

VORWERK Pipeline- und Anlagenservice GmbH

Bereich MESA Elektrotechnik
Fahrenberg 20
D-22885 Barsbüttel

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



Messen – Regeln - Automatisieren
Schaltanlagenbau
Elektromontagen
Service und Wartung
Kundendienst

Manual

Function Blocks
RK512 for S7-1200/1500

Note:

The content of this documentation has been checked for compliance with the hardware and software described. However, deviations cannot be excluded - so that no guarantee can be assumed for the complete agreement.



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	Folgeblöcke und Englische Version	1.6	JR
009	11.10.2017	Texte und Bilder überarbeitet, Anpassungen der FB und deren Parameter, Performance-Test	1.7	JR
010	14.05.2018	Kommunikationsprotokoll ‚Freeport‘	1.8	JR
011	04.09.2018	Logo	1.9	JR

Content

Abbreviations	V
Table Directory	6
Table of Figures.....	7
Introduction	8
1. General function Description of the RK512 procedure	10
2. Telegram Structures	11
2.1 Structure of the command telegram header	11
2.2 Structure of the reaction telegram.....	11
2.3 Sequence of a FETCH request	12
The fetch job runs in the following order:	12
2.4 Sequence of a Send request.....	13
2.5 Error-messages and error-number.....	14
3. Description of the building blocks and interface parameters	15
3.1. Release code for full range of functions.....	16
3.2. User data blocks.....	17
3.3. Joblist	17
3.4. MESA_RK512_MSTR	18
3.5. MESA_RK512_SLAV	21
4. Device configuration.....	24
4.1. HW-Id CPU.....	24
4.2. HW-ID CM	25
4.3. Parameterization of the serial interface.....	26
4.4. Communication Parameters CPU	26
4.5. Communication Parameters CM	27
5. Annex	29
5.1. Performance Test.....	29
5.2. Required Software and Firmware-Releases	30
5.3. Determining the CPU serial number	30
5.4. Order form for function modules	31

Abbreviations

Abkürzung	Erläuterung
NIC	Network Interface Connector
REATEL	Reaktionstelegramm
CM	Communication Modul
FETCH	RK512 Data Request
SEND	RK512 Data Send
S7 1200/1500	Automation-System Siemens

Table Directory

Table 1 : RK512 Command Telegramheader	11
Table 2 : RK512 Reaction Telegramheader.....	11
Table 3 : RK512 Fetch request	12
Table 4 : RK512 Send request.....	13
Table 5 : RK512 Error-Codes.....	14
Table 6 : RK512_JOB	16
Table 7 : MESA_RK512_MSTR – VAR_INPUT	18
Table 8 : MESA_RK512_MSTR – VAR_IN_OUT	19
Table 9 : MESA_RK512_MSTR – VAR_OUTPUT	19
Table 10 : MESA_RK512_SLAV – VAR_INPUT	21
Table 11 : MESA_RK512_SLAV – VAR_OUTPUT	21
Table 12 : Joblist for Performance-Test.....	29
Table 13 : Required Firmware-Releases	30
Table 14 : Order form for Software	31

Table of Figures

Figure 0-1: RK512 as Firewall.....	8
Figure 2-1: Ablauf Fetchanfrage	12
Figure 2-2: Ablauf Sendetelegramm	13
Figure 3-1: User data blocks without ‚optimize access‘	17
Figure 3-2: Structur of Joblist RK512 (Master)	17
Figure 3-3: Function block MESA_RK512_MSTR	19
Figure 3-4: Processdescription of Fetch-Request MESA_RK512_MSTR	20
Figure 3-5: Function block MESA_RK512_SLAV	22
Figure 3-6: Sequence Fetch-Request MESA_RK512_SLAV	23
Figure 4-1: Hardware CPU with CM.....	24
Figure 4-2: HW_Id_CPU	24
Figure 4-3: HW_ID_CM (characteristics CM)	25
Figure 4-4: HW_Id_CM (system constants)	25
Figure 4-5: Properties of the RS232 interface	26
Figure 4-6: Adjustments for communication performance	26
Figure 4-7: Settings CM1241 (RS232).....	28
Figure 5-1: Serial number of CPU	30

Introduction

The RK512 protocol is an extension of the 3964R protocol and uses it as a base. The RK512 enables data exchange between two communication partners using a point-to-point connection with a serial interface (RS232).

The distance between the communication partners depends on the transmission speed used. In general, distances of about 10-15 meters are bridged. The cable you use must be a null modem cable (RXD and TXD are rotated). The hardware parameters of the serial connection must be set identically for the data exchange on both coupling partners.

The exchange of data is based on a master-slave relationship. The master initiates a transaction and the slave responds. A RK512 transaction consists of a command telegram and a response telegram. The initiator of the transaction controls the data transfer to or from the partner system. It distinguishes between two order types, send (data transfer to the partner system) and fetch (request to partners and data transfer to the initiator).

The user data are sent in data blocks with 128 bytes. Each data block receives a checksum (block check can be deselected optionally). The slave responds to each data block with a reaction telegram. Large amounts of data are divided into individual blocks and merged into the target. The division into followup blocks is automatically made according to the RK512 protocol and is transparent to the user.

Because of the interface protocol and the physical point-to-point connection, a secure separation of networks is guaranteed without additional effort.

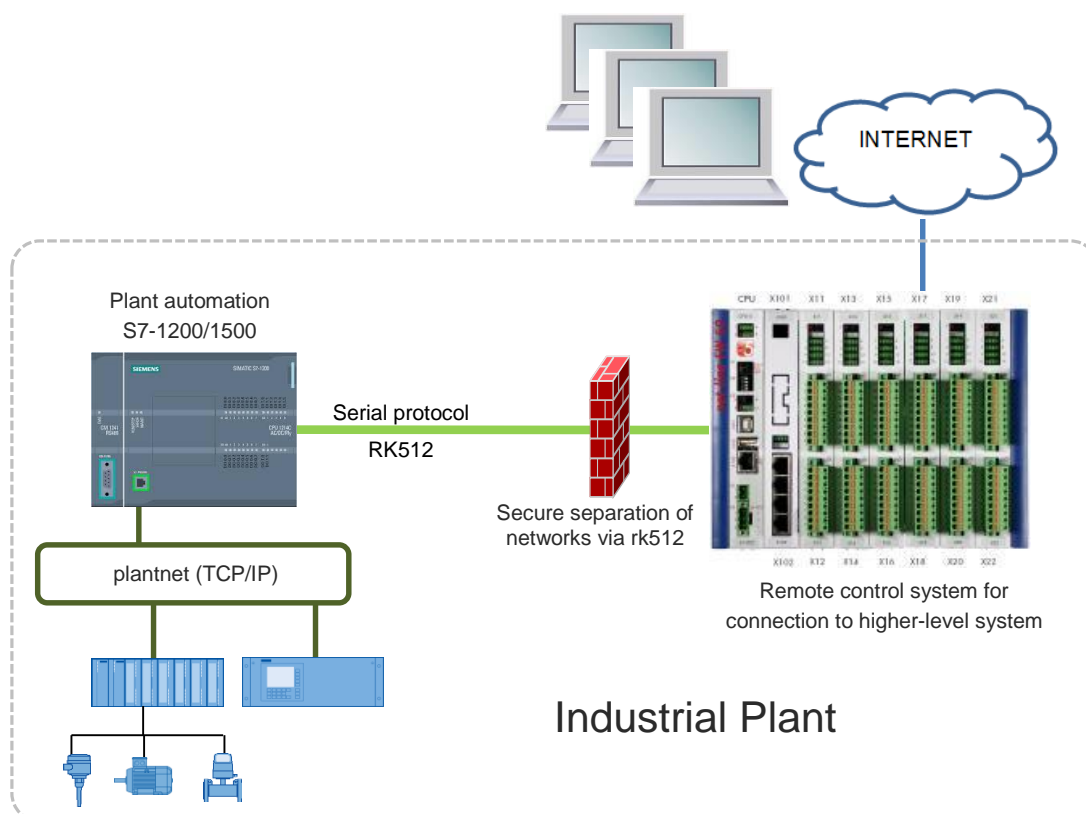


Figure 0-1: RK512 as Firewall

The following limitations are to be considered in the developed building blocks for the described properties of the RK512 protocol:

- Data is exchanged only via data blocks ('d' 44h)
- Data requests support only the functions
 - ,A' (41H) SEND job with Target DB
 - ,E' (45H) FETCH job with source DB.
- Koordinierungsmerker are not supported

Any number of data requests can be parameterized with the master module. The data requests or transmissions are defined in an order list. The transfer function 'Fetch' ('E') or 'Send' ('A') determines the data direction. The data requests or shipments are defined in a job list. The transfer function 'fetch' ('e') or 'send' ('a') determines the data direction.

The number of orders and the data blocks used to record the payload are only limited by the system memory. The individual orders are processed sequentially.

The Block check applies globally to all orders and is optionally adjustable, but must be the same for master and slave.

This documentation describes the function of the interface for the S7-1200 and S7-1500. Because of the different hardware platforms (processor types), the program code for the target platform must be compiled. To identify the different building blocks, the names are given a prefix. The function of the building blocks is identical and therefore the documentation is generally valid for all types. In this documentation, the prefix is generally referred to as 'Mesa' when referring to a building block.

Function Modul Name S7-1200 (Prefix="Bi001_F")

Bi001_F_RK512_MSTR
Bi001_F_RK512_SLAV

Function Modul Name S7-1500 (Prefix="Bi002_F")

Bi002_F_RK512_MSTR
Bi002_F_RK512_SLAV

The communication modules are activated using an release code. Without the release code, the scope of functions is limited. The release code is available from:

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

Please indicate the CPU serial number. The appendix contains an instruction to determine the required serial number and further information about the activation.

1. General function Description of the RK512 procedure

This documentation describes the RK512 protocol only in excerpts and does not claim to be exhaustive. For more in-depth information, refer to the original Siemens documentation.

In the following section you will find out how the data transmission with the computer coupling RK 512 runs in detail and the telegrams are built up.

Data transmission takes place at three levels:

- Hardware-Level
- Procedure-Level
- Telegram-Level

Hardware-Level:

The hardware level includes the physical conditions of the data transfer. These include the interface parameters, such as baud rate, bits, or parity. The hardware parameters must be set at the same time on both communication partners.

Procedure-Level:

All telegrams of the RK512 (command and reaction telegrams) are transmitted by the procedure level of the computer coupling, i.e. sent or received with the procedure 3964 or 3964r.

The procedure packages the telegrams with a starting and final sequence, provides the resulting data blocks with a block test mark (only at the procedure 3964r) and automatically handles telegram repetitions in case of faulty telegram receipt of the receiver.

Telegram-Level:

The telegram level corresponds to the transport layer of the ISO/OSI layer model. The communication master sends a command telegram to the partner. This sends a response telegram, possibly with an error code, back to the communication master.

In contrast to the 3964 (R) procedure, the computer coupling RK512 contains in addition to bit transfer layer and backup layer also the Transport Layer (Telegram level), in which each data transfer is acknowledged by the receiver.

2. Telegram Structures

For reason of clarity, the following telegram structures are limited to telegram data. This data is encapsulated during transmission in a frame, with a start and end ID. An optional block checksum can be added.

2.1 Structure of the command telegram header

The telegram header of a transfer request has a fixed size and always consists of 10 bytes.

Byte	Description *
1	Telegram identification for command telegrams (00h), Telegram identification for followup command telegrams (FFh)
2	Telegram Identification (00h)
3	' A ' (41h)-for send job with target db or ' E ' (45h)-for Fetch job
4	Data to be transferred is from: 'd ' (44h) = Data module
5	Data destination for Send order or source for fetch for example BYTE 5 = DB-No
6	BYTE 6 = DW-No
7	Length high-byte length of the data to be transferred depending on (0dez)
8	Length low-byte type in words (1.. 65535 dez)
9	Without Koordinierungsmerker, always FFH.
10	Bits 0 to 3: without Koordinierungsmerker Bit 4 to 7: Without CPU number (number from 1 to 4) This is FFH.

Table 1 : RK512 Command Telegramheader

(*) With the building blocks described here, a limited range of functions is implemented (see restrictions).

2.2 Structure of the reaction telegram

The response telegram consists of 4 bytes and contains information about the progress of the order. Transmission error describes byte 4.

Byte	Description
1	Telegram identification for reaction telegrams (00h), Telegram identification for followup reaction telegrams (FFh)
2	Telegram Identification (00h)
3	Fixed with 00h
4	Partner Error Number: 00h If no error occurred during transmission. In the event of an error, the error number is here. In the case of a communication using third-party products, the error number of the corresponding documentation can be consulted.

Table 2 : RK512 Reaction Telegramheader

2.3 Sequence of a FETCH request

The fetch job runs in the following order:

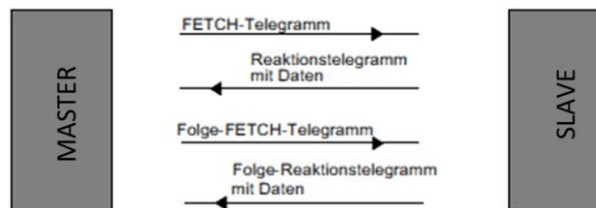


Figure 2-1: Ablauf Fetchanfrage

Active partner	sends a FETCH telegram. This contains the telegram head.
Passive partner	receives the telegram, checks the telegram head and acknowledges with a reaction telegram and the requested user data.
Active partner	receives the reaction telegram with the user data, acknowledges this and evaluates error messages from the reaction telegram
Passive partner	receives the followup FETCH telegram, checks the telegram header and acknowledges with a subsequent response telegram with further data.

Follouup-Telegrams are repeated until all data are transferred.



If an error number is not equal to 0 in the fourth byte of the reaction telegram, no data is stored.

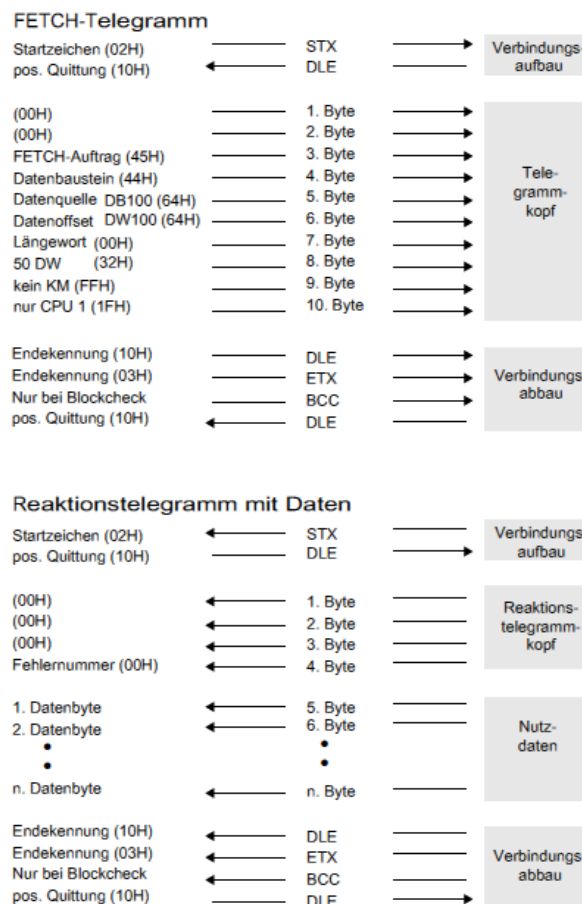


Table 3 : RK512 Fetch request

2.4 Sequence of a Send request

In the following image, the flow of data transmission when sending with a reaction telegram at the computer coupling RK 512 schematically mapped:

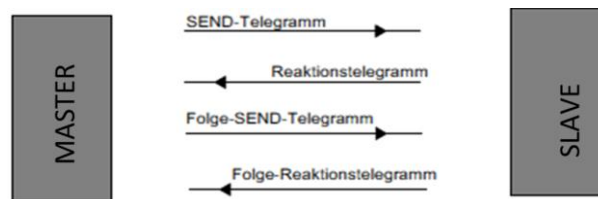


Figure 2-2: Ablauf Sendetelegramm

Active Partner	Master sends a SEND telegram to the slave. This contains the telegram header with the user data.
Passive partner	Slave receives the telegram, checks the telegram head and the data and acknowledges with a reaction telegram.
Active Partner	Master receives the response telegram.
Passive Partner	Receives the followup SEND telegram, checks the telegram header and the data, and acknowledges with a subsequent response telegram.

Followup-Telegrams are repeated until all data are transferred.



If the slave was unable to process the Send telegram without error, it enters an error number in the fourth byte of the reaction telegrams.

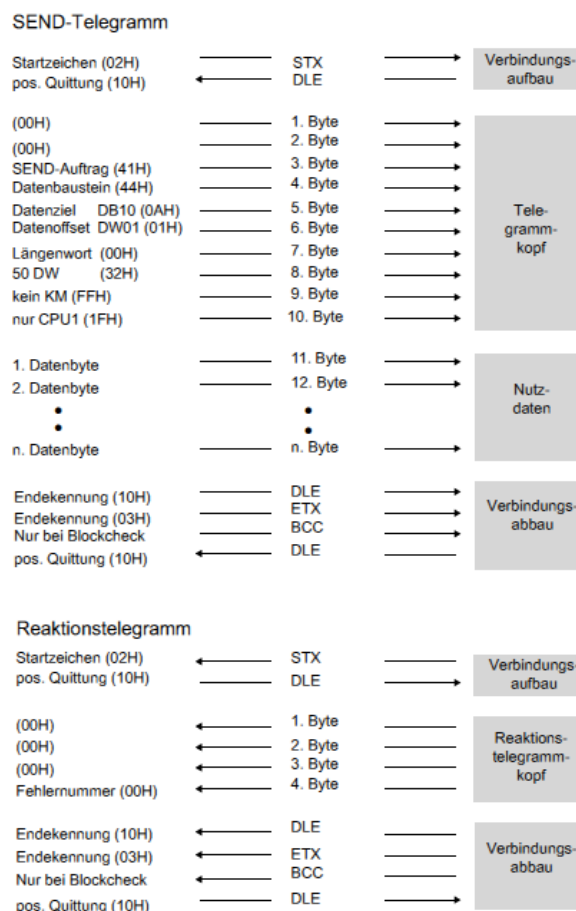


Table 4 : RK512 Send request

2.5 Error-messages and error-number

The error messages refer to the master and slave modules.

The error number describes the internal connection status and belong to the response telegram. The error numbers are not standardized and refer exclusively to the program modules described here.

Error number	description	troubleshooting
01H	Range (Request type, start address, length, data block) not allowed (possibly negative value)	Check parameterization for CPU and CP and possibly correct it. Data block for payload does not exist Requested data length is zero or greater than 64 Job type is not ' e ' or ' a ' The request data length and offset (length + DW_NR) exceed the size of the data block for the payload. (length and DW_NR are specified in Word).
2AH	Error in the construction of the received Reaction Telegram: 1. Bytes not 00h (subsequent telegrams are not supported)	First byte in the log header is not NULL. Malfunction of the partner device if necessary with interface test device Which is turned on in the transmission line.
32H	The wrong range of the ad word or Data range not available (except db/dx) or data range too short (except db/dx)	Check whether the desired data area is present at the partner and is sufficiently large, or check the length and offset (DW_NR)
36H	Partner recognizes incorrect telegram length (total length). The number of data received does not match the number of data in the log header.	Malfunction of the partner device if necessary with interface test device Which is turned on in the transmission line.
40H	Error in the first command byte: is not 00h or FFH	Principled head structure error in the partner. Malfunction of the partner device, if necessary, with an interface test device (FOXPG), which is switched on to the transmission line.
41H	Error in the 3rd command byte: Command letter is not "a" or "0" or "E"	
43H	Error in the 4th command byte: Command letter Incorrect	Principled head structure error in the partner or a command combination was required, which is not allowed. In the command tables, check the commands that are allowed. Malfunction of the partner device may be evidenced by an interface test device that is switched on to the transmission line.
45H	Error in the 5th command byte: DB-No. Invalid (e.g. 00 or data block not present)	Check and change the allowed DB numbers, start addresses, or lengths from the job tables.
46H	Error in the 5th or 6th command byte: Start address too high	
47H	Error in the 7th and 8th command byte: length is not allowed	
E2H	Receipt delay Time (QVZ) Exceeded: After sending STX or a request did not respond from the partner within the timeout period.	The partner device is too slow or not ready for reception or there is a fault of the transmission line. Faulty behaviour of the partner device, if necessary, with an interface test device that is switched on to the transmission line.
F4H	BCC block checkmark error (only for RK 512 with procedure 3964r): The internally formed value of the BCC does not match the BCC received by the partner at the connection end.	Check if the connection is severely disturbed, in this case you will also occasionally see error codes fe. Faulty behaviour of the partner device, if necessary, with an interface test device that is switched on to the transmission line.
F8H	Receive data too large, data cannot be evaluated. Receiving data will be discarded.	Check send job, analyze log

Table 5 : RK512 Error-Codes

3. Description of the building blocks and interface parameters

Standard building blocks and system requirements:

For serial communication, the function modules from the standard library for the Freeport communication are used. These modules are integrated into the RK512 function modules and are called up there. The building blocks for the Freeport communication are supplied with the serial cm of the S7-1200/1500 and must be available on the controller.

- Send_P2P
- Receive_P2P

Driver blocks RK512:

The RK512 driver modules are composed of 2 function modules, the order data structure and the data modules for the payload. The data modules for the payload are to be set in the appropriate size. The RK512 driver modules may only be used once for each serial port. Multiple serial ports are possible. The number of the hardware determines the logical connection of the communication module to the serial interface. The hardware is in the system configuration.

- MESA_RK512_MSTR Driver for RK512 as Master
- MESA_RK512_SLAV Driver for RK512 as Slave

Data structures for initializing the RK512 jobs (master):

A job list manages the data requests and data items that the master should execute. The data in the order list consists of individual orders. The job list and its parameters are stored in a data block. The number of orders is limited only by the maximum size of the data block.

- MESA_RK512_JOB data structure for managing SEND and FETCH communication.

An array summarizes the individual Orders in a job list.

Each job is described by a data structure. The structure sets the connection parameters with the data to be transferred and contains variables for error detection.

The structure requires only the building block ' MESA_RK512_MSTR ', the Communication module ' MESA_RK512_SLAV ' is passive and refers to the data request parameters via the telegram header of the requesting Protocol Master.

For a correct request, the parameters type, length, REM_DB_NR, REM_DW_NR, LOC_DB_NR, LOC_DW_NR must be supplied in the job structure.

The variables done, RK512ERR, SYSTAT serve as return value and error analysis.

The data structure and parameter description of an order:

Variable	Typ	Default value	Description																				
TYP *1	Char	'?'	' E ' = fetch; ' A ' = send. For an unknown character, the order is not processed! A '?' can be deactivated for individual orders. Deactivated jobs do not cause the group error ' QGRPERR ' to be set.																				
LENGTH *1	Int	0 (1 .. 65535)	Number of words to read/write																				
REM_DB_NR *1	Int	0 (1..255)	Remote scope of the data source or destination for send data. This data block must be present on the remote control. The size of the data blocks is the amount of data Length * 2 + REM_DW_NR to store all data in it.																				
REM_DW_NR *1	Int	0 (0..511)	<p>! Offset No. Of the data word in S7 format.</p> <p>Data in data blocks is addressed in S7 byte-by-case, i.e. word addresses at S5 (by multiplying by 2) are converted to byte addresses. The difference between S5 and S7 is to be observed.</p> <table border="1" style="float: right; margin-left: 20px;"> <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	Local data area of the S7-1200/1500 for fetch requests or as destination for send jobs. This data block must be present on the local Control panel. The size of the data block is to be adjusted to the data quantity length * 2 + LOC_DW_NR to store all data in it.																				
LOC_DW_NR *1	Int	0 (1..65535) *2	Offset No. Of the data word on the S7-1200/1500. The difference between S5 and S7 is to be observed.																				
DONE	Bool	false	Output: For a cycle done = ' true ' If job has been successfully processed																				
RK512ERR	Byte	16#0	Output: Error code from the response telegram, see error code.																				
SYSTAT	Byte	16#0	Output: Internal status and error messages, see error code																				

Table 6 : RK512_JOB

(*1) Parameter must be supplied!

(*2) The range information refers to an S7-1200

3.1. Release code for full range of functions

A release code is required to unlock the full range of functions. It created by serial number of the CPU. Without the unlock code, the communication modules work in demo mode with a timeout of 15 min.

The release code is assigned to the serial number of the CPU or the memory card and is only evaluated when the CPU is restarted.

The appendix describes how the serial number is to be determined and the activation is done for the full range of functions.

The modules provided for download on the Internet are exclusively demo versions. You cannot activate these with the unlock code. To obtain a full version please contact the company

MESA Gesellschaft für Elektrotechnik.

3.2. User data blocks

Data blocks are required to store the payload. The variable '... DB_NR ' In the order structure defines the data block number (see above). The size of the data blocks is to be adjusted to the amount of data to be transferred. In the data blocks, you can define an individual structure that has been adapted to the payload.



The ' optimized block access ' must be disabled for the data blocks for storing the payloads.

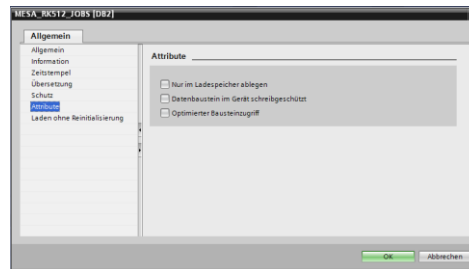


Figure 3-1: User data blocks without ,optimize access'

The parameterization of a non-existent data block results in an error message!

3.3. Joblist

The individual orders with the data requests or shipments are described in a job list and stored in a data module. The order Data module consists of a field with an unlimited number of jobs (' jobs '). The number of jobs is limited only by system memory.



The order Data module is passed as input parameter to the driver module ' MESA_RK512_MSTR '. The ' optimized block access ' must be released for the Order data module.

The order list is processed sequentially by the ' MESA_RK512_MSTR ' as a ring memory. Unused jobs can be flagged in the job list with type = ' ? '. When the job list is processed, these jobs are skipped.

MESA_RK512_JOBS					
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			

Figure 3-2: Structure of Joblist RK512 (Master)

Different data blocks and offset ranges can be specified for the data source and destination. The variables with the prefix "REM_..." always refer to the data of the partner station. The prefix "LOC_..." defines the data range of the RK512 master.

3.4. MESA_RK512_MSTR

The module coordinates the traffic for the RK512 protocol as master. The data requests or data shipments are configured by individual orders in an order data module. The order data must be passed as a call parameter to the 'MESA_RK512_MSTR'.

You can assign any number of jobs to a communication interface. A job is described with the structure 'RK512_JOB'. Further information on the structure 'RK512_JOB' and its parameters can be found in the previous chapter.

The parameters for a job specify the data range, the offset, the length, and the job type (send or receive). Additional variables are available for error evaluation.

The orders are listed in an Order data module. The list consists of a field with the individual orders ('Jobs'). The driver module 'MESA_RK512_MSTR' works sequentially down list of individual orders. The software automatically recognizes the number of orders that have been parameterized. Once the processing has reached the last parameterized order, the software starts again with the processing from the beginning with the order from the first field.

If there are no valid parameters for a job, this job is skipped during processing and continues with the next valid one.

Each order has a parameter for a data block number for storing receive or send data. The data module must be large enough and loaded. If the data block is missing, the job is not processed and an error code is set for this job.

The input 'start' = 'true' starts the processing of the order list with the parameterized data requests/data transmissions. When the input start is set 'start' = 'false' during an active data request, the current request is still fully processed. The withdrawal of the start request corresponds to a pause signal.

Resetting the start to 'start' = 'true' will cause the job list to continue processing. The processing of the orders does not start from the beginning, but with the following order after the stop signal.

VAR_INPUT

Input	Datotyp	Description
Registration_Code	DWORD	Release-code
Registration_Serialtyp	Bool	allocation of the release code: 0 = to CPU, 1 = to Memorycard
Timeout	Time	Maximum time for data responses for an expected response. If the connection is missing, the master sends a 'STX' cyclically and waits for the time 'timeout' for a response from the slave. If the master does not respond within the specified time, the master sends a 'NAK' and starts a new start sequence with a 'STX'. The timeout is re-triggered with each character received.
HW_Identifier_CPU	HW_INTERFACE	"HW-id" of the CPU, can be consulted in the device configuration (see Chapter Device Configuration).
HW_Identifier_CM	PORT	"HW-id" of the communication module is to be read in the device configuration (see Chapter Device Configuration).
BlkChk	Bool	Enable Block check, BLKCHKC = 'true' means with block check calculation
Start	Bool	At start = 'true', data is requested cyclically, at start = 'false' the order processing is terminated after the currently pending job.

Table 7 : MESA_RK512_MSTR – VAR_INPUT

VAR_IN_OUT

Input	Datotyp	Description
Jobs	Array [*] of „RK512_JOB“	Pointer to a field with the orders. An order contains the parameters of the data requests, as well as their error states. It is the pointer Transferred to the data block with the order list.

Table 8 : MESA_RK512_MSTR – VAR_IN_OUT

VAR_OUTPUT

Input	Datotyp	Description
QTmRemain	Time	Time remaining for timeout. A rk512 slave must respond to the request within the timeout period. The passing of time leads to an error and abortion of the current order processing and set the output ' QTmOut '.
QTmOut	Bool	Timeout error, set to True for a cycle if requests are not answered. A ' Nak ' is sent and the processing of the order list continues with the next order. The timeout time is set with the input parameter 'Timeout '.
QGrpErr	Bool	Group error, QGRPERR = ' true ' If an error occurs in any job. The error is reset only after successful processing of all requests. Disabled jobs do not result in a group error (see job structure variable type = ' ? ').
QTxtAction	String[32]	text display for the current processing cycle
QState	WORD	State of transmission communication. Returns the status of the Send_P2P function. The description of the status is the help for the building block ' SendP2P '.
QBusy	Bool	QBusy is ' true ' as long as a request is active, QBusy is temporarily reset between each new request
QJobIdx	Int	Current job index corresponds to the current position in the field of jobs (zero-based). If the input ' start ' is set to ' false ' during order Processing, QJOBIDX displays the last processed job index.
Registration_Code_OK	Bool	Set to ,true' if ,Registration_Code' is correct
DemoMode	Bool	Set to ,true' if ,Demomode'. Kommunikation-Funktion is limited to 15 min.

Table 9 : MESA_RK512_MSTR – VAR_OUTPUT

communication block MESA_RK512_MSTR:

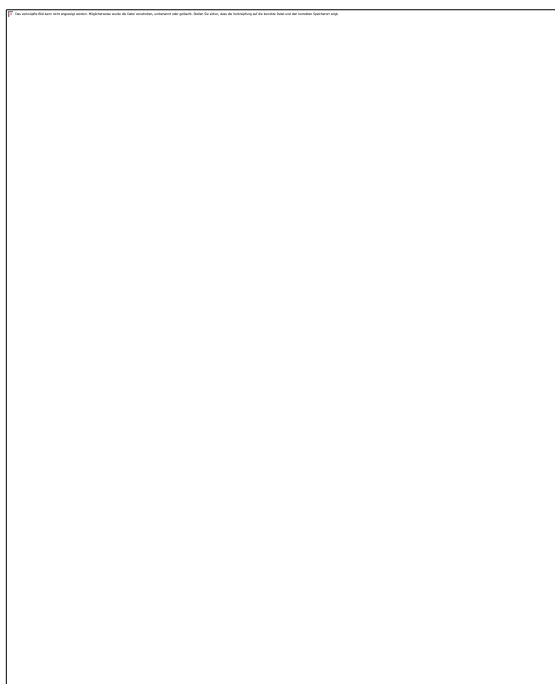


Figure 3-3: Function block MESA_RK512_MSTR

Process description of a FETCH request (master):

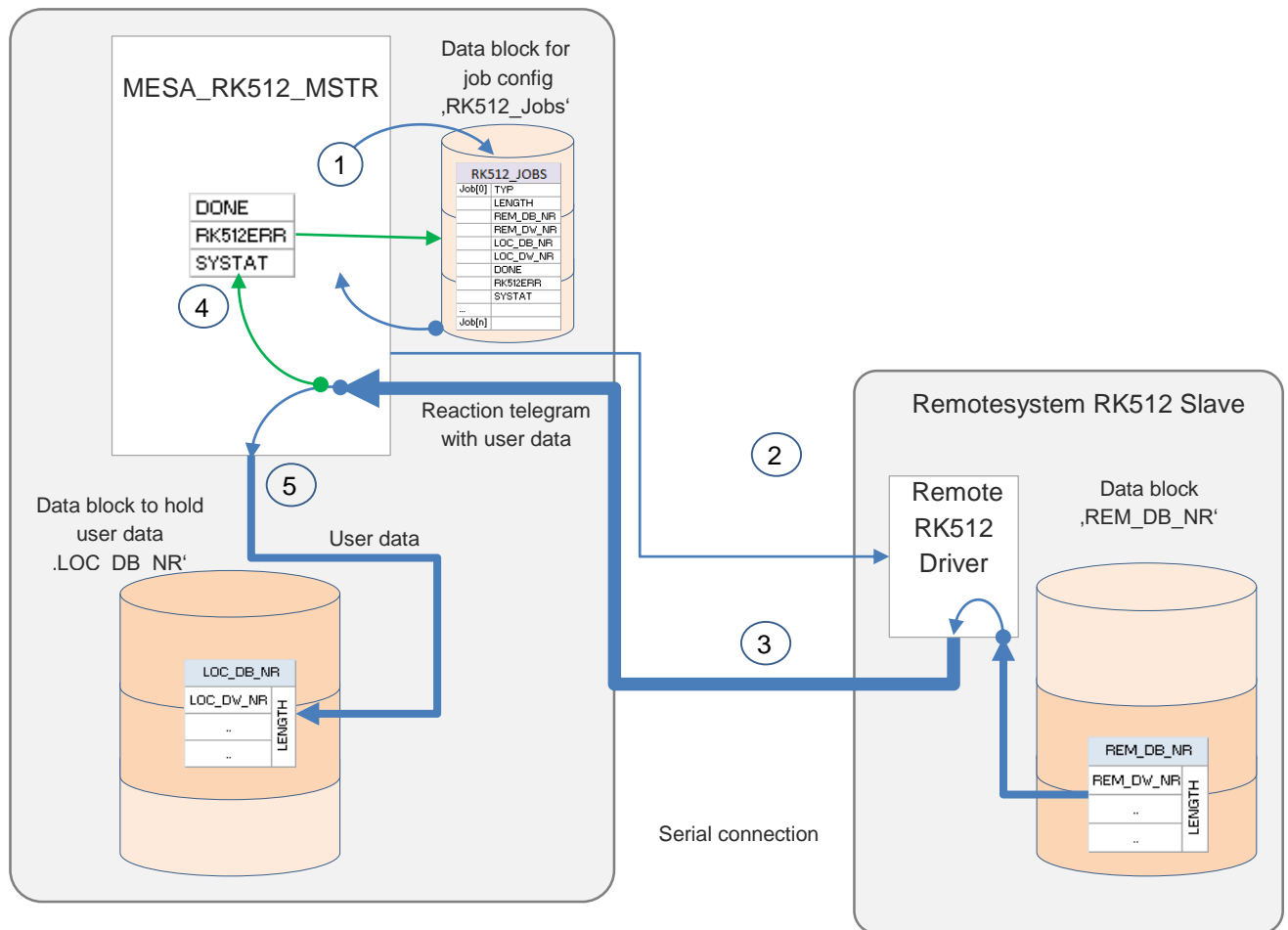


Figure 3-4: Processdescription of Fetch-Request MESA_RK512_MSTR

1. The ' MESA_RK512_MSTR ' polls the order list cyclically and generates a request for valid order data.
2. The request with the address data of the remote module (length, REM_DB_NR, REM_DW_NR) is submitted to the remote partner via a fetch request.
3. If the partner has understood the request, it responds with a response telegram and the requested data.
4. 'Done' is set after a successful transaction or the errors are displayed in the variable ' RK512ERR '/' SYSTAT ' for this job.
5. In the case of a positive response telegram, the ' MESA_RK512_MSTR ' stores the received data in the local range at the address as defined by length, LOC_DB_NR, LOC_DW_NR.

3.5. MESA_RK512_SLAV

The block responds to RK512 protocol requests as a slave. The data requests are analyzed and corresponding errors are reported. The ' MESA_RK512_SLAV ' requires no data configuration, but determines the address data based on the received request from the telegram header.

If the telegram header has been evaluated without errors and the data ranges are present on the CPU, the master is served with the data or it is attempted to save the received data. In any case, the data block from the telegram header must be present.

If the order was syntactically correct, a response telegram is generated. This indicates a corresponding error code if the data request could not be served.

The processing cycle can be stopped via the input ' start '. The current pending order is processed in full.

VAR_INPUT

Input	Datotyp	Description
Registration_Code	DWORD	Release-code
Registration_Serialtyp	Bool	allocation of the release code: 0 = to CPU, 1 = to Memorycard
Timeout	Time	Maximum time for data responses when the slave waits for a response from the master. If exceeded, the current request cycle is terminated and then switched back to hibernation to respond to a new request. The timeout is only active on a request that has already been started.
HW_Kennung_CPU	HW_INTERFACE	"HW-id" of the CPU, can be read from the device configuration (see Chapter Device Configuration).
HW_Kennung_CM	PORT	"HW-id" of the communication module is to be read in the device configuration (see Chapter Device Configuration).
BlkChk	Bool	Enable Block check, BLKCHKC = ' true ' means with Bcc
Start	Bool	At start = ' true ', waits for incoming request. When start = ' false ', the order processing is terminated after the current request.

Table 10 : MESA_RK512_SLAV – VAR_INPUT

VAR_OUTPUT

Input	Datotyp	Description
QTmRemain	Time	remaining time of Timeout
QTmOut	Bool	Time error, set to True for a cycle if a pending query has not been answered in a timely manner. Processing is canceled and waiting for a new request
QGrpErr	Bool	Group error, ' True ' if an error occurs in the request. ' QGRPERR ' will be reset after the error-free processing of the next job.
QTxtAction	String[32]	text display for the current processing cycle
QState	WORD	State of reception communication. Returns the status of the function ' Receive_P2P ' The description of the status is the help for the building block ' Receive_P2P '.
QBusy	Bool	Is true as long as a request is actively in progress. QBusy will be ' false ' while waiting for a request (STX) from the master.
Registration_Code_OK	Bool	Set to ,true' if ,Registration_Code' is correct
DemoMode	Bool	Set to ,true' if ,Demomode'. Kommunikation-Funktion is limited to 15 min.

Table 11 : MESA_RK512_SLAV – VAR_OUTPUT

Network of communication block MESA_RK512_SLAV:

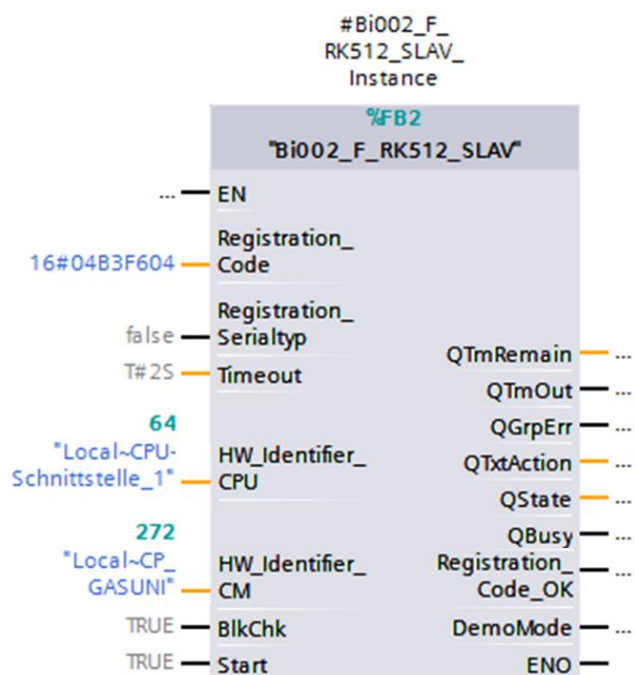


Figure 3-5: Function block MESA_RK512_SLAV

Process description of a FETCH-Request (Slave):

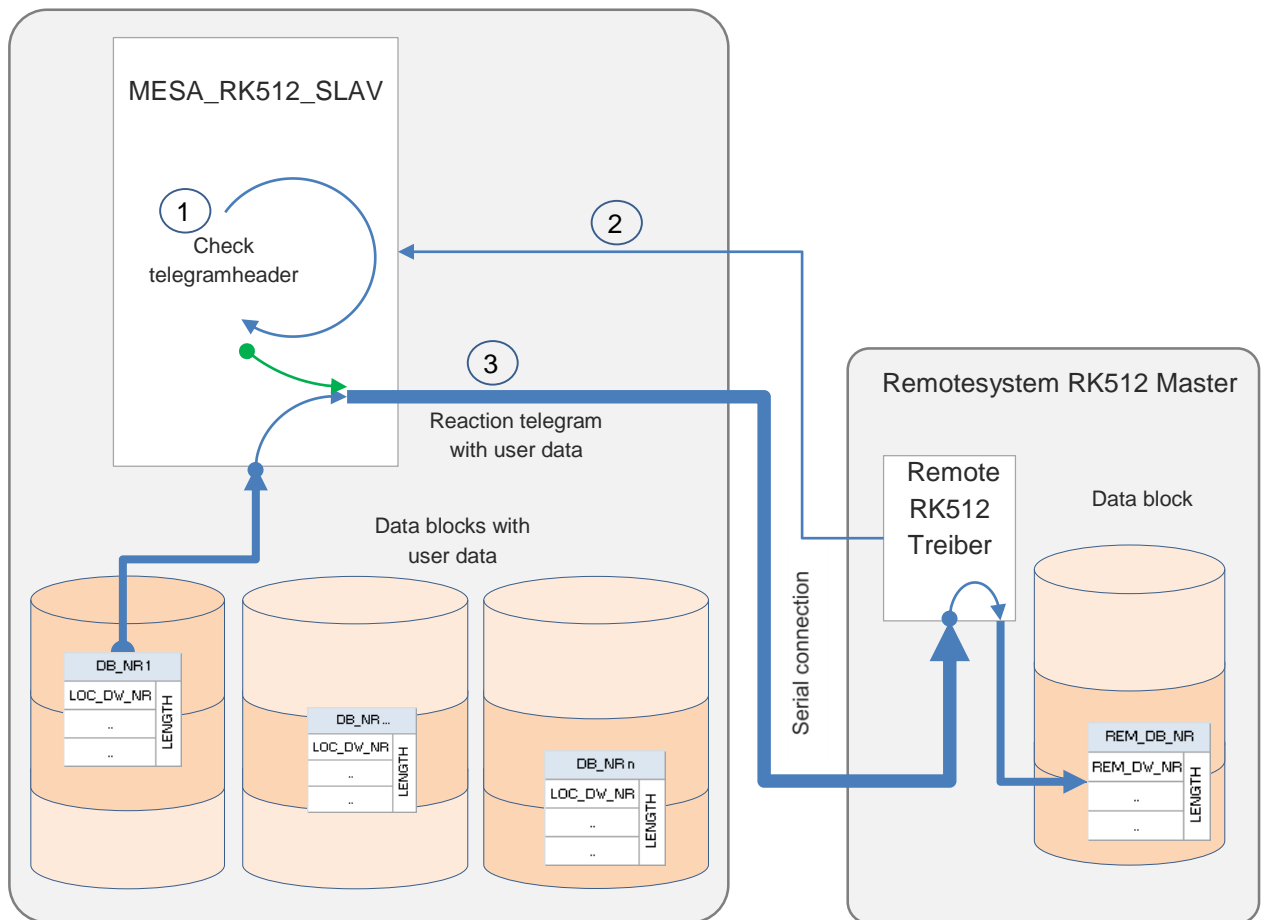


Figure 3-6: Sequence Fetch-Request MESA_RK512_SLAV

1. The ' MESA_RK512_SLAV ' is waiting for incoming data. If the software does not receive data within the timeout cycle, the output ' QTmOut ' is set for a cycle and the receive cycle is restarted.
2. When receiving data, the telegram head is analyzed. Checking if the data block exists and the requested data areas are available.
3. If the requested data is not available because, for example, the data block is not present or the requested data quantity is too large, the slave responds with a reaction telegram and an error code. If the request can be answered, the data from the requested address range (DB_NR, DW_NR, length) is read and appended to the response telegram as payload.

4. Device configuration

Settings on the hardware are required for error-free operation. The settings are made in the device configuration under the properties of the CPU and CM.

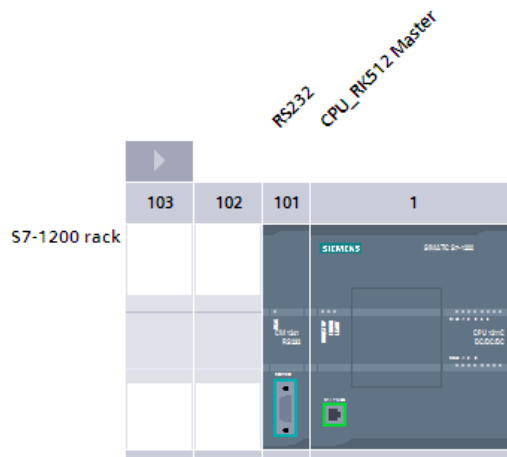


Figure 4-1: Hardware CPU with CM

4.1. HW-Id CPU

The hardware modules of the S7-1200/1500 are addressed by hardware identifiers. The identification must be determined and entered after the migration of the building blocks.

Follow these steps:

- Open the device configuration.
- Establish a connection to the CPU.
- Select the module you want to address.
- In the Inspector window, select tab "Eigenschaften > Systemkonstanten".
- The table contains constants for all the modules used with the required HW-IDs.
- Select the constant for the module you want to address and choose the Copy command from the context menu.
- Insert the constant at the parameter "HW_Kennung_CPU" of the communication module.

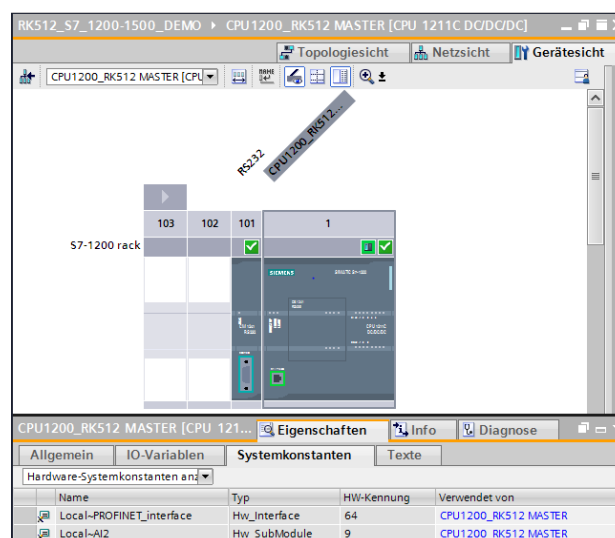


Figure 4-2: HW_Id_CPU

You can drag and drop the HW ID from the variable table to the module parameter "HW_Kennung_CPU".

Advantage: The HW data type is checked by the programming editor and, if necessary, rejected if an invalid HW object has been selected. No check is possible for numerical input.

4.2. HW-ID CM

The hardware identification for the communication module (CM) is to be discarded from the interface card.

Follow these steps:

- Open the device configuration.
- Establish a connection to the CPU.
- Select the CM module you want to address.
- In the Inspector window, select the tab "Allgemein -> HW-Kennung".

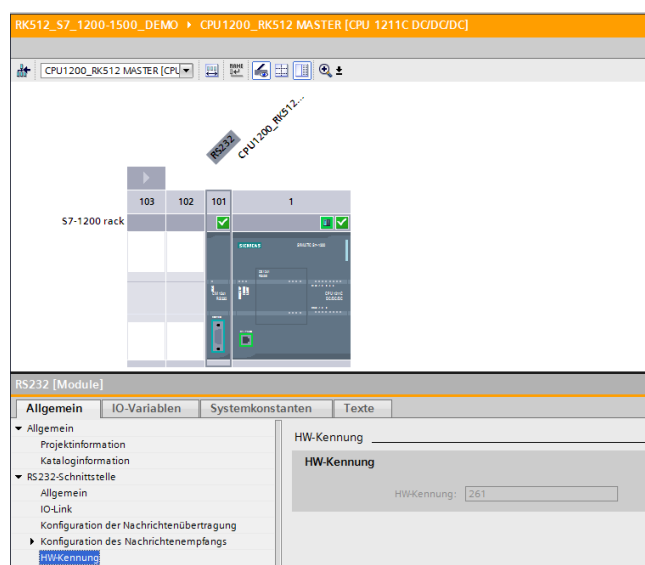


Figure 4-3: HW_ID_CM (characteristics CM)

Alternatively, the HW port can be determined from the system constants. To do this, open the system constants in the project under the entry ' PLC variables '. If you select ' Show all variables ' and the ' System Constants ' tab, you will find an entry for the port of the serial interface. This can be connected with drag & drop to the communication module.

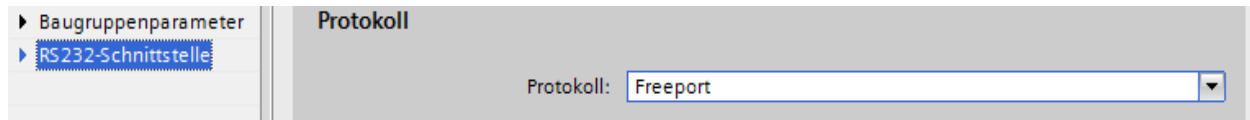
	Name	Datentyp	Wert
1	Local-PROFINET_interface	Hw_Interface	64
2	Local-AI2	Hw_SubModule	9
3	Local-DI14_DQ10_1	Hw_SubModule	257
4	Keiner	Pip	65535
5	Automatische Aktualisierung	Pip	0
6	TPA 1	Pip	1
7	TPA 2	Pip	2
8	TPA 3	Pip	3
9	TPA 4	Pip	4
10	TPA OB Servo	Pip	32768
11	Local-RS232	Port	261

Figure 4-4: HW_Id_CM (system constants)

4.3. Parameterization of the serial interface

The hardware parameters of the serial port must be set identically for both communication partners. The settings are performed in the hardware configuration. The following image shows a configuration example which shows the properties of the RS232 interface under the device configuration.

Communication-Protokoll must be set to 'Freeport'



Freely selectable parameters:

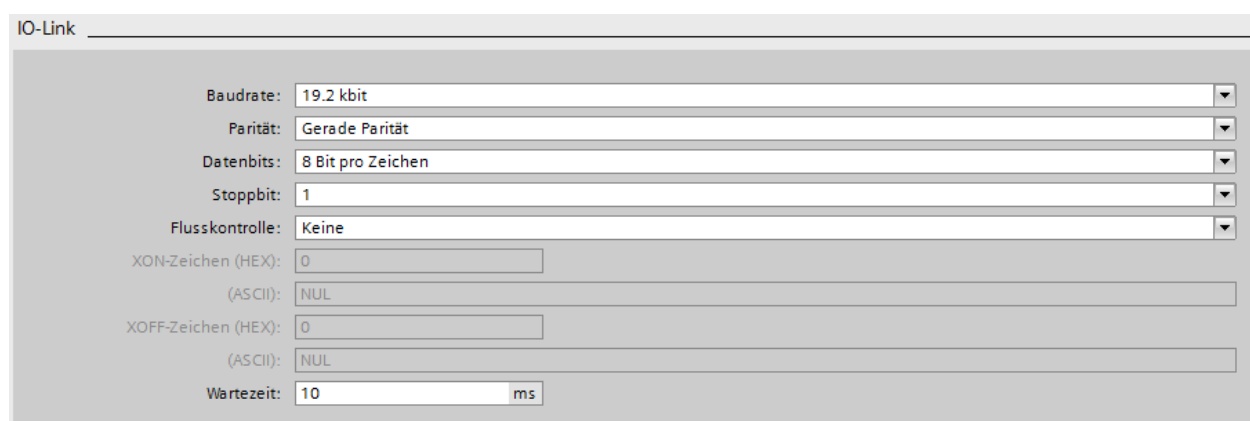


Figure 4-5: Properties of the RS232 interface

In addition to the interface parameters, the timeout times must be parameterized. The communication assemblies for the S7-1200/1500 have to be aware of the limitations of the transmission speed (baud rate). Not all communication assemblies can transmit data with a bitrate greater 19200 bits/sec!

4.4. Communication Parameters CPU

If there is an online connection from the programmer to the CPU parallel to the RK512 communication, the communication load can become so large that the delay times are not adhered to. This results in block repetitions and connection crashes in the RK512 protocol. Assigning additional computing power to the communication can bypass the problem. The allocation is done in the device configuration of the CPU.

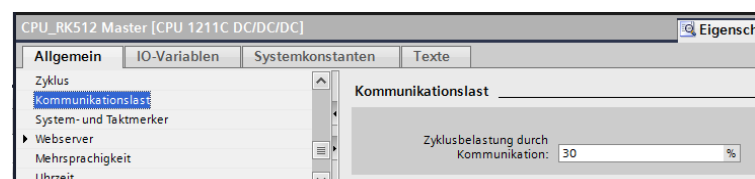


Figure 4-6: Adjustments for communication performance

4.5. Communication Parameters CM

The firmware of the communication modules CM allows the setting of multitude parameters to enable fully automatic processing of simple data protocols with the CM. For the RK512 protocol, these possibilities can only be used conditionally. The following pictures show the necessary parameters for the RK512 operation.

Parameters required by the data protocol:

Konfiguration der Nachrichtenübertragung

Übertragungseinstellungen

RTS ON Verzögerung: ms

RTS OFF Verzögerung: ms

Sende Pause zu Nachrichtenbeginn

Anzahl Bitzeiten in einer Pause: Bitzeiten

Sende Idle Line nach Pause

Idle Line nach Pause: Bitzeiten

Konfiguration des Nachrichtenempfangs

> **Nachrichtenbeginn**

Mit beliebigem Zeichen beginnen

Mit spezieller Bedingung beginnen

Erkenne Nachrichtenbeginn bei Line Break

Erkenne Nachrichtenbeginn bei Idle Line

Dauer von Idle Line: Bitzeiten

Erkenne Nachrichtenbeginn mit einzelmem Zeichen

Nachrichtenbeginn-Zeichen (HEX):

Nachrichtenbeginn-Zeichen (ASCII):

Erkenne Nachrichtenbeginn mit einer Zeichenkette

Anzahl der zu definierenden Zeichenfolgen:

5-Zeichen-Sequenz zum Nachrichtenbeginn

Nachrichtenbeginn Sequenz 1

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):

...

Bedingungen zum Nachrichtende definieren

Erkenne Nachrichtende durch Nachrichten-Zeitüberschreitung

Nachrichten-Zeitüberschreitung: 10 ms

Erkenne Nachrichtende durch Antwort-Zeitüberschreitung

Antwort-Zeitüberschreitung: 10 ms

Erkenne Nachrichtende durch Zeitüberschreitung innerhalb der Zeichen

Zeichenlücken-Zeitüberschreitung: 48 Bitzeiten

Erkenne Nachrichtende durch maximale Länge

Maximale Länge der Nachricht: 1 bytes

Nachrichtende durch feste Nachrichtenlänge erkennen

Feste Nachrichtenlänge: 1 bytes

Lese Nachrichtenlänge aus Nachricht

Offset des Längenfeldes in der Nachricht: 0 bytes

Größe des Längenfeldes: 1 bytes

Das den Daten folgende Längenfeld gehört nicht zur N...: 0 bytes

Erkenne Nachrichtende mit einer Zeichenkette

5-Zeichen-Sequenz zum Nachrichtende

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

Empfangspuffer

Erhaltene Telegramme im Puffer: 20

Überschreiben verhindern

Empfangspuffer im Anlauf löschen

HW-Kennung _____

HW-Kennung

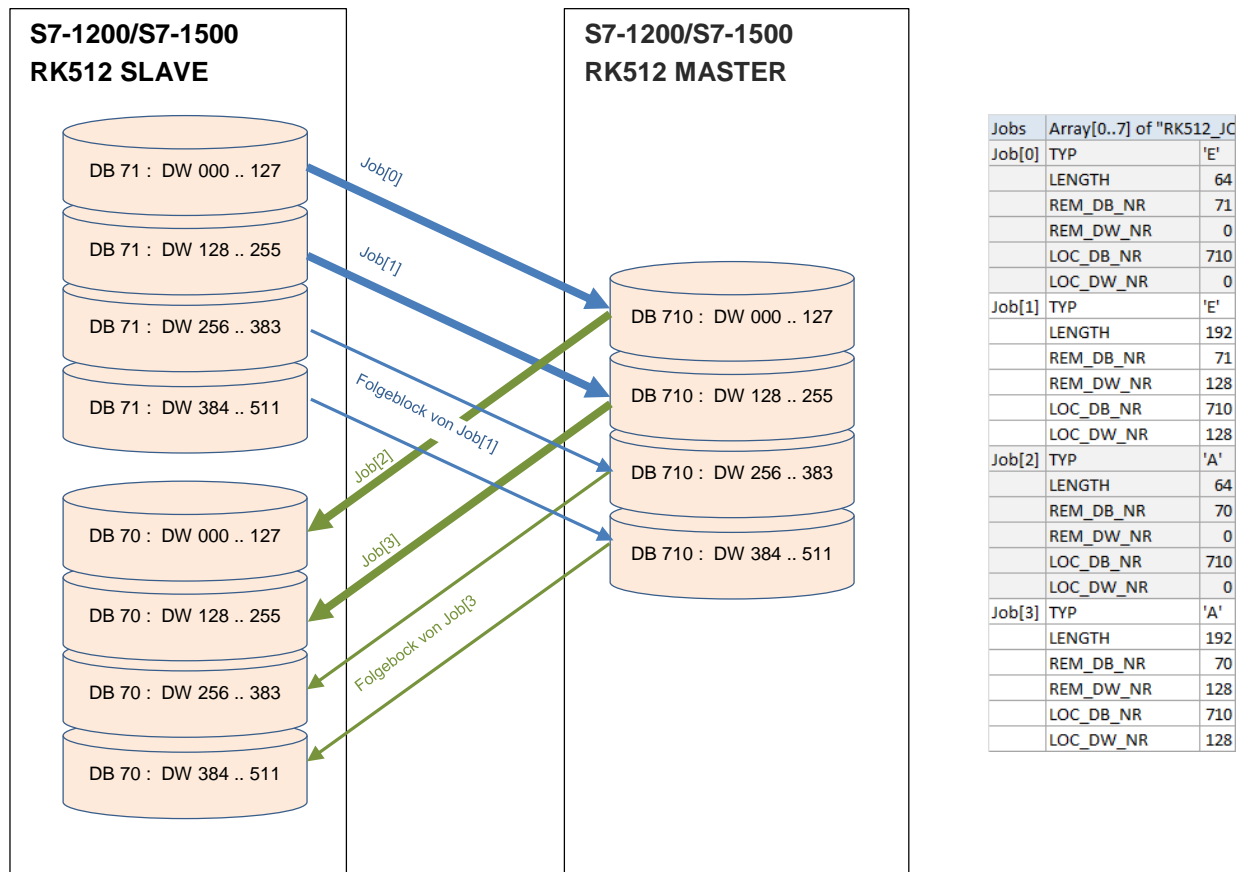
HW-Kennung: 261

Figure 4-7: Settings CM1241 (RS232)

5. Annex

5.1. Performance Test

The performance test was carried out with two S7-1200 (CPU1211C with CM1241). One control was used as master and the other as slave. 4 orders were created on the master, 2 read (FETCH) and 2 write (send).



With the parameterized order list (right) the master 512Bytes from the DB71 to the slave and writes it back to the slave in the DB70. For copying all data, 1,85s is required for an interface setting of 19200, 8E1. The RK512 Functionblocks run on both controls in the Ob1 with block check.

Table 12 : Joblist for Performance-Test

5.2. Required Software and Firmware-Releases

The required software versions are the minimum requirements for the program modules.



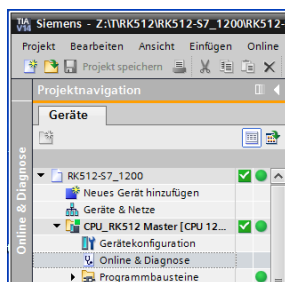
Bezeichnung	Version
TIA	V14 SP1
CPU S7-1200	Firmware V4.2
CM 1241	Firmware V2.1
CPU S7-1500	Firmware V2.0.5
CM PtP RS232 BA	Firmware V1.0

Table 13 : Required Firmware-Releases

5.3. Determining the CPU serial number

The communication modules are protected and can only be activated with an unlock-button.

The Unlock key is formed from the serial number of the CPU. For each CPU on which the communication modules are to run, an unlock-button is required. The key is identical for the communication modules used MESA_RK512_MSTR, MESA_RK512_SLAV.



Without the release key, the scope of functions is limited. The serial number of the CPU is obtained from the hardware configuration. In the TIA projects the function 'online & diagnosis' must be called up for an existing online connection to the CPU. In the tab 'Diagnostics-> General' The serial number is in the block under the manufacturer's information. The fields are readable and write access is not possible.

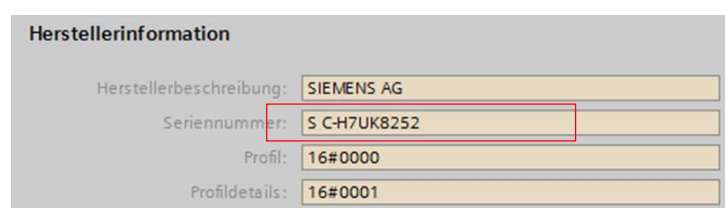


Figure 5-1: Serial number of CPU

To activate the software, the unlock key must be entered in the module input 'unlock code' and the PLC should be restarted.

Note: The release key code must be enclosed with commas (' code ').

5.4. Order form for function modules

Please use the following form to order. All fields except the field for the unlock code are to be filled in. A separate form sheet is to be created for each CPU.


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)	

Table 14 : Order form for Software