



Protocol interface manual

Standard Protocols

ASCII, 3964R, RK512

Modbus RTU, Modbus Plus

Modnet 1/N, Modnet 1/SFB

Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 78
D-65795 Hattersheim
Germany

Tel. +49 (0) 6190 / 9907 - 0
Fax. +49 (0) 6190 / 9907 - 50

Sales: +49 (0) 6190 / 9907 - 0
Hotline and Support: +49 (0) 6190 / 9907 - 99

e-mail: hilscher@hilscher.com
Homepage: <http://www.hilscher.com>

Index	Date	Version	Chapter	Revision
1	02.08.96	V2.100 * V2.000 * V2.010 * V2.100 * V1.113 * V2.000 * V2.000 *	all	ASCII, 3964R, RK512, Modbus RTU, Modbus Plus, Modnet 1/N, Modnet 1/SFB

* The version number refers to the protocol implementation

Although this software has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this software for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this software or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifications at any time in as far as this contributes to technical progress. The version of the manual supplied with the software applies.

1 Introduction	5
2 Conventions	6
3 User Interface	7
3.1 ASCII	7
3.1.1 Protocol Name	7
3.1.2 Protocol Parameter	8
3.1.3 Protocol State	9
3.1.4 Command Interface	9
3.1.4.1 Command Interface in Master Mode	9
3.1.4.1.1 Funktion 2, in Master Mode	10
3.1.4.1.2 Function 3, in Master Mode	11
3.1.4.2 User Interface in Slave Mode	12
3.2 3964R	14
3.2.1 Protocol Name	14
3.2.2 Protocol Parameter	14
3.2.3 Protocol State	15
3.2.4 Command Interface	16
3.2.4.1 Command Interface for Send Data	16
3.2.4.2 Command Interface for Receive Data	17
3.2.4.3 Downwards Compatible Command Interface	18
3.3 RK512	19
3.3.1 Protocol Name	19
3.3.2 Protocol Parameter	19
3.3.3 Protocol State	20
3.3.4 Command Interface	20
3.3.4.1 Command Interface Client	21
3.3.4.2 Command Interface Server	23
3.4 Modbus RTU	25
3.4.1 Protocol Name	25
3.4.2 Protocol Parameter	25
3.4.3 Protocol State	26
3.4.4 Command Interface	27
3.4.4.1 Command Interface in Master Mode	27
3.4.4.2 Command Interface in Slave Mode	29
3.5 Modbus Plus	31
3.5.1 Protocol Name	31
3.5.2 Protocol Parameter	31
3.5.3 Command Interface	32

3.5.3.1 Command Interface for the Master (Client)	32
3.5.3.2 Command Interface for the Slave (Server)	34
3.6 Modnet 1/N	36
3.6.1 Protocol Name	36
3.6.2 Protocol Parameter	36
3.6.3 Protocol State	37
3.6.4 Command Interface	38
3.7 Modnet 1/SFB	39
3.7.1 Protocol Name	39
3.7.2 Protocol Parameter	39
3.7.3 Protocol State	40
3.7.4 Command Interface	41
3.7.4.1 Command Interface for Sending Data	41
3.7.4.2 Command Interface for Receiving Data	43

1 Introduction

This manual describes the user interface for the communication interface and the communication modul for standard protocols. Standard protocols are ASCII, 3964R, RK512, Modbus RTU, Modbus Plus, Modnet 1/N and Modnet 1/SFB.

The aim of this manual is to support the integration of these devices into own applications based on device driver function or direct access to the dual-port memory.

The manual 'Device Driver' and this manual are needed for the integration based on device driver functions.

The manual 'Toolkit: general definitions' and this manual are needed for the integration with direct access to the dual-port memory. The general mechanism for the data transfer are protocol independent and described in the toolkit manual.

2 Conventions

- Protocol identification
Each protocol is identified by its name. This name is located in the variable TaskiName.
The name length is defined with 8 Bytes without a trailing zero. If the name is shorter than 8 Bytes, the name is padded with spaces. The name is represented by ASCII characters.
- Parameter and data representation
All of the parameters and datas are basicly in the representation LSB/MSB (known as Intel format). This corresponds to the Mocrosoft C-Compiler conventions. Sending and receiving of word oriented datas in the representation MSB/LSB (known as Motorola format) is configurable and performed by the protocol task.
- Protocol parameterization
The protocol specific parameters are passed by the dual-port memory to the protocol task. To activate the parameters, a WARMSTART command must be activated on the DEVICE.
- Extended task state
A protocol normaly provides extended task states. These states are located in the dual-port memory and can be read by the user. The description of the state entrys can be found in the protocol manual chapter 'Diagnosis function'.

3 User Interface

3.1 ASCII

Description of the ASCII protocol specific properties and the user interface.

3.1.1 Protocol Name

Protocol name: 'ASCII', located in the variable TasiName

3.1.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter '*Die Parametrierung mit Hilfe des ComPro*' of the manual '*ASCII-Kopplung*'.

The parameters of task 3 and task 4 are only available on the CIF-20 boards and located in the DEVICE mailbox area. So the application has to wait for the WARMSTART acknowledgement, before the first command can be sent to the device.

Parameter	Structure element	Type	Address Task 1	Address Task 2	Address Task 3	Address Task 4
Communication line	bScl	byte	680H	6C0H	400H	440H
RTS control	bRtsControl	byte	681H	6C1H	401H	441H
Baudrate	bBaudrate	byte	682H	6C2H	402H	442H
Data bits	bDataBits	byte	683H	6C3H	403H	443H
Stop bits	bStopBits	byte	684H	6C4H	404H	444H
Parity	bParityBit	byte	685H	6C5H	405H	445H
Mode	bMode	byte	686H	6C6H	406H	446H
End mode	bEndMode	byte	687H	6C7H	407H	447H
Checksum mode	bCheckMode	byte	688H	6C8H	408H	448H
Checksum area	bCheckArea	byte	689H	6C9H	409H	449H
Filter mode	bFilterMode	byte	68AH	6CAH	40AH	44AH
Filter characters	usFilterCharacter	word	68BH	6CBH	40BH	44BH
Telegram timeout	usTelTimeout	word	68DH	6CDH	40DH	44DH
Telegram start timeout	usStartTimeout	word	68FH	6CFH	40FH	44FH
Character timeout	usCharTimeout	word	691H	6D1H	411H	451H
Telegram retry count	bRetry	byte	693H	6D3H	413H	453H
Error LED mode	bErrorLed	byte	694H	6D4H	414H	454H
Telegram start length	bTelStartLen	byte	695H	6D5H	415H	455H
Telegram start	bTelStart[8]	byte	696H	6D6H	416H	456H
Telegram end length	bTelEndLen	byte	69EH	6DEH	41EH	45EH
Telegram end	bTelEnd[8]	byte	69FH	6DFH	41FH	45FH
Length ACK telegram	bTelAckLen	byte	6A7H	6E7H	427H	467H
ACK telegram	bTelAck[8]	byte	6A8H	6E8H	428H	468H
Length NACK telegram	bTelINackLen	byte	6B0H	6F0H	430H	470H
NACK telegram	bTelINack[8]	byte	6B1H	6F1H	431H	471H
Telegram length device	bTelDeviceLen	byte	6B9H	6F9H	439H	479H
Telegram following time	usTelFollowTime	word	6BAH	6FAH	43AH	47AH

Parameter values of the ASCII protocol task

The parameter structure can be found on the device driver disk.

3.1.3 Protocol State

The extended task state is also available in the dual-port memory. The meaning of the parameters are described in the chapter '*Diagnosefunktionen*' of the manual '*ASCII-Kopplung*'.

For task 3 and 4 are no task states in the dual-port memory, of the CIF-20 board, available.

Parameter	Structure element	Type	Address Task 1	Address Task 2
Task state	bTaskState	byte	700H	740H
Transmitt telegram count	ulTxCount	dword	701H	741H
Receive telegram count	ulRxCount	dword	705H	745H
Transmitt error count	usTxErrorCount	word	709H	749H
Receive error count	usRxErrorCount	word	70BH	74BH
Error bits	usErrorBits	word	70DH	74DH
Last error	bError	byte	70FH	74FH

Address table of the extended protocol state

3.1.4 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*'.

3.1.4.1 Command Interface in Master Mode

The master mode is command orientated. Each command consists of a command (send) and a acknowledge (receive) message. Received datas are transferred by the acknowledge message. The command and the acknowledge message are assigned together by the message number.

There are two different send functions. The difference between the functions are the differences between their acknowledge message.

- Function 2: The telegram header of the send command will be given back without changes. The length variable MSG.LN gives the length of user datas in the message.
- Function 3: The telegram header of the acknowledge telegram contains the length of the user datas (MSG.LN) in the telegram.

3.1.4.1.1 Funktion 2, in Master Mode

Variable	Type	Value	Description	
Msg.Rx	byte	1-4	Receiver is the ASCII-Task	Message header
Msg.Tx	byte	16	Transmitter is HOST	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Send command	
Msg.E	byte	0 2	No extension No acknowledgement if no error	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of send data	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	2	Function	
Msg.D(0-239)	byte		Send data	User data

Command message for function 2 in master mode, from, HOST to ASCII task

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the ASCII task	
Msg.Ln	byte	8-248	Message length	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	Answer	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0/2	Extension	
Msg.DeviceAdr	byte	0	unused	Telegram header take up from the command message
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of send data	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	2	Function	
Msg.D(0-239)	byte		Receive data if available	User data

Acknowledge message for function 2, in master mode from the ASCII Task to the HOST

3.1.4.1.2 Function 3, in Master Mode

Variable	Type	Value	Description	
Msg.Rx	byte	1-4	Receiver is the ASCII task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Send command	
Msg.E	byte	0 2	No extension No acknowledgement if no error	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0-240	Number of receive datas*	
Msg.DataCnt	byte	0-240	Number of send datas	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	3	Function	
Msg.D(0-239)	byte		Send data	User data

Command message for function 3, in master mode from HOST to the ASCII task

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the ASCII task	
Msg.Ln	byte	8-248	Message length	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	Answer	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0/2	Extension	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of receive datas	
Msg.DataType	byte	10	data type	
Msg.Function	byte	3	Function	
Msg.D(0-239)	byte		Receive datas if available	User data

Acknowledge message for function 3, in master mode from the ASCII task to the HOST

Note: * The number of receive datas are only tested in end mode 5

3.1.4.2 User Interface in Slave Mode

In slave mode, the user datas are transferd with command messages, in both directions. In send direction, if the application transfers its datas with a command message, it can also receive an acknowledgement for this message.

Variable	Type	Value	Description	
Msg.Rx	byte	1-4	Receiver is the ASCII task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Send command	
Msg.E	byte	0 2	No extension No Answer if no error	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of send data	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	2/3	Function	
Msg.D(0-239)	byte		Send data	User data

Command message in slave mode from the HOST to ASCII task

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the ASCII task	
Msg.Ln	byte	8-248	Message length	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	Answer	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0/2	Extension	
Msg.DeviceAdr	byte	0	unused	Telegram header take up from the send message
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of send data	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	2/3	Function	

Acknowledge message in slave mode from the ASCII task to the HOST

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the ASCII task	
Msg.Ln	byte	8-248	Message length	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	2	No acknowledge message	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	17	Receive command	
Msg.E	byte	0	No extension	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of receive datas	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	2	Function	
Msg.D(0-239)	byte		Receive data	User data

Receive message in slave mode, from the ASCII task to the HOST

Note: An acknowledgement message to the ASCII task is not necessary.

3.2 3964R

Description of the 3964R protocol specific properties and the user interface.

3.2.1 Protocol Name

Protocol name: '3964R', located in the variable TasiName

3.2.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter '*Die Parametrierung mit Hilfe des ComPro*' of the manual '*3964R-Kopplung*'.

The parameters of task 3 and task 4 are only available on the CIF-20 boards and located in the DE-VICE mailbox area. So the application has to wait for the WARM-START acknowledgement, before the first command can be sent to the device.

Parameter	Structure element	Type	Address Task 1	Address Task 2	Address Task 3	Address Task 4
Communication line	bScl	byte	680H	6C0H	400H	440H
RTS control	bRtsControl	byte	681H	6C1H	401H	441H
Baudrate	bBaudrate	byte	682H	6C2H	402H	442H
Data bits	bDataBits	byte	683H	6C3H	403H	443H
Stop bits	bStopBits	byte	684H	6C4H	404H	444H
Parity bits	bParityBit	byte	685H	6C5H	405H	445H
Priority	bPriority	byte	686H	6C6H	406H	446H
Timeout	usTimeout	word	687H	6C7H	407H	447H
Receive mode	bReceiveMode	byte	689H	6C9H	409H	449H
Send mode	bSendMode	byte	68AH	6CAH	40AH	44AH
Error LED mode	bErrorLed	byte	68BH	6CBH	409H	449H

Address table of the parameters in the dual-port memory of the DEVICE

The parameter structure can be found on the device driver disk.

3.2.3 Protocol State

The extended task state is also available in the dual-port memory. The meaning of the parameters are described in the chapter '*Diagnosefunktionen*' of the manual '*3964R-Kopplung*'.

For task 3 and 4 are no task states in the dual-port memory, of the CIF-20 board, available.

Parameter	Structure element	Type	Address Task 1	Address Task 2
Task state	bTaskState	byte	700H	740H
Send telegram count	ulTxCount	dword	701H	741H
Receive telegram count	ulRxCount	dword	705H	745H
Send retry count	bTxRetryCount	byte	709H	749H
Receive retry count	bRxRetryCount	byte	70AH	74AH
Send error count	usTxErrorCount	word	70BH	74BH
Receive error count	usRxErrorCount	word	71DH	74DH
Error bits	usErrorBits	word	71FH	75FH
Last error	bError	byte	721H	761H

Address table of the protocol state in the dual port memory of the DEVICE

3.2.4 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*' of this manual.

3.2.4.1 Command Interface for Send Data

Variable	Type	Value	Description	
Msg.Rx	byte	1-4	Number of the 3964R task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	8-248	Message length in bytes. Counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Send command	
Msg.E	byte	0 2	No extension No answer if error free	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240 0-120	Number of send datas Data type byte Data type word	
Msg.DataType	byte	6	Data types Word Byte	
Msg.Function	byte	2	Function	User data
Msg.D(0-239)	byte		Send data	

Command message in send direction from the HOST to the 3964R task

The telegram header is unchanged and taken up from the command message..

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the 3964R task	
Msg.Ln	byte	8	Message length in bytes	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	Answer	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0/2	Extension	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240	Number of send datas	
Msg.DataType	byte	6/10	Data type	
Msg.Function	byte	2	Function	

Acknowledge message from the 3964R task to the HOST

3.2.4.2 Command Interface for Receive Data

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the 3964R task	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	18	Receive command	
Msg.E	byte	2	No acknowledge message	
Msg.DeviceAdr	byte	0	unused	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	0	unused	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-240 0-120	Number of receive datas Receive mode: Byte telegram Word telegram	
Msg.DataType	byte	6 10	Data types Receive mode: Word telegram Byte telegram	
Msg.Function	byte	2	Function	
Msg.D(0-239)	byte		Receive datas	User data

Receive message from the 3964R task to the HOST

Note: An acknowledge message to the 3964R task is not necessary.

3.2.4.3 Downwards Compatible Command Interface

Older versions of the protocol implementation are not using a telegram header in their messages. For compatibility, these messages are also included.

To use these messages in receive direction, the receive mode must be switched into '*transparent*' mode.

In this mode the user datas at Msg.D(0) are transferred transparent, in both directions. User datas are not swapped.

For new applications, this interface should not be used any more.

Variable	Type	Value	Description	
Msg.Rx	byte	1-4	Receiver is the 3964R task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	0-240	Message length in bytes counted from Msg.D(0)	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	10	Send command	
Msg.E	byte	0 2	No Extension No acknowledgement if no error	
Msg.D(0-239)	byte		Send datas	User data

Downward compatible send message to the 3964R task

To receive this message, the parameter 'Receive mode' must be set to 'transparent'

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the 3964R task	
Msg.Ln	byte	0-240	Message length in bytes, counted from Msg.D(0)	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	10	Receive command	
Msg.E	byte	2	No acknowledge message	
Msg.D(0-239)	byte		Receive datas	User data

Downwards compatible receive message from the 3964R task

3.3 RK512

Description of the 3964R protocol specific properties and the user interface.

3.3.1 Protocol Name

Protocol name: 'RK512', located in the variables Task1Name/Task2Name

The name '3964R', is located in the variable names Task5Name/ Task6.

3.3.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter '*Die Parametrierung mit Hilfe des ComPro*' of the manual '*RK512-Kopplung*'.

Parameter	Structure element	Type	Address Task 1	Address Task 2
Communication line	bScl	byte	680H	6C0H
RTS control	bRtsControl	byte	681H	6C1H
Baudrate	bBaudrate	byte	682H	6C2H
Data bits	bDataBits	byte	683H	6C3H
Stop bits	bStopBits	byte	684H	6C4H
Parity bits	bParityBit	byte	685H	6C5H
Priority	bPriority	byte	686H	6C6H
Timeout	usTimeout	word	687H	6C7H
Receive mode	bReceiveMode	byte	689H	6C9H
Send mode	bSendMode	byte	68AH	6CAH
Error LED mode	bErrorLed	byte	68BH	6CBH

Address table of the parameters in the dual port of the DEVICE

The parameter structure can be found on the device driver disk.

3.3.3 Protocol State

The extended task state is also available in the dual-port memory. The meaning of the parameters are described in the chapter '*Diagnosefunktionen*' of the manual '*RK512-Kopplung*'.

Parameter	Structure element	Type	Address Task 1	Address Task 2
Task state	bTaskState	byte	700H	740H
Send telegram count	ulTxCount	dword	701H	741H
Receive Telegram count	ulRxCount	dword	705H	745H
Send retry count	bTxRetryCount	byte	709H	749H
Receive retry count	bRxRetryCount	byte	70AH	74AH
Send error count	usTxErrorCount	word	70BH	74BH
Receive error count	usRxErrorCount	word	71DH	75DH
Error bits	usErrorBits	word	71FH	75FH
Last error	bError	byte	721H	761H

Address table of the protocol state in the dual-port memory of the DEVICE

3.3.4 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*' of this manual.

The protocol implementation is designed, that only one server and one client command can be active at a time. All further commands are rejected.

3.3.4.1 Command Interface Client

Variable	Typ	Wert	Bedeutung	
Msg.Rx	byte	1-2	Receiver is RK512 task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	16-246	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	18	Command	
Msg.E	byte	0 ***	No extension Sequenz	
Msg.DeviceAdr	byte	0	unused	Extended telegram header
Msg.DataArea	byte	*	RK512 data type	
Msg.DataAdr	word	*	Data word	
Msg.DataIdx	byte	*	Data block number	
Msg.DataCnt	byte	*	Data count ****	
Msg.DataType	byte	0x06 0x86	Word (Motorola format, MSB-LSB) Word (Intel format, LSB-MSB)	
Msg.Function	byte	1 2	FETCH command SEND command	
Msg.Kby	byte	**	Coordination byte	
Msg.Kbi	byte	**	Coordination bits	
Msg.DataCntX	word	*	Data count ****	
Msg.Reserve1	byte	0	unused	
Msg.Reserve2	byte	0	unused	
Msg.Reserve3	byte	0	unused	
Msg.Reserve4	byte	0	unused	
Msg.D(0-229)	byte		Send datas	User data

Command message from the HOST to the RK512 task

- * Valid values can be taken from the table described in the chapter 'RK512 Datentypen' of the manual 'RK512-Kopplung'.
- ** Coordination bits in accordance to the Siemens handbook (a value of 255 means, coordination byte unused).
- *** Described in the chapter 'Definition des Messagekopfes' of this manual.
- **** The amount of Msg.DataCnt and Msg.DataCntX must be added to get the real data amount of the message. It is also possible to use only Msg.DataCnt while Msg.DataCntX is setting to zero.

The extended telegram header must returned unchanged also on error conditions. The header should be taken from the command message.

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-2	Transmitter is the RK512 task	
Msg.Ln	byte	16-246	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	18	Answer	
Msg.F	byte	0 1-255	No Error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0 ***	No extension Sequenz	
Msg.DeviceAdr	byte	0	unused	Extended telegram header
Msg.DataArea	byte	*	RK512 data type	
Msg.DataAdr	word	*	Data word	
Msg.DataIdx	byte	*	Data block number	
Msg.DataCnt	byte	*	Data count ****	
Msg.DataType	byte	0x06 0x86	Word (Motorola format, MSB-LSB) Word (Intel format, LSB-MSB)	
Msg.Function	byte	1 2	FETCH command SEND command	
Msg.Kby	byte	**	Coordination byte	
Msg.Kbi	byte	**	Coordination bits	
Msg.DataCntX	word	*	Data count ****	
Msg.Reserve1	byte	0	unused	
Msg.Reserve2	byte	0	unused	
Msg.Reserve3	byte	0	unused	
Msg.Reserve4	byte	0	unused	
Msg.D(0-229)	byte		Receive datas	User data

Acknowledge message from the RK512 task to the HOST

- * Valid values can be taken from the table described in the chapter 'RK512 Datentypen' of the manual 'RK512-Kopplung'.
- ** Coordination bits in accordance to the Siemens handbook (a value of 255 means, coordination byte unused).
- *** Described in the chapter 'Definition des Messagekopfes' of this manual.
- **** The amount of Msg.DataCnt and Msg.DataCntX must be added to get the real data amount of the message. It is also possible to use only Msg.DataCnt while Msg.DataCntX is setting to zero.

3.3.4.2 Command Interface Server

Variable	Typ	Wert	Bedeutung	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-2	Transmitter is the RK512 task	
Msg.Ln	byte	16-246	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	18	Command	
Msg.E	byte	0 ***	No extension Sequenz	
Msg.DeviceAdr	byte	0	unused	Extended telegram header
Msg.DataArea	byte	*	RK512 data type	
Msg.DataAdr	word	*	Data word	
Msg.DataIdx	byte	*	Data block number	
Msg.DataCnt	byte	*	Data count ****	
Msg.DataType	byte	0x06 0x86	Word (Motorola format, MSB-LSB) Word (Intel format, LSB-MSB)	
Msg.Function	byte	1 2	FETCH command SEND command	
Msg.Kby	byte	**	Coordination byte	
Msg.Kbi	byte	**	Coordination bits	
Msg.DataCntX	word	*	Data count ****	
Msg.Reserve1	byte	0	unused	
Msg.Reserve2	byte	0	unused	
Msg.Reserve3	byte	0	unused	
Msg.Reserve4	byte	0	unused	
Msg.D(0-229)	byte		Receice datas	User data

Command message from the RK512 task to the HOST

- * Valid values can be taken from the table described in the chapter 'RK512 Datentypen' of the manual 'RK512-Kopplung'.
- ** Coordination bits in accordance to the Siemens handbook (a value of 255 means, coordination byte unused).
- *** Described in the chapter 'Definition des Messagekopfes' of this manual.
- **** The amount of Msg.DataCnt and Msg.DataCntX must be added to get the real data amount of the message. It is also possible to use only Msg.DataCnt while Msg.DataCntX is setting to zero.

User datas, in the send commands, are given in the Motorola representation (MSB-LSB)

The extended telegram header must returned unchanged also on error conditions. The header should be taken from the command message.

Variable	Type	Value	Description	
Msg.Rx	byte	1-2	Receiver is the RK512 task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	16-246	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	18	Answer	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0 ***	No extension Sequenz	
Msg.DeviceAdr	byte	0	unused	Extended telegram header
Msg.DataArea	byte	*	RK512 data type	
Msg.DataAdr	word	*	Data word	
Msg.DataIdx	byte	*	Data block number	
Msg.DataCnt	byte	*	Data count ****	
Msg.DataType	byte	0x06 0x86	Word (Motorola format, MSB-LSB) Word (Intel format, LSB-MSB)	
Msg.Function	byte	1 2	FETCH command SEND command	
Msg.Kby	byte	**	Coordination byte	
Msg.Kbi	byte	**	Coordination bits	
Msg.DataCntX	word	*	Data count ****	
Msg.Reserve1	byte	0	unused	
Msg.Reserve2	byte	0	unused	
Msg.Reserve3	byte	0	unused	
Msg.Reserve4	byte	0	unused	
Msg.D(0-229)	byte		Send datas	User data

Acknowledge message from the HOST to the RK512 task

- * Valid values can be taken from the table described in the chapter 'RK512 Datentypen' of the manual 'RK512-Kopplung'.
- ** Coordination bits in accordance to the Siemens handbook (a value of 255 means, coordination byte unused).
- *** Described in the chapter 'Definition des Messagekopfes' of this manual.
- **** The amount of Msg.DataCnt and Msg.DataCntX must be added to get the real data amount of the message. It is also possible to use only Msg.DataCnt while Msg.DataCntX is setting to zero.

User datas, in a FETCH command, must be given in the Motorola representation (MSB-LSB).

3.4 Modbus RTU

Description of the Modbus RTU protocol specific properties and the user interface.

3.4.1 Protocol Name

Protocol name: 'MODBUS ', located in the variable TaskiName

3.4.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter '*Die Parametrierung mit Hilfe des ComPro*' of the manual '*Modbus RTU-Kopplung*'.

The parameters of task 3 and task 4 are only available on the CIF-20 boards and located in the DE-VICE mailbox area. So the application has to wait for the WARM-START acknowledgement, before the first command can be sent to the device.

Parameter	Structure element	Type	Address Task 1	Address Task 2	Address Task 3	Address Task 4
Communication line	bScl	byte	680H	6C0H	400H	440H
RTS control	bRtsControl	byte	681H	6C1H	401H	441H
Baudrate	bBaudrate	byte	682H	6C2H	402H	442H
Stop bits	bStopBts	byte	683H	6C3H	403H	443H
Parity bits	bParityBit	byte	684H	6C4H	404H	444H
Mode	bMode	byte	685H	6C5H	405H	445H
Modbus address	bAddress	byte	686H	6C6H	406H	446H
Timeout	usTimeout	word	687H	6C7H	407H	447H
Telegram retry count	bRetries	byte	689H	6C9H	409H	449H
Error LED mode	bErrorLed	byte	68AH	6CAH	40AH	44AH

Address table of the parameter in the dual port of the DEVICE

The parameter structure can be found on the device driver disk.

3.4.3 Protocol State

The extended task state is also available in the dual-port memory. The meaning of the parameters are described in the chapter '*Diagnosefunktionen*' of the manual '*Modbus RTU-Kopplung*'.

For task 3 and 4 are no task states in the dual-port memory, of the CIF-20 board, available.

Parameter	Structure element	Type	Address Task 1	Address Task 2
Task state	bTaskState	byte	700H	740H
Transmitt telegram count	ulTxCount	dword	701H	741H
Receive telegram count	ulRxCount	dword	705H	745H
Transmitt retry count	bTxRetryCount	byte	709H	749H
Transmitt error count	usErrorCount	word	70AH	74AH
Error bits	usErrorBits	word	70CH	74CH
Last error	bError	byte	70EH	74EH

Address table of the protocol states in the dual-port memory of the DEVICE

3.4.4 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*'.

3.4.4.1 Command Interface in Master Mode

In master mode, the protocol task is waiting for an incoming send message. This message will be converted to a Modbus telegram and transmitted. The acknowledge message of the slave will be given back as an answer message to the sender.

Variable	Type	Value	Description	
Msg.Rx	byte	1-4	Receiver is the Modbus task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Send command	
Msg.E	byte	0 2	Noextension No acknowledgement if no error	
Msg.DeviceAdr	byte	0-247	Slave address	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	
Msg.D(0-239)	byte	**	Datas, only on write functions	User data

Send message in master mode from the HOST to the Modbus task

- * Valid values can be taken from the table described in chapter 2 of the manual '*Modbus RTU-Kopplung*'.
- ** In function code 5 (write single coil), Msg.D[0] is checked for 0. If Msg.D[0] is equal to zero, the coil is set to 0 otherwise the coil is set to 1.

The telegram header must be returned unchanged also on error conditions. The header should be taken from the command message.

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the Modbus task	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	Answer number	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0/2	Extension	Telegram header
Msg.DeviceAdr	byte	0-247	Slave address	
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	User data
Msg.Function	byte	*	Function code	
Msg.D(0-239)	byte		Receive datas only on read functions	

Acknowledge message in master mode from the Modbus task to the HOST

- * Valid values can be taken from the table described in chapter 2 of the manual 'Modbus RTU-Kopplung'.

3.4.4.2 Command Interface in Slave Mode

In slave mode, the Modbus task is waiting for master messages. These messages are transmitted, as receive messages, to the coupling station. After transmission, the task waits for the acknowledge message. The acknowledge message will be converted to a Modbus telegram and transmitted back to the master.

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1-4	Transmitter is the Modbus task	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Command number	
Msg.E	byte	0	No extension	
Msg.DeviceAdr	byte	0-247	Slave address	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	
Msg.D(0-239)	byte		Receive datas only on write functions	User data

Receive message in slave mode from the Modbus task to the HOST

- * Valid values can be taken from the table described in chapter 2 of the manual 'Modbus RTU-Kopplung'.

The message number must correspond to the command message number.

The telegram header from the command message, must be given back unchanged. Also on error conditions.

Variable	Type	Value	Decsription	
Msg.Rx	byte	1-4	Receiver is the Modbus task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	8-248	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	No answer	
Msg.F	byte	0 1-255	No errorr Error number **	
Msg.B	byte	0	No command	
Msg.E	byte	0	No extension	Telegram header
Msg.DeviceAdr	byte	0-247	Slave address	
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	User data
Msg.D(0-239)	byte		Datas, only on read functions	

Acknowledge message in slave mode from the HOST to the Modbus task

Anmerkung:

- * Valid values can be taken from the table described in chapter 2 of the manual '*Modbus RTU-Kopplung*'.
- ** If the coupling station answer with an error message, so the Modbus task creates an 'exception response code 4' for the master. No further error checks are done by the Modbus task.

3.5 Modbus Plus

Description of the Modbus Plus protocol specific properties and the user interface.

3.5.1 Protocol Name

Protocol name: 'MBP_TASK', located in the variable TaskiName

3.5.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter '*Die Parametrierung mit Hilfe des ComPro*' of the manual '*Modbus Plus -Kopplung*'.

Parameter	Structure element	Type	Address Task 1
Node-Adresse	node_adr	Byte	280H
User-Mode	user_mode	Byte	281H
Routing-Mode	Rout_mode	Byte	282H
Timeout MASTER	ma_timeout	Word	283H
Timeout SLAVE	sl_timeout	Word	285H
Fehler-LED	err_led	Byte	287H
User 1	user1	Byte	288H
User 2	user2	Byte	289H
User 3	user3	Byte	28AH
User 4	user4	Byte	28BH
User 5	user5	Byte	28CH
User 6	user6	Byte	28DH
User 7	user7	Byte	28EH
User 8	user8	Byte	28FH

Parameter values of the DEVICE

3.5.3 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*'.

3.5.3.1 Command Interface for the Master (Client)

Master commands are command messages which are transmitted from the HOST to other Modbus Plus devices.

Variable	Type	Value	Description	
Msg.Rx	byte	1	Receiver is the Modbus Plus task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	16-216	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	18	Send command	
Msg.E	byte	0	No extension	
Msg.DeviceAdr	byte	0	unused	Extended telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	
Msg.Routing1	byte	1..64	Routingbyte 1	
Msg.Routing2	byte	0..64	Routingbyte 2	
Msg.Routing3	byte	0..64	Routingbyte 3	
Msg.Routing4	byte	0..64	Routingbyte 4	
Msg.Routing5	byte	0..64	Routingbyte 5	
Msg.Reserved	byte	0	unused	
Msg.Reserved	byte	0	unused	
Msg.Reserved	byte	0	unused	
Msg.D(0-199)	byte	**	Datas, only on write functions	User data

Send message from the HOST to the master of the Modbus Plus task

* Valid values can be taken from the table described in the chapter '*Funktionscode, Adressbereiche und Datenanzahl*' of the manual '*Modbus Plus-Kopplung*'.

** In function code 5 (write single coil), Msg.D[0] is checked for 0. If Msg.D[0] is equal to zero, the coil is set to 0 otherwise the coil is set to 1.

The telegram header from the command message, must be given back unchanged. Also on error conditions.

Variable	Typ	Wert	Bedeutung	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1	Transmitter is the Modbus Plus task	
Msg.Ln	byte	16-216	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	18	Anwer number	
Msg.F	byte	0 1-255	No Error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0	Extension	
Msg.DeviceAdr	byte	0	unused	Erweiterter Telegramm-kopf
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	
Msg.Routing1	byte	1..64	Routingbyte 1	
Msg.Routing2	byte	0..64	Routingbyte 2	
Msg.Routing3	byte	0..64	Routingbyte 3	
Msg.Routing4	byte	0..64	Routingbyte 4	
Msg.Routing5	byte	0..64	Routingbyte 5	
Msg.Reserved	byte	0	unused	
Msg.Reserved	byte	0	unused	
Msg.Reserved	byte	0	unused	
	byte		Receive datas, only on read functions	User data

Acknowledge message from the master of the Modbus Plus task to the HOST

- * Valid values can be taken from the table described in the chapter '*Funktionscode, Adressbereiche und Datenanzahl*' of the manual '*Modbus Plus-Kopplung*'.

3.5.3.2 Command Interface for the Slave (Server)

Slave commands are commands which are received from other Modbus Plus devices. These commands are transferred to the HOST. The Modbus Plus task waits for the acknowledge message from the HOST, converts the answer into a Modbus Plus telegram and gives this answer back to the Modbus Plus device.

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message header
Msg.Tx	byte	1	Transmitter is the Modbus Plus task	
Msg.Ln	byte	8-208	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	1-8	Path number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Command number	
Msg.E	byte	0	No extension	
Msg.DeviceAdr	byte	1..64	Node address of the Modbus Plus transmitter	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	
Msg.D(0-199)	byte		Receive data, only on write functions	User data

Command message in receive direction of the Modbus Plus slave to the HOST

- * Valid values can be taken from the table described in the chapter 'Funktionscode, Adressbereiche und Datenanzahl' of the manual 'Modbus Plus-Kopplung'.

The message number Msg.Nr must correspond to the command message number Msg.Nr.

The telegram header from the command message, must be given back unchanged. Also on error conditions.

Variable	Type	Value	Description	
Msg.Rx	byte	1	Number of the Modbus Plus task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	8-208	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	1-8	Path number	
Msg.A	byte	17	Answer command	
Msg.F	byte	0 1-255	No error Error number **	
Msg.B	byte	0	No command	
Msg.E	byte	0	No extension	Telegramm-kopf
Msg.DeviceAdr	byte	1..64	Node address of the Modbus Plus transmitter	
Msg.DataArea	byte	0	unused	
Msg.DataAdr	word	*	Data address	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	*	Data count	
Msg.DataType	byte	0	unused	
Msg.Function	byte	*	Function code	User data
Msg.D(0-199)	byte		Datas, only on read function	

Acknowledge message to the Modbus Plus slave from the HOST

- * Valid values can be taken from the table described in the chapter '*Funktionscode, Adressbereiche und Datenanzahl*' of the manual '*Modbus Plus-Kopplung*'.
- ** If the acknowledge message is an error message (Msg.F \neq 0), the Modbus Plus task creates an 'exception response code 3004 Hex' for the Modbus Plus transmitter. No further error checks are done by the Modbus Plus task.

3.6 Modnet 1/N

Description of the Modnet 1/N protocol specific properties and the user interface.

3.6.1 Protocol Name

Protocol name: 'MOD1N', located in the variable TaskiName

3.6.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter *'Die Parametrierung mit Hilfe des ComPro'* of the manual *'Modnet 1/N Kopplung'*.

On enumerations of parameter values, e.g. STERN-1-SL and STERN-1-MA, the first parameter value is 0 the second value is 2 and so on.

Parameter	Data type	Value	Address Task 1	Address Task 2
	Byte	1 = SEA1 2 = SEA2	280H	2C0H
Mode	Byte	0 = STERN-1-SL 1 = STERN-1-MA 2 = BUS-1-SL	281H	2C1H
Baudrate	Byte	0 .. 10	282H	2C2H
A-Byte own connection	Byte	0 .. 126	283H	2C3H
A-Byte partner device	Byte	0 .. 126	290H	2B0H
ZKA suppression	Word	10 .. 250 * 5msec	284H	2C4H
Retry count	Byte	1 .. 3	286H	2C6H
Error LED mode	Byte	0 = off 1 = set/clear 2 = only set	288H	2C8H

Parameter address in the dual-port memory

3.6.3 Protocol State

The extended task state is also available in the dual-port memory.

Parameter	Data type	Value	Address Task 1	Address Task 2
Task state	Word	counter	300H	340H
Send telegram counter	Dword	counter	302H	342H
Receive telegram counter	Dword	counter	306H	346H
Telegramm-Fehler	Dword	counter	30AH	34AH
Last error	Dword	value	30CH	34CH
Error bits	Dword	*	30EH	34EH

Address of the state information in the dual-port memory

* Refer to the table at the chapter '*Erweiterter Taskstatus*' of the manual '*Modnet 1/N Kopplung*'

3.6.4 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*' of this manual.

In send direction the command 11 (Msg.b = 11) and in receive direction the command 21 (Msg.b = 21) are defined.

Variable	Type	Value	Description
msg.rx	Byte	16	Receiver is the HOST
msg.tx	Byte	1	Transmitter is the Modnet 1/N task
msg.ln	Byte	8-134	Message length in bytes, counted from Msg.n(0)
msg.nr	Byte	0-255	Message number
msg.a	Byte	0	No answer
msg.f	Byte	0	No error
msg.b	Byte	16	Receive command (0x10 hex)
msg.e	Byte	2	No acknowledge
msg.kopf.device_adr	Byte	0-126	AByte of the partner station
msg.kopf.data_area	Byte	0	Data area always 0 (unused)
msg.kopf.data_adr	Word	0-65535	Logical message number
msg.kopf.data_idx	Byte	0	Index always 0 (unused)
msg.kopf.data_cnt	Byte	1-126	Data count in bytes
msg.kopf.data_type	Byte	10	Byte string
msg.kopf.function	Byte	2	Write command
msg.n(0-126)	Byte		User data

Command message in receive direction from the Modnet 1/N task

Variable	Type	Value	Description
msg.rx	Byte	1	Receiver is the Modnet 1/N task
msg.tx	Byte	16	Transmitter is the HOST
msg.ln	Byte	8-134	Message length in bytes, counted from Msg.n(0)
msg.nr	Byte	0-255	Message number
msg.a	Byte	0	No answer
msg.f	Byte	0	No error
msg.b	Byte	17	Send command (0x11, hex)
msg.e	Byte	0	Acknowledgment activ
	Byte	2	No error message
msg.kopf.device_adr	Byte	0-126	AByte of the partner station
msg.kopf.data_area	Byte	0	Data area always 0 (unused)
msg.kopf.data_adr	Word	0-65535	Logical message number
msg.kopf.data_idx	Byte	0	Index always 0 (unused)
msg.kopf.data_cnt	Byte	1-126	Data count in bytes
msg.kopf.data_type	Byte	10	Byte string
msg.kopf.function	Byte	2	Write command
msg.n(0-126)	Byte		User data, if available

Command message in send direction from the Modnet 1/N task

3.7 Modnet 1/SFB

Description of the Modnet 1/SFB protocol specific properties and the user interface.

3.7.1 Protocol Name

Protocol name: 'MOD1SFB', located in the variable TaskiName

3.7.2 Protocol Parameter

Bellow, you can find the various parameters and their offset addresses in the dual-port memory. The meaning of the parameters are described in the chapter '*Die Parametrierung mit Hilfe des ComPro*' of the manual '*Modnet 1/SFB-Kopplung*'.

The parameters existing only once in the dual-port memory, because the protocol can only run one time on the device.

Parameter	Structure element	Type	Address Task 1	Address Task 2	Adresse Task 3	Address Task 4
Communication line	bScl	byte	680H	-	-	-
Mode	bMode	byte	681H	-	-	-
Baudrate	bBaudrate	byte	682H	-	-	-
Bus address	bBusAdress	byte	683H	-	-	-
Cycle time	usCyletime	word	684H	-	-	-
Retry count	bRetrey	byte	686H	-	-	-
Timeout	usSlaveTimeout	word	687H	-	-	-
Error LED mode	bErrorLed	byte	689H	-	-	-
reserved	bReserve	byte[5]	68AH	-	-	-
Slave count	bSlaveCount	byte	68FH	-	-	-
Slave address	bSlaveAdress	byte[32]	670H	-	-	-

Parameter address table in the dual-port memory of the DEVICE

3.7.3 Protocol State

The extended task state is also available in the dual-port memory. The meaning of the parameters are described in the chapter '*Diagnosefunktionen*' of the manual '*Modnet 1/SFB-Kopplung*'.

Parameter	Structure element	Type	Address Task 1	Address Task 2
Task state	bTaskState	byte	700H	-
Transmitt telegram count	ulTxCount	dword	701H	-
Receive telegram count	ulRxCount	dword	705H	-
Telegram error count	usErrorCount	word	709H	-
Error bits	usErrorBits	word	70AH	-
Slave watchdog	bWatchtdog	byte	70CH	-
reserved	bReserve	byte[3]	70DH	-
Slave 1 state	bSlaveState	byte[32]	710H	-

Address table of the protocol state in the dual-port memory of the DEVICE

3.7.4 Command Interface

The send and receive telegrams are described in the chapter '*Definition of the message interface*' of this manual.

3.7.4.1 Command Interface for Sending Data

Variable	Type	Value	Description	
Msg.Rx	byte	1	Receiver is the Modnet 1/SFB task	Message header
Msg.Tx	byte	16	Transmitter is the HOST	
Msg.Ln	byte	9-134	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Send command	
Msg.E	byte	0 2	No extension No acknowledgement, if no error	
Msg.DeviceAdr	byte	1..126	Bus address (A-Byte)	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataIAdr	word	256 .. 65535	Logical message number (LNN)	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	1-126	Send data count if data type is byte	
Msg.DataType	byte	10	Data type byte	
Msg.Function	byte	2	Function send	
Msg.D(0-126)	byte		Send datas	User data

Command message in send direction from the HOST to the Modnet 1/SFB task

The telegram header from the command message, must be given back unchanged. Also on error conditions.

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the Host	Message header
Msg.Tx	byte	1	Transmitter is the Modnet 1/SFB task	
Msg.Ln	byte	8	Message length in bytes	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	17	Answer telegram	
Msg.F	byte	0 1-255	No error Error number	
Msg.B	byte	0	No command	
Msg.E	byte	0/2	Extension	
Msg.DeviceAdr	byte	1..126	Bus address (A-Byte)	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataIAdr	word	256 .. 65535	Logical message number (LNN)	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	0-126	Send data count	
Msg.DataType	byte	10	Data type	
Msg.Function	byte	2	Function	

Acknowledge message from the Modnet 1/SFB task to the HOST

3.7.4.2 Command Interface for Receiving Data

Variable	Type	Value	Description	
Msg.Rx	byte	16	Receiver is the HOST	Message-kopf
Msg.Tx	byte	1	Transmitter is the Modnet 1/SFB task	
Msg.Ln	byte	9-134	Message length in bytes, counted from Msg.DeviceAdr	
Msg.Nr	byte	0-255	Message number	
Msg.A	byte	0	No answer	
Msg.F	byte	0	No error	
Msg.B	byte	17	Receive command	
Msg.E	byte	2	No acknowledge message	
Msg.DeviceAdr	byte	1..126	Bus address (A-Byte)	Telegram header
Msg.DataArea	byte	0	unused	
Msg.DataIAdr	word	256 .. 65535	Logical message number (LNN)	
Msg.DataIdx	byte	0	unused	
Msg.DataCnt	byte	1-126	Receive data count Receive mode = Byte telegram	
Msg.DataType	byte	10	Data type Receive mode = Byte telegram	
Msg.Function	byte	2	Function	
Msg.D(0-239)	byte		Receive datas	User data

Receive message from the Modnet 1/SFB task to the HOST

Note: No acknowledgment to the Modnet 1/SFB task is necessary.