Protocol manual

Modbus Protocol

RTU format

(also J-Bus)

Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 15
D-65795 Hattersheim
Germany

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

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

Sales email: sales@hilscher.com
Hotline and Support email: hotline@hilscher.com

Homepage: http://www.hilscher.com

Index	Date	Version	Chapter	Revision
5	10.04.96 09.11.96 05.03.98	 CIF 10: V2.130 CIF 30: V2.200		
6	06.10.98	all	4	max baudrate is 19200 baud
7	04.10.99	V2.150	all	Expansion of function codes 1, 2 and 15 to a maximum data count of 1.920 bits. Add init error 55 Adaptions to PD file (parameter, error text, task state). Setting of stopbit is hardware dependend

Although this protocol implementation has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this protocol implementation 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 protocol implementation 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 protocol implementation applies.

1 Introduction

This description documents the implementation of the Modbus protocol in the RTU format on our devices. It must be noted, however, that the range of functions described may be limited depending on the device or utilization case.This is especially the case in the protocol conversions where a lesser functions range is usually used!

The Modbus protocol is used by the 984 control family of the AEG company. In this, it serves for the communication with the programming device as also among the controls. Depending on the selected physical driver interface it can be utlized as a Point-to Point connection or as Bus operation.

The Modbus protocol prefers an address range between 40001 and 49999 or for markers from 1 to 9999. These ranges can be extended from 0 to 65535 by means of the configuration. This corresponds to the J Bus which otherwise has the same protocol procedure.

2 Specification and Protocol Fundamentals

The implementation is carried out according to the manual:

Gould Modbus Protocol - Reference Guide
PI-MBUS-300 Rev B

The Master as well as the Slave operation were implemented. Thus, all the function codes that are necessary for the reading and writing of registers and markershave been realized. The table in the chapter "Telegram types and function codes" provides a listing of the imlpemented function codes.

2.1 Telegram Format

The Modbus telegram consists of a telegram header and a user data portion. The telegram is closed with a checksum.
The telegram header differs according to the function code. It comprises the Slave address, the function code, the starting address of the user data and the count of the user data .
If a 0 is given as the Slave address, then it is a Broadcast telegram.This is accepted and processed by every Slave.

2.2 Telegram Types and Function Codes

An individual telegram type is generated on the transfer distance for each of the function codes. In addition, so-called "Exception response codes" can occur in the case of error.

2.2.1 Brief Description

The following table provides a listing of the telegram types on the basis of their function codes:

Function code	Description
1 Read Coil Status	With this function, sequential Coils (bits, markers) are read out from the controls.
2 Read Input Status	With this function sequential discrete inputs (Coil, bit, markers) are read out.
3 Read Holding Registers	With this function, sequential output registers (Coils, marker words) are read out.
4 Read Input Registers	With this function, sequential input registers (Words, marker words) are read out.
5 Force Single Coil	With this function, a Coil (bit, marker) is placed or deleted. The altered Coil is read back for comparison.
6 Preset Single Register	With this function, a register (Word, marker word) is written. The register is read back for comparison.
7 Read Exception Status	With this function, 8 fixedly defined coils (bits, markers) are ead out from the controls. The address and the meaning of the Coils depends on the particular controls.
8 Loopback Test	With this function code, diagnostic values are read out. For further important details: see below!
15 Force Multiple Coils	With this function, sequential Coils (bits, markers) are written.
16 Preset Multiple Register	With this function, sequential Registers (Words, marker words), are written.

Brief description of the function codes

Further Explanation of the Function Code 8:

- Diagnostic values are read out with this function code. The diagnostic value is called up as diagnostic code in the address. The code 0 is defined as the actual "Loopback test". In this, the output data must be returned unaltered by the Slave that has been accessed. By measn of comparison with the output value, it can be determined whether the transfer distance works correctly.The comparison must be carried out in the user program..
  A code not equal to 0 addresses a diagnostic value in the accessed Slave and reads it out.
  Only the "Loopback test" is supported in the Slave mode. This is carried out independently without the user Ppogram being being asked. Every other diagnostic code is rejected with an "exception 7".

2.2.2 Function Code, Address Ranges and Data Count

Information: The given address range can be limited deppending on the type of device, e.g., on the PLC 984-685 only the range 40001..41000 is possible instead of to 49999!

Given here is the maximum address range as specified by the protocol.

Function code	Data type	max. Data count.	Address region***** Mode =			Broad-cast
			0/1	2/3	4/5	
1 - read output status	Marker	1..255 / 256, 264, 272, .. , 1.912, 1.920*	1-9999	1-65535	0-65535	no
2 - read input status	Marker	1..255 / 256, 264, 272, .. , 1.912, 1.920*	10001-19999	1-65535	0-65535	no
3 - read output registers	Register	1..100	40001-49999	1-65535	0-65535	no
4 - read input registers	Register	1..100	30001-39999	1-65535	0-65535	no
5 - force single coil	Marker	1**	1-9999	1-65535	0-65535	yes
6 - preset single registers	Register	1**	40001-49999	1-65535	0-65535	yes
7 - read exception status	Marker	8**	***	***	***	no
8 - loopback test	Register	1**	****	****	****	no
15 - force multiple coils	Marker	1..255 / 256, 264, 272, .. , 1.912, 1.920*	1-9999	1-65535	0-65535	yes
16 - preset multiple registers	Register	1..100	40001-49999	1-65535	0-65535	yes

Overview function codes, address range and data quantity

Remarks:

The data count gives the number of markers or registers in the telegram. It does not give the telegram length in Bytes or Words!
The depiction of the Registers and the parameters data count and address are determined by the configurationof the Modbus protocol.

Broadcast telegrams are output with the Slave address 0 and are valid simultaneously for all connected Slaves.

* From a data count of 256 bits, the data count must be a multiple of **eight**.

** This data count is an implicit result of the function code and is not checked.

*** This address is an implicit result of the respective set Modbus control and is not checked.

**** This address is used for calling up the diagnostic function .
The permissible diagnostic functions depend in the Master mode on the accessed Slave.
Only the Diagnostic function 0 is supported in the Slave mode.

***** The addressrange is determined by the "Mode" parameter in the configuration. See further in chapter 4.

2.2.3 Exception response codes

These are answer telegrams from the Slave that, although having received a telegram without error, has recognized an error in the content. This can occur, for instance, when access is made to a data range that is not available in the Slave.

The protocol implementation generates the following "Exception response modes" in the Slave mode:

01 - illegal function	The function code of the received Modbus telegram is not supported.
04 - failure associated device	None, or incorrect, answer received from the coupling partner to a modbus telegram.
06 - busy, rejected message	Application is not ready to receive. Data are not forwarded.
07 - negative acknowledge	Content error recognized in the received Modbus telegram.

2.3 Telegram Security

The telegrams are all secured by means of a CRC-16 polynomial. This checksum is tranmitted with the telegram and ensures a Hamming distance of four.

2.4 Protocol Procedure

A data transfer always consists of a task telegram from the Master to the Slave and the Slave's answering telegram. The written data is is given directly into the task telegram, whilst, when being read, the user data are contained in the answering telegram. .

The initiative for a transfer on the data line always comes from the Master. This means that the Slave may answer only when the Master has sent a telegram to a Slave and the Slave has received it without error.

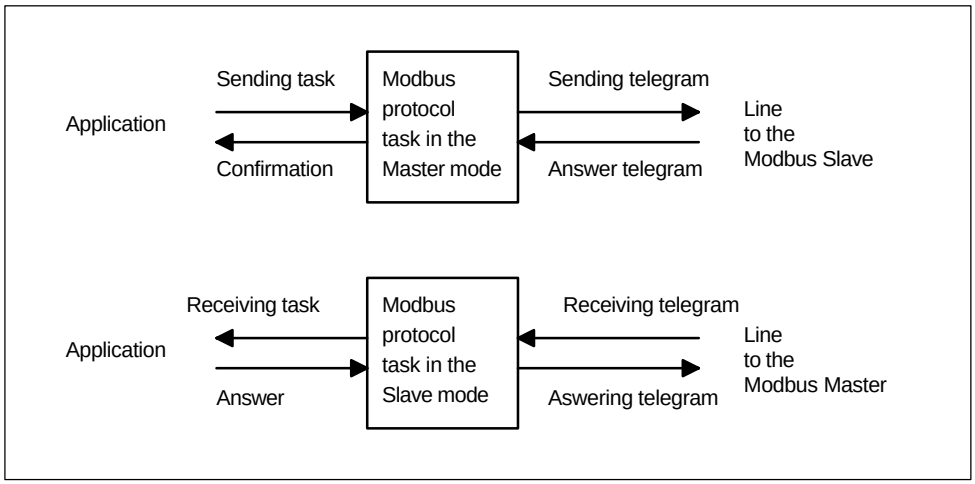
The Slave must confirm every telegram. The only exception is for Broadcast telegrams. These are not confirmed in principle. The absence of a confirmation is used for the display of a data transfer error. GOULD specifies no maximum waiting time for a confirmation. This depends on the respective controls and their loading.

The protocol implementation in the Slave and in the Master mode carries out a time monitoring whose duration is parametrizised. Besides the task and answering telegrams, no further data pass over the line.

2.5 Time Behavior

The telegram data are issued continuously. An interruption of the telegram data of more than the transmission time of three-and-a-half characters is recognized as the end of the telegram.

In the Master operation of the protocol converter, only the reaction time of the



Slaves is monitored. It starts after the last character has been output to the Slave. If, after the parametrizised Timeout, there is noanswer from the Slave, the transmission is aborted with an error message.

In Slave operation, the received telegrams are forwarded for processing to the coupling partner. Within the parametrizesed Timeout period, the data or a confirmation must have arrived from it. If this is not the case, the demand for a telegram from the Master is answered with the "Exception response code" 04 .
Later arriving answers from the coupling partner are rejected.

2.6 Receiving Preparedness of the Application

In slave mode the Modbus RTU checks, if the received telegram can be processed by the application. There are two causes for this:

- The application is not ready or it has not been started. Examples of this is the placing of the "NotRdy" bits on the communication interface or the PLC program is not running at the KPO 104.
- There are no more storage segments free to buffer the telegram.

In both cases the device waits for half the confirmation time and tests the readiness to receive once again. If this is not given, then the connection build-up is refused with the exception code 6.

2.7 Device-Internal Transfer Structure

Within the device, the data are transferred as Tasks and Confirmations in socalled Messages between the application and the Modbus RTU protocol task. Depending on the device, this can be via a special-function module, e.g. at the KPO 104 or by the call-up of a driver function at the communication interface. For all devices, the message structure is the same and fixedly defined.

Data interface of the Modbus RTU protocol task

Variable	Type	Meaning	
Msg.Rx	Byte	Number of the sender	Message header
Msg.Tx	Byte	Number of the recipient	
Msg.Ln	Byte	Length of the message	
Msg.Nr	Byte	Running number of the message	
Msg.A	Byte	Number of the answer	
Msg.F	Byte	Error number of answer	
Msg.B	Byte	Number of command	
Msg.E	Byte	Extension bits	
Msg.DeviceAdr	Byte	Modbus address	Telegram header
Msg.DataArea	Byte	Unused	
Msg.DataIAdr	Word	Data address	
Msg.DataIdx	Byte	Unused	
Msg.DataCnt	Byte	Data count	
Msg.DataType	Byte	Unused	
Msg.Function	Byte	Function code	
Msg.D(0-239)	Byte	User data, if there are	User data

Device-internal transfer structure between the Modbus RTU protocol task and the application.

Here the message header is utilized for the internal forwarding of the message, while the telegram header provides the actual sending or receipt task. The Message itself can contain a maximum of 200 Bytes of user data.

Individually, the user interface for the respective device is separate and described in its own manual.

3 Physical Interface

As a physical interface there can be utilized a RS232C interface for a Point-to-Point connection or a RS422 or RS485 interface for a Bus operation.

The "RTS Control" parameter must be determined in accordance with the physical interface.

The connection positions of the individual interfaces is taken from the device manual.

4 Parametrizing

The parameters necessary for the processing of the Modbus RTU protocol are stored inside the device in a database. These are created on a PC with the aid of the ComPro[1] set-up and diagnostic program and transferred by means of the diagnosis interface into the internal FLASH memory of the device. From here they are retrieved when the operating voltage is switched on or after a cold start and used for initializing the protocol .
Depending on the device, these default parameters can be overwritten by the user program and activated with a warm start command.

The RUN - LED at the device only comes on when all protocols have been initialized without error.

During the initialization, the parameters are tested for their completeness and their permissible limits. If an error is found during testing, it is entered into the trace buffer and the protocol is not activated. This condition can only be ended with a renewed intialization.
When the initialization has been successfully completed, the coupling oeration is started.

The initialization of the Modbus RTU coupling on the KPO 104 communication expert is carried out with another procedure. This is described in the "The Modbus RTU Coupling on the KPO 104" chapter in the KPO 104 manual. The parameters described in the following are also valid for the KPO 104.

Individually, the following parameters must be defined. The default parameters are characterized by being in bold and underlined:

Care must be taken that several protocols are not configured on the same interface. In this case the initializing error 10 will be shown.

Parameter	Meaning	Value rane
Communication line	Device interface being served by the protocol.	0 = Interface is locked all other parameters are no longer relevant 1 = SCL 1 2 = SCL 2 3 = SCL 3 (if available) 4 = SCL 4 (if available) Maximum number of interfaces depending on device, Default interface = Number of the protocol task.
RTS-Control	RS232: RTS is fixed to high, CTS is not evaluated. RS422, RS485: Data driver always switched through. RS232: RTS high only for sending, CTS is evaluated. RS422, RS485: Data driver switched through only for sending.	0 = **no** 1 =yes

[1] The operation of the ComPro program is described in the "Compro Configuration and Diagnosis Program" manual.

Baudrate	Defines the transmission rate.	0 = 50 Baud 1 = 100 Baud 2 = 110 Baud 3 = 150 Baud 4 = 200 Baud 5 = 300 Baud 6 = 600 Baud 7 = 1.200 Baud 8 = 2.400 Baud 9 = 4.800 Baud 10 = **9.600 Baud** 11 = 19.200 Baud
Stopbits	Defines the quantity of stop bits.	**1**, 2 (See on the left)
Parity	Defines the paritybit.	0 = none 1 = **even** 2 = odd
Mode	Determines the operating mode. Differentiates between Slave and Master operation. Also defines the valid address range of the data. Modes 2/3 are for the newer Modicon controls that have more than 10000 Registers. Modes 4/5 correspond to the J-Bus.	0 = Slave / Address:40001-49999 1 = **Master / Address:40001-49999** 2 = Slave / Address:1-65535 3 = Master / Address:1-65535 4 = Slave / Address: 0-65535 5 = Master / Address: 0-65535
Modbus address	Defines the own address of the Modbus.	1,**2**..247
Timeout	Master mode: Defines the maximum waiting time in milliseconds of an answer telegram from the slave. Slave mode: Defines the maximum waiting time in milliseconds of an answer from the coupling partner.	10...**1000**...10000
Retries	Defines the quantity of telegram retries. Only valid in master mode.	0...**3**...10
Error-LED	Define the function of the error LED.	**set/clear** only set

Using PKV30-PB, PKV30-COS, PKV30-DNS and NetNode only the setting 1 stopbit is possible. The setting 2 stopbits is not possible.

Parameter of the Modbus protocol

According to the protocol, the quantity of data bits is fixed to 8.

5 Error Handling

If an error occurs during a data transfer, then the error LED of the interface comes into operation. Depending on the configuration, it is set and extinguished again either with the next error-free data transfer or only after an initialization.

The trace buffer must be evaluated for further error analysis. All the system and initialization errors are automatically entered here. If data transfer or other protocol errors are also to be entered, it must be explicitly switched on via the ComPro[1] configuration and diagnosis program by means of the Tracefilter.

During the initialization, all parameters are tested for their permissible limits. If the parameters are entered over the ComPro configuration and diagnosis program, then no errors can occur here. The testing is necessary, as the user has direct access to the parameters on the communication interface:

Error number	Error	Meaning
0	No error	Positive confirmation on carrying out of command.
4	Task not existing	The addressed task is not available. Msg.Rx has error.
5	Task not initialized	A command to the tesk has been rejected because the task has not been, or was incorrectly, initialized.

Error message by the operating system

[1] The operation of the ComPro program is described in the "Compro Configuration and Diagnosis Program" manual .

Error number	Error	Meaning
10	Serial interface occupied	The serial interface has already been intialized by another task. Error in the "Communication line" parameter.
11	Sum of all baudrates too high	The sum of all Baud rates on all intialized interfaces is too great.
12	Error 'Communication line'	Parametrized interface at the device not available.
13	Error 'Baudrate'	Invalid value for the "Baudrate" intialization parameter.
14	Error 'Parity'	Invalid value for the "Parity" intialization parameter .
15	Error 'Databits'	Invalid value for the "Databits" intialization parameter.
16	Error 'Stopbits'	Invalid value for the "Stopbits" intialization parameter.
17	Error 'RTS-Control'	Invalid value for the "RTS-Control" intialization parameter.
50	Error 'Mode'	Invalid value for the "Mode" intialization parameter.
51	Error 'Modbus address'	Invalid value for the "Modbus address" intialization parameter.
52	Error 'Timeout'	Invalid value for the "Timeout" intialization parameter.
53	Error 'Retries'	Invalid value for the "Retries" intialization parameter.
54	Error 'Error-LED'	Invalid value for the "Error-LED" intialization parameter.
	Error 'Baudrate'	Invalid value for the "Baudrate" intialization parameter.

Initialization Errors (Modbus RTU-Protokoll)

Error number	Error	Meaning
100	Parity error	The interface controller has detected a parity error.
101	Framin error	The interface controller has detected a "Framing error".
102	Overrun error	The interface controller has detected an "Overrun error".
103	To much/less data received	More than 240 user data have been received.
104	CRC error	A CRC error has been determined in the received telegram.
105	Timeout telegram	The coupling partner has not answered within the monitoring time.
110	Unknown exception received	This is an undefined "Exception response code", i.e. 0 or greater than 9 have been received.
111	Exception 1 received	The accessed Slave has answered with an "Exception response code".
112	Exception 2 received	
113	Exception 3 received	
114	Exception 4 received	
115	Exception 5 received	
116	Exception 6 received	
117	Exception 7 received	
118	Exception 8 received	
119	Exception 9 received	
120	Invalid 'Slave address' received	The Slave address in the answering telegram is not that of the addressed Slave.
121	Invalid 'Function code' received	The function information in the received Modbus telegram does not correspond with the issued function.
122	Invalid 'Bytecount' received	The given data count in the received Modbus telegram (bytecount) does not correspond with the transmitted data count.
123	To much/less data received	Incorrect data count in the received Modbus telegram.
124	Invalid data address received	Incorrect data address in the received Modbus telegram.
125	Invalid answer data received	The answering telegram of the Slave has been correctly received but does not fit to the task telegram.
126	Invalid diagnostic code received	Only the 0 diagnostic code, i.e. "Loopback test" , is permissible.

Errors in receiving or evaluating Modbus telegrams

Error number	Error	Meaning
151	Invalid length of message	The message length Msg.Ln is shorter than 8 Bytes and is insufficient for evaluating the telegram header.
152	Unknown message command	The message command Msg.B is invalid.
154	Message error received	In the Slave mode, the coupling partner indicates an error instead of the required data or a confirmation.
155	Timeout message	A time monitoring error has accurred as no answer has arrived from the application program in the parametrized Timeout.
160	Invalid telegram header in acknowledge message	The application program has transmitted a different telegram header in the task confirmation.
161	Error 'Device address'	The Slave addresss in Msg.DeviceAdr was entered incorrectly by the application program.
162	Error 'Data area'	The data area in Msg.DataArea was entered incorrectly by the application program.
163	Error 'Data address'	The data address in Msg.DataAdr was entered incorrectly by the application program.
165	Error 'Data quantity'	The given data count for the Modbus telegram to be created is impermissible.
166	Error 'Data type'	The data type in Msg.DataTyp was entered incorrectly by the application program.
167	Error 'Function'	The function code in Msg.Function was entered incorrectly by the application program.

Error messages when checking incoming sending commands or receipt confirmations

Error number	Meaning	Possible cause
200	Task is not initialized	Task is not initialized and therefor rejects the command message.
201	Task is not ready	Task is waiting for a memory segment and therefor rejects the command message.
210	Error opening the data base	Incorrect or mixed-up database.
212	Error reading the data base	Incorrect or mixed-up database.
213	Systemerror 'RcsPutStructure'	Internal operating system error.

System errors

6 Diagnostic Functions

The error LEDs on the device are available for the diagnostic functions as also is the possibbility of the ComPro via the diagnosis interface.

6.1 Error-LED

The method of working of the error LED must be parametrizised. In the "Set/delete" method of operation, it is switched on with every recognized error and switched off again with the next error-free transmission. In the "Set only" storing method of operation, the error display is extinguished only by switching on or resetting the device.

6.2 Trace Function

System and intialization errors, e.g. incorrect parametrization, are always entered into the Trace buffer independently of the selected conditions!

A trace function can be activated for test and check purposes. In this method, all the sending tasks to the Modbus RTU protocol or all the tasks with received data from the Modbus RTU protocol are entered in a Trace buffer. Receiving telegramms are only entered when no errors have been recognized. Otherwise only an error message is entered. Also the issued or received task confirmations are stored in the Trace buffer. Thus, with the aid of the time stamp, the complete reaction time in the sending as well as the receiving direction can be checked.

The entry into the Trace buffer always occurs from the task structure immediately after the input or after the output of an receiving task. In this way, the data between the application and the Modbus RTU protocol are documented. The exact telegram structure, the CRC16 checksum and a possible swapping of the data words thus cannot be seen in the Trace entry.

By means of setting the Trace filter in ComPro, the entry can be limited to particular events and portions of the task structure.

The operation of the Trace function is dealt with in the ComPro manual. The following paragraphs detail which conditions and data the Modbus RTU protocol makes available for a Trace input.

6.2.1 Inputting the Talegram Data into the Trace Buffer

Besides events and errors, also the complete telegeam data can be input into the Trace buffer. In order to reduce the amount of data, there is the possibility of inputing only the telegram header. The telegram header in the Trace is structured as follows for the Modbus RTU protocol:

1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7thByte	8th Byte
Modbus address	0 Unused	Data adress in format LSB-MSB		0 Unused	1-240 Data count	0 Unused	Function code

Data arrangement of the telegram header in the Trace display

6.2.2 Inputting Protocol Errors into the Trace Buffer

Protocol errors are input into the Trace buffer with an error number and converted to clear text when being read out by the Monitor program. The possible error entries are given in the "Error Handling" chapter.

6.2.3 Entering the Time Stamp into the Trace Buffer

For each input the current condition of the cyclic system timer is stored as well and displayed when reading out the Trace buffer. Here there is always displayed the difference between the first input of the Trace buffer read out.
The elapsed time is calculated by multiplication with the cycling time. This is called up from the on-line system hardware configuration menu.

6.3 Extended Task Status

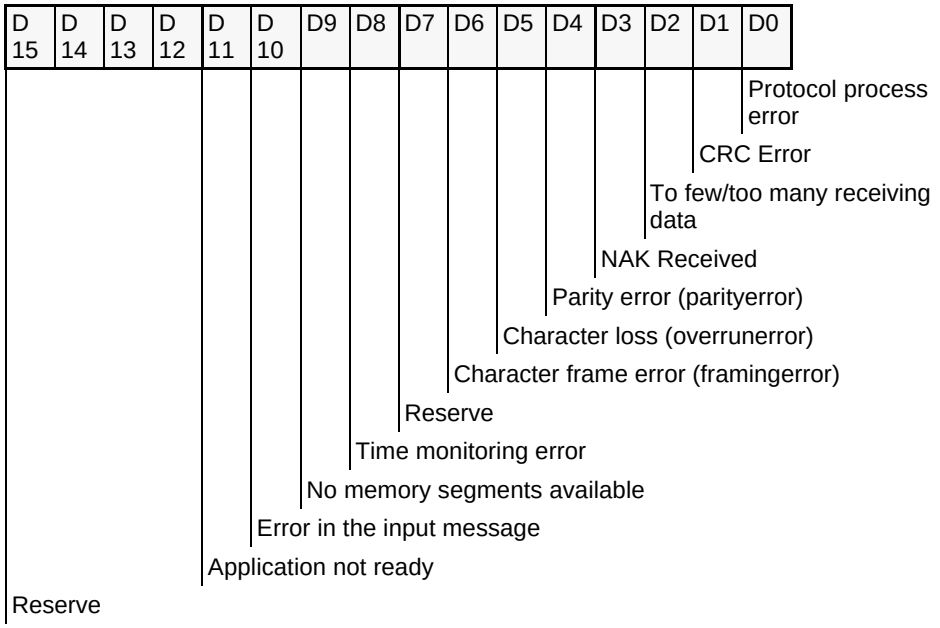
In the ComPro configuration and diagnosis program an extended task status can be displayed over the on-line task status menu path. From this there can be taken the current processing condition of the protocol and statistical information on the previous protocol procedure. This information is always updated by the coupling protocol. The complete data can be set to 0 by means of the on-line task status delete menu path.

Status	Meaning
Task state	Current condition of the protocol procedure. For definition see below.
Send telegrams	Gives the number of error-free sent telegrams.
Receive telegrams	Gives the number of error-free received telegrams.
Send retries	Tells how often the send telegram was repeated before it was transferred error-free or before the maximum number of repeats was reached.
Transmission error	Tells how many send tasks have been lost because of syntax or transferrance errors.
Error bits [Hex]	Allocates every incoming error to an error class and shows this by placing a bit . A display only comes about when, despite repeats, this has led to a loss of telegram. For definition see below.
Last error	Gives the number of the last incomming error . A display only comes about when, despite repeats, this has led to a loss of telegram .

Definition of the extended task status

Task state	Meaning
0	Protocol not initialized.
10	Normal state in Master mode, that means waiting for a send message
11	Syntax check of send message in master mode
12	Sending telegram or waiting for slave answer telegram respectively receiving the slave answer telegram
13	Syntax check of received telegram in master mode
20	Normal state in Slave mode, that means waiting for a receiving telegram
21	Syntax check of received telegram in slave mode
22	Waiting for answer message from application
23	Sends NAK and waits for telegram repeat.
24	Sending answer telegram to master
25	Waiting for memory segment

Definition of the task state (Modbus RTU-Protocol)



Definition of the error bits

The corresponding bit is set for each error. Deletion occurs only after a renewed starting of the device or by means of targeted reset with the aid of ComPro.