5	MOBY U/D, RF200, RF300												
			6	MOBY U												
			7	MOBY D												

Byte	Wert	RFID-System	Bedeutung
			Hinweis: Beachten Sie, dass nach der Parameter-Änderung das CM neu gestartet werden muss.

Byte	Wert	RFID-System	Bedeutung										
Byte 11	option_1	RF200, RF300, Freeport, MOBY D/U	Dieses Byte ist bitweise kodiert. Es hat standardmäßig den Wert "B#16#0". Mit diesem Byte können Sie besondere Steuerungen im CM/Reader vorgenommen werden.										
			<table border="1"> <thead> <tr> <th>Bit</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>1</td> <td> <p>Mögliche Werte:</p> <ul style="list-style-type: none"> 0 = Das Blinken der "ERR"-LED im Fehlerfall wird nicht zurückgesetzt. 1 = Das Blinken der "ERR"-LED im Fehlerfall wird durch einen "WRITE-CONFIG" zurückgesetzt. <p>Bei RF200/RF300 wird damit das Blinken der "ERR"-LED auf dem Kommunikationsmodul und auf dem Reader zurückgesetzt. Bei Freeport wird ausschließlich das Blinken der "ERR"-LED auf dem Kommunikationsmodul zurückgesetzt.</p> </td> </tr> </tbody> </table>	Bit	Beschreibung	1	<p>Mögliche Werte:</p> <ul style="list-style-type: none"> 0 = Das Blinken der "ERR"-LED im Fehlerfall wird nicht zurückgesetzt. 1 = Das Blinken der "ERR"-LED im Fehlerfall wird durch einen "WRITE-CONFIG" zurückgesetzt. <p>Bei RF200/RF300 wird damit das Blinken der "ERR"-LED auf dem Kommunikationsmodul und auf dem Reader zurückgesetzt. Bei Freeport wird ausschließlich das Blinken der "ERR"-LED auf dem Kommunikationsmodul zurückgesetzt.</p>						
		Bit	Beschreibung										
		1	<p>Mögliche Werte:</p> <ul style="list-style-type: none"> 0 = Das Blinken der "ERR"-LED im Fehlerfall wird nicht zurückgesetzt. 1 = Das Blinken der "ERR"-LED im Fehlerfall wird durch einen "WRITE-CONFIG" zurückgesetzt. <p>Bei RF200/RF300 wird damit das Blinken der "ERR"-LED auf dem Kommunikationsmodul und auf dem Reader zurückgesetzt. Bei Freeport wird ausschließlich das Blinken der "ERR"-LED auf dem Kommunikationsmodul zurückgesetzt.</p>										
		RF300 (2. Generation)	Folgende Werte sind zulässig:										
			<table border="1"> <thead> <tr> <th>Wert</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>Das Blinken der LED im Fehlerfall wird nur durch Abschalten der Versorgungsspannung am Reader zurückgesetzt.</td> </tr> <tr> <td>0x01</td> <td>Das Blinken der LED im Fehlerfall wird nach dem erfolgreichen Ausführen eines Befehls zurückgesetzt. Dies betrifft ausschließlich Kommunikationsfehler (z. B. 0xE1FE0200, 0xE2FE0100), andere Fehler werden dadurch nicht zurückgesetzt.</td> </tr> <tr> <td>0x02</td> <td>Das Blinken der LED im Fehlerfall wird durch einen "init_run" bzw. "WRITE-CONFIG" mit "Init" (RESET) zurückgesetzt.</td> </tr> <tr> <td>0x03</td> <td>Das Blinken der LED im Fehlerfall wird durch einen "init_run" bzw. "WRITE-CONFIG" mit "Init" (RESET) oder nach dem erfolgreichen Ausführen eines Befehls zurückgesetzt.</td> </tr> </tbody> </table>	Wert	Beschreibung	0x00	Das Blinken der LED im Fehlerfall wird nur durch Abschalten der Versorgungsspannung am Reader zurückgesetzt.	0x01	Das Blinken der LED im Fehlerfall wird nach dem erfolgreichen Ausführen eines Befehls zurückgesetzt. Dies betrifft ausschließlich Kommunikationsfehler (z. B. 0xE1FE0200, 0xE2FE0100), andere Fehler werden dadurch nicht zurückgesetzt.	0x02	Das Blinken der LED im Fehlerfall wird durch einen "init_run" bzw. "WRITE-CONFIG" mit "Init" (RESET) zurückgesetzt.	0x03	Das Blinken der LED im Fehlerfall wird durch einen "init_run" bzw. "WRITE-CONFIG" mit "Init" (RESET) oder nach dem erfolgreichen Ausführen eines Befehls zurückgesetzt.
			Wert	Beschreibung									
0x00	Das Blinken der LED im Fehlerfall wird nur durch Abschalten der Versorgungsspannung am Reader zurückgesetzt.												
0x01	Das Blinken der LED im Fehlerfall wird nach dem erfolgreichen Ausführen eines Befehls zurückgesetzt. Dies betrifft ausschließlich Kommunikationsfehler (z. B. 0xE1FE0200, 0xE2FE0100), andere Fehler werden dadurch nicht zurückgesetzt.												
0x02	Das Blinken der LED im Fehlerfall wird durch einen "init_run" bzw. "WRITE-CONFIG" mit "Init" (RESET) zurückgesetzt.												
0x03	Das Blinken der LED im Fehlerfall wird durch einen "init_run" bzw. "WRITE-CONFIG" mit "Init" (RESET) oder nach dem erfolgreichen Ausführen eines Befehls zurückgesetzt.												
0x00 (reserviert)													
RF200													
RF300 (nur RF350R, RF380R)	<p>Hinweis: Dieser Parameter richtet sich an geschulte Anwender. Nicht geschulte Anwender empfiehlt Siemens den Default-Wert zu verwenden.</p> <p>Im Bit 0...3 kann die Ausgangsleitung eingestellt werden. Diese muss bei dem Readern RF380R der 2. Generation (6GT2801-3BAx0) nicht manuell eingestellt werden, da die Leistungsgrenzen abhängig vom Reader-Transponder-Abstand automatisch optimiert werden. Aus Gründen der Kompatibilität kann diese Einstellung trotzdem gesetzt werden. Beachten Sie, dass die Werte "02", "03" und "04" eine Reduktion der Leistung um ca. 50 % bewirken. Einstellungen außerhalb des Bereichs (02 ... 08) haben den Effekt, dass der Standardwert (0,6 W) festgelegt wird. In diesem Fall gibt es aus Kompatibilitätsgründen keine Fehlermeldung.</p>												
Byte 12	distance_limiting	RF200	0x00 (reserviert)										

Byte	Wert	RFID-System	Bedeutung																
			Bit 4...7: Antennentyp (Reader RF350R der 2. Generation) Zur Verbesserung der Kommunikationsstabilität. Folgende Werte sind zulässig: <ul style="list-style-type: none"> • 0: Unspezifiziert (Default) • 1: ANT 1 • 2: ANT 3 • 3: ANT 3S • 4: ANT 8 • 5: ANT 12 • 6: ANT 12 (6GT2398-1CC10 mit integriertem Antennenanschlusskabel 0,6 m) • 7: ANT 18 • 8: ANT 18 (6GT2398-1CA10 mit integriertem Antennenanschlusskabel 0,6 m) • 9: ANT 30 																
		MOBY U	Reichweitenbegrenzung <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">normale Sendeleistung¹⁾</th> <th style="width: 50%;">reduzierte Sendeleistung</th> </tr> </thead> <tbody> <tr> <td>0x05 = 0,5 m</td> <td>0x85</td> </tr> <tr> <td>0x0A = 1,0 m</td> <td>0x8A</td> </tr> <tr> <td>0x0F = 1,5 m</td> <td>0x8F</td> </tr> <tr> <td>0x14 = 2,0 m</td> <td>0x91</td> </tr> <tr> <td>0x19 = 2,5 m</td> <td>0x99</td> </tr> <tr> <td>0x1E = 3,0 m</td> <td>0x9E</td> </tr> <tr> <td>0x23 = 3,5 m</td> <td>0xA3</td> </tr> </tbody> </table> <p>Stellen Sie die reduzierte Sendeleistung ein, wenn mehrere Reader nahe beieinander positioniert sind oder wenn Transponder, die sich in der Nähe eines Readers aufhalten, später oder nicht mehr erkannt werden sollen. Nachteil: Das Antennenfeld wird kleiner und damit steht für die Kommunikation weniger Zeit zur Verfügung bzw. es muss genauer positioniert werden.</p> <p>¹⁾ Zwischenwerte in Abstufungen von 0,1 m sind möglich (0x02, 0x03, ..., 0x23)</p>	normale Sendeleistung ¹⁾	reduzierte Sendeleistung	0x05 = 0,5 m	0x85	0x0A = 1,0 m	0x8A	0x0F = 1,5 m	0x8F	0x14 = 2,0 m	0x91	0x19 = 2,5 m	0x99	0x1E = 3,0 m	0x9E	0x23 = 3,5 m	0xA3
normale Sendeleistung ¹⁾	reduzierte Sendeleistung																		
0x05 = 0,5 m	0x85																		
0x0A = 1,0 m	0x8A																		
0x0F = 1,5 m	0x8F																		
0x14 = 2,0 m	0x91																		
0x19 = 2,5 m	0x99																		
0x1E = 3,0 m	0x9E																		
0x23 = 3,5 m	0xA3																		
		MOBY D	Sendeleistung von 0,5 W bis 10 W in 0,25 W-Schritten nur wirksam bei SLG D10S; bei SLG D11S / D12S ist eine Leistung von 1 W (04 hex) eingestellt und kann nicht geändert werden. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Wert</th> <th>Sendeleistung</th> </tr> </thead> <tbody> <tr> <td>0x02</td> <td>0,5 W</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>0x10</td> <td>4,0 W (Default)</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>0x28</td> <td>10,0 W</td> </tr> </tbody> </table>	Wert	Sendeleistung	0x02	0,5 W	0x10	4,0 W (Default)	0x28	10,0 W				
Wert	Sendeleistung																		
0x02	0,5 W																		
...	...																		
0x10	4,0 W (Default)																		
...	...																		
0x28	10,0 W																		
Byte 13...14	Anzahl der Transponder	RF300	Mit Wert "0x0001" zu belegen.																
Byte 15	field_on_control	RF200, RF300, MOBY D	Mit Wert "0x00" zu belegen.																
		MOBY U	BERO-Betriebsart; automatisches Ein-/Ausschalten des Antennenfeldes. Der Befehl "Antenne EIN/AUS" wird von der BERO-Betriebsart überlagert. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Wert</th> <th>Beschreibung der Betriebsart</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Wert	Beschreibung der Betriebsart														
Wert	Beschreibung der Betriebsart																		

Byte	Wert	RFID-System	Bedeutung
			0x00 ohne BEROs; keine Reader-Synchronisation
			0x01 ein oder zwei BEROs Die BEROs sind logisch ODER verknüpft. Während der Zeit der Betätigung eines BEROs ist das Antennenfeld eingeschaltet.
			0x02 ein oder zwei BEROs Der 1. BERO schaltet das Antennenfeld ein und der 2. BERO schaltet das Antennenfeld aus. Wenn zwei BEROs vorhanden sind und ein "field_ON_time" parametrier ist, wird das Antennenfeld automatisch ausgeschaltet, wenn der 2. BERO nicht innerhalb dieser BERO-Zeit schaltet. Ist keine "field_ON_time" parametrier, bleibt das Antennenfeld bis zur Betätigung des 2. BERO eingeschaltet.
			0x03 Reader-Synchronisation über Kabelverbindung aktivieren Weitere Informationen finden Sie im MOBY U-Handbuch für Projektierung, Montage und Service.

Byte	Wert	RFID-System	Bedeutung	
Byte 16	field_on_time	RF200	Transponder-Typ	
		0x01	Beliebiger ISO-Transponder	
		RF300	Auswahl der verwendeten Transponder-Typen Die Reader der 2. Generation verwenden bei dem Wert "0x01"/"0x31" (ISO 15693 allgemein) immer die ISO-Befehle, mit denen die größte Performance bei dem jeweiligen Transponder erreicht werden kann. Bei den Readern der 1. Generation aktiviert der Wert "0x01" den allgemeinen ISO-Modus mit rudimentären ISO-Befehlen. Mit dieser Einstellung ist in der Regel die Performance eingeschränkt, der Betrieb aber mit jedem ISO-kompatiblen Transponder grundsätzlich gewährleistet. Alle ISO 15693-Transponder (MDS D), die im Systemhandbuch "SIMATIC RF300" angegebenen sind, unterstützen diese ISO-Befehle. Folgende Werte können eingestellt werden:	
		Wert	Transponder-Typ	Beschreibung
		0x00	RF300 (RF3xxT)	Für alle Transponder des Typs "RF3xxT"
		0x01	ISO 15693 allgemein	Beliebiger ISO-Transponder Aktivierung des allgemeinen ISO-Modus mit rudimentären ISO-Befehlen. Mit dieser Einstellung ist in der Regel der Betrieb mit jedem ISO-kompatiblen Transponder grundsätzlich gewährleistet.
		0x03	ISO 15693 (MDS D3xx; Infineon)	z. B. MDS D324, D339
		0x04	ISO 15693 (MDS D4xx, Fujitsu - 2 kB)	z. B. MDS D421, D422, D423, D424, D425, D426, D428, D460
		0x05	ISO 15693 (MDS D1xx, NXP)	z. B. MDS D100, D124, D126, D139, D150, D165
		0x06	ISO 15693 (MDS D2xx, TI)	z. B. MDS D200
		0x07	ISO 15693 (MDS D261, STM)	z. B. MDS D261
		0x08 ¹⁾	ISO 15693 (MDS D5xx, Fujitsu - 8 kB)	z. B. MDS D521, D522, D524, D525, D528
		0x10 ¹⁾	RF300 (RF3xxT)	Für alle Transponder des Typs "RF3xxT"
		0x20 ¹⁾	ISO 14443 (MIFARE Classic MOBY E, E6xx)	z. B. MDS E600, E611, E623, E624
0x2F ¹⁾	ISO 14443 (MIFARE Classic)	Offener Modus zum Ändern des Schlüsselmaterials auf dem MIFARE Classic-Transponder. Informationen zum offenen Modus können Sie der Produktinformation "Input-Parameter für das RF300-System" entnehmen.		

Byte	Wert	RFID-System	Bedeutung	
			0x31 ¹⁾	General Mode (ISO, RF300, MOBY E) Aktivierung des "General Mode" für die Bearbeitung der Transponder-Typen RF300, ISO 15693 und ISO 14443 (MIFARE Classic, MOBY E). Mit dieser Einstellung ist der Betrieb mit jedem der kompatiblen Transponder grundsätzlich gewährleistet.
			0x40 ¹⁾	P2P-Master Reader wird zu einem P2P-Master
			0x4F ¹⁾	P2P-Slave Reader wird zu einem P2P-Slave
			¹⁾ Betrifft ausschließlich die Reader der 2. Generation. Beachten Sie, dass einzelne Einstellungen bzw. Transponder-Familien kombiniert werden können (z. B. ISO 15693 allgemein + RF300). In diesem Fall müssen die entsprechenden hex-Werte kombiniert werden (ISO 15693 allgemein [0x01] + RF300 [0x10] = 0x11). Hinweis: <ul style="list-style-type: none"> ISO 15693: Folgende ISO-Spezialfunktionen werden nicht unterstützt: <ul style="list-style-type: none"> AFI (Application Family Identifier) DSFID (Data Storage Format Identifier) Chip-spezifische Zusatzfunktionen wie EAS, Kill-Kommandos, etc. 	
		MOBY U	Zeit für BERO-Betriebsart (field_ON_control = 02)	
			Wert	Beschreibung
			0x00	Die Zeitüberwachung ist abgeschaltet. Für die Feldabschaltung wird der 2. BERO benötigt.
			0x01 ... 0xFF	1 ... 255 s Einschaltzeit für das Reader-Antennenfeld
		MOBY D	Transponder-Typ	
			0 ... 255	Transponder-Typ
			0x00	I-Code 1 (z. B. MDS D139)
			0x01	ISO-Transponder
			0x02	I-Code 1 und ISO-Transponder
			0x03	ISO-my-D (nur bei SLG D10S; bei SLG D11S / D12S wird für den ISO-my-D der Wert "0x01" parametrieret)
			0x04	ISO-FRAM (nur bei SLG D11S / D12S; bei SLG D10S wird für den ISOFRAM der Wert "0x01" parametrieret)

3.5.3.3 Erweiterte Befehle für Optische Lesesysteme (MV400/MV500)

Der Befehl "WRITE-CONFIG"

Während der Initialisierung ("INIT") wird vom Ident-Profil automatisch der Befehl "WRITE-CONFIG" ausgeführt. Die Parameterwerte des "WRITE-CONFIG"-Befehls sind davon abhängig, ob das Ident-Profil mit oder ohne Kommunikationsmodul verwendet wird.

Tabelle 3- 71 WRITE-CONFIG

CMD	OFFSET BUFFER	LEN_DATA	CONFIG	TXREF
0x78	Offset im Sendepuffer "TXREF"	Länge der Parameterdaten	<ul style="list-style-type: none"> 0x01 ≙ Kommunikations-Reset ohne Konfigurationsdaten (LEN_DATA = 0) 0x03 ≙ Kommunikations-Reset mit zu sendenden Konfigurationsdaten 	Zu sendende Konfigurationsdaten

Beachten Sie, dass der Kommunikations-Reset ohne Konfigurationsdaten (CONFIG = 0x01) ausschließlich dann verwenden können, wenn Sie das Lesegerät ohne Kommunikationsmodul über PROFINET IO betreiben. Der Reset ohne Konfigurationsdaten entspricht dabei dem "INIT"-Befehl ohne Programmanwahl.

Aufbau der Konfigurationsdatenanhangs von "WRITE-CONFIG"

Tabelle 3- 72 MV400/MV500 bei "CONFIG = 0x03"; "LEN_DATA = 0x10"

Byte	1	2...5	6	7...8	9	10	11	12 ... 13	14	15	16
Wert	0x04	0x00	0x0A	0x00	0x00	0x25	0x02	0x00	0x01	0x00	0x00 ... 0x0F ¹⁾

¹⁾ 0x00: "INIT" ohne Programmanwahl

0x01 ... 0x0F: Nummer des zu startenden Programms ("INIT" mit Programmanwahl)

Der Befehl "PHYSICAL-WRITE"

Die Optischen Lesesysteme MV400/MV500 besitzen weitere Befehle, die mit dem Befehl "PHYSICAL-WRITE" übertragen werden.

Tabelle 3- 73 PHYSICAL-WRITE

CMD	OFFSET BUFFER	ADDR_T AG	LEN_DATA	TXREF
0x71	Offset im Sende- puffer "TXREF"	0x0000	Länge der an das Lesegerät zu übertragenden Daten:	An das Lesegerät zu übertragende Attribute mit Daten. Das erste "Byte" beinhaltet den Befehlsbezeichner:
			• 0x02	• 0x01 = Programmwechsel
			• 0x01	• 0x02 = Programmnummer lesen aktivieren
			• Matchstring-Länge + 3	• 0x03 = Matchstring schreiben
			• 0x01	• 0x04 = Matchstring lesen aktivieren
			• 0x01	• 0x05 = DISA-Bit setzen
			• 0x01	• 0x06 = DISA-Bit zurücksetzen
			• Gesamtlänge in XMATCH-Nutzdaten + 4	• 0x07 = Triggersynchrones Matchstring schreiben (XMATCH)
• 0x07	• 0x08 = Digital Out setzen			

Tabelle 3- 74 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x01 (Programmwechsel)

Adresse	Wert	Bedeutung
0x0000	0x01	Befehlbezeichner "Programmwechsel"
0x0001	0x00 ... 0x0F	Nummer des Programms

Tabelle 3- 75 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x02 (Programmnummer lesen aktivieren)

Adresse	Wert	Bedeutung
0x0000	0x02	Befehlbezeichner "Programmnummer lesen"

Tabelle 3- 76 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x03 (Matchstring schreiben)

Adresse	Wert	Bedeutung
0x0000	0x03	Befehlbezeichner "Matchstring schreiben"
0x0001	0x00...0xFF	Länge Matchstring High Byte
0x0002	0x00...0xFF	Länge Matchstring Low Byte
0x0003	--	1. Zeichen des Matchstrings
...
n + 2	--	(n-1). Zeichen des Matchstrings
n + 3	--	n. Zeichen des Matchstrings

Tabelle 3- 77 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x04 (Matchstring lesen aktivieren)

Adresse	Wert	Bedeutung
0x0000	0x04	Befehlbezeichner "Match-String lesen"

Tabelle 3- 78 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x05 (DISA-Bit setzen)

Adresse	Wert	Bedeutung
0x0000	0x05	Befehlbezeichner "DISA-Bit setzen"

Tabelle 3- 79 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x06 (DISA-Bit zurücksetzen)

Adresse	Wert	Bedeutung
0x0000	0x06	Befehlbezeichner "DISA-Bit zurücksetzen"

Tabelle 3- 80 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x07 (XMATCH)

Adresse	Wert	Bedeutung
0x0000	0x07	Befehlsbezeichner "XMATCH"
0x0001	0x00	Reserviert
0x0002	Ausführliche Informationen finden Sie im Handbuch "SIMATIC MV420 / SIMATIC MV440" oder "SIMATIC MV500".	XMATCH-Nutzdaten
...		
0xN		

Tabelle 3- 81 Befehl-Datenbereich "TXREF" Befehlsbezeichner 0x08 (Digital Out setzen)

Adresse	Wert	Bedeutung
0x0000	0x08	Befehlbezeichner "Digital Out setzen"
0x0001	0x01...0x04	Nummer des logischen externen Signals. Entspricht "EXT_1", "EXT_2", "EXT_3" und "EXT_4".
0x0002	0x00...0x02	Pegel des Signals <ul style="list-style-type: none"> • 0x00: Pegel statisch auf "low" setzen. • 0x01: Pegel statisch auf "high" setzen. • 0x02: Pegel für die konfigurierte Pulszeit auf "high" setzen.
0x0003	0x01...0x07	Verknüpfungsart <ul style="list-style-type: none"> • 0x01: logisches "Oder" • 0x02: logisches "Und" • 0x03: logisches "Exklusiv-Oder" • 0x04: keine Verknüpfung • 0x05: logisches "Oder Nicht" • 0x06: logisches "Und Nicht" • 0x07: logisches "Exklusiv-Oder Nicht"
0x0004	0x00...0x05	Logisches Signal mit dem verknüpft wird. Wenn die Verknüpfungsart 0x4 ist, hat der Parameter keine Bedeutung. <ul style="list-style-type: none"> • 0x00: logisches Signal "IN_OP" • 0x01: logisches Signal "TRD" • 0x02: logisches Signal "RDY" • 0x03: logisches Signal "READ" • 0x04: logisches Signal "MATCH" • 0x05: logisches Signal "NOK"
0x0005	0x00	Reserviert, muss zur Wahrung der Aufwärtskompatibilität 0x00 sein.
0x0006	0x00	Reserviert, muss zur Wahrung der Aufwärtskompatibilität 0x00 sein.

Der Befehl "PHYSICAL-READ"

Der Befehl "PHYSICAL-READ" wird für folgende Funktionen verwendet:

- Lesen von Codes
- Folgebefehl nach "Programmnummer lesen aktivieren" zum Auslesen der Programmnummer
- Folgebefehl nach "Matchstring Lesen aktivieren" zum Auslesen des Matchstrings

Tabelle 3- 82 PHYSICAL-READ

CMD	OFFSET BUFFER	ADDR_T AG	LEN_DATA	RXREF
0x70	Offset im Sendepuffer "RXREF"	0x0000	Länge der vom Lesegerät abzuholenden Daten:	Vom Lesegerät abgeholte Daten:
			• \geq Codelänge +2	• Code-Daten
			• = 0x01	• Programmnummer
			• \geq Matchstring-Länge +2	• Matchstring

3.5.3.4 Wirkung der Befehle

Die verwendeten Befehle wirken wie folgt:

- Die Eingangsparameter "INIT" und "SRESET" unterbrechen die Befehlsausführung innerhalb des Kommunikationsmoduls/Readers.
- Die Fertigmeldung die dem "INIT" oder "SRESET" folgt ("DONE" bzw. "ERROR") bezieht sich immer auf den Eingangsparameter "INIT" oder "SRESET" und nicht auf den unterbrochenen Befehl.
- Der Eingangsparameter "INIT" setzt die Kommunikation zwischen Ident-Profil und dem Kommunikationsmodul zurück. Nach dem "harten" Zurücksetzen des Kommunikationsmoduls überträgt das Ident-Profil automatisch den Befehl "WRITE-CONFIG" zum Kommunikationsmodul. Deswegen ist es zwingend erforderlich, dass Sie im ersten Element des Befehlspeuffers "CMDREF" den Befehl "WRITE-CONFIG" hinterlegen.
- Der Befehl "WRITE-CONFIG" setzt alle Funktionen innerhalb des Kommunikationsmoduls zurück, ausgenommen die Kommunikation.
- Der Parameter "SRESET" unterbricht den laufenden Befehl.

3.5.3.5 Befehlsbearbeitung

Gehen Sie wie folgt vor, um die Befehle zu bearbeiten:

1. Beschreiben Sie den Parameter "CMDREF" (Array [1...n]) mit den gewünschten Befehlen.

Der Inhalt von "CMDREF" = [1] ist für die Initialisierung reserviert. Er wird ausgeführt, wenn der Eingang "INIT" des Ident-Profil gesetzt wird und "CMDSEL" = [1] ist.

2. Übertragen Sie die zu schreibenden Daten in den Sendedaten-Puffer "TXBUF".
3. Wählen Sie mit dem Parameter "CMDSEL" den zuvor beschriebenen Befehl aus (Array [1...n]).
4. Führen Sie mit Hilfe des Parameters "EXECUTE" den Befehl aus ("EXECUTE" = 1).

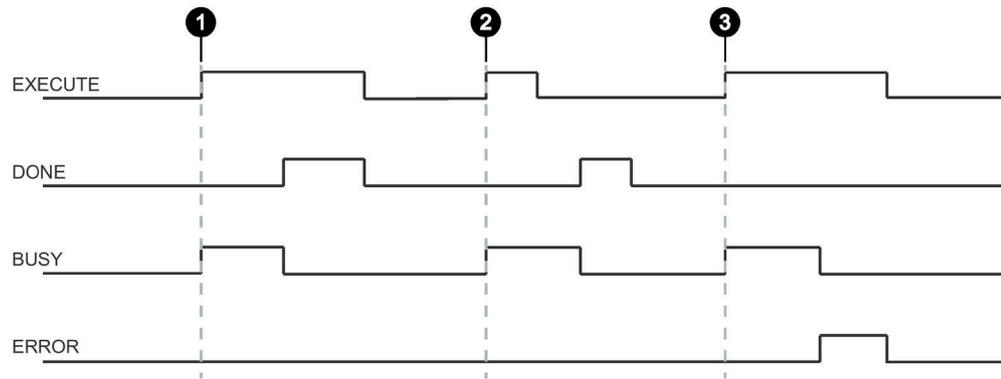
Warten Sie, bis die Bits "BUSY = FALSE" und "DONE = TRUE" gesetzt werden.

Der Befehl ist nun fehlerfrei abgearbeitet.

Wenn "ERROR = TRUE" gesetzt wird, dann fahren Sie mit Punkt 5 fort. Andernfalls fahren Sie mit Punkt 6 fort.

5. Werten Sie aufgetretene Fehler aus.
6. Setzen Sie das Bit "EXECUTE" zurück.

Das folgende Diagramm zeigt den zeitlichen Ablauf des Ident-Profils. Ein Befehlsstart erfolgt immer mit der positiven Flanke von "EXECUTE", "INIT" oder "SRESET".



- Fall ① Durch Setzen von EXECUTE (EXECUTE = 1) wird die Funktion/Anweisung gestartet. Wurde der Auftrag erfolgreich beendet (DONE = 1) müssen Sie EXECUTE wieder zurücksetzen. DONE wird zeitgleich zurückgesetzt.
- Fall ② EXECUTE wird nur für einen Zyklus gesetzt. Sobald BUSY gesetzt ist (und DONE zurückgesetzt ist) können Sie EXECUTE wieder zurücksetzen. Wurde der Auftrag erfolgreich beendet, wird DONE für einen Zyklus gesetzt.
- Fall ③ Handhabung wie Fall 1, jedoch mit Fehlerausgabe. Sobald ERROR gesetzt ist, steht in der Ausgabe STATUS der genaue Fehlercode. ERROR und STATUS behalten ihren Wert solange bei, wie EXECUTE gesetzt ist.

Bild 3-31 Allgemeiner Ablauf des Ident-Profils

3.5.3.6 Neustart und Wiederanlauf parametrieren

Ein Neustart des Kommunikationsmoduls und des Readers erfolgt durch das Setzen des Parameters "INIT". Mit dem Parameter werden das CM bzw. der Reader und das Ident-Profil neu parametrieren und synchronisiert.

Ein "INIT" ist notwendig nach:

- dem Einschalten oder Wiederanlauf der SIMATIC Steuerung (OB 100 / Startup)
- dem Einschalten der Versorgungsspannung am CM/Reader
- dem Anstecken des Readers am CM
- der Unterbrechung der PROFIBUS/PROFINET-Kommunikation
- einer Fehlermeldung durch den Parameter "STATUS"

3.5.3.7 Verkettung

Mit dem Ident-Profil sowie dem "Advanced"-Baustein ist es möglich verkettete Befehle zu versenden. Verkettete Befehle werden komplett an den Reader geschickt, ohne dass auf das Ergebnis des ersten Befehls gewartet wird. Diese Funktion ermöglicht es Ihnen verschiedene Transponder-Befehle mit einem Befehlsstart auszuführen.

Bei beiden Bausteinen steht Ihnen ein Befehlspeicher von 100 Befehlen zur Verfügung ("Array [1...n]" des "IID_CMD_STRUCT"). In jeder Befehlsstruktur befindet sich ein "Chained"-Bit. Dieses Bit muss für jeden verketteten Befehl gesetzt werden. Im letzten verketteten Befehl darf dieses Bit nicht gesetzt werden, damit der Baustein erkennt, dass die Kette beendet ist.

Hinweis

Verkettungsfunktion ist gerätespezifisch

Prüfen Sie bitte, ob das von Ihnen verwendete Ident-Gerät Verkettung unterstützt.

Die Verkettung wird zum Veröffentlichungsdatum dieses Handbuchs von den Kommunikationsmodulen RF120C, RF170C und RF180C nicht unterstützt.

Befehlsübersicht

Tabelle 3- 83 Übersicht der Befehle, bei denen eine Verkettung möglich ist

Befehl	Befehlscode		Beschreibung
	HEX	ASCII	
PHYSICAL-READ	0x70	'p'	Liest Daten von einem Transponder / optischen Lesesystem durch Angabe der physikalischen Anfangsadresse, der Länge und des Passworts aus.
PHYSICAL-WRITE	0x71	'q'	Schreibt Daten auf einen Transponder / optischen Lesesystem durch Angabe der physikalischen Anfangsadresse, der Länge und des Passworts.
READER-STATUS	0x74	't'	Liest den Status eines Kommunikationsmoduls aus. Dieser Befehl darf nicht der letzte Befehl innerhalb einer Verkettung sein.
TAG-STATUS	0x73	's'	Liest den Status eines Transponders aus.
INVENTORY	0x69	'i'	Fordert eine Liste aller gegenwärtig zugänglichen Transponder innerhalb der Antennenreichweite an.
FORMAT	0x66	'f'	Initialisiert den Transponder.
PUT	0x65	'e'	Überträgt weitere Befehle, die nicht im Normprofil-Standard festgelegt sind. Dazu ist für jeden Befehl eine entsprechende Datenstruktur im Sendedaten-Puffer definiert.
WRITE-ID	0x67	'g'	RF61xR/RF68xR: Schreibt eine neue EPC-ID in den Transponder.
KILL-TAG	0x6A	'j'	RF61xR/RF68xR: Der Transponder wird permanent deaktiviert.
LOCK-TAG-BANK	0x79	'y'	RF61xR/RF68xR: Definiert ein Passwort für den Transponder-Zugriff.

Beispiel einer Befehlsstruktur

Tabelle 3- 84 Beispiel einer Befehlsstruktur mit 3 Befehlen (ohne EPC-ID)

Befehl	Parameter	Wert	Beschreibung
Befehl 1	IID_CMD_STRUCT[2].CMD	0x69	Durchführen eines Inventory mit der Dauer von 2 Inventories.
	IID_CMD_STRUCT[2].ATTRIBUTES	0x80	
	IID_CMD_STRUCT[2].EXT_UHF. INVENTORY.DURATION	2	
	IID_CMD_STRUCT[2].EXT_UHF. INVENTORY.DUR_UNIT	1	
	IID_CMD_STRUCT[2].OPTIONS.CHAINED	true	
Befehl 2	IID_CMD_STRUCT[3].CMD	0x70	Lesen von 10 Byte aus der User-Bank ab Adresse 0.
	IID_CMD_STRUCT[3].EXT_UHF. MEM_BANK	3	
	IID_CMD_STRUCT[3].LEN_DATA	10	
	IID_CMD_STRUCT[3].ADDR_TAG	0	
	IID_CMD_STRUCT[3].OPTIONS.CHAINED	true	
Befehl 3	IID_CMD_STRUCT[4].CMD	0x71	Schreiben von 10 Byte in die User-Bank ab Adresse 20.
	IID_CMD_STRUCT[4].EXT_UHF. MEM_BANK	3	
	IID_CMD_STRUCT[4].LEN_DATA	10	
	IID_CMD_STRUCT[4].ADDR_TAG	20	
	IID_CMD_STRUCT[4].OPTIONS.CHAINED	false	

Bei der Verkettung kann der komplette "IID_CMD_STRUCT"-Puffer ("IID_CMD_STRUCT[1...n]") genutzt werden. Der Kettenanfang wird über den Parameter "CMDSEL" eingestellt.

Wenn mehrere Befehle in der Kette ausgeführt werden, bei denen Daten zurückgeliefert werden, kann die Position der Daten im Empfangspuffer "RXREF" über den Parameter "IID_CMD_STRUCT[x].OFFSETBUFFER" für jeden einzelnen Befehl eingestellt werden.

Hinweis

"IID_CMD_STRUCT[1]" für "INIT" reserviert

Bei dem Ident-Profil ist der Parameter "IID_CMD_STRUCT[1]" im Regelfall für "INIT" reserviert. Wenn "IID_CMD_STRUCT[1]" für einen anderen Befehl genutzt werden soll, müssen Sie sicherstellen, dass bei einem "INIT" die Reset-Parameter in diesen Parameter geschrieben werden.

3.5.3.8 Befehlswiederholung

Das Ident-Profil unterstützt die Befehlswiederholung (Repeat-Kommando).

Hinweis

Befehlswiederholung ist gerätespezifisch

Prüfen Sie bitte, ob das von Ihnen verwendete Ident-Gerät Befehlswiederholung unterstützt.

Die Befehlswiederholung wird zum Veröffentlichungsdatum dieses Handbuchs von den Readern RF61xR/RF68xR (ab V3.0) und RF360R (ab FW V2.0), sowie den Kommunikationsmodulen ASM 456 (ab V5.0), RF166C und RF18xC/RF18xCI unterstützt.

Arbeitsweise

Nach Neustart (bzw. "INIT") des Readers überträgt das Ident-Profil einmalig den Befehl bzw. die Befehlskette zum Reader. Die Befehlsübertragung geschieht automatisch mit dem ersten durchgeführten "EXECUTE". Dieser Befehl (bzw. der letzte Befehl oder die Befehlskette) bleibt immer im Reader zwischengespeichert. Wird die Befehlswiederholung gestartet, so wird der zwischengespeicherte Befehl im Reader erneut ausgeführt und das/die Ergebnis/se zum Ident-Profil übertragen.

Stellen Sie sicher, dass die zu wiederholenden Befehle im Parameter "EPCID_UID" den Wert 0 aufweisen. Hat die EPCID_UID einen anderen Wert, wird eine Fehlermeldung erzeugt.

Auswirkungen der Befehlswiederholung

- Der PROFIBUS-/PROFINET-Datentransfer wird minimiert. Diese Minimierung wirkt sich besonders bei großen Bus-Konfigurationen und langsamen Übertragungsraten positiv aus.
- Der Reader bearbeitet unabhängig von dem Ident-Profil jeden Transponder. Das wirkt sich vor allem auf Gate-Applikationen vorteilhaft aus, da alle Transponder immer mit der vollen Reader-Scan-Geschwindigkeit erfasst werden.
- Bei Steuerungen mit nur wenigen Systemressourcen für azyklische Telegramme wird der Gesamtdatendurchsatz erheblich erhöht.

Befehlsübersicht

Tabelle 3- 85 Übersicht der Befehle, bei denen eine Befehlswiederholung möglich ist

Befehl	Befehlscode		Beschreibung
	HEX	ASCII	
PHYSICAL-READ	0x70	'p'	Liest Daten von einem Transponder / optischen Lesesystem durch Angabe der physikalischen Anfangsadresse, der Länge und des Passworts aus.
PHYSICAL-WRITE	0x71	'q'	Schreibt Daten auf einen Transponder / optischen Lesesystem durch Angabe der physikalischen Anfangsadresse, der Länge und des Passworts.
READER-STATUS	0x74	't'	Liest den Status des Readers aus.
TAG-STATUS	0x73	's'	Liest den Status eines Transponders aus.
INVENTORY	0x69	'i'	Fordert eine Liste aller gegenwärtig zugänglichen Transponder innerhalb der Antennenreichweite an.
FORMAT	0x66	'f'	Initialisiert den Transponder.
KILL-TAG	0x6A	'j'	Der Transponder wird permanent deaktiviert.
LOCK-TAG-BANK	0x79	'y'	Definiert ein Passwort für den Transponder-Zugriff.

Befehlswiederholung starten

Sie haben die Möglichkeit die Befehlswiederholung mit oder ohne Befehlsübertragung auszuführen. Nachfolgend werden die verschiedenen Vorgehensweisen beschrieben.

Ablauf des Repeat-Kommandos mit gleichzeitiger Befehlsübertragung:

1. Aktivieren Sie ggf. den "Presence_Mode".
2. Starten Sie den Befehl mit Hilfe des Eingangsparameters "EXECUTE" und gleichzeitig gesetztem "RPTCMD". ①
 Der Befehl wird bearbeitet und das Ergebnis an das Ident-Profil übertragen.
 Das Repeat-Kommando wird auf dem Reader aktiviert.
3. Der Reader bestätigt die Aktivierung über den Ausgangsparameters "RPTACT" am Ident-Profil. Die Bestätigung erfolgt erst, nachdem der erste Befehl abgearbeitet wurde. ②
 Der Reader führt den Befehl automatisch aus, sobald ein Transponder im Antennenfeld erkannt wird.
 Wird das Repeat-Kommando vom Reader nicht unterstützt, bleibt "RPTACT" inaktiv. Wird "EXECUTE" trotzdem gesetzt, wird der Fehler "0xE7FE0900" nach einem Timeout von 10 Sekunden ausgegeben.
4. Die einzelnen Ergebnisse können Sie über wiederholtes Setzen des Eingangsparameters "EXECUTE" auslesen. ③

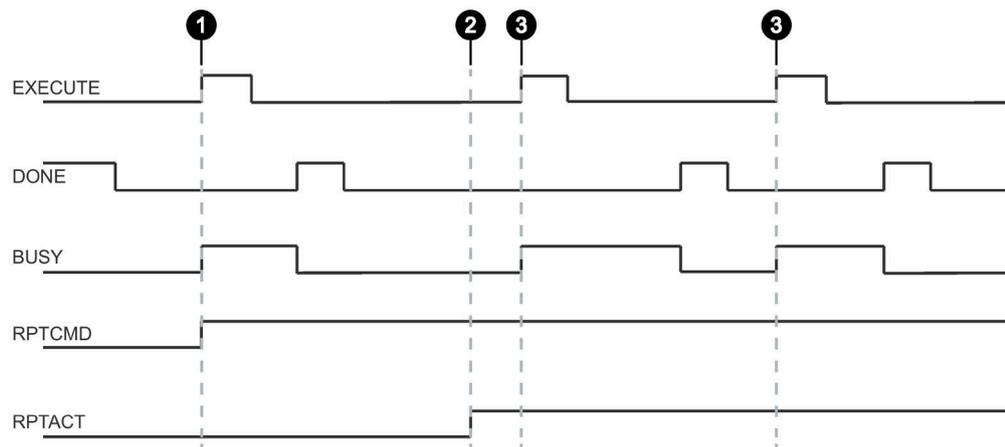


Bild 3-32 Ablauf des Repeat-Kommandos mit gleichzeitiger Befehlsübertragung

Ablauf des Repeat-Kommandos ohne Befehlsübertragung:

Dieser Ablauf ist nur möglich, wenn der betreffende Befehl bereits übertragen wurde.

1. Aktivieren Sie ggf. den "Presence_Mode".
2. Setzen Sie den Eingangsparameter "RPTCMD". ①
Das Repeat-Kommando wird auf dem Reader aktiviert.
3. Der Reader bestätigt die Aktivierung über den Ausgangsparameter "RPTACT" am Ident-Profil. Die Bestätigung erfolgt erst, nachdem der erste Befehl abgearbeitet wurde. ②

Wird das Repeat-Kommando vom Reader nicht unterstützt, bleibt "RPTACT" inaktiv. Wird "EXECUTE" trotzdem gesetzt, wird der Fehler "0xE7FE0900" nach einem Timeout von 10 Sekunden ausgegeben.

4. Die einzelnen Ergebnisse können Sie über wiederholtes Setzen des Eingangsparameters "EXECUTE" auslesen. ③

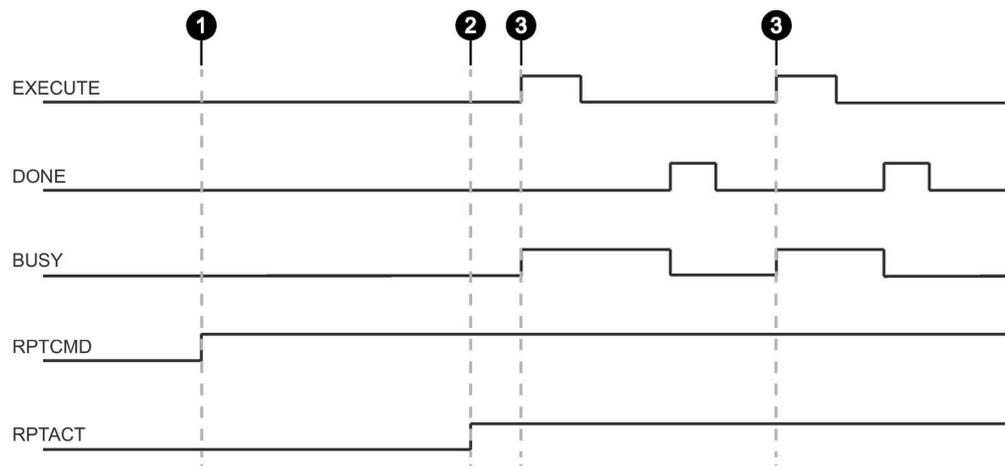


Bild 3-33 Ablauf des Repeat-Kommandos ohne Befehlsübertragung

Befehlswiederholung beenden

Sie haben die Möglichkeit die Befehlswiederholung durch Zurücknahme von "RPTCMD" oder mit Hilfe der Befehle "INIT" oder "SRESET" zu beenden. Nachfolgend werden die verschiedenen Vorgehensweisen beschrieben

Repeat-Kommando beenden und Zurücknahme von "RPTCMD":

1. Setzen Sie den Eingangsparameter "RPTCMD" zurück. ①
2. Holen Sie evtl. vorhandene Quittungen über den Eingangsparameter "EXECUTE" ab. ②
Der Ausgangsparameter "RPTACT" bleibt solange vom Reader gesetzt, wie Quittungen vorhanden sind.
3. Sind keine Quittungen mehr vorhanden, wird "RPTACT" vom Reader zurückgesetzt. ③

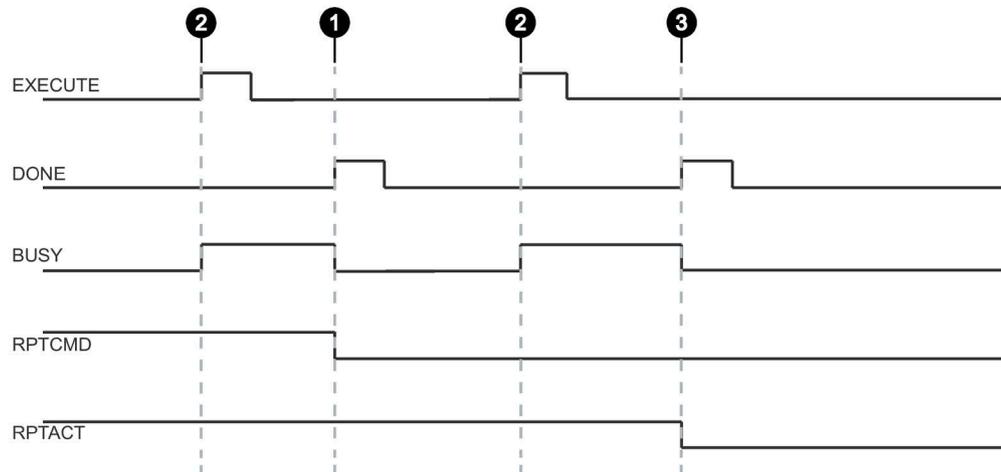


Bild 3-34 Repeat-Kommando beenden durch Zurücknahme von "RPTCMD" (regulär beendet)

Der Ausgangsparameter "RPTACT" wird vom Reader zurückgesetzt. Unter Umständen kann es passieren, dass "RPTACT" verzögert zurückgesetzt wird. D. h. nicht zeitgleich mit dem "DONE" der letzten Quittung. Wenn der Baustein nun erneut mit "EXECUTE" gestartet wird und "RPTACT" noch gesetzt ist, obwohl keine Ergebnisse mehr im Puffer sind, wird der Baustein nicht beendet (BUSY = 1). In diesem Fall können Sie warten, bis die nächsten Transponder ausgelesen werden. Alternativ kann der Baustein mit "INIT" oder "SRESET" beendet werden.

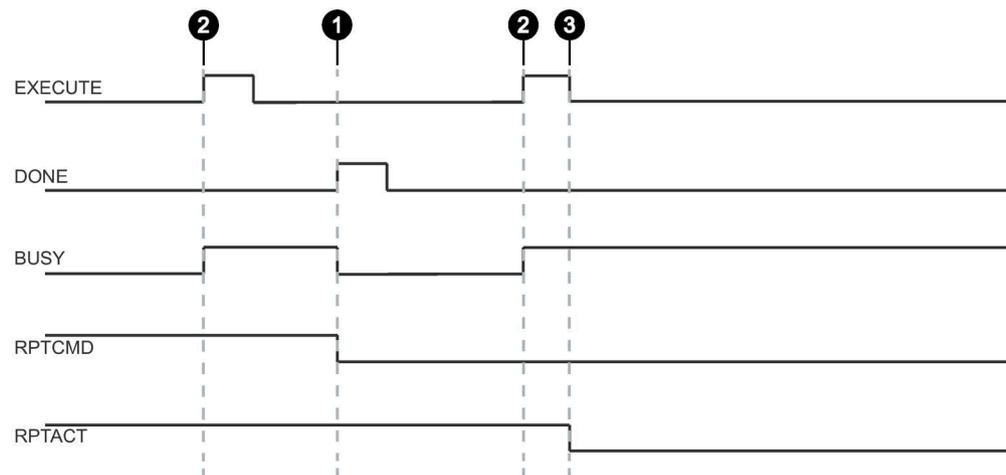


Bild 3-35 Repeat-Kommando beenden durch Zurücknahme von "RPTCMD" (der letzte Befehl bleibt anstehen)

Hinweis

Repeat-Kommando mit "INIT" oder "SRESET" beenden

Beenden Sie das Repeat-Kommando mit Hilfe der Eingangsparameter "INIT" oder "SRESET", wenn nicht bekannt ist, wie viele Transponder nach dem Zurücksetzen des Eingangsparameter "RPTCMD", noch bearbeitet wurden.

In der Regel wird ein "SRESET" deutlich schneller ausgeführt, da keine Reset-Prozedur durchlaufen wird.

Repeat-Kommando durch "INIT" beenden:

1. Setzen Sie den Eingangsparameter "RPTCMD" zurück und setzen Sie den Eingangsparameter "INIT". ①

Wird "RPTCMD" nicht zurückgesetzt, wird das Repeat-Kommando wieder auf dem Reader aktiviert. Dieses Verhalten löst eine Fehlermeldung aus, da kein Befehl vorhanden ist.

2. Der Reader setzt aufgrund des Eingangsparameters "INIT" den Ausgangsparameter "RPTACT" zurück. ②

Repeat-Kommando durch "SRESET" beenden:

1. Setzen Sie den Eingangsparameter "RPTCMD" zurück und setzen Sie den Ausgangsparameter "SRESET". ①
2. Der Ausgangsparameter "DONE" wird gesetzt und der Reader nimmt den Ausgangsparameter "RPTACT" zurück. ②

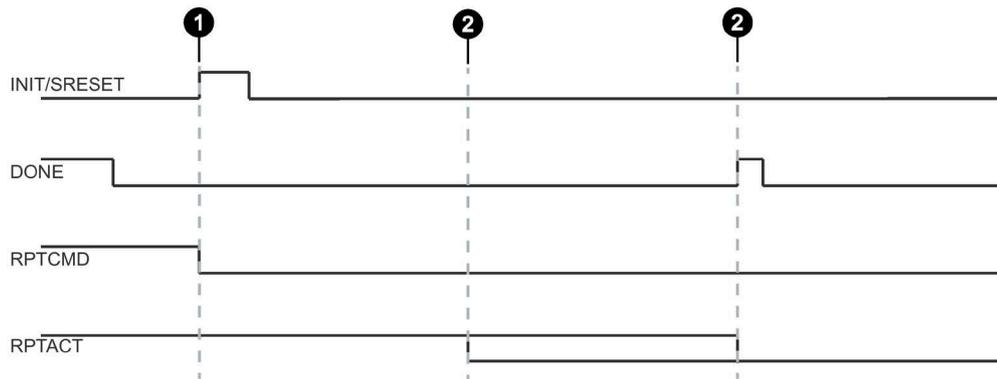


Bild 3-36 Repeat-Kommando durch "INIT"/"SRESET" beenden

Datenpuffer

Eine permanente Befehlswiederholung kann dazu führen, dass die Daten langsamer zum Ident-Profil übertragen werden, als neue Transponder bearbeitet werden. Alle Kommunikationsmodule, sowie einige Reader verfügen über Datenpuffer, die als Zwischenspeicher für die Befehlsfolge (Verkettung), als auch für die Ergebnisse genutzt wird (z. B. Repeat-Kommando). Sind die Datenpuffer voll, werden keine neuen Daten von dem Transponder gelesen bzw. auf diesen geschrieben, d. h. neu ankommende Transponder werden nicht mehr bearbeitet.

Tabelle 3- 86 Geräte, die die Befehlswiederholung unterstützen und Ergebnisse speichern

Gerätetyp	Pufferanzahl für verkettete Befehle und Ergebnisse	Max. bearbeitbare Nutzdaten mit Befehlswiederholung
RF360R ¹⁾	270	270 × 1034 Byte = 279 180 Byte
	150	150 × 480 Byte = 72 000 Byte
RF61xR/RF68xR	250	250 × 1034 Byte = 258 500 Byte
RF180C	150	150 × 233 Byte = 34 950 Byte
RF18xC/RF18xCI	270	270 × 1034 Byte = 279 180 Byte
RF166C	270	270 × 229 Byte = 61 830 Byte
ASM 456	150	150 × 233 Byte = 34 950 Byte

¹⁾ Ab der FW > V2.0 wird das Repeat-Kommando unterstützt. Bei der Verwendung des Repeat-Kommandos wird die Anzahl der Puffer auf 150 reduziert.

Die nachfolgend genannten Geräte unterstützen die Befehlswiederholung und verfügen entsprechend über einen Datenpuffer. Der Datenpuffer wird jedoch ausschließlich als Zwischenspeicher für die Befehlsfolge (Verkettung) und nicht zur Speicherung von Ergebnissen genutzt.

Tabelle 3- 87 Geräte, die die Befehlswiederholung unterstützen

Gerätetyp	Pufferanzahl für einen verketteten Befehl	Max. bearbeitbare Nutzdaten mit Befehlswiederholung
RF300-Reader; 1. Gen.	245	245 × 233 Byte = 57 085 Byte
RF300-Reader; 2. Gen. ¹⁾	299	299 × 233 Byte = 69 667 Byte
	150	150 × 480 Byte = 72 000 Byte

¹⁾ Ab der FW > V1.7 wird durch die Verwendung des "RF300 fast protocol" die Anzahl der Puffer auf 150 reduziert.

Hinweis

Einschränkung der Befehlswiederholung

Bei RFID-Systemen mit eindeutigen Tag-IDs (UID bzw. EPC-ID) (z. B. RF300, RF600, MOBY U) wird der gespeicherte Befehl nur dann wiederholt, wenn unterschiedliche Transponder in das Antennenfeld gebracht werden. Wenn hingegen immer wieder der gleiche Transponder (gleiche UID / EPC-ID) ins Antennenfeld gebracht wird, so erfolgt keine weitere Bearbeitung des Transponders.

3.6 Transponder-Adressierung

Adressierung

Die Adressierung der Transponder erfolgt linear von Adresse "0x0000" (oder der angegebenen Anfangsadresse) bis zur Endadresse. Das CM bzw. der Reader erkennt automatisch die Größe des Speichers auf dem Transponder. Wird die Endadresse auf dem Transponder überschritten, erhalten Sie eine Fehlermeldung.

Die nachfolgenden Tabellen zeigen den Adressraum der einzelnen Transponder-Parameter. Die Parameter "ADDR_TAG" und "LEN_DATA" müssen gemäß diesem Adressraum parametrieren werden.

Adressräume der ISO 15693-Transponder (MDS D)

Nachfolgend sind die Adressräume der ISO 15693-Transponder (MDS D) abgebildet. Diese Transponder können in den Ident-Systemen RF200, RF300 und MOBY D eingesetzt werden.

Die UID der Transponder ist festadressiert und kann ausschließlich komplett ausgelesen werden.

- Startadresse der UID: 0xFFFF0
- Länge der UID: 0x0008

Tabelle 3- 88 Adressräume der ISO 15693-Transponder (MDS D)

Anwenderspeicher	Transponder-Typ	Chip	Adressbereich (16 Bit)
112 Byte EEPROM	MDS D1xx	NXP I-Code SLI	0x0000 ... 0x006F
256 Byte EEPROM	MDS D2xx	Texas Inst. Tag-it HFI STM ST LRI 2K	0x0000 ... 0x00FF
992 Byte EEPROM	MDS D3xx	Infineon my-d	0x0000 ... 0x03DF
2000 Byte FRAM	MDS D4xx	Fujitsu MB89R118	0x0000 ... 0x07CF
8192 Byte FRAM	MDS D5xx	Fujitsu MB89R112	0x0000 ... 0x1FFF

Adressräume der RF300-Transponder (RF3xxT)

Nachfolgend sind die Adressräume der RF300-Transponder (RF3xxT) abgebildet. Diese Transponder können in dem Ident-System RF300 eingesetzt werden. Beachten Sie, dass der EEPROM-Anwenderspeicher bei RF300 sowohl als R/W-Speicher, als auch als OTP-Speicher genutzt werden kann. Ausführliche Informationen dazu finden Sie im Systemhandbuch "SIMATIC RF300".

Die UID der Transponder ist festadressiert und kann ausschließlich komplett ausgelesen werden.

- Startadresse der UID: 0xFFFF0
- Länge der UID: 0x0008

Tabelle 3- 89 Adressräume der RF300-Transponder (RF3xxT)

Anwenderspeicher	Transponder-Typ	Adressbereich (16 Bit)
20 Byte EEPROM	RF320T	0xFF00 ... 0xFF13
8 kByte FRAM/EEPROM	RF340T / RF360T	FRAM: 0x0000 ... 0x1FFC
		EEPROM: 0xFF00 ... 0xFF13
32 kByte FRAM/EEPROM	RF330T / RF340T / RF350T / RF360T / RF370T / RF380T	FRAM: 0x0000 ... 0x7FFC
		EEPROM: 0xFF00 ... 0xFF13
64 kByte FRAM/EEPROM	RF370T	FRAM: 0x0000 ... 0xFEFC
		EEPROM: 0xFF00 ... 0xFF13

Allgemeine Hinweise zur Bedeutung des OTP-Speichers bei RF300-Transpondern

RF300-Transponder und ISO-Transponder besitzen einen Speicherbereich, der gegen erneutes Überschreiben geschützt werden kann. Dieser Speicherbereich heißt OTP. Für die Aktivierung der OTP-Funktion sind folgende 5 Blockadressen vorgesehen:

- 0xFF80
- 0xFF84
- 0xFF88
- 0xFF8C
- 0xFF90

Ein Schreibbefehl an diese Blockadresse, mit zulässiger Länge (4, 8, 12, 16, 20 je nach Blockadresse), schützt die geschriebenen Daten vor einem anschließenden Überschreiben.

Hinweis

OTP-Bereich nur im statischen Betrieb nutzen

Nutzen Sie den OTP-Bereich nur im statischen Betrieb.

Hinweis

Nutzung des OTP-Bereichs ist nicht reversibel

Wenn Sie den OTP-Bereich nutzen, können Sie diese Belegung nicht mehr rückgängig machen, da der OTP-Bereich nur einmalig beschrieben werden kann.

Adressmapping des OTP-Speichers auf RF300-Transpondern

Ein EEPROM-Speicher (R/W) und OTP-Speicher sind in den Transpondern nur einmal vorhanden.

Die folgende Tabelle zeigt das Mapping der Adressen auf dem Transponder.

Das Lesen der Daten kann sowohl über die EEPROM-Adresse als auch über die OTP-Adresse erfolgen.

Tabelle 3- 90 Adressmapping des OTP-Speichers

EEPROM-Speicher (R/W)		OTP-Speicher (einmalig beschreiben)	
Adresse	Länge	Adresse	Länge
0xFF00	1 .. 20	0xFF80	4, 8, 12, 16, 20
0xFF01	1 .. 19		--
0xFF02	1 .. 18		--
0xFF03	1 .. 17		--
0xFF04	1 .. 16	0xFF84	4, 8, 12, 16
0xFF05	1 .. 15		--
0xFF06	1 .. 14		--

EEPROM-Speicher (R/W)		OTP-Speicher (einmalig beschreiben)	
0xFF07	1 .. 13		--

EEPROM-Speicher (R/W)		OTP-Speicher (einmalig beschreiben)	
0xFF08	1 .. 12	0xFF88	4, 8, 12
0xFF09	1 .. 11		--
0xFF0A	1 .. 10		--
0xFF0B	1 .. 9		--
0xFF0C	1 .. 8	0xFF8C	4, 8
0xFF0D	1 .. 7		--
0xFF0E	1 .. 6		--
0xFF0F	1 .. 5		--
0xFF10	1 .. 4	0xFF90	4
0xFF11	1 .. 3		--
0xFF12	1 .. 2		--
0xFF13	1		--

Hinweis**Schreibschutz aktivieren**

Ein Schreiben ab der Adresse "0xFF80" bis "0xFF93" setzt den EEPROM-Anwenderspeicher auf Schreibschutz (OTP-Funktion). Der Vorgang ist nicht reversibel. Die Einschaltung des Schreibschutzes muss immer in lückenlos aufsteigender Reihenfolge erfolgen, beginnend mit Adresse "0xFF80".

Adressräume der MIFARE-Transponder (MDS E)

Nachfolgend sind die Adressräume der MIFARE-Transponder (MDS E) abgebildet. Diese Transponder können in dem Ident-System RF300 eingesetzt werden.

Die UID der Transponder ist festadressiert und kann ausschließlich komplett ausgelesen werden.

- Startadresse der UID: 0xFFFF0
- Länge der UID: 0x0008

Tabelle 3- 91 Adressräume der MIFARE-Transponder (MDS E)

Anwenderspeicher	Transponder-Typ	Adressbereich (16 Bit)
752 Byte EEPROM	MDS E600 / MDS E611 / MDS E623 / MDS E624	0x0000 ... 0x02EF

Adressräume der RF600-Transponder

Bei den Readern RF61xR/RF650R/RF68xR werden die Anwenderdaten, TID, EPC und Passwörter über die betreffenden Memory Banks ausgelesen. Um die gewünschten Daten auszulesen, muss die betreffende Memory Bank ausgewählt werden.

Bereich und Länge der Anwenderdaten können Sie aus der obigen Tabelle (Spalte "USER") entnehmen. Die EPC-ID können Sie über einen Inventory-Befehl auslesen. Alternativ können Sie EPC-ID auch über einen Read-Befehl auf die Memory Bank 1, Startadresse 0x04 auslesen.

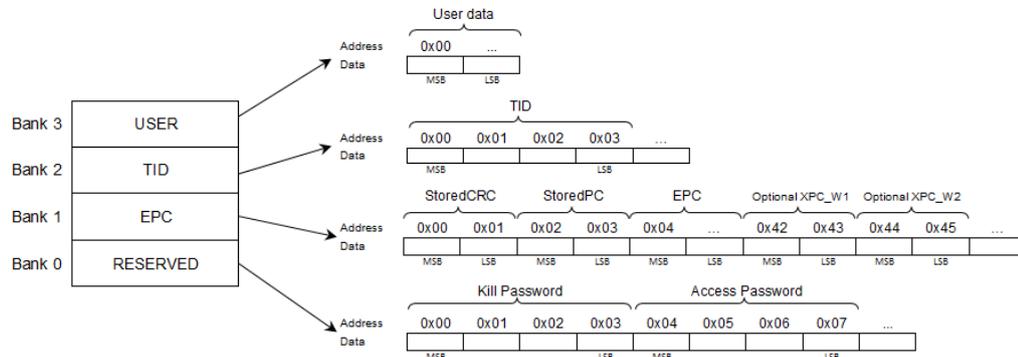


Bild 3-37 Speicheraufbau

Tabelle 3- 92 Adressräume der RF600-Transponder

Chip-Typ	Transponder	USER Anwenderspeicher	EPC		TID (read only)	RESERVED (Passwörter)
		Bereich / Länge	Bereich / Länge (Default und max.)	Zugriff	Bereich / Länge	Bereich / Länge
Alien Higgs 3	RF630L-Varianten RF640L-Varianten RF690L-Varianten	16/64 Byte ¹⁾	96 / 480 Bit	read/write	8 Byte	8 Byte
Alien Higgs 4	RF630L-Varianten	16 Byte	96 / 128 Bit	read/write	24 Byte	8 Byte
Impinj Monza 4 QT ²⁾	RF630L-Varianten RF620T RF625T	64 Byte	96 / 128 Bit	read/write	4 Byte	8 Byte
NXP G2iL	RF630L-Varianten	0 Byte	96 / 128 Bit	read/write	8 Byte	8 Byte
NXP G2iM ³⁾	RF630L-Varianten RF680L-Varianten	64 Byte	96 / 256 Bit	read/write	26 Byte	8 Byte

Chip-Typ	Transponder	USER Anwen- derspei- cher	EPC		TID (read only)	RESERVED (Passwörter)
		Bereich / Länge	Bereich / Län- ge (Default und max.)	Zugriff	Bereich / Länge	Bereich / Län- ge
NXP G2XM	RF630L-Varianten RF680L-Varianten RF610T RF630T RF640T	64 Byte	96 / 240 Bit	read/ write	8 Byte	8 Byte
NXP UCODE 7xm-2k	RF630L-Varianten RF642L-Varianten RF682L-Varianten RF645T	256 Byte	96 / 448 Bit	read/ write	12 Byte	8 Byte
NXP UCODE DNA	RF682T	384 Byte	96 / 224 Bit	read/ write	12 Byte	8 Byte

- 1) Der EPC-Speicherbereich der Alien Higgs 3-Chips kann zu Lasten des Anwenderspeichers (USER) vergrößert werden. Weitere Informationen dazu finden Sie in den entsprechenden Transponder-Kapiteln des Systemhandbuchs "SIMATIC RF600".
- 2) Verwendet User Memory Indicator (UMI).
- 3) Der TID-Bereich enthält einen beschreibbaren Bereich von 112 bit, der bei Bedarf genutzt werden kann.

Alle RF600-Transponder und -Smartlabels verfügen über ein Kill-Passwort und eine Lock-Funktion.

Adressraum der Transponder-/MDS-Varianten für MOBY U

System	Adressierung	Adressbereich (16 Bit)
MOBY U	2 KByte Datenspeicher	
	Anfangsadresse	0x0000
	Endadresse	0x07FF
	OTP-Speicher lesen (Schreiben ist nur einmalig möglich. Der OTP-Speicher von MOBY U kann nur komplett bearbeitet werden. D. h. die Anfangsadresse muss immer mit dem Wert 0xFFFF0 und die Länge mit dem Wert 0x10 angegeben werden.)	
	Anfangsadresse	0xFFFF0
	Länge	0x10
	ID-Nr.: (festcodiert 4 Byte; das Auslesen ist nur mit dem Befehl "MDS-Status" möglich)	
	32 KByte Datenspeicher	
	Anfangsadresse	0x0000
	Endadresse	0x7FFF
OTP-Speicher lesen (Schreiben ist nur einmalig möglich)		

System	Adressierung	Adressbereich (16 Bit)
	Anfangsadresse	0xFFF0
	Länge	0x10
	ID-Nr.: (festcodiert 4 Byte; das Auslesen ist nur mit dem Befehl "MDS-Status" möglich)	

Fehlermeldungen

4.1 Aufbau des Ausgangsparameters "STATUS"

Ein Fehlerzustand beim Ident-Profil ist immer dann gegeben, wenn im Ausgangsparameter "ERROR = TRUE" erscheint. Die Analyse (Dekodierung) des Fehlers kann dann über den Ausgangsparameter "STATUS" erfolgen.

Der Ausgangsparameter "STATUS" umfasst folgende 4 Byte:

Tabelle 4- 1 Bytes des Ausgangsparameters "STATUS"

Byte	Bedeutung
Byte 3 (niederwertigstes Byte)	Warnungen In diesem Byte hat jedes Bit eine eigene Bedeutung.
Byte 2	Fehlercode
Byte 1	Fehlernummer Dieses Byte definiert die Bedeutung der Fehlercodes und der Warnungen. Die Fehlernummern haben folgende Bedeutung: <ul style="list-style-type: none"> • 0x00 - kein Fehler, keine Warnung • 0x81...0x8F - Die Steuerung meldet einen Fehler gemäß dem Parameter "x" (0x8x). • 0xFE - Fehler vom Ident-Profil bzw. Kommunikationsmodul / Reader
Byte 0	Anweisungsnummern <ul style="list-style-type: none"> • Cx - Fehler in der Buskommunikation • E1 - Transponderbezogene Fehler • E2 - Fehler der Luftschnittstelle • E4 - Fehler der Reader Hardware • E5 - Fehler in der Kommunikation zwischen Reader und FB • E6 - Fehler im Befehl des Anwenders • E7 - vom FB erzeugte Fehlermeldung • F0 - Warnungen

4.2 STEP 7 - Fehlermeldungen

Sollten Sie die Bausteine und Datentypen in Ihr Projekt eingefügt haben und Probleme bei der Übersetzung haben, prüfen Sie bitte folgende Punkte:

- Der Bausteinname, die Bausteinnummer und der Datentypname dürfen nicht verändert werden.
- Die Datentypen "IID_CMD_STRUCT" und "IID_HW_CONNECT" dürfen in Ihrem Inhalt nicht verändert werden. Erst wenn diese in einen Datenbaustein als Variable eingebunden wurden, dürfen die Inhalte angepasst werden.
- Wenn Sie die Ident-Bausteine nutzen, müssen immer die Datentypen "IID_HW_CONNECT" und "IID_CMD_STRUCT" sowie der Baustein "Ident_Profile" in Ihrem Projekt vorhanden sein.

4.3 Fehler vom Kommunikationsmodul/Reader

Ursachen für diese Fehler können z. B. sein:

- Die Kommunikation zwischen CM und Reader oder zwischen Reader und Transponder ist fehlerhaft.
- Das Kommunikationsmodul kann den Befehl nicht bearbeiten.

Für die Fehlermeldungen ist Byte 3 des "STATUS" nicht relevant.

Tabelle 4- 2 Fehlermeldungen von Kommunikationsmodul/Reader oder vom Ident-Profil über den Ausgangsparameter "STATUS"

Fehlermeldung	Beschreibung
0xE1FE0100	Speicher des Transponders kann nicht beschrieben werden <ul style="list-style-type: none"> • Speicher des Transponders ist defekt. • EEPROM-Transponder wurde zu oft beschrieben und hat sein Lebensende erreicht. • Transponder ist schreibgeschützt (Memory Lock).
0xE1FE0200	Anwesenheitsfehler Der Transponder ist aus dem Übertragungsfenster des Readers gefahren. Der Befehl wurde nur teilweise abgearbeitet. Lesebefehl: Es sind keine gültigen Daten bei "IDENT_DATA" vorhanden. Schreibbefehl: Der Transponder, der gerade das Antennenfeld verlassen hat, beinhaltet einen unvollständigen Datensatz. Mögliche Ursachen: <ul style="list-style-type: none"> • Arbeitsabstand von Reader zu Transponder wird nicht eingehalten. • Projektierungsfehler: Zu bearbeitender Datensatz ist zu groß (im dynamischen Betrieb). • Mit Timeout: kein Transponder im Antennenfeld

Fehlermeldung	Beschreibung
0xE1FE0300	Adressfehler Der Adressbereich des Transponders wird überschritten. Mögliche Ursachen: <ul style="list-style-type: none"> • Anfangsadresse beim Befehlsstart ist falsch aufgesetzt. • Transponder ist nicht vom richtigen Typ. • Der zu schreibende Bereich ist schreibgeschützt. Freeport: Adresse ≠ "0x0000", "0x0002", "0xFFFF"
0xE1FE0400	Initialisierungsfehler Transponder kann den Initialisierungsbefehl nicht durchführen. Mögliche Ursachen: <ul style="list-style-type: none"> • Transponder ist defekt.
0xE1FE0500	Der Speicher des Transponders ist voll.
0xE1FE0600	Fehler im Speicher des Transponders <ul style="list-style-type: none"> • Der Transponder wurde noch nie beschrieben oder hat durch einen Ausfall der Batterie seinen Speicherinhalt verloren. Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Transponder wechseln (wenn das Batterie-Bit gesetzt ist). – Transponder neu initialisieren. • RF300: Beim Semaphore-Verfahren wurde die Datenspeicherung nicht korrekt abgeschlossen.
0xE1FE0700	<ul style="list-style-type: none"> • Passwort-Fehler • RF300: ECC-Fehler Mögliche Ursachen: <ul style="list-style-type: none"> – Der Bit-Fehler konnte nicht korrigiert werden. – Daten des Transponders sind verlorengegangen (Transponder defekt). – Der Transponder wurde nicht mit dem ECC-Treiber initialisiert. – Der EEPROM-Speicher des Transponders hat sein Lebensende erreicht und die Daten sind verloren gegangen. – Der Transponder (mit FRAM-Speicher) wurde zu heiß gelagert und hat Informationen verloren. Möglicherweise kann der Transponder nach Initialisierung weiterverwendet werden.
0xE1FE0800	Der Transponder im Antennenfeld hat nicht die erwartete UID / EPC-ID bzw. hat keine UID / EPC-ID.
0xE1FE0900	Der Befehl wird von dem Transponder nicht unterstützt.
0xE1FE0A00	Der Transponder ist lese-/schreibgeschützt.
0xE1FE8100	Der Transponder antwortet nicht.
0xE1FE8200	Das Transponder-Passwort ist falsch. Zugriff wird verweigert.
0xE1FE8300	Die Verifikation der geschriebenen Transponder-Daten ist fehlgeschlagen. Mögliche Ursachen: <ul style="list-style-type: none"> • Transponder ist defekt. • Transponder befindet sich im Grenzbereich.
0xE1FE8400	Allgemeiner Transponder-Fehler Wenden Sie sich ggf. an den Support.

Fehlermeldung	Beschreibung
0xE1FE8500	<p>Der Transponder hat zu wenig Leistung, um den Befehl auszuführen.</p> <p>Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Transponder befindet sich im Grenzbereich. • Die Sende-/Strahlungsleistung des Readers bzw. der Antenne ist zu gering.
0xE2FE0100	<p>Feldstörung am Reader</p> <p>Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Der Reader empfängt Störimpulse aus der Umgebung. <ul style="list-style-type: none"> – Externes Störfeld; das Störfeld kann mit dem "induktiven Feldindikator" des mobilen Readers nachgewiesen werden. – Der Abstand zwischen zwei Readern ist zu klein und entspricht nicht den Projektierungsrichtlinien. – Das Verbindungskabel zum Reader wird gestört, ist zu lang oder entspricht nicht der Spezifikation. • Zu viele Sendefehler <p>Der Transponder konnte den Befehl oder die Schreibdaten vom Kommunikationsmodul nach mehreren Versuchen nicht richtig empfangen.</p> <ul style="list-style-type: none"> – Transponder steht genau im Grenzbereich des Übertragungsfensters. – Datenübertragung zum Transponder wird durch externe Störungen beeinflusst. <ul style="list-style-type: none"> • CRC-Sendefehler <ul style="list-style-type: none"> – Der Transponder meldet sehr oft CRC-Fehler (Transponder steht im Grenzbereich des Readers; Transponder und/oder Reader haben einen Hardwaredefekt). • Nur bei Initialisierung: CRC-Fehler beim Quittungsempfang vom Transponder (Ursache wie bei Feldstörung am Reader). • Bei der Formatierung muss der Transponder im Übertragungsfenster des Readers stehen, ansonsten erfolgt ein Timeout-Fehler, d. h.: <ul style="list-style-type: none"> – Der Transponder steht genau im Grenzbereich des Übertragungsfensters. – Der Transponder ist defekt und verbraucht zu viel Strom. – Der EEPROM-Transponder wurde durch "FORMAT" falsch parametrieret. • RF600: <ul style="list-style-type: none"> – Es ist kein ETSI-Kanal frei. – Es wurde ein falscher Kommunikationsstandard im "INIT"-Befehl ausgewählt. – Der Expertenparameter ist fehlerhaft. – Die Leistungsprüfung des ETSI-Funkprofils ist fehlerhaft. • Freeport: Negative Quittung bzw. fehlerhaftes Telegramm vom Lesegerät
0xE2FE0200	<ul style="list-style-type: none"> • Es sind mehr Transponder im Übertragungsfenster, als der Reader gleichzeitig bearbeiten kann. • Mehr als ein Transponder befindet sich im Übertragungsfenster.
0xE2FE8100	<p>Kein Transponder mit der gewünschten EPC-ID/UID befindet sich im Übertragungsfenster, bzw. es befindet sich gar kein Transponder im Antennenfeld.</p>
0xE2FE8200	<p>Die angeforderten Daten sind nicht verfügbar.</p>
0xE2FE8300	<p>CRC-Fehler in der Reader-Transponder-Kommunikation.</p>
0xE2FE8400	<p>Die ausgewählte Antenne ist nicht aktiviert.</p>
0xE2FE8500	<p>Die ausgewählte Frequenz ist nicht aktiviert.</p>

Fehlermeldung	Beschreibung
0xE2FE8600	Das Trägersignal ist nicht aktiviert.
0xE2FE8700	Mehr als ein Transponder befindet sich im Übertragungsfenster.
0xE2FE8800	Allgemeiner Funkprotokoll-Fehler
0xE4FE0100	<p>Kurzschluss oder Überlastung der 24 V-Ausgänge</p> <ul style="list-style-type: none"> • Der Reader verbraucht zu viel Strom. • Das Reader-Kabel verursacht einen Kurzschluss. <p>Mögliche Folgen:</p> <ul style="list-style-type: none"> • Der betroffene Ausgang wird abgeschaltet • Bei Gesamtüberlastung werden alle Ausgänge abgeschaltet • Ein Rücksetzen ist nur durch das Aus- und Wiedereinschalten der 24 V-Versorgungsspannung möglich • anschließend "Reset_Reader" starten
0xE4FE0300	<ul style="list-style-type: none"> • Fehler in der Verbindung zum Reader; Reader antwortet nicht. <ul style="list-style-type: none"> – Kabel zwischen Kommunikationsmodul und Reader ist falsch verdrahtet oder Kabelbruch. – 24 V-Versorgungsspannung ist nicht angeschlossen oder abgeschaltet bzw. kurzzeitig ausgefallen. – Automatische Sicherung auf dem Kommunikationsmodul hat angesprochen. – Hardware defekt. – Anderer Reader ist in der Nähe und ist aktiv geschaltet. – Nach der Fehlerbehebung "INIT"/"WRITE-CONFIG" durchführen. • Die Antenne am Reader ist abgeschaltet. In diesem Zustand wurde ein Tag-Befehl zum Kommunikationsmodul gestartet. <ul style="list-style-type: none"> – Mit dem Befehl "Antenne Ein/Aus" die Antenne einschalten. – Die Antenne ist eingeschaltet (ausgeschaltet) und hat einen weiteren Einschaltbefehl (Aus-schaltbefehl) erhalten. • Der Modus im Befehl "SET_ANT" ist unbekannt. • Die Antenne am Reader ist abgeschaltet oder die Antenne bzw. das Antennenkabel ist defekt.
0xE4FE0400	<ul style="list-style-type: none"> • Der Puffer im Kommunikationsmodul oder Reader zur Zwischenspeicherung des Befehls reicht nicht aus. • Freeport: Es wurden neue Daten empfangen, bevor vorhandene Daten abgeholt wurden. <ul style="list-style-type: none"> – Scan-Betrieb: Die neuesten Daten gehen verloren. <p>Beim "Write"-Befehl gehen die ältesten Quittungen verloren, ohne dass eine Fehlermeldung generiert wird.</p>
0xE4FE0500	Der Puffer im Kommunikationsmodul oder Reader zur Zwischenspeicherung der Daten reicht nicht aus.
0xE4FE0600	<p>Dieser Befehl ist in diesem Status nicht erlaubt, bzw. wird nicht unterstützt.</p> <p>Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Der Befehl "INIT" wurde verkettet. • RF600: Die Befehlswiederholung wurde ohne "Presence-Mode" gestartet.

Fehlermeldung	Beschreibung
0xE4FE0700	<p>Hochlaufmeldung vom Reader/Kommunikationsmodul</p> <p>Der Reader bzw. das Kommunikationsmodul war ausgeschaltet und hat noch keinen Befehl "Reset_Reader" ("WRITE-CONFIG") erhalten.</p> <p>Mögliche Ursachen / weiteres Vorgehen:</p> <ul style="list-style-type: none"> • Führen Sie den Befehl "INIT" durch. • Die gleiche physikalische Adresse im Parameter "IID_HW_CONNECT" wird mehrmals verwendet. Überprüfen Sie ihre "IID_HW_CONNECT"-Parametrierungen. • Überprüfen Sie die Verbindung zum Reader. • Nachdem die Übertragungsgeschwindigkeit geändert wurde, wurde das Gerät noch nicht neu gestartet.
0xE4FE8100	<p>Das angegebene Tag-Feld des Transponders ist nicht bekannt.</p>
0xE4FE8A00	<p>Allgemeiner Fehler</p>
0xE4FE8B00	<p>Es wurden keine oder fehlerhafte Konfigurationsdaten/Parameter übertragen.</p> <p>Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Sie greifen auf eine nicht projektierte Lesestelle zu.
0xE4FE8C00	<ul style="list-style-type: none"> • Kommunikationsfehler zwischen Ident-Profil und Kommunikationsmodul. Handshake-Fehler. <p>Mögliche Ursachen / weiteres Vorgehen:</p> <ul style="list-style-type: none"> – UDT dieses Kommunikationsmoduls wird durch andere Programmteile überschrieben – Parametrierung des Kommunikationsmoduls im UDT überprüfen – Befehl des Ident-Profiles überprüfen, der zu diesem Fehler führt – Nach Fehlerbehebung "INIT" starten <ul style="list-style-type: none"> • Rückwandbus- / PROFIBUS DP¹⁾ / PROFINET-Fehler aufgetreten <p>Dieser Fehler wird nur angezeigt, wenn die "Ansprechüberwachung" bei der PROFIBUS-Konfigurierung eingeschaltet wurde.</p> <p>Mögliche Ursachen / weiteres Vorgehen:</p> <ul style="list-style-type: none"> – Rückwandbus/PROFIBUS DP / PROFINET-Busverbindung war unterbrochen (Drahtbruch am Bus; Busstecker am Kommunikationsmodul war kurzzeitig gezogen). – Rückwandbus/PROFIBUS DP / PROFINET-Master spricht Kommunikationsmodul nicht mehr an – "INIT" durchführen – Das Kommunikationsmodul hat am Bus eine Telegrammunterbrechung festgestellt. Eventuell wurde der Rückwandbus, PROFIBUS bzw. PROFINET neu konfiguriert (z. B. mit HW-Konfig oder TIA Portal) <p>¹⁾ Dieser Fehler wird nur angezeigt, wenn die "Ansprechüberwachung" bei der PROFIBUS-Konfigurierung eingeschaltet wurde.</p>

Fehlermeldung	Beschreibung
0xE4FE8D00	<ul style="list-style-type: none"> • Kommunikationsfehler des Kommunikationsmoduls/Readers Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Stecker-Kontaktproblem auf dem Kommunikationsmodul/Reader – Hardware des Kommunikationsmoduls/Readers hat einen Defekt; → Kommunikationsmodul/Reader zur Reparatur einschicken. – Führen Sie nach der Fehlerbehebung den Befehl "INIT" durch. • Überwachungsfehler des Kommunikationsmoduls/Readers Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Programmablauffehler auf dem Kommunikationsmodul/Reader – Versorgungsspannung des Kommunikationsmoduls/Readers aus- und wieder einschalten. – Führen Sie nach der Fehlerbehebung den Befehl "INIT" durch. • Firmware-Fehler Mögliche Ursachen: Das Firmware-Update wurde nicht vollständig durchgeführt.
0xE4FE8E00	<p>Der laufende Befehl wurde durch den Befehl "WRITE-CONFIG" ("INIT" oder "RESET") abgebrochen bzw. der Busstecker wurde abgezogen.</p> <p>Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Die Kommunikation mit dem Transponder wurde mit "INIT" abgebrochen. • Dieser Fehler kann nur bei einem "INIT" oder "RESET" zurückgemeldet werden.
0xE5FE0100	Falsche Sequenz-Nummernfolge (SN) im Reader/Kommunikationsmodul
0xE5FE0200	<p>Falsche Sequenz-Nummernfolge (SN) im Ident-Profil</p> <p>Mögliche Ursache: In der Gerätekonfiguration ist nicht der User-Mode "Ident-Profil / RFID-Normprofil" eingestellt.</p>
0xE5FE0400	Ungültige Datenblock-Nummer (DBN) im Reader/Kommunikationsmodul
0xE5FE0500	Ungültige Datenblock-Nummer (DBN) im Ident-Profil
0xE5FE0600	Ungültige Datenblock-Länge (DBL) im Reader/Kommunikationsmodul
0xE5FE0700	Ungültige Datenblock-Länge (DBL) im Ident-Profil
0xE5FE0800	<p>Vorheriger Befehl ist noch aktiv bzw. der Puffer ist voll.</p> <p>An den Reader bzw. das Kommunikationsmodul wurde ein neuer Befehl geschickt, obwohl der letzte Befehl noch aktiv ist.</p> <ul style="list-style-type: none"> • Der aktive Befehl kann nur durch "INIT" abgebrochen werden. • Vor dem Start eines neuen Befehls muss das "DONE-Bit = 1" sein (Ausnahme "INIT"). • Zwei Ident-Profil-Aufrufe wurden mit den gleichen Parametern "HW_ID", "CM_CHANNEL" und "LADDR" parametrisiert. • Zwei Ident-Profil-Aufrufe arbeiten mit dem gleichen Zeiger. • Führen Sie nach der Fehlerbehebung den Befehl "INIT" durch. • Beim Arbeiten mit Befehlswiederholung (z. B. Festcode-Transponder) werden keine Daten vom Transponder abgeholt. Der Datenpuffer im Reader/Kommunikationsmodul ist übergelaufen. Es sind Transponder-Daten verloren gegangen.
0xE5FE0900	Der Reader bzw. das Kommunikationsmodul führt einen Hardware-Reset aus ("INIT_ACTIVE" auf "1" gesetzt), vom Ident-Profil wird "INIT" erwartet (Bit 15 im zyklischen Steuerwort).
0xE5FE0A00	Der Befehlscode "CMD" und die entsprechende Bestätigung stimmen nicht überein. Hierbei kann es sich um einen Software- oder Synchronisationsfehler handeln, der im Normalbetrieb nicht auftreten kann.

Fehlermeldung	Beschreibung
0xE5FE0B00	Falsche Reihenfolge der Quittungstelegramme (TDB / DBN)
0xE5FE0C00	Synchronisationsfehler (falsches Inkrement von "AC_H / AC_L" und "CC_H / CC_L" im zyklischen Steuerwort). "INIT" musste ausgeführt werden.
0xE5FE8100	Kommunikationsfehler zwischen Reader und Kommunikationsmodul Zugriff verweigert
0xE5FE8200	Kommunikationsfehler zwischen Reader und Kommunikationsmodul Ressource belegt
0xE5FE8300	Kommunikationsfehler zwischen Reader und Kommunikationsmodul Funktionsfehler der seriellen Schnittstelle
0xE5FE8400	Kommunikationsfehler zwischen Reader und Kommunikationsmodul Sonstige Fehler
0xE6FE0100	Unbekannter Befehl Das Ident-Profil sendet einen nicht interpretierbaren Befehl an den Reader. Mögliche Ursachen: <ul style="list-style-type: none"> • Der Baustein "AdvancedCmd" wurde mit einem falschen "CMD" versorgt. • Der Eingang "CMD" des Bausteins "AdvancedCmd" wurde überschrieben. Freeport: Falsche Befehlsparameter oder nicht zugelassene Befehlsverkettung (falsche Längenangabe)
0xE6FE0200	Ungültiger Kommandoindex (CI)

Fehlermeldung	Beschreibung
0xE6FE0300	<ul style="list-style-type: none"> • Das Kommunikationsmoduls bzw. der Reader wurde falsch parametrieret. Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Überprüfen Sie die "INPUT"-Parameter im Ident-Profil. – Überprüfen Sie die Parametrierung in HW-Konfig / STEP 7 (TIA Portal). – Befehl "WRITE-CONFIG" ist falsch parametrieret. – Nach einem Hochlauf hat der Reader bzw. das Kommunikationsmodul noch keinen "INIT" erhalten. • Der Reader bzw. das Kommunikationsmodul am PROFIBUS/PROFINET wurde falsch parametrieret und der Befehl kann nicht abgearbeitet werden. Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Länge der Ein-/Ausgangsbereiche zu klein für das zyklische Wort E/A. Richtige GSD-Datei verwendet? – Befehl (z. B. "READ") mit zu großer Länge der Nutzdaten aufgesetzt. • Fehler beim Bearbeiten des Befehls Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Die Daten im "AdvancedCmd" bzw. "IID_CMD_STRUCT" sind fehlerhaft (z. B. "WRITE"-Befehl mit Länge = 0); "AdvancedCmd" bzw. "IID_CMD_STRUCT" überprüfen und "INIT" durchführen. – Hardware des Reader/Kommunikationsmoduls defekt: Bei "INIT" erhält der Reader bzw. das Kommunikationsmodul falsche Daten. – Inkonsistente Längenangaben im Befehl • Falscher Reset-Baustein wurde ausgewählt. Mögliche Ursachen / weiteres Vorgehen: <ul style="list-style-type: none"> – Verwenden Sie, den zu dem Ident-Gerät passenden Reset-Baustein. – Technologieobjekt mit RF68xR und RF120C: Verwenden Sie, unabhängig vom gewählten Ident-Gerät/-System, den Funktionsbaustein "Reset_Reader".
0xE6FE0400	<p>Anwesenheitsfehler Ein Transponder hat das Übertragungsfenster eines Readers durchquert, ohne bearbeitet zu werden.</p> <ul style="list-style-type: none"> • Diese Fehlermeldung wird nicht sofort gemeldet. Vielmehr wartet der Reader bzw. das Kommunikationsmodul auf den nächsten Schreib-/Lesebefehl. Dieser Befehl wird sofort mit diesem Fehler beantwortet und der Schreib-/Lesebefehl wird nicht bearbeitet. Erst der nächste Befehl wird wieder regulär vom Reader/Kommunikationsmodul ausgeführt. • Sie können diesen Fehlerzustand mit Hilfe eines "INIT" zurücksetzen.
0xE6FE0500	<p>Es ist ein Fehler aufgetreten, der ein Reset_Reader ("WRITE-CONFIG") erforderlich macht. Mögliche Ursachen / weiteres Vorgehen:</p> <ul style="list-style-type: none"> • Der Befehl "WRITE-CONFIG" ist fehlerhaft. • Führen Sie nach der Fehlerbehebung einen "INIT" durch. • Überprüfen Sie den Parameter "IID_HW_CONNECT".
0xE6FE8100	Ein Parameter fehlt.
0xE6FE8200	Der Parameter hat ein ungültiges Format.
0xE6FE8300	Der Parameter-Typ ist ungültig.
0xE6FE8400	Unbekannter Parameter

Fehlermeldung	Beschreibung
0xE6FE8500	Der Befehl bzw. das Telegramm hat ein ungültiges Format.
0xE6FE8600	Der "Inventory"-Befehl ist fehlgeschlagen.
0xE6FE8700	Der Lesezugriff auf den Transponder ist fehlgeschlagen.
0xE6FE8800	Der Schreibzugriff auf den Transponder ist fehlgeschlagen.
0xE6FE8900	Das Schreiben der EPC-ID/UID auf dem Transponder ist fehlgeschlagen.
0xE6FE8A00	Das Aktivieren des Schreibschutzes auf dem Transponder ist fehlgeschlagen.
0xE6FE8B00	Der "Kill"-Befehl ist fehlgeschlagen.
0xE7FE0100	In diesem Zustand ist nur der Befehl "Reset_Reader" ("WRITE-CONFIG" mit "CMDSEL =1" und "CMD = 0x78") zulässig.
0xE7FE0200	<ul style="list-style-type: none"> • Der Befehlscode "CMD" oder der Wert in "CMDSEL" ist nicht zulässig. <p>Mögliche Ursache / weiteres Vorgehen:</p> <ul style="list-style-type: none"> – Die Parameter "CMDSEL" oder "CMDDIM" wurde falsch parametrierd. Überprüfen Sie die Parameter. – Der Wert des Parameters "CMDSEL" muss zwischen 1 ... 100 liegen und darf nie größer als der Wert des Parameters "CMDDIM" (max. 100) sein. <ul style="list-style-type: none"> • Bei "Read_Tagfield/Write_Tagfield": <p>Unzulässiger Wert im Parameter "TAGFIELD". Der Wert muss zwischen 1 ... 64 liegen.</p>
0xE7FE0300	Der Wert des Parameters "LEN_DATA" des Befehls ist zu lang und passt nicht zu den globalen Daten, die innerhalb des Sendedaten-Puffers (TXBUF) reserviert wurden.
0xE7FE0400	Der Empfangsdaten-Puffer (RXBUF) oder der Sendedaten-Puffer (TXBUF) ist zu klein, der angelegte Puffer an TXBUF/RXBUF/IDENT_DATA hat nicht den richtigen Datentypen oder der Parameter "LEN_DATA" hat einen negativen Wert.
0xE7FE0500	Als nächster Befehl ist ausschließlich ein "INIT"-Befehl zulässig. Alle anderen Befehle werden zurückgewiesen.
0xE7FE0600	Falscher Datensatzindex eines azyklischen Datensatzes Erlaubter Index liegt in den Bereichen "101 ... 108" und "-20401 ... -20418".
0xE7FE0700	Der Reader bzw. das Kommunikationsmodul antwortet nicht auf "INIT" (in zyklischer Statusmeldung wird "INIT_ACTIVE" erwartet).
0xE7FE0800	Zeitüberschreitung während des "INIT" (60 Sekunden)
0xE7FE0900	Befehlswiederholung wird nicht unterstützt.
0xE7FE0A00	Fehler während der Übertragung der PDU (Protocol Data Unit).

Fehlermeldung	Beschreibung
0xE7FE0B00	Der Parameter "CMDREF" wurde falsch parametrieret. Überprüfen Sie den Parameter. Der Parameter "CMDREF" muss als "ARRAY of IID_CMD_STRUCT" angelegt sein und darf max. 100 Elemente enthalten.
0xFxFExx00	Ein Fehler "FxFExxxx" ist identisch mit dem entsprechenden Fehler "ExFExxxx" (siehe Fehler "ExFExxxx"). Zusätzlich stehen im Byte 3 Warnhinweise.

4.4 Fehler vom optischen Lesegerät

Bei Fehlermeldungen wird das Signal "IN_OP" (In Operation) zurückgesetzt und die LED "STATE/SF" leuchtet dauerhaft rot. Bei Anbindung über das Ident-Profil wird zusätzlich das "Ready"- bzw. "Done"-Bit zurückgesetzt.

Falls das optische Lesegerät an PROFINET IO angeschlossen ist, lösen diese Fehlermeldungen am zugehörigen I/O-Controller einen Diagnosealarm aus. Wie die auf dem optischen Lesegerät zur Verfügung gestellten Diagnoseinformationen ausgelesen und anschließend ausgewertet werden, finden Sie unter I/O-Diagnose.

Tabelle 4- 3 Fehlermeldungen über den Ausgangsparameter "STATUS" ("IN_OP" wird zurückgesetzt)

Fehlermeldung	Beschreibung
0xE1FE0200	Die Verbindung der internen Schnittstelle zum Bildsensor ist gestört. Wenn der Fehler nach erneutem Einschalten immer noch auftritt, kontaktieren Sie den technischen Support.
0xE1FE0400	Übertragungsfehler Der Sende-Puffer ist voll, weil die Daten nicht in ausreichend kurzer Zeit abgefragt werden. Verringern Sie die Triggerfrequenz oder verarbeiten Sie die Ergebnisse schneller. Ändern Sie ggf. die Aktualisierungszeit bei der PROFINET-Konfiguration. Gegebenenfalls kann die Übertragungsgeschwindigkeit der CM-Verbindung erhöht werden. Um kurzfristige Spitzenlasten abzudecken, kann die Größe des Bildpuffers des Programms unter "Bildaufnahme", Register "Steuerung" vergrößert werden.
0xE1FE0600	Programm kann nicht gestartet werden, da nicht genügend Speicher verfügbar oder das Programm beschädigt ist. Reduzieren Sie den Speicherbedarf und führen Sie den Vorgang "Programm speichern" erneut durch. Programm erfordert EF-Objektiv Im Programm ist ein EF-Objektiv konfiguriert, dieses ist aber nicht angeschlossen. Schließen Sie das EF-Objektiv wieder an oder ändern Sie die Parametrierung des Programms.
0xE1FE0700	Vergleichsfehler Das Programm konnte wegen fehlerhafter Match-Einstellungen nicht erstellt werden. Passen Sie die Match-Einstellungen an oder verwenden Sie einen passenden Prüfling.
0xE4FE0400	Interner Datei-Fehler Beim Speichern in den Festwertspeicher ist ein Fehler aufgetreten. Bei vermehrtem Auftreten dieses Fehlers wenden Sie sich bitte an den technischen Support.

Fehlermeldung	Beschreibung
0xE4FE0600	<p>Lampe überlastet</p> <p>Die angeschlossene Lampe wurde überlastet. Der parametrierte oder vorgegebene "maximale Tastgrad" unter "Optionen", Register "Beleuchtung" wurde überschritten.</p> <p>Verringern Sie die Triggerfrequenz, verkleinern Sie die Belichtungszeit oder setzen Sie eine leistungsfähigere Lampe ein.</p>
0xE4FE8400	<p>Befehlssequenz war fehlerhaft.</p> <p>Kann beispielsweise durch zu schnelles Triggern auftreten. Das Ident-Profil kann nur jeweils einen Befehl abarbeiten, ehe ein neuer Befehl ausgeführt wird.</p>
0xE6FE0400	<p>Das Programm konnte nicht erstellt bzw. gespeichert werden.</p> <ul style="list-style-type: none"> • Beim Programm speichern wird das DISA-Signal an ungültiger Stelle verändert oder die zeitliche Abfolge der angelegten Signale nicht eingehalten. <p>Überprüfen Sie die zeitliche Abfolge der angelegten Signale. Starten Sie den Programmspeichern-Vorgang erneut.</p> <ul style="list-style-type: none"> • Das Programm konnte nicht erstellt werden. <p>Passen Sie die Parametrierung, die Platzierung der Prüflinge im Bild oder die Bildqualität an.</p> <ul style="list-style-type: none"> • Über die in "Steuerung" eingestellte Schnittstelle wird versucht, ein Programm unter einer ungültigen Nummer zu speichern. <p>Wählen Sie eine Programmnummer zwischen 1 und 15. Bei SIMATIC MV420 SR-B sind 1 bis 5 und 15 gültige Programmnummern.</p>

Bei den nachfolgenden Meldungen wird das Signal "IN_OP" (In Operation) nicht zurückgesetzt und die LED "STATE/SF" leuchtet nicht rot.

Tabelle 4- 4 Meldungen über den Ausgangsparameter "STATUS" ("IN_OP" wird nicht zurückgesetzt)

Meldung	Beschreibung
0xE1FE0300	<p>Falscher Parameter im MV-Befehl</p> <p>Der Befehl wurde falsch aufgebaut. Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Die spezifizierte Adresse bei einem "WRITE"-Befehl ist \neq "0x0000". • MV-Befehl Programmwechsel <ul style="list-style-type: none"> – Länge der zu schreibenden Daten sind $>$ "0x1". – Übermittelte Programmnummer ist $>$ "0xF". <p>Die übermittelte Programmnummer ist nicht gespeichert.</p>
0xE6FE0100	<p>Befehl nicht zulässig oder der Befehl wurde abgebrochen.</p> <p>Die genaue Fehlermeldung kann mit "INIT" ohne Programmanwahl ermittelt werden. Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Der Sendepuffer ist voll. • Das Programm ist beschädigt. • Die Ethernet-Schnittstelle ist in Betrieb und hat ein Problem. • Die angeschlossene Lampe ist überlastet. • Der Match-String-Zugriff ist aufgrund fehlender Parametrierung fehlgeschlagen.
0xE6FE0300	<p>Die Initialisierung mit Programmanwahl ("INIT"/"WRITE-CONFIG") ist nicht möglich. Mögliche Ursachen:</p> <ul style="list-style-type: none"> • Übermittelte Programmnummer ist nicht gespeichert. • Lesegerät befindet sich noch im Selbsttest.

4.5 Fehler vom Bus/Rückwandbus

Die Transportschicht vom verwendeten Bussystem (Rückwandbus, PROFIBUS, PROFINET) meldet einen Fehler. Zur genauen Fehlersuche und -analyse ist ein PROFIBUS-Tracer eine wertvolle Hilfe. Für PROFINET kann die open Source-Software "Wireshark" verwendet werden. Die Systemdiagnose des PROFIBUS bzw. PROFINET kann weiteren Aufschluss über die Fehlerursache liefern.

Fehlermeldungen die mit "80/C0/DE/DF" beginnen beziehen sich auf Fehler vom Bus/Rückwandbus. Details zur Fehlermeldung können der STEP 7-Hilfe zu den Bausteinen "WRREC" bzw. "RDREC" (SFB52/SFB53) entnommen werden.

4.6 Warnungen

Das Byte 3 des Ausgangsparameters "STATUS" zeigt Warnungen an, wenn das Byte 0 des "STATUS" (Funktionsnummern) den Wert "Fx" bzw. "Dx" aufweist.

Tabelle 4- 5 Mögliche Warnungen beim Arbeiten mit dem Ident-Profil

Byte 0...2	Byte 3	Bedeutung
FxFExx	Bit 0	Das Bit ist fest auf "0" gesetzt.
	Bit 1	Der ECC-Modus hat einen 1-Bit-Fehler korrigiert.
	Bit 2	Batterie niedrig
	Bit 3	Herstellerabhängig
	Bit 4	Herstellerabhängig
	Bit 5	Herstellerabhängig
	Bit 6	Herstellerabhängig
	Bit 7	Herstellerabhängig

A.1 Interne Statusparameter

Status-Variablen

Jeder Ident-Baustein verfügt über Statusausgänge, um im Fehlerfall im Anwenderprogramm entsprechend zu reagieren und um eine Fehlerdiagnose am Gerät zu vereinfachen. Zusätzlich dazu beinhaltet jeder Ident-Baustein einen Zeitstempel und einen Fehlerspeicher, um vergangene Probleme besser nachzuvollziehen.

Diese Variablen sind gespeichert im jeweiligen Instanz-DB des Bausteins.

Tabelle A- 1 Status-Variablen im Instanz-Datenbaustein

Name	Datentyp	Beschreibung
statLastErrorStatus	DWORD	Diese Variable beinhaltet den letzten Anweisungsstatus im Fehlerfall. Dieser Wert wird immer überschrieben, wenn ein neuer Fehler am Baustein auftritt.
statLastErrorTimestamp	DTL (S7-1200/-1500) DATE_AND_TIME (S7-300/-400)	Diese Variable speichert den Zeitstempel des letzten aufgetretenen Fehlers ("Last_error_status") an der Anweisung.

Sie finden die Status-Variablen unter folgendem Pfad: "Instanz-Datenbaustein/Ident_Instance/Static/*Name*".

Weitere Status-Variablen sind in der Variablen "IID_HW_CONNECT" vorhanden.

Tabelle A- 2 Status-Variablen in "IID_HW_CONNECT"

Name	Datentyp	Beschreibung
STATUS_IN_WORK	BOOL	Befehl wird aktuell ausgeführt <ul style="list-style-type: none"> • True = Ein Baustein bzw. das Ident-Profil greift auf diesen Kanal/Reader zu. • False = Der Kanal/Reader wird aktuell nicht verwendet.
STATUS_INITIALISATION	BOOL	Reset-Anzeige <ul style="list-style-type: none"> • True = An diesem Reader/Kanal ist ein Reset aktiv. • False = An diesem Reader/Kanal ist kein Reset aktiv.
LAST_CMD_INIT	BOOL	Dieses Bit zeigt an, dass der zuletzt ausgeführte Befehl ein Reset war. <ul style="list-style-type: none"> • True = Letzter Befehl war ein Reset • False = Letzter Befehl war kein Reset Dieses Bit wird beim nächsten Befehlsstart zurückgesetzt

Sie finden die Status-Variablen unter folgendem Pfad: "IID_HW_CONNECT-Variable/Static/*Name*".

A.2 Zyklische Daten

Mit Hilfe der zyklischen Daten können Sie im Problemfall schnell die Ursache identifizieren. Bei der Verwendung von Fremdsteuerungen können Sie über die zyklische Daten das Ident-Profil anpassen.

Eine ausführliche Beschreibung der Kommunikationsschnittstelle finden Sie in dem Dokument "Identification Systems, Proxy Ident Function Block" auf den Seiten der PROFIBUS-Nutzerorganisation (<https://www.profibus.com/download/>).

A.2.1 Zyklisches Steuerwort

Die nachfolgende Tabelle zeigt die Codierung des zyklischen Steuerworts.

Tabelle A- 3 Codierung des Steuerworts: Zyklische Ausgangsdaten

Bit	Beschreibung
0 ... 5	Reserviert
6	RPTCMD
7	SRESET
8 ... 14	Reserviert
15	INIT

Tabelle A- 4 Bedeutung der Bits des zyklischen Steuerworts

Befehl / Bit	Bedeutung für RFID-Systeme	Bedeutung für optische Lesesysteme
RPTCMD	1 = Wiederholung des aktuell ausgeführten oder als nächstes auszuführenden Befehls durch das Kommunikationsmodul	
SRESET	1 = Abbruch des aktuell im Kommunikationsmodul verarbeiteten Befehls	
INIT	1 = Kommunikationsmodul führt einen Neustart aus und wird neu parametrisiert	
Reserviert	Reservierte Bits	

A.2.2 Zyklisches Statuswort

Die nachfolgende Tabelle zeigt die Codierung des zyklischen Statusworts.

Tabelle A- 5 Codierung des Statusworts: Zyklische Eingangsdaten

Bit	Beschreibung
0 ... 3	UIN0 ... UIN3
4	ERR_IREQ
5	Reserviert
6	RPTACT
7	SRESETACT
8	TP
9	TPC
10	Reserviert
11	Auftragszähler (CC_L)
12	Auftragszähler (CC_H)
13	Quittierzähler (AC_L)
14	Quittierzähler (AC_H)
15	INITACT

Tabelle A- 6 Bedeutung der Bits des zyklischen Steuerworts

Befehl / Bit	Bedeutung für RFID-Systeme	Bedeutung für optische Lesesysteme
UIN0 ... UIN3	Bei RFID-Readern wird die Anzahl der im Antennenfeld befindlichen Transponder angezeigt. UIN0 ... UIN3 sind als Binärwert zu interpretieren. Beispiel: UIN3 = 0; UIN2 = 0; UIN1 = 1; UIN0 = 1 ⇒ 3 Transponder	Bei optischen Lesegeräten werden die verschiedenen Zustände des Lesegeräts angezeigt. <ul style="list-style-type: none"> • UIN0: Entspricht "IN_OP"-Bit des Lesegeräts • UIN1: Entspricht "RDY"-Bit des Lesegeräts • UIN2 + UIN3: Diese zwei Bits werden als unsigned Wert interpretiert (Bit 2 ist das niederwertigere Bit) welcher die Anzahl verfügbarer decodierter Codes darstellt. Wenn der Wert = 3 ist, dann sind drei oder mehr decodierte Codes verfügbar.
ERR_IREQ	1 = am Kommunikationsmodul, Reader oder Lesegerät ist ein Fehler aufgetreten (z. B. beim Hochlauf des Geräts, Verbindungsunterbrechung oder bei RF61xR, RF68xR bei Antennenfehlern oder Konfigurationsänderungen)	

Befehl / Bit	Bedeutung für RFID-Systeme	Bedeutung für optische Lesesysteme
RPTACT	1 = "RPTCMD" ist aktiv Das Quittierungs-Bit zeigt an, dass der "Repeat-Mode" des CMs/Readers/Lesegeräts aktiv ist.	
SRESETACT	1= Zeigt an, dass der momentan innerhalb des Readers / optischen Lesegeräts verarbeitete Befehl abgebrochen wurde. Das Bit wird auf "0" zurückgesetzt, nachdem "SRESET" zurückgesetzt wurde.	
TP	Transponder Presence 1 = ein Transponder befindet sich im Antennenfeld des Readers.	Herstellerspezifisch
TPC	Transponder Presence Changed (nur bei aktivem Presence_Mode) 1 = Ein neuer Transponder kommt ins Antennenfeld des Readers oder ein Transponder hat das Antennenfeld verlassen. Der Parameter wird nach erfolgreicher Ausführung des nächsten "INVENTORY"- (0x80, 0x81, 0x87) oder "INIT"-Befehls auf "FALSE" gesetzt.	Nicht verwendet
Auftragszähler (CC_L)	Niederwertiges Bit des Befehlszählers	
Auftragszähler (CC_H)	Höherwertiges Bit des Befehlszählers	
Quittierzähler (AC_L)	Niederwertiges Bit des Quittierzählers	
Quittierzähler (AC_H)	Höherwertiges Bit des Quittierzählers	
INITACT	1= Zeigt an, dass der Reader / das optische Lesegerät gerade einen Startvorgang ausführt. Nach Rücksetzen des Bits auf "0" wird erwartet, dass ein vom Baustein gesendeter Befehl "Write-Config" empfangen wird. Andere Befehle als "Write-Config" werden mit einem Fehler zurückgewiesen. Das Bit ist zur Verwendung bei der Neusynchronisierung des Quittier-- und des Befehlszählers vorgesehen.	
Reserviert	Reservierte Bits	

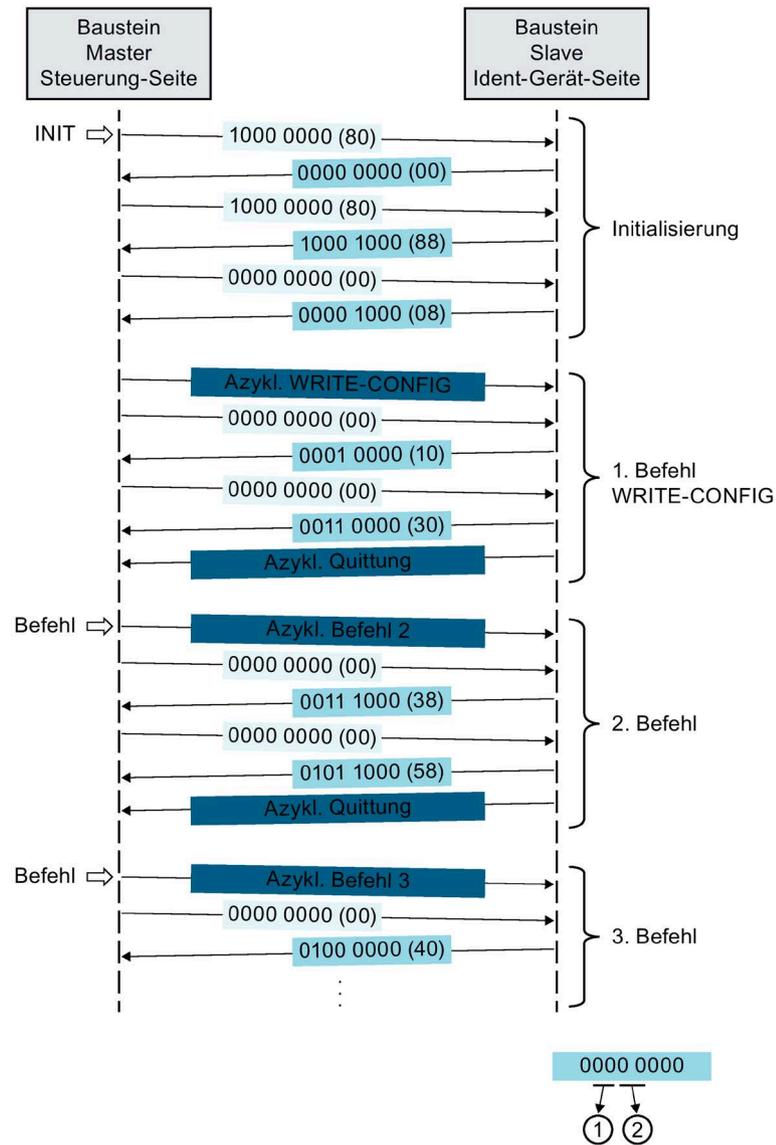
A.2.3 Verwendung des Quittier- und des Befehlszählers

Das zyklische Steuerwort dient zur Synchronisierung des Telegrammaustauschs zwischen DP-Master / IO Controller (der Steuerung, welche den Baustein ausführt) und Slave / IO Device (Ident-Gerät). Die Übertragung der nichtzyklischen Befehls- und Quittiertelegramme über eine MS1- oder Record Data CR-Verbindung darf erst gestartet werden, wenn der Start durch das zyklische Statuswort des Ident-Geräts im Quittierzähler angezeigt wird.

Die Quittier- und Befehlszählerbits dienen zur Synchronisierung der Befehlsausführung. Die Änderung des Befehlszählers (vom Ident-Gerät ausgeführt) zeigt dem Baustein an, dass er den nächsten Befehl absenden darf. Durch die Änderung des Quittierzählers wird angezeigt, dass ein Befehl (innerhalb des Ident-Geräts) beendet wurde und der Baustein die Quittung aus dem Ident-Gerät auslesen kann. Die Reihenfolge der Änderung der

Zähler (vom Ident-Gerät ausgeführt und vom Baustein überwacht) ist bei beiden Zählern (Quittier- und Befehlszähler) gleich und wird in der Abbildung unten dargestellt.

Nach dem Startvorgang wird das zyklische Statusbyte wie folgt codiert:

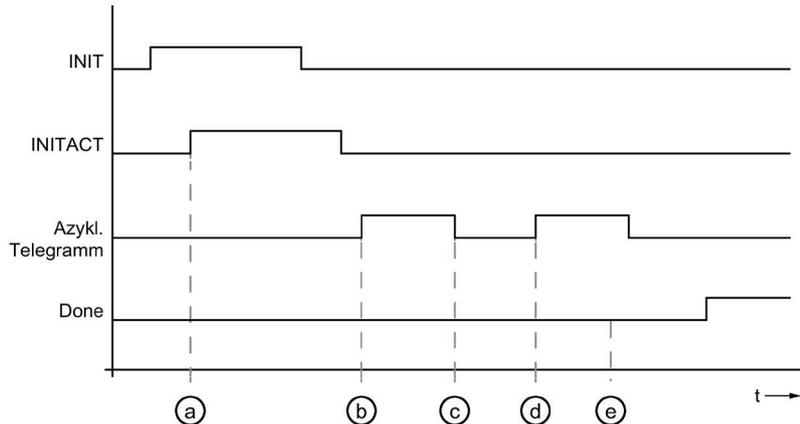


- ① AC_H und AC_L
- ② CC_H und CC_L

Bild A-1 Verwendung des Befehls- und des Quittierzählers

Synchronisierung zwischen Befehls- und Quittierzähler

Die Befehlszähler (CC_H/CC_L) und die Quittierzähler (AC_H/AC_L) werden während eines Starts synchronisiert. Das Ident-Gerät setzt "AC_H/AC_L = 0" und "CC_H/CC_L = 1". Die Initialisierung kann sowohl von dem Ident-Gerät (durch Wiedereinschalten) als auch vom Baustein (durch "Init") ausgelöst werden.



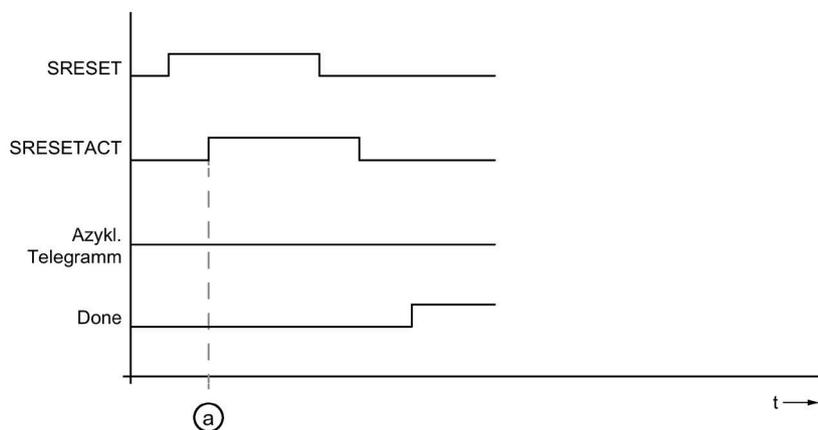
- a CC_H/CC_L = 1, AC_H/AC_L = 0
Festgelegter Reset
- b "WRITE-CONFIG"-Befehl zum Ident-Gerät
- c CC_H/CC_L = 2
- d AC_H/AC_L = 1
"WRITE-CONFIG"-Quittung vom Ident-Gerät
- e Bei Unterbrechung eines Befehls mit "WRITE-CONFIG" oder "Init" erfolgt eine Fehlermeldung.

Bild A-2 Zeitpunkt des Starts (durch Baustein eingeleitet)

Befehlsabbruch mit "SRESET"

Der Befehl "SRESET" bricht die Ausführung des momentan verarbeiteten Befehls ab. Er wird vom Baustein (Steuerung) zum Baustein (Reader / optisches Lesegerät) über zyklische Kommunikation (Steuer- und Statuswort) übertragen. Der Handshake wird über "SRESET" und "Soft_Reset_Active" durchgeführt. Nach dem Setzen des Baustein-Parameters "SRESET" ordnet der Baustein den Befehl dem Steuerbit "Soft_Reset" zu. Währenddessen, kann die Ausführung von "RD_REC" und "WR_REC" nicht unterbrochen werden. Dadurch kann der Zuordnungsvorgang erst nach beendeter Ausführung von "RD_REC" oder "WR_REC" stattfinden.

Voraussetzung für den Befehl "SRESET" ist, dass der Befehls- und der Quittierzähler innerhalb des Bausteins (Reader / optisches Lesegerät) und des Statusworts initialisiert werden. Auf die Sequenznummer (SN innerhalb des Telegrammheaders) wirkt sich "SRESET" nicht aus.



- a $CC_H/CC_L = 1, AC_H/AC_L = 0$
 Festgelegter Reset

Bild A-3 Zeitlicher Ablauf von "SRESET"

A.3 Anschluss serieller Geräte via Freeport-Protokoll

A.3.1 Kompatible Ident-Geräte

Folgende Ident-Geräte sind mithilfe von Kommunikationsmodulen über das Freeport-Protokoll und die RS232-Schnittstelle betreibbar:

- SIMATIC MV320
- SIMATIC RF1040R/RF1070R

Sie können auch beliebige serielle Geräte an das Kommunikationsmodul anschließen, da mithilfe des Freeport-Protokolls das Kommunikationsmodul mit seriellen Geräten kommunizieren kann (z. B. Barcode-Scanner, intelligenten Sensor oder anderen seriell angeschlossenen Automatisierungskomponenten). Bei Kommunikationsmodulen mit vier seriellen Schnittstellen unterstützen nur zwei Schnittstellen RS232.

A.3.2 Funktionen und Befehle

Mit Hilfe der Ident-Bausteine / des Ident Profils können Sie mit den Ident-Geräten kommunizieren. Dabei übernimmt der "Write"-Befehl das Senden und der "Read"-Befehl das Empfangen von Daten.

Die folgende Tabelle zeigt die bausteinspezifischen Befehle, die Sie mit seriellen Feldgeräten durchführen können.

Tabelle A- 7 Befehlsübersicht

Befehl	Beschreibung
Read	Lesen von Daten <ul style="list-style-type: none"> • Adresse "0x0000" In den ersten 2 Byte des Ergebnisses wird die Länge der gültigen Daten eingetragen. Die Nutzdaten werden ab dem 3. Byte eingetragen. Max. Nettodaten 227 Byte; keine Fehlermeldung bei Längenüberschreitung • Adresse "0x0002" Keine Längeninformatio n im Ergebnis; max. Nettodaten 229 Byte
Write	Senden von Daten Die Adresse muss mit "0xFFFF" angegeben werden. Max. Nettodaten 194 Byte (ohne Längenangabe); keine Fehlermeldung bei Längenüberschreitung

Befehl	Beschreibung
Reset_Univ	Kommunikation zwischen CM und Gerät zurücksetzen und Puffer leeren. Der Parameter "PARAM" kann im 11. Byte ("option_1") mit "0x00" oder "0x02" belegt werden. Im 1. Byte muss der Wert "0x04" und im 6. Byte "0x0A" eingetragen werden. Bei Verwendung des Technologieobjekts kann alternativ der Befehl "Reset_Reader" verwendet werden.
Reset_Reader	Alternativ zu dem Befehl "Reset_Univ" kann, bei Verwendung des Technologieobjekts "SIMATIC Ident > TO_Ident", dieser Befehl verwendet werden.

Ausführliche Informationen zu den Befehlen sowie Fehlermeldungen des RF1040R/RF1070R finden Sie im Kapitel "Programmierung" im Handbuch "SIMATIC RF1000". Die abgebildeten Auftragstelegramme der Systemaufträge müssen Sie im Datenbereich für den "Write"-Befehl hinterlegen. Die Antworttelegramme werden im Datenbaustein nach einem "Read"-Befehl ausgegeben.

Anmerkungen zu den Befehlen und Funktionen

Reset_*

Durch den Reset-Befehl wird die Kommunikation zum CM initialisiert. Der Reset-Befehl erfolgt ohne readerspezifische Parameter und löst ein Löschen der Puffer aus.

Nach dem "Reset_*" ist das Kommunikationsmodul einsatzbereit.

Read

Wird der Befehl "Read" gestartet, werden die Daten, die das CM bereits empfangen hat, an die Steuerung übermittelt.

Wird der Befehl "Read" gestartet, bevor das CM Daten empfangen hat, bleibt der Befehl aktiv bis das CM Daten empfangen hat, die dann an die Steuerung übermittelt werden. Ein aktiver "Read"-Befehl kann durch einen "Reset_*/"Write_Config"-Befehl abgebrochen werden.

Das Anwesenheitsbit ("TP") zeigt an, dass das CM neue Daten empfangen hat und diese zur Abholung durch einen "Read"-Befehl bereitstehen. Nach der Datenübertragung an die Steuerung wird das Anwesenheitsbit zurückgesetzt - soweit nicht bereits neue Daten empfangen wurden.

Beachten Sie dabei die Puffergröße und den Pufferüberlauf. Ist der Empfangspuffer des CMs voll, werden bestehende Daten durch neu empfangene überschrieben.

Service & Support

Industry Online Support

Zusätzlich zur Produktdokumentation unterstützt Sie die umfassende Online-Plattform des Siemens Industry Online Support unter folgender Internet-Adresse:

Link: (<https://support.industry.siemens.com/cs/de/de/>)

Neben Neuigkeiten finden Sie dort:

- Produktinformationen: Handbücher, FAQs, Downloads, Anwendungsbeispiele etc.
- Ansprechpartner, Technisches Forum
- Die Möglichkeit, eine Support-Anfrage zu stellen:
Link: (<https://support.industry.siemens.com/My/ww/de/requests>)

- Unser Service-Angebot:

Rund um unsere Produkte und Systeme bieten wir eine Vielzahl von Dienstleistungen an, die Sie in jeder Lebensphase Ihrer Maschine oder Anlage unterstützen - von der Planung und Realisierung über die Inbetriebnahme bis zur Instandhaltung und Modernisierung.

Kontaktdaten finden Sie im Internet unter folgender Adresse:

Link: (https://www.automation.siemens.com/aspa_app/?ci=yes&lang=de)

Homepage "Industrielle Identifikation"

Allgemeine Neuigkeiten zu unseren Identifikationssystemen finden Sie im Internet auf unserer Homepage (www.siemens.com/ident).

Online-Katalog und -Bestellsystem

Den Online-Katalog und das Online-Bestellsystem finden Sie ebenfalls auf der Industry Mall-Homepage (<https://mall.industry.siemens.com>).

SITRAIN - Training for Industry

Das Schulungsangebot umfasst mehr als 300 Kurse zu Grundlagenthemen, Aufbauwissen und Spezialwissen, sowie Weiterbildungsmaßnahmen zu einzelnen Branchen - verfügbar an über 130 Standorten weltweit. Zudem können die Kurse individuell gestaltet und bei Ihnen vor Ort abgehalten werden.

Ausführliche Informationen zum Schulungsangebot und Kontaktdaten unserer Kundenberater finden Sie unter folgender Internet-Adresse:

Link: (<https://new.siemens.com/global/de/produkte/services/industrie/sitrain.html>)

