



Messen – Regeln - Automatisieren
Schaltanlagenbau
Elektromontagen
Service und Wartung
Kundendienst

Bedienungsanleitung

Funktionsbausteine
RK512 für S7-1200/1500

Hinweis:

Der Inhalt dieser Dokumentation wurde auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Gewähr übernommen werden kann.



ELEKTROTECHNIK

**VORWERK Pipeline- und
Anlagenservice GmbH
Bereich MESA Elektrotechnik**

Fahrenberg 20
D-22885 Barsbüttel

T +49.(0) 40.6759504-0
F +49.(0) 40.6759504-29
E info@mesa-automation.de

Büro Erkner
Julius-Rütgers-Straße 10
D-15537 Erkner

T +49.(0) 3362.500-375
F +49.(0) 3362.500-376
E info@mesa-automation.de

www.friedrich-vorwerk.de
www.mesa-automation.de

Rechnungsanschrift
VORWERK Pipeline- und Anlagenservice GmbH
c/o FRIEDRICH VORWERK KG (GmbH & Co.)
Niedersachsenstraße 19-21
D-21255 Tostedt

Dokumentversionen

Nr.	Datum	Bemerkung	Version	Bearbeiter
000	18.04.2017	Dokument erzeugt	0.2	JR
001	10.05.2017	Erweitert für RK512_SLAVE	0.4	JR
002	11.05.2017	Performance-Test	1.0	JR
003	12.05.2017	Bausteinnamen an Bibliotheksstrukturen angepasst, Firmware-Versionen	1.1	JR
004	15.05.2017	Beschreibung der VAR_OUTPUT überarbeitet	1.2	JR
005	16.05.2017	Freischaltcode	1.3	JR
006	06.06.2017	Texte überarbeitet	1.4	JR
007	19.06.2017	Formular für Bestellung	1.5	JR
008	28.08.2017	Datenübertragung mit Folgeblöcke	1.6	JR
009	11.10.2017	Texte und Bilder überarbeitet, Anpassungen der FB und deren Parameter	1.7	JR
010	08.03.2018	Textformulierung und Bilder des Performancetest überarbeitet	1.8	JR
011		Version skipped	1.9	JR
012	17.04.2018	Hardware-Einstellungen ‚Freeport‘	2.0	JR
013	16.05.2018	Seriennummer im FB und ‚warmstart‘ nach der Aktivierung	2.1	JR
014	04.09.2018	Logo	2.2	JR

Inhalt

Abkürzungen	V
Tabellenverzeichnis	6
Abbildungsverzeichnis	7
Einleitung	8
1. Allgemeine Funktionsbeschreibung der RK512-Prozedur	10
2. Telegrammstrukturen	11
2.1 Aufbau des Befehls-Telegrammkopfes	11
2.2 Aufbau des Reaktionstelegrammes	11
2.3 Ablauf einer Fetchanfrage	12
2.4 Ablauf eines Sendeauftrag	13
2.5 Fehlermeldungen	14
3. Beschreibung der Bausteine und Schnittstellenparameter	15
3.1. Freischaltcode für vollen Funktionsumfang	16
3.2. Nutzdatenspeicher	18
3.3. Auftragsliste (Jobs)	18
3.4. MESA_RK512_MSTR	19
3.5. MESA_RK512_SLAV	22
4. Gerätekonfiguration	25
4.1. HW-Kennung CPU	25
4.2. HW-Kennung CM	26
4.3. Parametrierung der seriellen Schnittstelle	27
4.4. Kommunikations-Parameter CPU	27
4.5. Kommunikations-Parameter CM	28
5. Anhang	30
5.1. Performance Test	30
5.2. Systemvoraussetzungen	31
5.3. Ermittlung der CPU-Seriennummer	31
5.4. Bestellformular für Funktionsbausteine	32

Abkürzungen

Abkürzung	Erläuterung
NIC	Network Interface Connector
REATEL	Reaktionstelegramm
CM	Communication Modul
FETCH	RK512 Auftrag für die Anfrage von Daten
SEND	RK512 Auftrag für das Senden von Daten
S7 1200/1500	Automatisierungssysteme der Fa. Siemens

Tabellenverzeichnis

Tabelle 1 : RK512 Telegrammkopf	11
Tabelle 2 : RK512 Reaktionstelegramm	11
Tabelle 3 : RK512 Fetch Prozedur	12
Tabelle 4 : RK512 Sende Prozedur	13
Tabelle 5 : RK512 Fehlercodes.....	14
Tabelle 6 : RK512_JOB	16
Tabelle 7 : MESA_RK512_MSTR – VAR_INPUT.....	19
Tabelle 8 : MESA_RK512_MSTR – VAR_IN_OUT	20
Tabelle 9 : MESA_RK512_MSTR – VAR_OUTPUT.....	20
Tabelle 10 : MESA_RK512_SLAV – VAR_INPUT.....	22
Tabelle 11 : MESA_RK512_SLAV – VAR_OUTPUT.....	22
Tabelle 12 : Auftragsliste für Performance-Test	30
Tabelle 13 : Erforderliche Firmware-Stände	31
Tabelle 14 : Bestellformular für Funktionsbausteine	32

Abbildungsverzeichnis

Abbildung 0-1: RK512 als Firewall	8
Abbildung 2-1: Ablauf Fetchanfrage.....	12
Abbildung 2-2: Ablauf Sendetelegramm	13
Abbildung 3-1: Datenbaustein ohne ‚Optimierten Zugriff‘	18
Abbildung 3-2: Auftragsdatenstrukturen RK512 (Master)	18
Abbildung 3-3: Funktionsbaustein MESA_RK512_MSTR	20
Abbildung 3-4: Ablauf einer Fetch-Anfrage MESA_RK512_MSTR	21
Abbildung 3-5: Funktionsbaustein MESA_RK512_SLAV	23
Abbildung 3-6: Ablauf einer Fetch-Anfrage MESA_RK512_SLAV	24
Abbildung 4-1: Hardware CPU mit CM	25
Abbildung 4-2: HW_Kennung_CPU	25
Abbildung 4-3: HW_Kennung_CM (Eigenschaften CM)	26
Abbildung 4-4: HW_Kennung_CM (Systemkonstanten).....	26
Abbildung 4-5: Eigenschaften der RS232-Schnittstelle	27
Abbildung 4-6: Einstellung der Kommunikationslast.....	27
Abbildung 4-7: Einstellung CM1241 (RS323)	29
Abbildung 5-1: Seriennummer der CPU.....	31

Einleitung

Das RK512 Protokoll ist eine Erweiterung des 3964R Protokolls und verwendet dieses als Basis. Beim RK512 können zwei Kommunikationspartner über eine Punkt-zu-Punkt Verbindung mittels serieller Schnittstelle (RS232) Daten austauschen.

Die Distanz zwischen den Kommunikationspartnern ist von der verwendeten Übertragungsgeschwindigkeit abhängig. In der Regel sind Entfernungen von ca. 10 - 15 Meter überbrückbar. Beim verwendeten Kabel muss es sich um ein Nullmodemkabel (RXD und TXD sind gedreht) handeln. Die Hardwareparameter der seriellen Verbindung ist für den Datenaustausch an beiden Koppelpartner identisch einzustellen.

Der Datenaustausch beruht auf einer Master–Slave Beziehung. Der Master initiiert eine Transaktion auf die der Slave reagiert. Eine RK512-Transaktion besteht aus einem Befehlstelegramm und einem Reaktionstelegramm. Der Initiator der Transaktion steuert dabei die Datenübertragung zum oder vom Partnersystem. Dabei wird zwischen zwei Auftragsarten unterschieden, SEND (Datenübertragung zum Partnersystem) und FETCH (Anfrage an Partner und Datenübertragung zum Initiator).

Die Nutzdaten werden in Datenblock mit jeweils 128 Byte versendet. Jeder Datenblock erhält eine Prüfsumme (Blockscheck optional abwählbar). Der Slave beantwortet jeden Datenblock mit einem Reaktionstelegramm. Große Datenmengen werden in einzelne Folgeblöcke aufgeteilt und im Ziel wieder zusammen geführt. Die Aufteilung in Folgeblöcke erfolgt automatisch nach dem RK512 Protokoll und ist für den Anwender transparent.

Aufgrund des Schnittstellenprotokolls und der physikalischen Punkt-zu-Punkt-Verbindung ist eine sichere Trennung von Netzwerken ohne Mehraufwand gewährleistet.

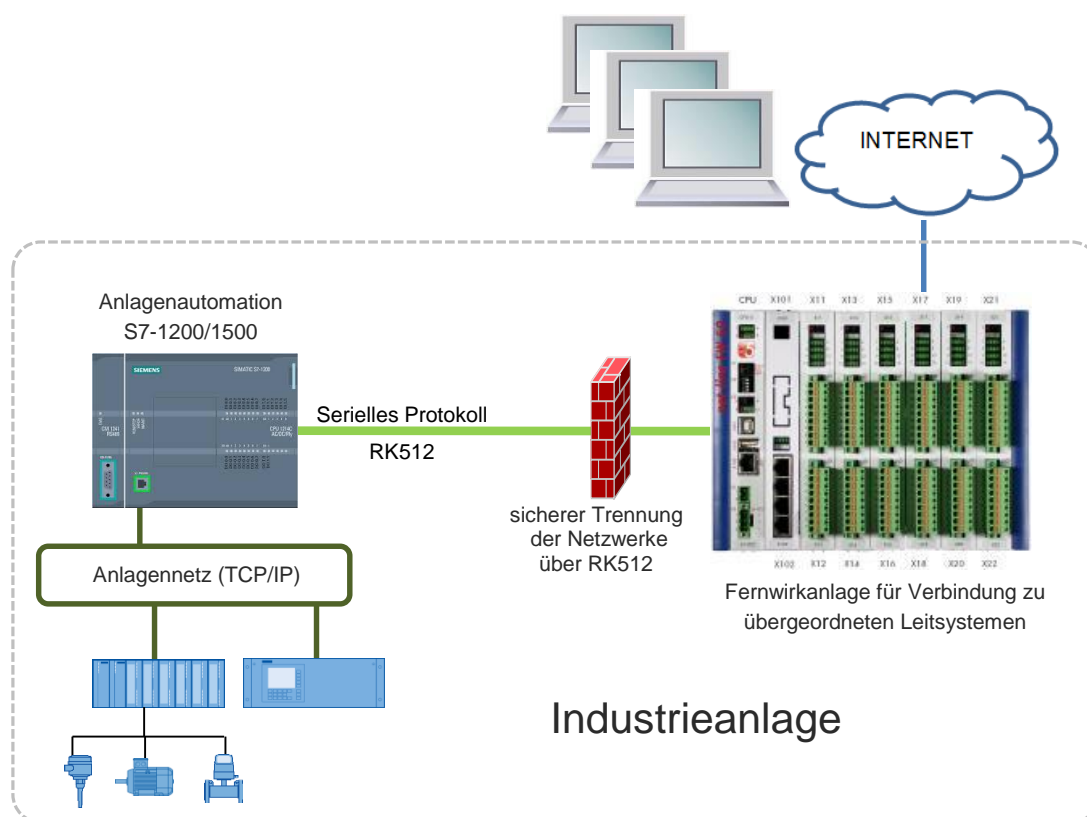


Abbildung 0-1: RK512 als Firewall

Zu den beschriebenen Eigenschaften des RK512 Protokolls sind bei den entwickelten Bausteinen folgende Einschränkungen zu berücksichtigen:

- *Der Datenaustausch erfolgt ausschließlich über Datenbausteine (,D' 44H)*
- *Datenaufträge unterstützen nur die Funktionen*
 - ,A' (41H) SEND-Auftrag mit Ziel DB*
 - ,E' (45H) FETCH-Auftrag mit Quelle DB.*
- *Koordinierungsmerker werden nicht verwendet*

Bei dem Masterbaustein sind beliebig viele Datenanfragen parametrierbar. Die Datenanfragen oder -Sendungen werden in einer Auftragsliste definiert. Die Übertragungsfunktion ‚Fetch‘ (,E‘) oder ‚Send‘ (,A‘) legt dabei die Datenrichtung fest.

Die Anzahl der Aufträge und der Datenbausteine für die Aufnahme der Nutzdaten werden nur durch den Systemspeicher begrenzt. Die einzelnen Aufträge werden sequentiell abgearbeitet.

Der Blockscheck gilt global für alle Aufträge und ist optional einstellbar, muss aber bei beiden Kommunikationsxxi Master und Slave identisch sein.

Diese Dokumentation beschreibt die Funktion der Interfacebausteine für die S7-1200 und S7-1500. Aufgrund der unterschiedlichen Hardwareplattformen (Prozessortypen) muss der Programmcode für die unterschiedlichen Zielplattformen kompiliert werden. Zur Kennzeichnung der unterschiedlichen Bausteine erhalten die Namen ein Präfix. Die Funktion der Bausteine ist identisch und somit die Dokumentation allgemein für alle Typen gültig. In dieser Dokumentation wird bei Bezug auf einen Baustein der Präfix allgemein als ‚MESA‘ genannt.

Bausteinbezeichnung S7-1200 (Präfix=„Bi001_F“)

Bi001_F_RK512_MSTR
Bi001_F_RK512_SLAV

Bausteinbezeichnung S7-1500 (Präfix=„Bi002_F“)

Bi002_F_RK512_MSTR
Bi002_F_RK512_SLAV

Die Kommunikationsbausteine werden über einen Freischaltcode aktiviert. Ohne Freischaltcode ist der Funktionsumfang zeitlich eingeschränkt. Der Freischaltcode ist bei

*Firma MESA Gesellschaft für Elektrotechnik
Fahrenberg 20
22885 Barsbüttel
Tel. (+49) 40 6759504-0
info@mesa-automation.de*

mit Angabe der CPU-Seriennummer erhältlich. Im Anhang befindet sich eine Anweisung zur Ermittlung der benötigten Seriennummer und weitere Informationen zur Freischaltung.

1. Allgemeine Funktionsbeschreibung der RK512-Prozedur

Diese Dokumentation beschreibt das RK512-Protokoll nur auszugsweise und hegt keinen Anspruch auf Vollständigkeit. Für tieferegehende Informationen sei auf die Original-Dokumentation von Siemens verwiesen.

Im folgenden Abschnitt erfahren Sie, wie die Datenübertragung mit der Rechnerkopplung RK 512 im Einzelnen abläuft und die Telegramme aufgebaut sind.

Die Datenübertragung findet auf drei Ebenen statt:

- der Hardware-Ebene
- der Prozedur-Ebene
- der Telegramm-Ebene

Hardware-Ebene:

Die Hardware-Ebene umfasst die physikalischen Bedingungen der Datenübertragung. Hierzu gehören die Schnittstellenparameter, wie Baudrate, Bitanzahl oder das Paritätsbit. Die Hardwareparameter sind auf beiden Kommunikationspartnern gleich einzustellen.

Prozedur-Ebene:

Sämtliche Telegramme der RK512 (Befehls- und Reaktionstelegramme) werden von der Prozedur-Ebene der Rechnerkopplung übertragen, d.h. mit der Prozedur 3964 oder 3964R gesendet oder empfangen.

Die Prozedur verpackt die Telegramme mit einer Start- und Endsequenz, versieht die entstehenden Datenblöcke mit einem Blockprüfzeichen (nur bei der Prozedur 3964R) und wickelt bei fehlerhafter Telegrammquittung des Empfängers selbständig Telegrammwiederholungen ab.

Telegramm-Ebene:

Die Telegramm-Ebene entspricht der Transportschicht des ISO/OSI-Schichtenmodells. Der Kommunikationsmaster sendet ein Befehlstelegramm an den Partner. Dieser sendet ein Reaktionstelegramm, eventuell mit einem Fehlercode, an den Kommunikationsmaster zurück.

Im Unterschied zur Prozedur 3964(R) enthält die Rechnerkopplung RK512 neben der Bit-Übertragungsschicht und der Sicherungsschicht zusätzlich die Transportschicht (Telegramm-Ebene), bei der jede Datenübertragung vom Empfänger quittiert wird.

2. Telegrammstrukturen

Die folgenden Telegrammstrukturen enthalten aufgrund der Übersichtlichkeit nur die Telegrammdaten. Diese Daten werden bei der Übertragung in einem Frame, mit Start- und Endkennung gekapselt. Optional kann noch eine Blockprüfsumme beigefügt sein.

2.1 Aufbau des Befehls-Telegrammkopfes

Der Telegrammkopf einer Übertragungsanfrage hat eine feste Größe und besteht immer aus 10 Bytes.

Byte	Bedeutung *
1	Telegrammkennung bei Befehls-Telegrammen (00H), bei Folge-Befehls-Telegramm (FFh)
2	Telegrammkennung (00H)
3	'A' (41H) - für SEND-Auftrag mit Ziel-DB oder 'E' (45H) - für FETCH-Auftrag
4	zu übertragende Daten sind aus: 'D' (44H) = Datenbaustein
5	Datenziel bei SEND-Auftrag bzw. Datenquelle bei FETCH-Auftrag z.B. Byte 5 = DB-Nr
6	Byte 6 = DW-Nr
7	Länge High-Byte Länge der zu übertragenden Daten
8	Länge Low-Byte Typ in Wörtern für Datenbausteine (max. 65535 Datenwörter)
9	ohne Koordinierungsmerker, immer FFH.
10	Bit 0 bis 3: ohne Koordinierungsmerker Bit 4 bis 7: ohne CPU-Nummer (Zahl von 1 bis 4) hier steht FFH.

Tabelle 1 : RK512 Telegrammkopf

(*) Mit den hier beschriebenen Bausteinen ist ein begrenzter Funktionsumfang realisiert (siehe Einschränkungen).

2.2 Aufbau des Reaktionstelegrammes

Das Reaktionstelegramm besteht aus 4 Byte und enthält Informationen über den Verlauf des Auftrages. Übertragungsfehler beschreibt das Byte 4.

Byte	Bedeutung
1	Telegrammkennung bei Reaktionstelegrammen (00H), bei Folge-Reaktionstelegramm (FFh)
2	Telegrammkennung (00H)
3	Fest belegt mit 00H
4	Fehlernummer des Partners: 00H wenn bei Übertragung kein Fehler aufgetreten ist. Im Fehlerfall steht hier die Fehlernummer. Bei einer Kommunikation mittels Third-Party Produkten ist die Fehlernummer der entsprechenden Dokumentation zu entnehmen.

Tabelle 2 : RK512 Reaktionstelegramm

2.3 Ablauf einer Fetchanfrage

Der FETCH-Auftrag läuft in folgender Reihenfolge ab:

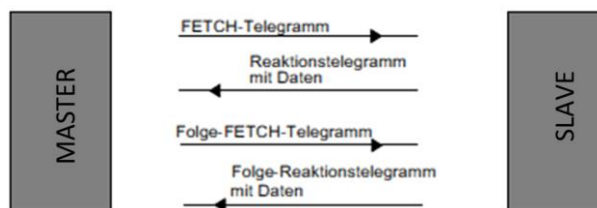


Abbildung 2-1: Ablauf Fetchanfrage

Aktiver Partner	Sendet ein FETCH-Telegramm. Dieses enthält den Telegrammkopf.
Passiver Partner	Empfängt das Telegramm, überprüft den Telegrammkopf und quittiert mit einem Reaktionstelegramm. Dieses enthält die angefragten Daten.
Aktiver Partner	Empfängt das Reaktionstelegramm. Überschreitet die Nutzdatenmenge 128Bytes, sendet er ein Folge-FETCH-Telegramm.
Passiver Partner	Empfängt das Folge-FETCH-Telegramm, überprüft den Telegrammkopf und quittiert mit einem Folge-Reaktionstelegramm mit weiteren Daten.

Die Wiederholung der Folgetelegramme erfolgt bis alle Daten übertragen sind.



Bei einer Fehlernummer ungleich 0 im 4. Byte des Reaktionstelegramms werden keine Daten gespeichert. Die Datenanfrage muss neu gestellt werden.

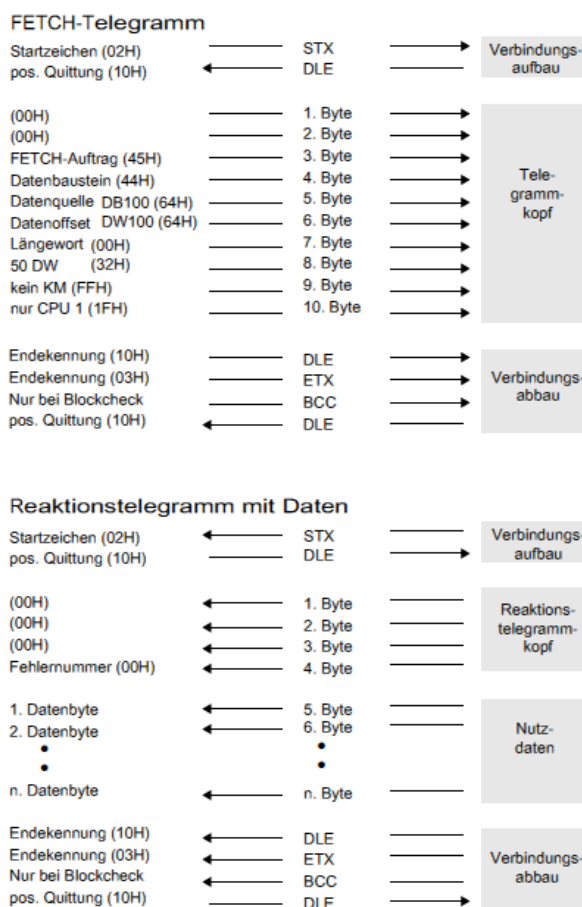


Tabelle 3 : RK512 Fetch Prozedur

2.4 Ablauf eines Sendeauftrag

Im folgenden Bild ist der Ablauf der Datenübertragung beim Senden mit einem Reaktionstelegramm bei der Rechnerkopplung RK 512 schematisch abgebildet:

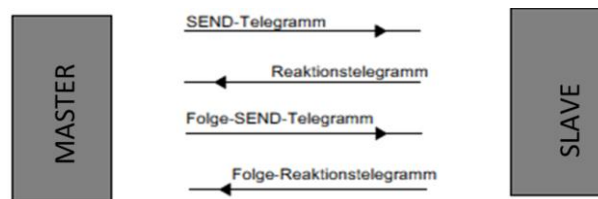


Abbildung 2-2: Ablauf Sendetelegramm

Aktiver Partner	Sendet ein SEND-Telegramm. Dieses enthält den Telegrammkopf und Daten.
Passiver Partner	Empfängt das Telegramm, überprüft den Telegrammkopf und die Daten und quittiert mit einem Reaktionstelegramm.
Aktiver Partner	Empfängt das Reaktionstelegramm. Wenn die Nutzdatenmenge 128 Byte überschreitet, sendet er ein Folge-SEND-Telegramm.
Passiver Partner	Empfängt das Folge-SEND-Telegramm, überprüft den Telegrammkopf und die Daten und quittiert mit einem Folge-Reaktionstelegramm.

Die Wiederholung der Folgetelegramme erfolgt bis alle Daten übertragen sind.



Falls das SEND-Telegramm vom Slave nicht fehlerfrei verarbeitet werden konnte, trägt dieser eine Fehlernummer ins 4. Byte des Reaktionstelegramms ein.

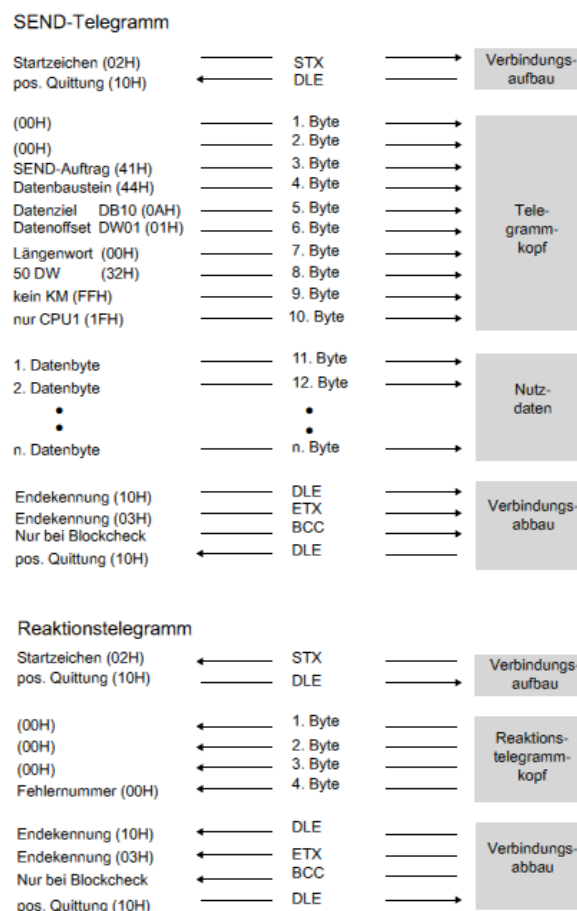


Tabelle 4 : RK512 Sende Prozedur

2.5 Fehlermeldungen

Die Fehlermeldungen beziehen sich auf den Master- und Slave-Baustein.

Die Fehlernummer beschreibt den internen Verbindungsstatus und die Antworten des Reaktionstelegrammes. Die Fehlernummern sind nicht genormt und beziehen sich ausschließlich auf die hier beschriebenen Programmbausteine.

Fehlernummer	Fehlerbeschreibung	Abhilfe
01H	Bereich (Anfragetyp , Anfangsadresse, Länge, Datenbaustein) nicht zulässig (eventuell negativer Wert)	Parametrierung auf CPU und CP prüfen und eventuell korrigieren. Datenbaustein für Nutzdaten existiert nicht Angefragte Datenlänge (LENGTH) ist NULL oder grösser 64 Auftragstyp (TYP) ist nicht ‚E‘ oder ‚A‘ Die Anfragedatenlänge und Offset (LENGTH + DW_NR) überschreiten die Grösse des Datenbausteines für die Nutzdaten. (LENGTH und DW_NR werden in WORT angegeben).
2AH	Fehler im Aufbau des empfangenen Reaktionstelegrammes: 1. Byte nicht 00H	Erste Byte im Protokollkopf ist nicht NULL. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
32H	falscher Bereich beim Anzeigenwort oder Datenbereich nicht vorhanden (außer DB/DX) oder Datenbereich zu kurz (außer DB/DX)	Kontrollieren Sie, ob der gewünschte Datenbereich beim Partner vorhanden und ausreichend groß ist bzw. prüfen Sie die Datenlänge (LENGTH) und den Offset (DW_NR)
36H	Partner erkennt falsche Telegrammlänge (Gesamtlänge). Die Anzahl der Empfangenen Daten passt nicht mit der Datenanzahl im Protokollkopf überein.	Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
40H	Fehler im 1. Befehlsbyte: ist nicht 00H oder FFH	Prinzipieller Kopfaufbaufehler beim Partner. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) nachweisen, das in die Übertragungsleitung eingeschaltet wird.
41H	Fehler im 3. Befehlsbyte: Befehlsbuchstabe ist nicht "A" oder "O" oder "E"	Prinzipieller Kopfaufbaufehler beim Partner bzw. eine Befehlskombination wurde verlangt, die nicht erlaubt ist. Kontrollieren Sie in den Befehlstabellen die zulässigen Befehle. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
43H	Fehler im 4. Befehlsbyte: Befehlsbuchstabe falsch	Prinzipieller Kopfaufbaufehler beim Partner bzw. eine Befehlskombination wurde verlangt, die nicht erlaubt ist. Kontrollieren Sie in den Befehlstabellen die zulässigen Befehle. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
45H	Fehler im 5. Befehlsbyte: DB-Nr. unzulässig (z. B. 00 oder Datenbaustein nicht vorhanden)	Entnehmen Sie den Auftragstabellen die zulässigen DB-Nummern, Anfangsadressen oder Längen.
46H	Fehler im 5. bzw. 6. Befehlsbyte: Anfangsadresse zu hoch	
47H	Fehler im 7. und 8. Befehlsbyte: Länge ist unzulässig	
E2H	Quittungsverzugszeit (QVZ) überschritten: nach Senden von STX oder einer Anfrage kam keine Antwort vom Partner innerhalb der Timeout-Zeit.	Partnergerät ist zu langsam oder nicht empfangsbereit oder es liegt z.B. ein Bruch der Sendeleitung vor. Fehlverhalten des Partnergerätes ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
F4H	Fehler beim Blockprüfzeichen BCC (nur bei RK 512 mit Prozedur 3964R): Der intern gebildete Wert des BCC stimmt nicht mit dem vom Partner am Verbindungsende empfangenen BCC überein.	Prüfen Sie, ob die Verbindung stark gestört ist, in diesem Fall werden auch gelegentlich Fehlercodes FE zu beobachten sein. Fehlverhalten des Partnergerätes ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
F8H	Empfangsdatenmenge zu groß, Daten können nicht ausgewertet werden. Empfangsdaten werden verworfen.	Sendeauftrag überprüfen, Protokoll analysieren

Tabelle 5 : RK512 Fehlercodes

3. Beschreibung der Bausteine und Schnittstellenparameter

Standardbausteine und Systemvoraussetzungen:

Für die serielle Kommunikation werden die Funktionsbausteine aus der Standard-Bibliothek für die Freeport-Kommunikation verwendet. Diese Bausteine sind in die RK512-Funktionsbausteine eingebunden und werden dort aufgerufen. Die Bausteine für die Freeport-Kommunikation gehören zum Lieferumfang der seriellen CM der S7-1200/1500 und müssen auf der Steuerung vorhanden sein.

- Send_P2P
- Receive_P2P

Treiberbausteine RK512:

Die RK512-Treiberbausteine bestehen aus 2 Funktionsbausteinen, der Auftragsdaten-Struktur und den Datenbausteinen für die Nutzdaten. Die Daten-Bausteine für die Nutzdaten sind in der entsprechenden Größe anzulegen.

Die RK512-Treiberbausteine dürfen für jede serielle Schnittstelle nur einmal verwendet werden. Die Nummer des Hardwareport legt die logische Verbindung des Kommunikationsbausteines zur seriellen Schnittstelle fest. Der Hardwareport ist der Systemkonfiguration zu entnehmen.

- MESA_RK512_MSTR Treiber für RK512 als Master
- MESA_RK512_SLAV Treiber für RK512 als Slave

Datenstrukturen für die Initialisierung der RK512 Aufträge (nur Master):

Eine Auftragsliste verwaltet die Datenanfragen und Datensendungen die der Master ausführen soll. Die Daten der Auftragsliste bestehen aus einzelnen Aufträgen. Die Auftragsliste und deren Parameter werden in einem Datenbaustein gespeichert. Die Anzahl der Aufträge ist nur durch die max. Größe des Datenbausteines begrenzt.

- MESA_RK512_JOB Datenstruktur für die Auftragsverwaltung. Ein Array fasst die einzelnen Aufträge in einer Auftragsliste zusammen.

Jeder Auftrag („Job“) wird durch eine Datenstruktur beschrieben. Die Struktur legt die Verbindungsparameter mit den zu übertragenen Daten fest und enthält Variablen für die Fehlererkennung.

Die Struktur benötigt nur der Baustein ‚MESA_RK512_MSTR‘, der Kommunikationsbaustein ‚MESA_RK512_SLAV‘ ist passiv und bezieht die Datenanfrageparameter über den Telegrammkopf des anfragenden Protokoll-Masters.

Für eine korrekte Anfrage müssen die Parameter TYP, LENGTH, REM_DB_NR, REM_DW_NR, LOC_DB_NR, LOC_DW_NR in der Auftragsliste versorgt werden.

Die Variablen DONE, RK512ERR, SYSTAT dienen als Rückgabewert und Fehleranalyse.

Die Datenstruktur und Parameterbeschreibung eines Auftrages (Jobs[?]):

Variable	Typ	Vorbelegung	Beschreibung																				
TYP *1	Char	'?'	'E' = Fetch; 'A' = Send. Bei einem unbekanntem Zeichen wird der Auftrag nicht bearbeitet! Über ein ',' lassen sich so einzelne Aufträge deaktivieren. Deaktive Aufträge führen nicht zum Setzen des Gruppenfehler ,QGrpErr'.																				
LENGTH *1	Int	0 (1 .. 65535)	Anzahl der zu lesenden/schreibenden WORD's																				
REM_DB_NR *1	Int	0 (1..255)	Remote Bereich der Datenquelle oder Ziel für Sendedaten. Dieser Datenbaustein muss auf der remote Steuerung vorhanden sein. Die Größe der Datenbausteine ist der Datenmenge LENGTH*2 + REM_DW_NR anzupassen, um alle Daten darin zu speichern.																				
REM_DW_NR *1	Int	0 (0..511)	<p>! Offset Nr. des Datenwortes im S7-Format. Daten in Datenbausteinen werden in S7 byteweise adressiert, d. h. Wort-Adressen bei S5 werden (durch Multiplikation mit 2) in Byte-Adressen gewandelt. Es ist der Unterschied zwischen S5 und S7 zu beachten.</p> <table border="1"> <thead> <tr> <th>Step 5</th> <th>Step7</th> <th colspan="2">Bytes</th> </tr> </thead> <tbody> <tr> <td>DW 0</td> <td>DBW 0</td> <td>BYTE 0</td> <td>BYTE 1</td> </tr> <tr> <td>DW 1</td> <td>DBW 2</td> <td>BYTE 2</td> <td>BYTE 3</td> </tr> <tr> <td>DW 2</td> <td>DBW 4</td> <td>BYTE 4</td> <td>BYTE 5</td> </tr> <tr> <td>DW 3</td> <td>DBW 6</td> <td>BYTE 6</td> <td>BYTE 7</td> </tr> </tbody> </table>	Step 5	Step7	Bytes		DW 0	DBW 0	BYTE 0	BYTE 1	DW 1	DBW 2	BYTE 2	BYTE 3	DW 2	DBW 4	BYTE 4	BYTE 5	DW 3	DBW 6	BYTE 6	BYTE 7
Step 5	Step7	Bytes																					
DW 0	DBW 0	BYTE 0	BYTE 1																				
DW 1	DBW 2	BYTE 2	BYTE 3																				
DW 2	DBW 4	BYTE 4	BYTE 5																				
DW 3	DBW 6	BYTE 6	BYTE 7																				
LOC_DB_NR *1	Int	0 (1..59999) *2	Lokaler Datenbereich der S7-1200/1500 für Fetch-Anfragen oder als Ziel für Sendeaufträge. Dieser Datenbaustein muss auf der lokalen Steuerung vorhanden sein. Die Größe des Datenbausteines ist der Datenmenge LENGTH*2 + LOC_DW_NR anzupassen, um alle Daten darin zu speichern.																				
LOC_DW_NR *1	Int	0 (1..65535) *2	Offset Nr. des Datenwortes auf der S7-1200/1500. Es ist der Unterschied zwischen S5 und S7 zu beachten.																				
DONE	Bool	false	AUSGANG: Wird für einen Zyklus DONE = ,true', wenn Auftrag erfolgreich abgearbeitet wurde.																				
RK512ERR	Byte	16#0	AUSGANG: Fehlercode vom Reaktionstelegramm, siehe Fehlercode																				
SYSTAT	Byte	16#0	AUSGANG: interne Status- und Fehlermeldungen, siehe Fehlercode																				

Tabelle 6 : RK512_JOB

(*1) Parameter muss versorgt werden!

(*2) Die Bereichsangaben beziehen sich auf eine S7-1200

Im Anhang ,Performance Test' wird eine Beispielkonfiguration für die Parametrierung der Datenaufträge beschrieben.

3.1. Freischaltcode für vollen Funktionsumfang

Um den vollständigen Funktionsumfang zu erhalten ist ein Freischaltcode erforderlich. Dieser wird aus der Seriennummer der CPU gebildet. Ohne Freischaltcode ist die Funktion der Kommunikationsbausteine zeitlich auf 15 Minuten limitiert.

Der Freischaltcode ist der Seriennummer der CPU oder der Speicherkarte zugeordnet und wird nur bei Neustart der CPU ausgewertet.

Im Anhang ist beschrieben, wie die Seriennummer zu ermitteln ist und die Freischaltung für den vollen Funktionsumfang erfolgt.

Bei den im Internet zum Download zur Verfügung gestellten Bausteinen handelt es sich ausschließlich um Demo-Versionen. Diese können mit dem Freischaltcode nicht aktiviert werden. Zum Erhalt einer Vollversion wenden Sie sich bitte an die Firma *MESA Gesellschaft für Elektrotechnik*.

Der Freischaltcode ist am Eingang ‚Registration_Code‘ einzustellen.

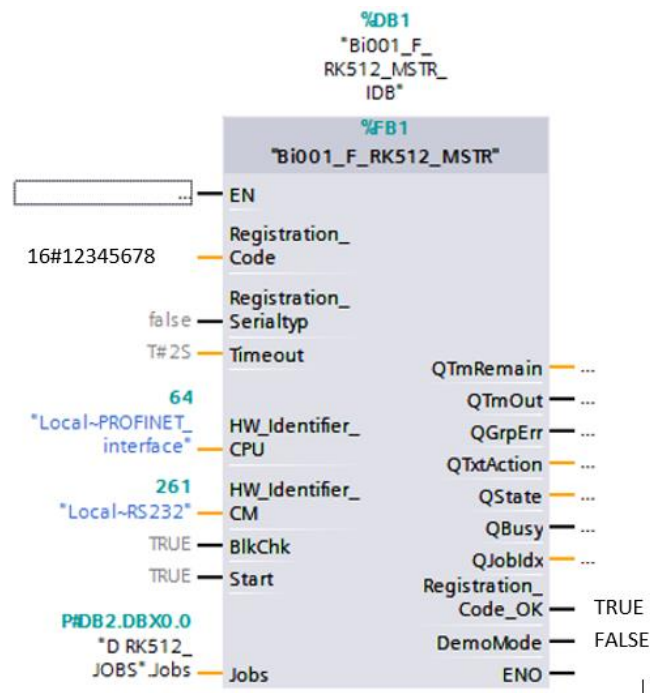


Abbildung 3-1: Registration_Code

Nach dem Ändern des ‚Registration_Code‘ ist die Software zu laden und ein Warmstart durchzuführen.

Hinweis: Der ‚Registration_Code‘ wird nur nach einem Warmstart gelesen.

Die Einstellparameter können im Speicherbereich des Instanz-Datenbaustein kontrolliert werden.

Input				
Registration_Code	DWord	16#0		16#12345678
Registration_Serialtyp	Bool	false		TRUE
Timeout	Time	T#2S		T#2S
HW_Identifier_CPU	HW_INTERFACE	0		64
HW_Identifier_CM	PORT	0		265
BlkChk	Bool	TRUE		TRUE
Start	Bool	TRUE		TRUE
Output				
QmRemain	Time	T#0ms		T#0MS
QmOut	Bool	false		FALSE
QGrpErr	Bool	false		TRUE
QTxtAction	String[32]	''		'Start'
QState	Word	16#0		16#7002
QBusy	Bool	false		TRUE
QJobIdx	Int	0		3
Registration_Code_OK	Bool	false		TRUE
DemoMode	Bool	false		FALSE
InOut				
Jobs	Array[*] of *Bi002_...			
Static				
DemoTime	Time	T#0ms		T#0MS
SerialNumber	String[16]	''		S C-ABCD12345678

Abbildung 3-2: Registration_Code im Instanzdatenbaustein

3.2. Nutzdatenspeicher

Zur Speicherung der Nutzdaten sind Datenbausteine erforderlich. Die Variable ‚...DB_NR‘ in der Auftragsstruktur legt die Datenbausteinnummer fest (siehe oben). Die Größe der Datenbausteine ist der zu übertragenen Datenmenge anzupassen. In den Datenbausteinen kann eine, der Nutzdaten angepasste Struktur hinterlegt sein.



Bei den Datenbausteinen für die Speicherung der Nutzdaten muss der ‚optimierte Bausteinzugriff‘ deaktiviert sein.

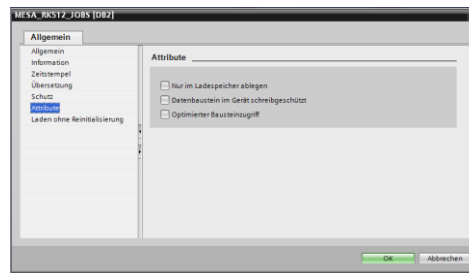


Abbildung 3-3: Datenbaustein ohne ‚Optimierten Zugriff‘

Die Parametrierung eines nicht vorhandenen Datenbausteines führt zu einer Fehlermeldung!

3.3. Auftragsliste (Jobs)

Die einzelnen Aufträge mit den Datenanfragen oder -sendungen werden in einer Auftragsliste beschrieben und in einem Datenbaustein gespeichert. Der Auftragsdatenbaustein besteht aus einem Feld mit beliebig vielen Aufträgen („Jobs“). Die Anzahl der Aufträge ist nur vom Systemspeicher begrenzt.



Der Auftragsdatenbaustein wird als Eingangsparameter dem Treiberbaustein ‚MESA_RK512_MSTR‘ übergeben. Beim Auftragsdatenbaustein muss der ‚optimierte Bausteinzugriff‘ freigegeben sein.

Die Auftragsliste wird vom ‚MESA_RK512_MSTR‘ sequentiell als Ringspeicher abgearbeitet. In der Auftragsliste können unbenutzte Aufträge („Jobs“) mit dem Typ =“?” gekennzeichnet werden. Bei der Abarbeitung der Auftragsliste werden diese Aufträge übersprungen.

Name	Datentyp	Offset	Beobachtungswert	Kommentar
Static				
Jobs	Array[0..7] of 'RK512_JOB'	0.0		
Jobs[0]	'RK512_JOB'	0.0		
TYP	Char	0.0	'E'	'E' = Fetch; 'A' = Send
LENGTH	Int	2.0	64	Anzahl der zu lesendenschreibenden WORD's.
REM_DB_NR	Int	4.0	70	Remote Datenbaustein für Fetchanfrage oder als Ziel für Send : REM_DB_NR < 256
REM_DW_NR	Int	6.0	0	Remote Offset Nr. des Datenwortes : REM_DW_NR < 512
LOC_DB_NR	Int	8.0	700	Lokale Datenbaustein
LOC_DW_NR	Int	10.0	0	Lokale Offset Nr. des Datenwortes
DONE	Bool	12.0	TRUE	True, wenn Auftrag erfolgreich abgearbeitet wurde
RK512ERR	Byte	13.0	16#00	Fehlercode vom Reaktionstelegramm
SYSTAT	Byte	14.0	16#00	interne Status- und Fehlermeldungen
Jobs[1]	'RK512_JOB'	16.0		
Jobs[2]	'RK512_JOB'	32.0		
TYP	Char	32.0	'E'	'E' = Fetch; 'A' = Send
LENGTH	Int	34.0	64	Anzahl der zu lesendenschreibenden WORD's
REM_DB_NR	Int	36.0	70	Remote Datenbaustein für Fetchanfrage oder als Ziel für Send : REM_DB_NR < 256
REM_DW_NR	Int	38.0	128	Remote Offset Nr. des Datenwortes : REM_DW_NR < 512
LOC_DB_NR	Int	40.0	700	Lokale Datenbaustein
LOC_DW_NR	Int	42.0	0	Lokale Offset Nr. des Datenwortes
DONE	Bool	44.0	FALSE	True, wenn Auftrag erfolgreich abgearbeitet wurde
RK512ERR	Byte	45.0	16#00	Fehlercode vom Reaktionstelegramm
SYSTAT	Byte	46.0	16#00	interne Status- und Fehlermeldungen
Jobs[3]	'RK512_JOB'	48.0		
Jobs[4]	'RK512_JOB'	64.0		
Jobs[5]	'RK512_JOB'	80.0		
Jobs[6]	'RK512_JOB'	96.0		
Jobs[7]	'RK512_JOB'	112.0		

Abbildung 3-4: Auftragsdatenstrukturen RK512 (Master)

Für Datenquelle und Ziel können unterschiedliche Datenbausteine und Offsetbereiche angegeben werden. Die Variablen mit dem Prefix „*REM...*“ beziehen sich immer auf die Daten der Partnerstation. Der Prefix „*LOC...*“ definiert den eigenen Datenbereich, also den Datenbereich des RK512 Masters.

3.4. MESA_RK512_MSTR

Der Baustein koordiniert als Master den Datenverkehr für das RK512 Protokoll. Die Datenanfragen oder Daten-sendungen werden durch einzelne Aufträge in einen Auftragsdatenbaustein konfiguriert. Die Auftragsdaten sind als Aufrufparameter dem ‚MESA_RK512_MSTR‘ zu übergeben.

Es können beliebig viele Aufträge („Jobs“) einer Kommunikationsschnittstelle zugeordnet werden. Ein Auftrag wird mit der Struktur ‚RK512_JOB‘ beschrieben. Weitere Information zur Struktur ‚RK512_JOB‘ und deren Parameter finden sich im vorherigen Kapitel.

Die Parameter für einen Auftrag legen die Datenbereiche, den Offset, die Länge und den Auftragsstyp (senden oder empfangen) fest. Für die Fehlerauswertung stehen zusätzliche Variablen zur Verfügung.

Die Aufträge werden in einem Auftragsdatenbaustein gelistet. Die Liste besteht aus einem Feld mit den einzelnen Aufträgen („Jobs“). Der Treiberbaustein ‚MESA_RK512_MSTR‘ arbeitet die Liste mit den einzelnen Aufträgen sequentiell ab. Dabei erkennt die Software selbstständig die Anzahl der parametrisierten Aufträge. Ist die Abarbeitung bei dem letzten parametrisierten Auftrag angelangt, beginnt die Software mit der Abarbeitung wieder von vorn mit dem Auftrag aus dem ersten Feld.

Sind für einen Auftrag keine gültigen Parameter vorhanden, wird dieser Auftrag bei der Abarbeitung übersprungen und mit dem nächstgültigen fortgefahren.

Jeder Auftrag besitzt einen Parameter für eine Datenbausteinnummer zur Ablage der Empfangs- oder Sendedaten. Der Datenbaustein muss ausreichend groß dimensioniert und geladen sein. Fehlt der Datenbaustein, wird der Auftrag nicht bearbeitet und ein Fehlercode für diesen Auftrag gesetzt.

Der Eingang ‚Start‘ = ‚TRUE‘ startet die Abarbeitung der Auftragsliste mit den parametrisierten Datenanfragen/Datensendungen. Beim Rücksetzen der Startanforderung ‚Start‘ = ‚FALSE‘ während einer aktiven Datenanfrage, wird die aktive Anfrage noch vollständig abgearbeitet. Die Wegnahme der Startanforderung entspricht einem Pause-Signal.

Ein erneutes Setzen der Startfreigabe ‚Start‘ = ‚TRUE‘ bewirkt die Fortsetzung der Abarbeitung der Auftragsliste. Die Abarbeitung der Aufträge beginnt dann nicht von vorn, sondern mit dem nach dem Stoppsignal folgenden Auftrag.

VAR_INPUT

Eingang	Datentyp	Beschreibung
Registration_Code	DWORD	Freischaltcode für den Funktionsbaustein
Registration_Serialtyp	Bool	Zuordnung des Freischaltcode: 0 = an die CPU, 1 = an die Speicherkarte
Timeout	Time	Max. Zeit für Datenantworten bei einer erwarteten Rückantwort. Bei einer fehlenden Verbindung sendet der MASTER zyklisch ein ‚STX‘ und wartet für die Zeit ‚Timeout‘ auf eine Antwort vom Slave. Antwortet der Master nicht innerhalb der angegebenen Zeit, sendet der Master ein ‚NAK‘ und beginnt eine neue Startsequenz mit einem ‚STX‘. Das Timeout wird mit jedem Empfangenen Zeichen neu angestoßen.
HW_Identifizier_CPU	HW_INTERFACE	"HW-Kennung" der CPU, ist der Gerätekonfiguration zu entnehmen (siehe Kapitel Gerätekonfiguration).
HW_Identifizier_CM	PORT	"HW-Kennung" des Kommunikationsmoduls, ist der Gerätekonfiguration zu entnehmen (siehe Kapitel Gerätekonfiguration).
BlkChk	Bool	Enable Blockcheck, BLKCHKC = ‚true‘ bedeutet mit Blockcheck-Berechnung
Start	Bool	bei Start = ‚true‘, werden Daten zyklisch angefragt, Bei Start = ‚false‘ wird die Auftragsbearbeitung nach dem aktuell anstehenden Auftrag beendet.

Tabelle 7 : MESA_RK512_MSTR – VAR_INPUT

VAR_IN_OUT

Eingang	Datentyp	Beschreibung
Jobs	Array [*] of „RK512_JOB“	Zeiger auf ein Feld mit den Aufträgen. Ein Auftrag enthält die Parameter der Datenanfragen, sowie deren Fehlerzustände. Es ist wird der Zeiger auf den Datenbaustein mit der Auftragsliste übergeben.

Tabelle 8 : MESA_RK512_MSTR – VAR_IN_OUT

VAR_OUTPUT

Ausgang	Datentyp	Beschreibung
QTmRemain	Time	Verbleibende Zeit für Timeout. Ein RK512-Slave muss innerhalb der Timeoutzeit die Anfrage beantworten. Das Überschreiten der Zeit führt zu einem Fehler und Abbruch der aktuellen Auftragsbearbeitung und setzen des Ausgangs ‚QTmOut‘.
QTmOut	Bool	Timeout-Fehler, wird für einen Zyklus ‚true‘ gesetzt wenn Anfragen nicht beantwortet werden. Es wird ein ‚NAK‘ gesendet und die Abarbeitung der Auftragsliste mit dem nächsten Auftrag fortgeführt. Die Timeout-Zeit wird mit dem Eingangsparameter ‚Timeout‘ eingestellt.
QGrpErr	Bool	Gruppenfehler, QGrpErr = ‚true‘ wenn in irgendeinem Auftrag ein Fehler auftritt. Der Fehler wird erst nach erfolgreicher Abarbeitung aller Anfrage zurückgesetzt. Deaktivierte Aufträge führen nicht zu einem Gruppenfehler (siehe Auftragsstruktur Variable TYP=‘?’).
QTxtAction	String[32]	Klartextanzeige für den aktuellen Bearbeitungszyklus
QState	WORD	Zustand der Sende Kommunikation. Liefert den Status der Funktion Send_P2P. Die Beschreibung des Status ist der Hilfe für den Bausteines ‚SendP2P‘ zu entnehmen.
QBusy	Bool	QBusy ist ‚true‘ solange eine Anfrage aktiv ist, QBusy wird zwischen jeder neuen Anfrage kurzzeitig zurückgesetzt
QJobIdx	Int	aktuelle Job-Index entspricht der aktuelle Position im Feld der Aufträge (nullbasieren). Wird der Eingang ‚Start‘ während der Auftragsbearbeitung auf ‚false‘ gesetzt, zeigt QJobIdx den zuletzt abgearbeiteten Auftragsindex an.
Registration_Code_OK	Bool	wird gesetzt bei korrekten ‚Registration_Code‘
DemoMode	Bool	Ist gesetzt bei aktiven Demomode oder fehlerhaften ‚Registration_Code‘. Funktionsumfang ist dann zeitlich begrenzt auf 15 Minuten.

Tabelle 9 : MESA_RK512_MSTR – VAR_OUTPUT

Kommunikationsbaustein MESA_RK512_MSTR:

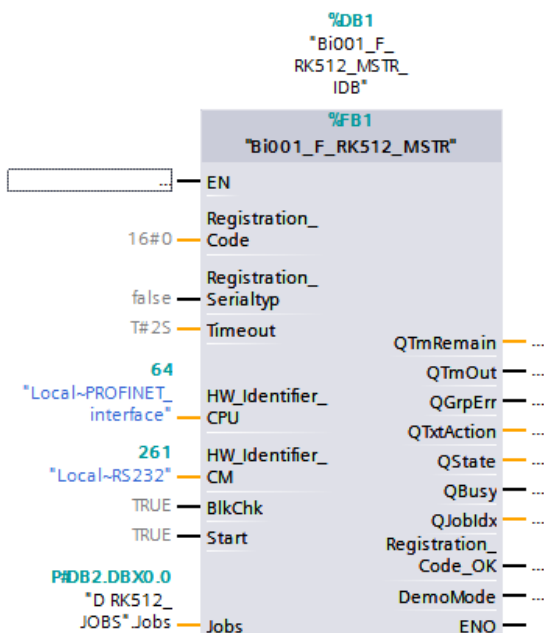


Abbildung 3-5: Funktionsbaustein MESA_RK512_MSTR

Ablaufbeschreibung einer Fetch-Anfrage (Master):

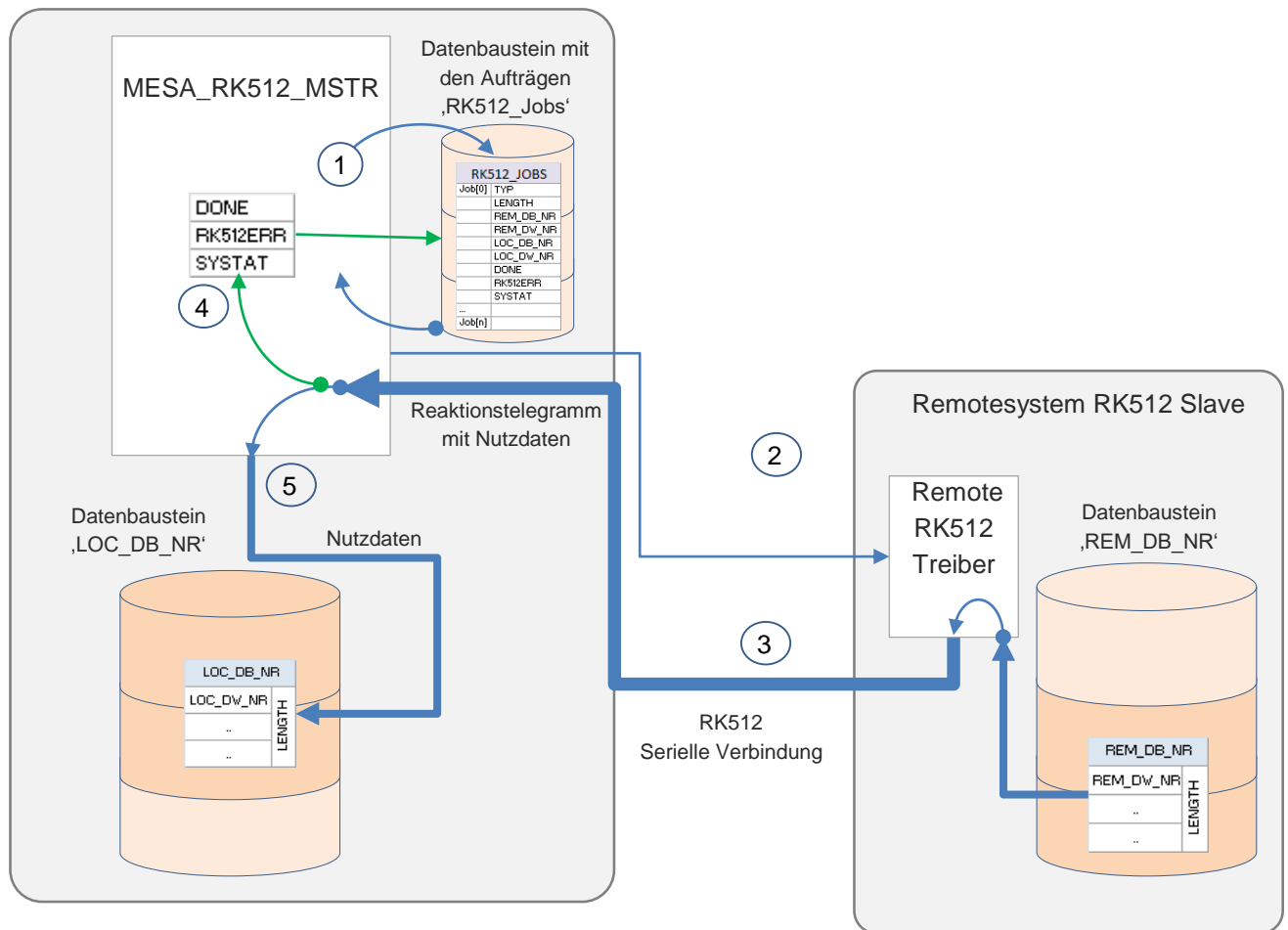


Abbildung 3-6: Ablauf einer Fetch-Anfrage MESA_RK512_MSTR

1. Der 'MESA_RK512_MSTR' polt zyklisch die Auftragsliste und generiert bei gültigen Auftragsdaten eine Anfrage.
2. Die Anfrage mit der Adressdaten des Remotebausteines (LENGTH, REM_DB_NR, REM_DW_NR) wird dem Remote-Partner über eine Fetch-Anfrage übermittelt.
3. Wenn der Partner die Anfrage verstanden hat, antwortet dieser mit einem Reaktionstelegramm und den angefragten Daten.
4. Im Auftragsdatenbaustein wird nach einer erfolgreichen Transaktion das 'DONE' gesetzt oder die Fehler in den Variablen 'RK512ERR' / 'SYSTAT' für diesen Auftrag angezeigt.
5. Bei einem positiven Reaktionstelegramm speichert der 'MESA_RK512_MSTR' die empfangenen Nutzdaten in dem lokalen Datenbereich an der Adresse wie sie mit LENGTH, LOC_DB_NR, LOC_DW_NR definiert wurde.

3.5. MESA_RK512_SLAV

Der Baustein reagiert auf RK512-Protokoll Anfragen als Slave. Die Datenanfragen werden analysiert und entsprechende Fehler gemeldet. Der , MESA_RK512_SLAV' benötigt keine Datenkonfiguration sondern ermittelt die Adressdaten anhand der empfangenen Anfrage aus dem Telegrammkopf.

Wenn der Telegrammkopf fehlerfrei ausgewertet wurde und die Datenbereiche auf der CPU vorhanden sind, wird der Master mit den Daten bedient oder es wird versucht die empfangenen Daten zu speichern. In jedem Falle muss der aus dem Telegrammkopf ermittelte Datenbaustein vorhanden sein.

Wenn der Auftrag syntaktisch korrekt war, wird ein Reaktionstelegramm generiert. In diesem steht ein entsprechender Fehlercode, wenn die Datenanfrage nicht bedient werden konnte.

Über den Eingang ,Start' kann der Abarbeitungszyklus angehalten werden. Der aktuelle anstehende Auftrag wird zuvor noch vollständig abgearbeitet.

VAR_INPUT

Eingang	Datentyp	Beschreibung
Registration_Code	DWORD	Freischaltcode für den Funktionsbaustein. Ohne Freischaltcode ist der Demomodus aktiv.
Registration_Serialtyp	Bool	Zuordnung des Freischaltcode: 0 = an die CPU, 1 = an die Speicherkarte
Timeout	Time	Max. Zeit für Datenantworten wenn der Slave auf eine Rückantwort des Masters wartet. Bei Überschreitung wird der aktuelle Anfragezyklus beendet und wieder in den Ruhezustand gewechselt um auf eine neue Anfrage zu reagieren. Das Timeout ist nur aktiv, bei einer bereits gestarteten Anfrage.
HW_Kennung_CPU	HW_INTERFACE	"HW-Kennung" der CPU, ist aus der Gerätekonfiguration zu entnehmen (siehe Kapitel Gerätekonfiguration).
HW_Kennung_CM	PORT	"HW-Kennung" des Kommunikationsmoduls, ist der Gerätekonfiguration zu entnehmen (siehe Kapitel Gerätekonfiguration).
BlkChk	Bool	Enable Blockcheck, BLKCHKC = ,true' bedeutet mit BCC
Start	Bool	bei Start = ,true', wartet auf eingehende Anfrage. Bei Start = ,false' wird die Auftragsbearbeitung nach der aktuell anstehenden Anfrage beendet.

Tabelle 10 : MESA_RK512_SLAV – VAR_INPUT

VAR_OUTPUT

Ausgang	Datentyp	Beschreibung
QTmRemain	Time	Verbleibende Zeit für TimeOut
QTmOut	Bool	Zeitfehler, wird für einen Zyklus ,true' gesetzt wenn eine ausstehende Rückfrage nicht rechtzeitig beantwortet wurde. Die Bearbeitung wird abgebrochen und auf eine neue Anfrage gewartet
QGrpErr	Bool	Gruppenfehler, 'true' wenn in der Anfrage ein Fehler auftritt. ,QGrpErr' wird nach der fehlerfreien Abarbeitung des nächsten Auftrages zurückgesetzt.
QTxtAction	String[32]	Klartextanzeige für den aktuellen Abarbeitungszyklus
QState	WORD	Zustand der Empfangs Kommunikation. Liefert den Status der Funktion ,Receive_P2P' Die Beschreibung des Status ist der Hilfe für den Bausteines ,Receive_P2P' zu entnehmen.
QBusy	Bool	ist 'true' solange eine Anfrage aktiv in Bearbeitung ist. QBusy wird ,false' solange auf eine Anfrage (STX) vom Master gewartet wird.
Registration_Code_OK	Bool	wird gesetzt bei korrekten ,Registration_Code'
DemoMode	Bool	Ist gesetzt bei aktiven Demomode oder fehlerhaften ,Registration_Code'. Funktionsumfang ist dann zeitlich begrenzt auf 15 Minuten.

Tabelle 11 : MESA_RK512_SLAV – VAR_OUTPUT

Netzwerk des Kommunikationsbausteines MESA_RK512_SLAV:

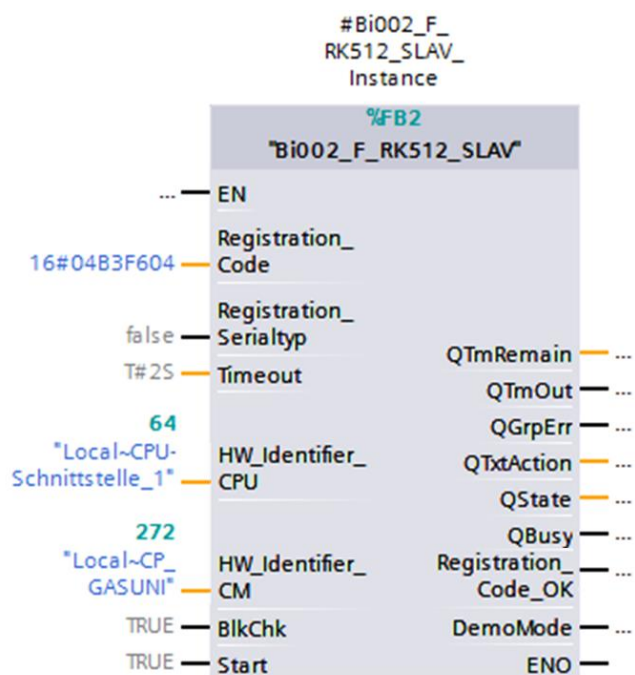


Abbildung 3-7: Funktionsbaustein MESA_RK512_SLAV

Ablaufbeschreibung eines Fetch-Auftrages (Slave):

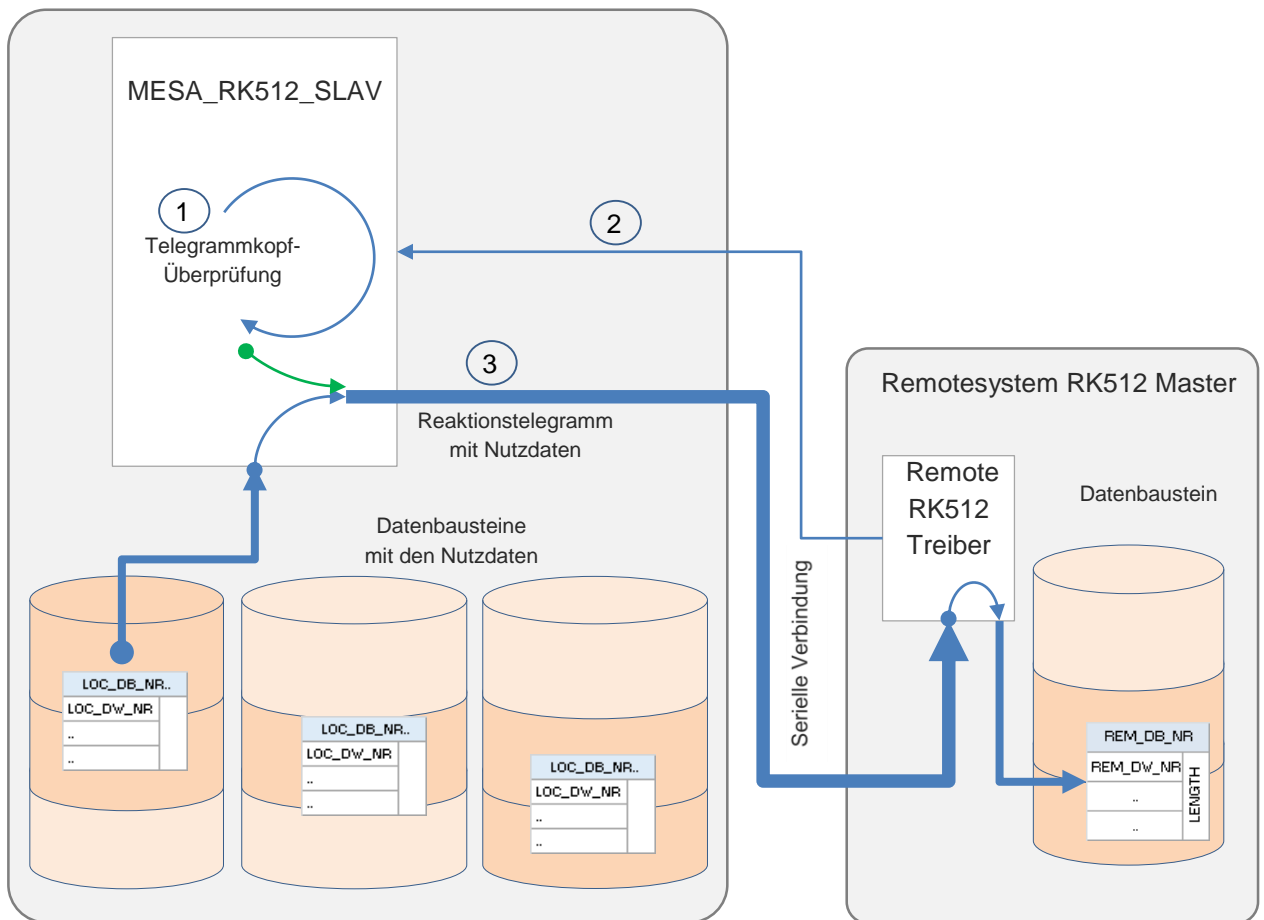


Abbildung 3-8: Ablauf einer Fetch-Anfrage MESA_RK512_SLAV

1. Der ‚MESA_RK512_SLAV‘ wartet auf eingehende Daten. Empfängt die Software innerhalb des Timeout-Zyklus keine Daten, wird der Ausgang ‚QTmOut‘ für einen Zyklus gesetzt und der Empfangszyklus neu begonnen.
2. Bei Datenempfang wird der Telegrammkopf analysiert. Es wird überprüft ob der Datenbaustein vorhanden ist und die angefragten Datenbereiche verfügbar sind
3. Wenn die angefragten Daten nicht zur Verfügung stehen, weil z. B. der Datenbaustein nicht vorhanden ist oder die angefragte Datenmenge zu groß ist, antwortet der Slave mit einem Reaktionstelegramm und einem Fehlercode.
Kann die Anfrage beantwortet werden, werden die Daten aus dem angefragten Adressbereich (DB_NR, DW_NR, LENGHT) gelesen und dem Reaktionstelegramm als Nutzdaten angehängt.

4. Gerätekonfiguration

Für einen fehlerfreien Betrieb sind Einstellungen an der Hardware erforderlich. Die Einstellungen werden in der Gerätekonfiguration unter den Eigenschaften der CPU und des CM vorgenommen.

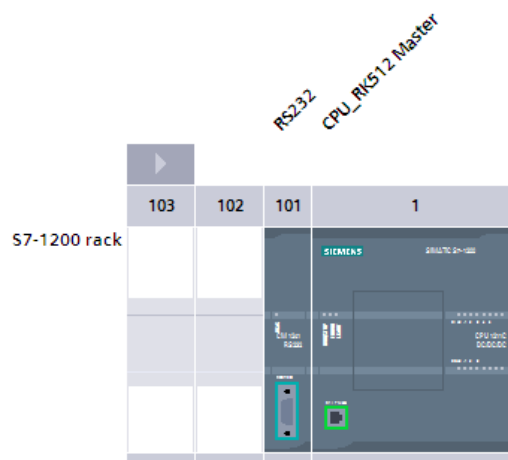


Abbildung 4-1: Hardware CPU mit CM

4.1. HW-Kennung CPU

Die Hardwaremodule der S7-1200/1500 werden durch eine Hardware-Kennung adressiert. Die Kennung ist nach der Migration der Bausteine zu ermitteln und einzutragen.

Gehen Sie folgendermaßen vor:

- Öffnen Sie die Gerätekonfiguration.
- Stellen Sie eine Verbindung zur CPU her.
- Markieren Sie das Modul, das Sie adressieren wollen.
- Wählen Sie im Inspektorfenster das Register "Eigenschaften > Systemkonstanten".
- Die Tabelle enthält Konstanten für alle verwendeten Module mit den benötigten HW-Kennungen.
- Markieren Sie die Konstante für das zu adressierende Modul und wählen Sie im Kontextmenü den Befehl "Kopieren".
- Fügen Sie die Konstante am Parameter "HW_Kennung_CPU" des Kommunikationsbausteines ein.

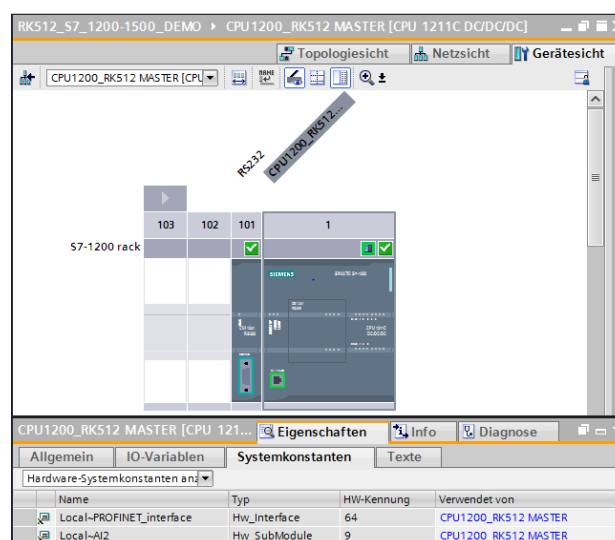


Abbildung 4-2: HW_Kennung_CPU

Sie haben die Möglichkeit, per Drag & Drop die HW-Kennung aus der Variablen-tabelle auf den Bausteinparameter "HW_Kennung_CPU" zu ziehen.

Vorteil: Der HW-Datentyp wird vom Programmiereditor geprüft und gegebenenfalls abgewiesen, falls ein nicht zulässiges HW-Objekt gewählt wurde. Bei einer numerischen Eingabe ist keine Prüfung möglich.

4.2. HW-Kennung CM

Die Hardwarekennung für das Kommunikationsmodul (CM) ist den Eigenschaften der Interfacekarte zu entnehmen.

Gehen Sie folgendermaßen vor:

- Öffnen Sie die Gerätekonfiguration.
- Stellen Sie eine Verbindung zur CPU her.
- Markieren Sie das CM-Modul, das Sie adressieren wollen.
- Wählen Sie im Inspektorfenster das Register "Allgemein -> HW-Kennung".

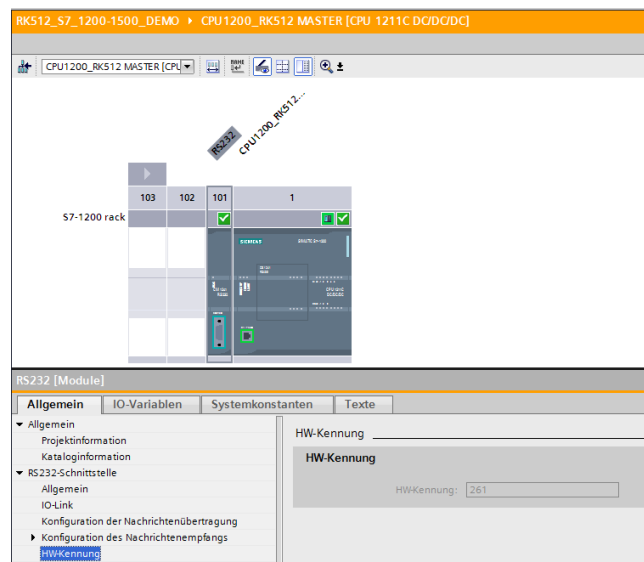


Abbildung 4-3: HW_Kennung_CM (Eigenschaften CM)

Alternativ kann der HW-Port aus den Systemkonstanten ermittelt werden. Dazu öffnen Sie die Systemkonstanten im Projekt unter dem Eintrag ‚PLC-Variablen‘. Wenn sie hier ‚alle Variablen anzeigen‘ und das Register ‚Systemkonstanten‘ auswählen, finden Sie einen Eintrag für den Port der seriellen Schnittstelle. Dieser kann mit Drag & Drop an den Kommunikationsbaustein verschaltet werden.

	Name	Datentyp	Wert
1	Local-PROFINET_interface	Hw_Interface	64
2	Local-AI2	Hw_SubModule	9
3	Local-DI14_DQ10_1	Hw_SubModule	257
4	Keiner	Pip	65535
5	Automatische Aktualisierung	Pip	0
6	TPA 1	Pip	1
7	TPA 2	Pip	2
8	TPA 3	Pip	3
9	TPA 4	Pip	4
10	TPA OB Servo	Pip	32768
11	Local-RS232	Port	261

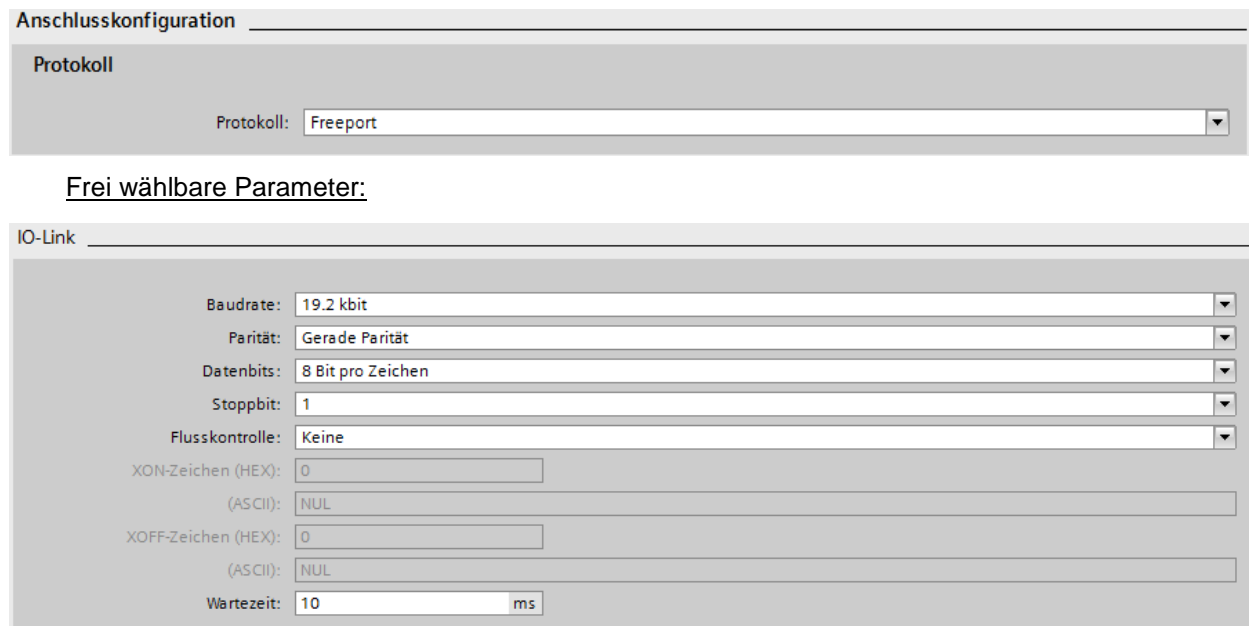
Abbildung 4-4: HW_Kennung_CM (Systemkonstanten)

4.3. Parametrierung der seriellen Schnittstelle

Die Hardwareparameter der seriellen Schnittstelle sind für beide Kommunikationspartner identisch einzustellen. Die Einstellungen werden in der Hardwarekonfiguration durchgeführt. Das folgende Bild zeigt ein Konfigurationsbeispiel für eine S7-1200, welches unter der Gerätekonfiguration die Eigenschaften der RS232-Schnittstelle zeigt.

Bei der Konfiguration einer S7-1500 findet man die Einstellungen wieder. Hier ist darauf zu achten, dass als Protokoll ‚Freeport‘ eingestellt ist. Die Einstellwerte von nicht aufgeführten Parametern sind leer zu lassen oder mit NULL zu belegen.

Der Anschluss muss als Freeport konfiguriert werden (nur CPU S7-1500):



Anschlusskonfiguration

Protokoll

Protokoll: Freeport

Frei wählbare Parameter:

IO-Link

Baudrate: 19.2 kbit

Parität: Gerade Parität

Datenbits: 8 Bit pro Zeichen

Stoppbit: 1

Flusskontrolle: Keine

XON-Zeichen (HEX): 0
(ASCII): NUL

XOFF-Zeichen (HEX): 0
(ASCII): NUL

Wartezeit: 10 ms

Abbildung 4-5: Eigenschaften der RS232-Schnittstelle

Zusätzlich zu den Schnittstellenparametern sind die Timeout-Zeiten zu parametrieren.

Bei den Kommunikationsbaugruppen für die S7-1200/1500 ist auf die Einschränkungen der Übertragungsgeschwindigkeit (Baudrate) zu achten. Nicht alle Kommunikationsbaugruppen können Daten mit einer Bitrate grösser 19200 Bits/sec übertragen!

4.4. Kommunikations-Parameter CPU

Wenn parallel zur RK512-Kommunikation eine Onlineverbindung vom Programmiergerät zur CPU besteht, kann die Kommunikationslast so groß werden, dass die Zeichenverzugszeiten nicht eingehalten werden. Dies führt zu Blockwiederholungen und Verbindungsabbrüchen im RK512-Protokoll. Durch die Zuweisung von zusätzlicher Rechenleistung für die Kommunikation kann das Problem umgangen werden. Die Zuweisung erfolgt in der Gerätekonfiguration der CPU.

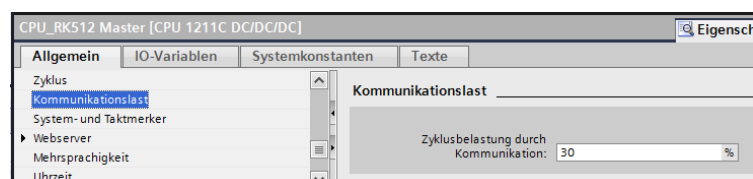


Abbildung 4-6: Einstellung der Kommunikationslast

4.5. Kommunikations-Parameter CM

Die Firmware der Kommunikationsmodule CM bieten eine Vielzahl an möglichen Einstellparameter um einfache Daten-Protokolle bereits vollautomatisch vom CM bearbeiten zu lassen. Für das RK512-Protokoll können diese Möglichkeiten nur bedingt genutzt werden. Die folgenden Bilder zeigen die notwendigen Parameter für den RK512 Betrieb.

Für das Datenprotokoll erforderliche Parameter:

Konfiguration der Nachrichtenübertragung

Übertragungseinstellungen

RTS ON Verzögerung: 0 ms

RTS OFF Verzögerung: 0 ms

Sende Pause zu Nachrichtenbeginn

Anzahl Bitzeiten in einer Pause: 0 Bitzeiten

Sende Idle Line nach Pause

Idle Line nach Pause: 0 Bitzeiten

Konfiguration des Nachrichteneingangs

> Nachrichtenbeginn

Mit beliebigem Zeichen beginnen

Mit spezieller Bedingung beginnen

Erkenne Nachrichtenbeginn bei Line Break

Erkenne Nachrichtenbeginn bei Idle Line

Dauer von Idle Line: 0 Bitzeiten

Erkenne Nachrichtenbeginn mit einzelmem Zeichen

Nachrichtenbeginn-Zeichen (HEX): 0

Nachrichtenbeginn-Zeichen (ASCII): NUL

Erkenne Nachrichtenbeginn mit einer Zeichenkette

Anzahl der zu definierenden Zeichenfolgen: 1

5-Zeichen-Sequenz zum Nachrichtenbeginn

Nachrichtenbeginn Sequenz 1

Prüfe Zeichen 1

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 2

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 3

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 4

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 5

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

...

Bedingungen zum Nachrichtenende definieren

Erkenne Nachrichtenende durch Nachrichten-Zeitüberschreitung

Nachrichten-Zeitüberschreitung: ms

Erkenne Nachrichtenende durch Antwort-Zeitüberschreitung

Antwort-Zeitüberschreitung: ms

Erkenne Nachrichtenende durch Zeitüberschreitung innerhalb der Zeichen

Zeichenlücken-Zeitüberschreitung: Bitzeiten

Erkenne Nachrichtenende durch maximale Länge

Maximale Länge der Nachricht: bytes

Nachrichtenende durch feste Nachrichtenlänge erkennen

Feste Nachrichtenlänge: bytes

Lesen Nachrichtenlänge aus Nachricht

Offset des Längenfeldes in der Nachricht: bytes

Größe des Längenfeldes: bytes

Das den Daten folgende Längenfeld gehört nicht zur N... bytes

Erkenne Nachrichtenende mit einer Zeichenkette

5-Zeichen-Sequenz zum Nachrichtenende

Prüfe Zeichen 1

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 2

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 3

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 4

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 5

Zeichenwert (HEX):

Zeichenwert (ASCII):

Empfangspuffer

Erhaltene Telegramme im Puffer:

Überschreiben verhindern

Empfangspuffer im Anlauf löschen

HW-Kennung _____

HW-Kennung

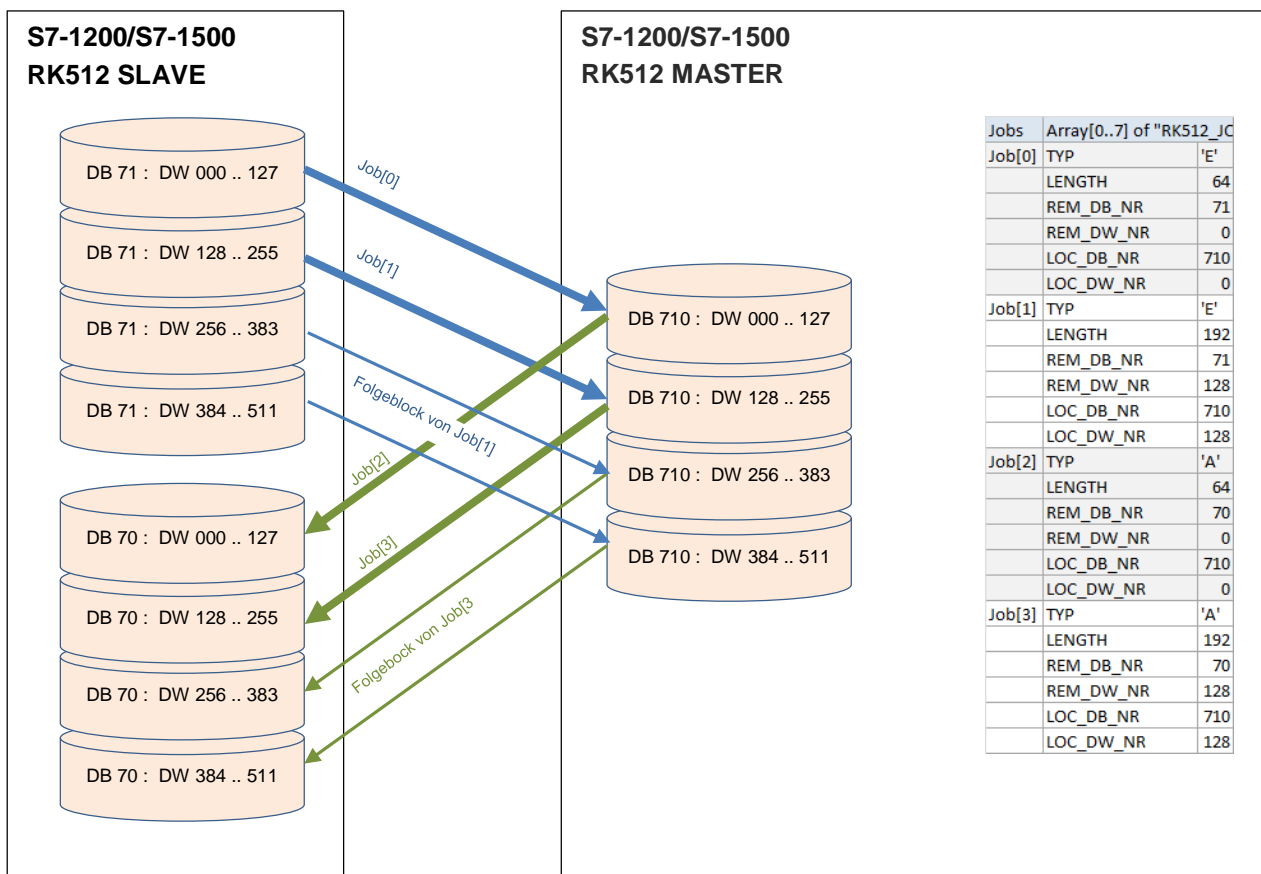
HW-Kennung:

Abbildung 4-7: Einstellung CM1241 (RS323)

5. Anhang

5.1. Performance Test

Der Leistungstest wurde mit zwei S7-1200 (CPU1211C mit CM1241) durchgeführt. Es wurde jeweils eine Steuerung als Master und die andere als Slave betrieben. Auf dem Master wurden 4 Aufträge angelegt, 2 lesende (FETCH) und 2 schreibende (SEND).



Mit der Auftragsliste (rechts) führt der Master 2 Datenanfragen ,(E)' beim Slave auf den DB71 durch und schreibt diese mit 2 Schreibkommandos ,(A)' zurück zum Slave in den DB70.

Die erste Anfrage besteht aus 64 Wörtern. Die zweite Anfrage besteht aus 192 Wörtern. Für die zweite Anfrage werden Folgeblöcke generiert, da das Datenvolumen grösser als die maximale Blockgröße des RK512-Protokolls ist. Insgesamt findet ein Datenaustausch von 256 Wörtern (512Bytes) in lesender und schreibender Richtungen statt.

Für den Datenaustausch werden ca. 1,85s benötigt bei einer Schnittstelleneinstellung von 19200,8E1. Die RK512-Bauteile wurden auf beiden Steuerungen im OB1 mit Blockscheck abgearbeitet.

Tabelle 12 : Auftragsliste für Performance-Test

5.2. Systemvoraussetzungen

Die erforderlichen Software-Versionen sind die Mindestanforderungen für die Programmbausteine.



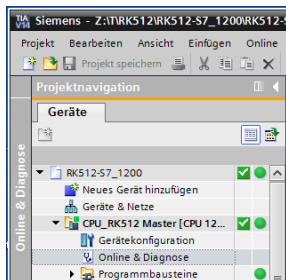
Bezeichnung	Version
TIA	V14 SP1
CPU S7-1200	Firmware V4.0
CM 1241	Firmware V2.1
CPU S7-1500	Firmware V2.0.5
CM PtP RS232 BA	Firmware V1.0

Tabelle 13 : Erforderliche Firmware-Stände

5.3. Ermittlung der CPU-Seriennummer

Die Kommunikationsbausteine sind geschützt und nur mit einem Freischaltsschlüssel zu aktivieren.

Der Freischaltsschlüssel wird aus der Seriennummer der CPU gebildet. Für jede CPU, auf der die Kommunikationsbausteine laufen, wird ein Freischaltsschlüssel benötigt. Der Schlüssel ist für die verwendeten Kommunikationsbausteine MESA_RK512_MSTR, MESA_RK512_SLAV identisch.



Ohne Freischaltsschlüssel ist der Funktionsumfang eingeschränkt.

Die Seriennummer der CPU erhält man aus der Hardwarekonfiguration. Im TIA-Projekte ist bei einer bestehenden Onlineverbindung zur CPU die Funktion ‚Online & Diagnose‘ aufzurufen.

Im Reiter ‚Diagnose -> Allgemein‘ steht im Block unter der Herstellerinformation die Seriennummer. Die Felder sind nur lesbar, Schreibzugriffe sind nicht möglich.

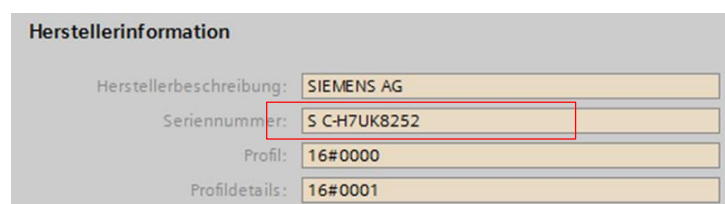


Abbildung 5-1: Seriennummer der CPU

Für die Aktivierung der Software ist der Freischaltsschlüssel am Bausteineingang ‚Freischaltcode‘ einzugeben und die SPS neu zu starten.

Hinweis: Der Freischaltcode ist mit Hochkommas (‚freischaltcode‘) zu umschließen.

5.4. Bestellformular für Funktionsbausteine

Bitte verwenden Sie für die Bestellung das folgende Formular. Es sind alle Felder bis auf das Feld für den Freischaltcode auszufüllen. Für jede CPU ist ein separates Formularblatt zu erstellen.


VORWERK Pipeline- und Anlagenservice GmbH Bereich MESA Elektrotechnik Fahrenberg 20 D- 22885 Barsbüttel		Büro Erkner Julius- Rütgers- Straße 10 D- 15537 Erkner	
Anforderung eines Freischaltcodes für Software-Funktionsbausteine Bitte die Felder ausfüllen (please fill out the fields)			
Bestellung von Softwaremodulen		Funktionsbaustein RK512 (Master- und Slave)	
Auftragsnummer:			
Anlagenbeschreibung (Project description)			
Endkunde (Customer)			
Land			
S7-1200 <input type="checkbox"/> S7-1500 <input type="checkbox"/>			
Seriennummer der CPU oder Speicherkarte, Groß- Kleinschreibung beachten (Serial number of CPU or Memorycard, case sensitiv)			
Freischaltcode, wird von MESA ausgefüllt (Release code)			
Datum: (Date)		01/01/2018	
Unterschrift: (Signature)			

Tabelle 14 : Bestellformular für Funktionsbausteine