Feeder and Transformer Terminals REF 54_, RET 54_

Modbus Communication Protocol

Technical Description

Contents

1. About this manual

1.1. Copyrights

1.2. Trademarks

1.3. General

This manual describes how the Modbus protocol and its parameterization are implemented in the REF 541/543/545 Feeder Terminals and the RET 541/543/545 Transformer Terminals.

This manual is valid for REF 54_ and RET 54_, release 3.0 and later.

1.4. Use of symbols

This document includes information icons that point out important information. The corresponding icons should be interpreted as follows:

The information icon alerts the reader to relevant facts and conditions.

1.5. Abbreviations

ASCII	American Standard Code for Information Interchange
CD	Change Detect
CO	Coil
CRC	Cyclic Redundancy Check
DI	Digital Input
EOF	End Of Frame
HMI	Human-machine interface
HR	Holding Register
IR	Input Register
LRC	Longitudinal Redundancy Check
PLC	Programmable Logic Controller
PMT	Protocol Mapping Tool
POD	Protocol Object Dictionary
RTU	Remote Terminal Unit
UDR	User-Defined Register

1.6. Related documents

Name of the manual	**MRS number**
General Manuals	
Installation Manual	1MRS750526-MUM
Technical Descriptions of Functions (CD-ROM)	1MRS750889-MCD
Manuals for REF 54_ and RET 54_	
RE_ 54_ Operator's Manual	1MRS750500-MUM
Protection & Control Terminals REF 54_, RET 54_, REM 54_, REC 523 Configuration Guideline	1MRS750745-MUM
Bus Connection Module RER 123, Technical Description	1MRS751143-MUM
Bus Connection Module RER 133, Technical Description	1MRS755163
Manuals for REF 54_	
Technical Reference Manual, General	1MRS750527-MUM
Parameter and event lists for REF 54_	
Parameter List for REF 541 and REF 543	1MRS751774-RTI
Parameter List for REF 545	1MRS751775-RTI
Event List for REF 541 and REF 543	1MRS751776-RTI
Event List for REF 545	1MRS751777-RTI
Manuals for RET 54_	
Technical Reference Manual, General	1MRS755225
Parameter and event lists for RET 54_	
Parameter List for RET 541 and RET 543	1MRS755228
Parameter List for RET 545	1MRS755229
Event List for RET 541 and RET 543	1MRS755226
Event List for RET 545	1MRS755227
Technical Reference Manual, Standard Configurations	1MRS 751802-MUM

Tool-specific manuals	
CAP505 Installation and Commissioning Manual	1MRS751901-MEN
CAP505 User's Guide	1MRS752292-MUM
CAP505 Protocol Mapping Tool, Operation Manual	1MRS755277
LIB, CAP, SMS, Tools for Relays and Terminals, User's Guide	1MRS752008-MUM
CAP 501 Installation and Commissioning Manual	1MRS751899-MEN
CAP 501 User's Guide	1MRS751900-MUM

1.7. Document revisions

Version	**Date**	**History**
A	02.04.2004	Document created
B	20.01.2005	RET 54_ added to manual
C	08.07.2005	RER 123 support added

2. Protocol overview

The Modbus protocol was first introduced by Modicon Inc. and is widely accepted as a communication standard for industrial device controllers and programmable logic controllers (PLCs). The protocol determines how each controller connected to a Modbus network will recognize a message addressed to it. It also determines the task to be performed and extracts any data or other information contained in the message. If a reply is required, the controller will construct a reply message and send it using the Modbus protocol.

A master device can be connected to slave devices either directly, or via modems using a compatible serial interface. The interface defines the connector pinouts, cabling, signal levels, transmission baud rates, and parity checking.

The communication technique used in the Modbus protocol is a master-slave technique. This means that only one device can be the master and initiate transactions while other devices connected to the network are slaves and can accordingly not initiate any transactions.

A message sent by the master to the slave is called a query. The master can address a query to an individual slave or to all slaves, that is, to broadcast the query. After the slave has received a query, it attempts to perform the requested task. If a query has been sent to an individual slave, the slave will send a message, that is, a response to the master. However, if it has been broadcast, no response will be sent. The response can be either a normal response (in case of performing the requested task) or an exception response (other cases).

There are four types of Modbus data: digital inputs (DI), input registers (IR), coils (CO), and holding registers (HR). Each type of data consists of either scan or control points, which all have separate 16-bit addresses.

All the data addresses in the Modbus protocol are referenced to zero. The first occurrence of a data item will be addressed as item number zero. For instance, coil 1 will consequently be addressed as coil 0, and coil 127 as coil 126 (7E in hex format).

2.1. Format of Modbus query and response

The query of the Modbus protocol includes the following fields:

- address of the slave to which the query is directed (if the query is broadcast, the address is 00),
- a function code defining the requested task to be performed,
- data to be sent, and
- an error-checking field.

The response includes fields containing

- the address of the slave,
- confirmation of the performed task in the form of a function code,
- any data to be returned, and
- an error-checking field.

If an error occurs in the receipt of the message, or if the slave is unable to perform the requested task, the slave will construct an error message and end it as its response.

The Modbus protocol has two serial transmission modes: ASCII and RTU. The transmission modes define the bit contents of the message fields transmitted in the network. They also determine how the information is packed into message fields and how it is decoded. The selected mode and the serial parameters must be the same for all devices in a Modbus network.

2.1.1. ASCII mode

In the ASCII (American Standard Code for Information Interchange) mode, each byte in a message is sent as two ASCII characters forming a hexadecimal number. The binary format of the characters is:

- one start bit,
- seven data bits,
- one even, odd or no parity bit, and
- one or two stop bits.

In ASCII mode each frame has a start character (:) and an end character (<CR><LF>)

2.1.2. RTU mode

In the RTU (Remote Terminal Unit) mode each message character is sent in binary format. Each character has:

- one start bit,
- eight data bits,
- one even, odd or no parity bit, and
- one or two stop bits.

In RTU mode, the end of frame (EOF) detection is time based. End of frame timeout is 3.5 times one character time, while the Next character timeout is 1.5 times one character time.

In both ASCII and RTU mode, the number of stop bits depends on whether a parity bit is used. If odd or even parity is used, the character will have one stop bit. If parity is not used, however, there will be two stop bits.

For further information about the Modbus protocol, please go to www.modbus.org or www.modicon.com on the Internet.

2.2. Physical interface

REF 54_ and RET 54_ can be connected to a fibre-optic system using the RER 123 Bus Connection Module, or to a 2- or 4-wire RS-485 system using the RER 133 Bus Connection Module. For further information, refer to the Bus Connection Module Technical Descriptions (see “Related documents” on page 5).

3. Interface configuration

This section describes the communication parameters required to configure the terminal to communicate using the Modbus protocol.

The Modbus protocol can be used only when the protocol is properly configured. The protocol must be selected in CAP 505 as an add-on protocol. For additional information, refer to the CAP 505 Operator's Manual.

When the protocol is selected and the relay configuration is created (refer to Relay Configuration Tool in CAP 505), the protocol mapping must be created or modified using Protocol Mapping Tool (PMT). The protocol parameters (as described in Section 3.2.) are available only after the protocol is first selected and then activated. When the protocol mapping is first downloaded and stored in the relay, a reset of the relay activates the protocol.

The protocol parameters can be uploaded, reviewed and modified using the Relay Setting Tool from the CAP 501/505 package by choosing the Communication library and the Modbus pages.

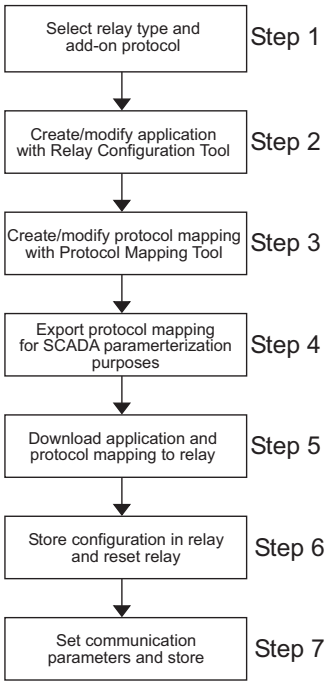

Fig. 3.-1 Interface configuration

- If the application is changed, start over from step 2.
- If you wish to keep the existing protocol mapping, you should select a new name for the protocol mapping or skip step 3.
- If you create a new protocol mapping with the wizard, the protocol mapping addresses will be changed.
- Application downloading overwrites existing add-on protocol parameters and protocol mapping. A back up of parameters can be created using the Relay Setting Tool in CAP 505.

3.1. Protocol mapping

Protocol mapping is a cross-reference table between the application and the remote control protocol, e.g. Modbus. This table defines what information can be accessed using the protocol interface. As the terminal is progammable and may run various application setups with different combination of function blocks, the protocol mapping is fully re-configurable to adapt the device to the requirements of the SCADA system. Protocol mapping can be referred to as Protocol Object Dictionary or POD in REC 523 and REX 521 product documentation.

3.1.1. General guidelines on how the application data is seen on the Modbus protocol

The following figure and table describe how the process data in a REF 54_ or a RET 54_ is seen on the Modbus protocol.

In the application example below, all the possible process data is present. The grey boxes show to which Modbus data category the signals belong.

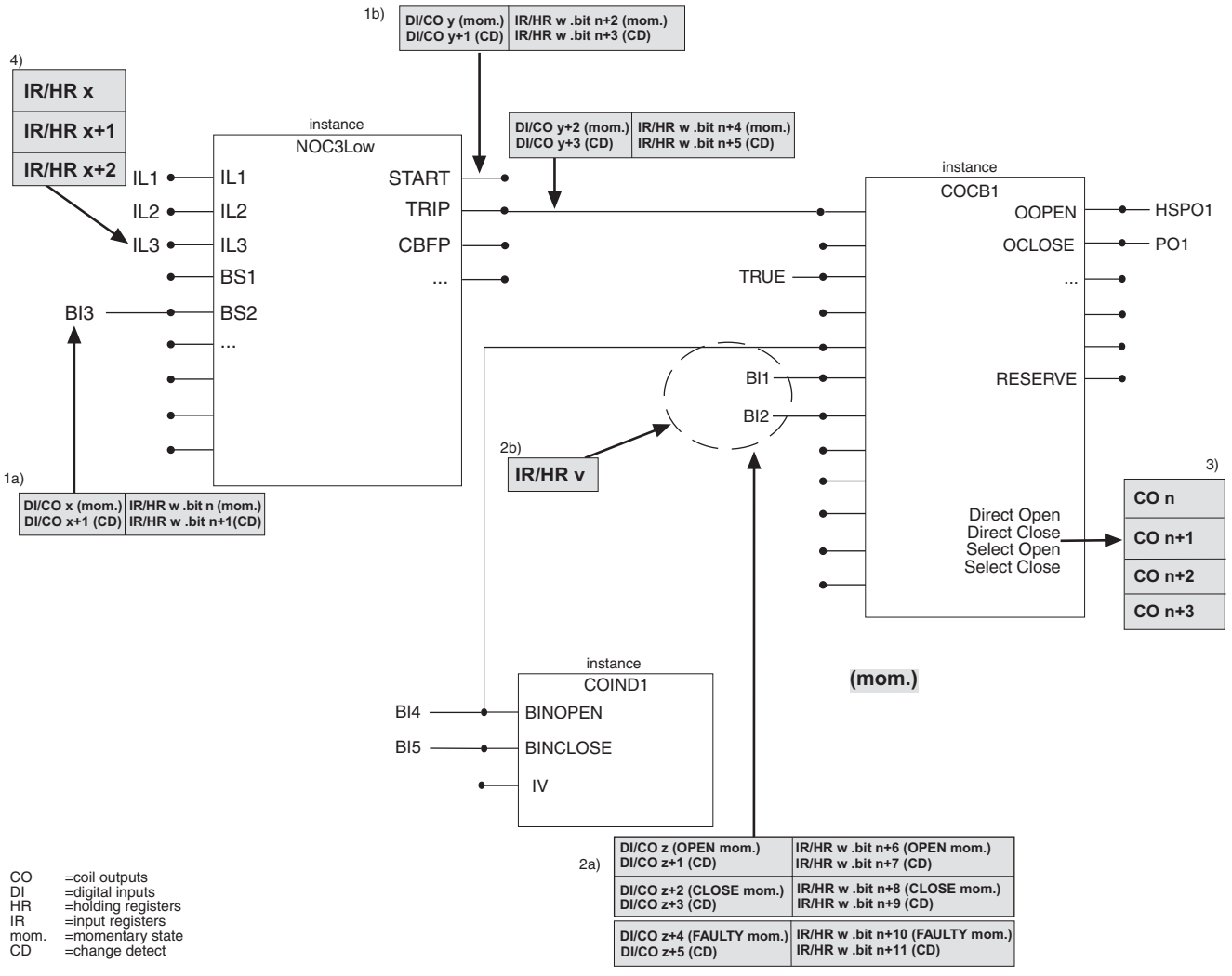

Fig. 3.1.-1 Application example, explained in Table 3.1.-1

Table 3.1.-1 Explanations to the application example in Fig. 3.1.-1

No	Application data type	Explanation	Modbus data type
1a)	One Bit Input	Binary input to a function block, e.g. blocking input.	Coils (0x references) Digital inputs (1x references) Input Registers (3x references) Holding Registers (4x references)
1b)	One Bit Output	Binary output from function block, e.g. START or TRIP signals.	Coils (0x references) Digital inputs (1x references) Input Registers (3x references) Holding Registers (4x references)
2a)	Two Bit Input	Binary position data coded in two bits (OPEN, CLOSE).	Coils (0x references) Digital inputs (1x references) Input Registers (3x references) Holding Registers (4x references) Coded in three bits: Bit 1: OPEN Bit 2: CLOSE Bit 3: FAULTY (validity): 1 if corresponding binary inputs of both Bit 1 and Bit 2 = 1. In case validity bit = 1, both OPEN value and CLOSE value are set to 0 in Modbus data.
2b)	Two Bit Input	In addition to 2a) the OPEN and the CLOSE bit values are also coded as least significant bits in an input and holding register. (One register per object.)	Input Registers (3x references) Holding Registers (4x references) Values: 1 = CLOSE 2 = OPEN 3 = Undefined 0 = Undefined
3)	Control output points	Outputs controlled from the Modbus master.	Coils (0x references) Digital inputs (3x references) Note! Only coils are writable.
4)	Measurement inputs	Measurement inputs to the function blocks.	Input Registers (3x references) Holding Registers (4x references)
Not visible in the figure	Parameters, settings, etc.	Some parameters of the device and function blocks may be adjustable (look in the Modbus point list of the Modbus configuration).	Input Registers (3x references) Holding Registers (4x references) Note! Only the Holding Registers are writable.

3.2. Protocol parameters

The protocol and link parameters of the Modbus interface can be programmed by means of a local HMI by selecting `Configuration\Communication\Comm.settings\Modbus`.

Table 3.2.-1 Modbus protocol parameters

Parameter name	DB name	Values	Default	Visible	Read/ write	Explanation
Unit address	F504V001	1...247	1	Remote HMI, HMI, SPA	R/W	Address of the terminal in the Modbus network. Must be same as configured in the master station.
CRC order	F504V002	0...1 [0=LO/HI, 1=HI/LO]	0	Remote HMI, HMI, SPA	R/W	The order of CRC bytes in protocol frame. Not used in ASCII mode.
Protocol mode	F504V003	0...1 [0=ASCII, 1=RTU]	1	Remote HMI, HMI, SPA	R/W	Chooses whether the terminal uses ASCII or RTU mode.
Password	F504V004	ASCII codes	4 space characters	Remote HMI, HMI, SPA	R/W	Password for control operations
Protocol mapping diagnostic parameter	F504V060		0	SPA	R	Total entries counter
Protocol mapping diagnostic parameter	F504V061		0	SPA	R	Number of entries not in use
Protocol mapping diagnostic parameter	F504V062		0	SPA	R	Number of entries with invalid, uncorrectable contents (INV).
Protocol mapping diagnostic parameter	F504V063		0	SPA	R	Number of entries with corrected contents (COR).
Protocol mapping diagnostic parameter	F504V064		0	SPA	R	Number of entries referring to a nonexistent block (NBL).
Protocol mapping diagnostic parameter	F504V065		0	SPA	R	Number of entries referring to invalid objects of an existing block (NOB).
Protocol mapping diagnostic parameter	F504V066		0	SPA	R	Number of entries translated to operational protocol mapping.
Baud rate	F504V211	0...6 [1) 2)] [0=300 1=600 2=1200 3=2400 4=4800 5=9600 6=19200]	6	Remote HMI, HMI, SPA	R/W	
Number of stop bits	F504V212	0...2 [3)]	1	HMI, SPA	R/W	
Next character timeout	F504V215	0=Not in use, 2...65535 [ms] [1)]	1000 (ASCII) 0 (RTU)	HMI, SPA	R/W	
End Of Frame timeout	F504V216	2....65535 [ms] [2)]	1000 (ASCII) 2 (RTU)	HMI, SPA	R/W	
Parity	F504V230	0...2 [3)] [0=None 1=Odd 2=Even]	2	Remote HMI, HMI, SPA	R/W	
Number of data bits	F504V231	5...8	7 (ASCII) 8 (RTU)	SPA	R/W	

Table 3.2.-1 Modbus protocol parameters (Continued)

Parameter name	DB name	Values	Default	Visible	Read/ write	Explanation
Frame error counter	F504V261	0...65535	0	SPA	R	Frame error counter 4)
Parity error counter	F504V262	0...65535	0	SPA	R	Parity error counter 4)
Overrun error counter	F504V263	0...65535	0	SPA	R	Overrun error counter 4)
Protocol mapping diagnostic parameter	F504V700			SPA	R/W	Protocol mapping name
Protocol mapping diagnostic parameter	F504M001			SPA	R/W	Modbus protocol mapping file

1) The change of Baud rate parameter value forces an automatic update of the Next character timeout parameter value to 1,5 character times. If the timeout value is lower then 2ms then it is disabled (set to 0 - not in use).
2) The change of Baud rate parameter value forces an automatic update of End Of Frame timeout parameter value to 3,5 character times.
3) The change of Parity parameter value forces an automatic update of the number of stop bits to 1, with parity used, and to 2 for parity none.
4) The counters are 16 bit cyclical counters, after reaching 0xFFFF (65535) next increment changes the value back to 0x0000. The default setting is reset to 0 at system start.

The General Modbus parameters (F504V001-004, 211-212, 215-216 and 230-231) can be changed at any time and will be taken into use directly after the write command.

3.3. Supported application functions

The implementation of the Modbus protocol in REF 54_ and RET 54_ supports the following functions:

Table 3.3.-1 Supported application functions

Function code (HEX)	Function description
01	Read coil status Reads the status of discrete outputs.
02	Read digital input status Reads the status of discrete inputs.
03	Read holding registers Reads the contents of output registers.
04	Read input registers Reads the contents of input registers.
05	Force single coil Sets the status of a discrete output.
06	Preset single register Sets the value of a holding register.
08	Diagnostics Checks the communication system between master and slave.
0B	Get comm event counters Returns amount of successful read/write operations on data points.
0F	Force multiple coils Sets the status of multiple discrete outputs.
10	Preset multiple registers Sets the value of multiple holding registers.
17	Read/write holding registers Exchanges holding registers in one query.
18	Read FIFO queue Reads set of multiple holding registers interpreted as FIFO queue

3.4. Supported diagnostic subfunctions

The implementation of the Modbus protocol in REF 54_ and RET 54_ supports the following subfunction codes:

Table 3.4.-1 Supported diagnostic subfunctions

Code (HEX)	Name	Subfunction description
00	Return query data	The data in query data field is returned (looped back) in response. The entire response is identical to the query.
01	Restart communication option	The peripheral port of the slave is initialized and restarted, and the communication event counters are cleared. Before this, a normal response is sent unless the port is in the listen only mode. If the port is in the listen only mode, no response will be sent.
02	Return Diagnostic Register	The contents of the slave's diagnostic register is returned in response.
04	Force listen only mode	The slave is forced to enter the listen only mode for the Modbus communication.
10	Clear counters and diagnostic register	All the counters and the diagnostic register are cleared.
11	Return bus message count	The number of messages in the communications system detected by the slave since its restart, clear counters operation or power-up is returned in response.
12	Return bus communication error count	The number of CRC errors encountered by the slave since its restart, clear counters operation or power-up is returned in response.
13	Return bus exception error count	The number of Modbus exception responses sent by the slave since its restart, clear counters operation or power-up is returned in response.
14	Return slave message count	The number of messages addressed to the slave or broadcast which the slave has processed since its restart, clear counters operation or power-up is returned in response.
15	Return slave no response count	The number of messages addressed to the slave for which a response (neither a normal response nor an exception response) has not been sent since its restart, clear counters operation or power-up is returned in response.
16	Return slave NACK response count	The number of messages addressed to the slave (for which a NACK response has been sent) is returned in response.
17	Return slave busy response count	The number of messages addressed to the slave (for which a Slave Busy response has been sent) is returned in response.
18	Return bus character overrun count	The number of messages addressed to the slave for which it has not been able to send a response due to a character overrun since its last restart, clear counters operation or power-up is returned in response.

Note: Please see “Supported exception responses” on page 18 for supported exceptions.

Sending other subfunction codes than the ones listed above causes an “Illegal data value” response

3.5. Diagnostic counters

The Modbus protocol provides the following diagnostic counters:

Table 3.5.-1 Diagnostic counters

Name	Meaning
Bus message count	The number of messages in the communications system detected by the slave since its restart, clear counters operation or power up.
Bus communication error count	The number of CRC or LRC errors encountered by the slave since its restart, clear counters operation or power up.
Bus exception error count	The number of Modbus exception responses sent by the slave since its restart, clear counters operation or power up.
Slave message count	The number of messages addressed to the slave or broadcast which the slave has processed since its restart, clear counters operation or power up.
Slave no response count	The number of messages addressed to the slave for which a response (neither a normal response nor an exception response) has not been sent since its restart, clear counters operation or power up.
Bus character overrun count	The number of messages addressed to the slave for which it has not been able to send a response due to a character overrun since its restart, clear counters operation or power up.
Slave NACK response count	The number of messages addressed to the slave (for which a NACK response has been sent) is returned in response.
Slave busy response count	The number of messages addressed to the slave (for which a Slave Busy response has been sent) is returned in response.

3.6. Possible exception codes

The following exception codes may be generated by the Modbus protocol:

Table 3.6.-1 Exception codes

Code (HEX)	Name	Meaning
01	Illegal function	The slave does not support the requested function.
02	Illegal data address	The slave does not support the data address, or the number of items in the query is incorrect.
03	Illegal data value	The value in the query data field is out of range.

If an “Illegal data value” exception response is generated when attempting to preset multiple registers, the contents of the register to which an illegal value has been imposed and the following registers will not be changed. The registers which have already been preset will not be restored.

3.7. User-defined registers

User-defined registers (UDRs) are used to map different already existing data points into a shared data category. REF 54_ and RET 54_ support up to 100 UDRs.

UDRs are created in the Protocol Mapping Tool (PMT) as a copy of other registers (input or holding registers) or as a set of up to 16 binary points (coils or digital inputs). For instructions on how to create UDRs, refer to Protocol Mapping Tool, Operator’s Manual. (See “Related documents” on page 5.)

3.8. Digital inputs

As the master may not detect the changes of states of all digital signals when scanning, an additional change detect (CD) indication bit is created for every momentary indication point; see the example in Fig. 3.8.-1.

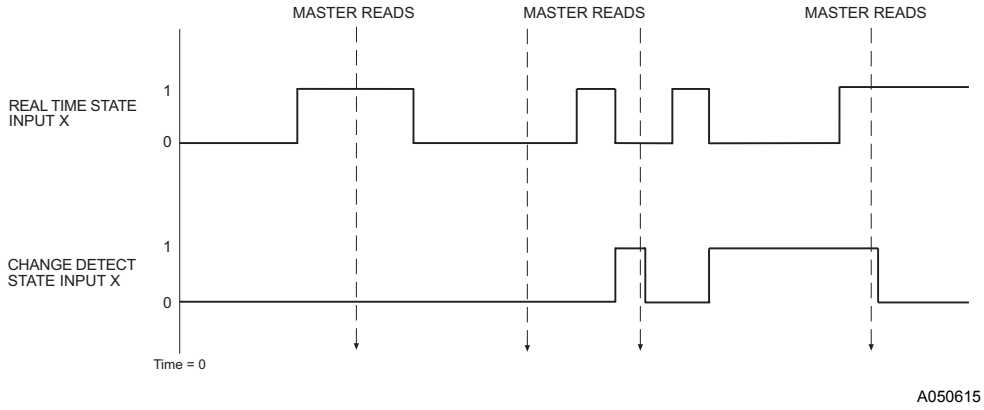

Fig. 3.8.-1 Change detection bit

If the momentary value of an indication bit has changed two or more times since the master last read it, the CD bit will be set to 1. When the CD bit has been read, it will be set to 0.

The momentary and the CD bit of a certain indication point always occur as a pair in the Modbus memory map.

Typically, a digital input change generates an event. These events update a digital register. A register can be updated by several different events. For an example on this, please see Section “Double bit indication - case 1: mapping into a holding register (and an input register)” on page 22, where four different events use the same register.

3.9. Modbus data mapping

There are two types of monitoring data: digital indications (DI) and measured values. For convenience and efficiency, the same data can be read from different data areas. Measured values and other 16-bit values can be read either from the IR or HR (read-only) area and digital indication values from either the DI or coil (read-only) area. It is also possible to read the status of the DIs as packed 16-bit registers from both the IR and HR area.

Consequently, all the monitoring data can be read as consecutive blocks of data from the IR or HR area.

However, there are also write-only coils in the Modbus data mapping of the REF 54_ and the RET 54_. Those coils are marked as writable (W) in the point lists of the protocol mappings.

The response time for the data listed under Slowly changing and Device data categories is longer than with other data categories. If possible, it is recommended to poll Slowly changing and Device data categories less frequently than others to save the bandwidth.

4. Appendix A: Profile checklist

MODBUS **DEVICE PROFILE DOCUMENT**	
Vendor Name:	ABB Oy, Distribution Automation
Device Name:	REF 54_, Release 3.0 RET 54_, Release 3.0
Device Function:	Slave
Modes:	RTU ASCII

Supported function codes

Code (HEX)	Function	Supported
01	Read coil Status	Yes
02	Read Input Status	Yes
03	Read Holding Register	Yes
04	Read Input Registers	Yes
05	Force Single Coil	Yes
06	Preset Single Register	Yes
07	Read Exception Status	No
08	Diagnostics	Yes
0B	Get Comm Event Counter	Yes
0C	Get Comm Event Log	No
0F	Force Multiple Coils	Yes
10	Preset Multiple Registers	Yes
11	Report Slave ID	No
14	Read General Reference	No
15	Write General Reference	No
16	Mask Write 4x Register	No
17	Read/Write 4x Registers	Yes
18	Read FIFO queue	Yes

Supported diagnostics subfunction codes

Code (HEX)	Name	Supported
00	Return Query Data	Yes
01	Restart Communication Option	Yes
02	Return Diagnostic Register	Yes
03	Change ASCII Delimiter	No
04	Force Listen Only Mode	Yes
10	Clear Counters and Diagnostics Register	Yes
11	Return Bus Message Count	Yes
12	Return Bus Communication Error Count	Yes
13	Return Bus Exception Error Count	Yes
14	Return Slave Message Count	Yes
15	Return Slave No Response Count	Yes
16	Return Slave NAK Count	Yes
17	Return Slave Busy Count	Yes
18	Return Bus Character Overrun Count	Yes

Supported exception responses

Code (HEX)	Name	Supported
01	ILLEGAL FUNCTION	Yes
02	ILLEGAL DATA ADDRESS	Yes
03	ILLEGAL DATA VALUE	Yes
04	SLAVE DEVICE FAILURE	No
05	ACKNOWLEDGE	No
06	SLAVE DEVICE BUSY	No
07	NEGATIVE ACKNOWLEDGE	No
08	MEMORY PARITY ERROR	No

Supported data types

Code (HEX)	Name	Supported
YY00XXXX	Coil number XXXX of data category YY	Yes
YY01XXXX	Digital input number XXXX of data category YY	Yes
YY03XXXX	Input register number XXXX of data category YY	Yes
YY04XXXX	Holding register number XXXX of data category YY	Yes
YY06XXXX	General reference	No

Supported event reporting methods

Name	Supported
Momentary change detect on digital input	Yes

5. Appendix B: Examples on data mapping

In the following examples Modbus addresses are only defined as offsets in the range assigned to the function block. The physical address will be assigned by the Protocol Mapping Tool (PMT) when building the protocol mapping table.

Table 5.-1 Abbreviations and glossary for examples

Name	Signal name
DB Name	Object name in SPA
Type	Type of item in the database or type of event, see Table 5.-4
Conversion	Type of conversion, see Table 5.-3
Point (HEX)	MODBUS point number representing the data item
Use (bin)	Use field functions, see Table 5.-4
CO	Coil
CTL	Controls
DDE	Device data
HR	Holding register
PDB	Process data: basic range
PDE	Process data: extended range (with added momentary change detect bits)
PRG	Pack register points

Table 5.-2 Data types used in Modbus

Name	Code	Data type
BOOL	0	Boolean value - 0 or 1
DPBOOL	1	Double point value: 00-middle, 01-closed, 10-open, 11-faulty
SINT	2	16-bit signed integer
INT	3	16-bit signed integer
DINT	4	32-bit signed integer
USINT	5	16-bit unsigned integer with range limited to 0...255, values over 255 are truncated to 255
UINT	6	16-bit unsigned integer
UDINT	7	32-bit unsigned integer
REAL	8	32-bit floating point
TIME	9	32-bit unsigned integer containing number of milliseconds
TOD	10	32-bit unsigned integer containing time of the day since midnight in 100us units
DATE	11	32-bit unsigned integer containing number of days since 01-01-1980
STRING	13	String value
BYTE	15	8-bit unsigned integer
WORD	16	16-bit unsigned integer
DWORD	17	32-bit unsigned integer
EV_NODAT	18	Event without data
EV_1BIT	19	Event with 1-bit data
EV_2BIT	20	Event with 2-bit data
EV_3BIT	21	Event with 3-bit data (treated as EV_NODAT - phase information will be ignored)
EV_FLOAT	22	Event with floating point value
EV_INT16	23	Event with 16-bit integer value
EV_INT32	24	Event with 32-bit integer value
EV_COUNT	25	Event with 32-bit counter value
EV_32BIT	26	Event with 32-bit value

Table 5.-2 Data types used in Modbus (Continued)

Name	Code	Data type
TIME_YR	54	Device clock, year field
TIME_MON	55	Device clock, month field
TIME_DAY	56	Device clock, day field
TIME_HR	57	Device clock, hour field
TIME_MIN	58	Device clock, minute field
TIME_SEC	59	Device clock, second field
TIME_10MS	61	Device clock, 10 milliseconds field
TIME_YRNN	62	Device clock, year field (range 0...99)

Table 5.-3 Conversion types used in the examples

Conversion type	Name in PMT	Code	Description
No operation	No_operation	0	No special handling all actions according to Modbus and database type
Scale by 10	Mul_10	1	Value will be multiplied by 10 and treated as unsigned
Scale by 1000	Mul_1000	3	Value will be multiplied by 1000 and treated as unsigned
Always close	Always_close	9	Conversion for events of type EV_NODAT or EV_3BIT used to set analog value of 1
Always open	Always_open	10	Conversion for events of type EV_NODAT or EV_3BIT used to set analog value of 2
Always middle	Always_middle	11	Conversion for events of type EV_NODAT or EV_3BIT used to set analog value of 0
Always faulty	Always_faulty	12	Conversion for events of type EV_NODAT or EV_3BIT used to set analog value of 3
Always off	Always_off	13	Conversion for events of type EV_NODAT or EV_3BIT used to set analog value of 0
Always on	Always_on	14	Conversion for events of type EV_NODAT or EV_3BIT used to set analog value of 1
Timer	Timer	27	Device clock will be read or written
Change detect	Ch_detect	28	Entry containing change detect bit informing about 2 or more changes of digital input value between scans
Is close	IS_CLOSE	82	Value will be set to 1 if close state is detected (coverts value 1 to 1 and any other to 0)
Is open	IS_OPEN	83	Value will be set to 1 if open state is detected (coverts value 2 to 1 and any other to 0)
Is faulty	Is_faulty	84	Value will be set to 1 if faulty state is detected (coverts value 3 to 1 and any other to 0)
Pack registers	Pack_reg	110	Register with packed status of binary signals
Point number of the first binary signal	First_coil	111	Modbus point number of the first binary signal (coil) packed into the special register on lsb position
Number of signals packed	Num_coils	112	Number of binary signals (coils) packed into the special register

Table 5.-4 Use field bit mask values (bin)

Bit number	Value	Description
Bit 0	0	Entry not in use [1)]
"in use"	1	Entry in use
Bit 1	0	Writing is allowed
"read-only"	1	Writing is not allowed
Bit 2	0	Data is read from the device database
"on request"	1	Data is read from the application on request
Bit 3	0	Data is updated from device internal events
"background scan"	1	Data is updated using a cyclic poll

1) When entry is set to not in use, it is removed from the protocol mapping

5.1. Single bit with change detect: mapping into 2 coils (and 2 digital inputs)

This example uses the NOC3Low function block.

	Name	DB Name	Data type	Conversion type	Point (HEX)	Use (bin)
1	Status of START signal from 3I>	F031O001	BOOL	No operation	PDE CO 0800	0011
2	START signal from 3I> signal reset	F031E000		Always off		0011
3	START signal from 3I> signal activated	F031E001		Always on		0011
4	Status of START signal from 3I>	F031O001	BOOL	Change detect	PDE CO 0801	0011

Row 1 defines the signal and the corresponding database object connected to this point on the Modbus protocol.

Rows 2 and 3 define the events used to update the Modbus data point defined at row 1 and the associated change detect data point (row 4).

Row 4 defines an associated momentary change detect point (see Section "Digital inputs" on page 16).

The same information is also accessible from the two digital input points of the same point numbers as these coils.

5.2. Double bit indication - case 1: mapping into a holding register (and an input register)

This example uses the COCB1 function block.

	Name	DB Name	Data type	Conversion type	Point (HEX)	Use (bin)
1	Object state	F120V001	USINT	No operation	PDB HR 0000	0011
2	Breaker position open	F120E000		Always open		0011
3	Breaker position close	F120E001		Always close		0011
4	Breaker position faulty	F120E002		Always faulty		0011
5	Breaker position middle	F120E003		Always middle		0011

Row 1 defines the signal and the corresponding database object connected to this point on the Modbus protocol.

Rows 2 to 5 define the events used to update the Modbus data point defined at row 1.

This holding register point is read-only. The same information is also accessible from the input register of the same point number as this holding register.

5.3. Double bit indication - case 2: mapping into a set of coils (and a set of digital outputs)

This example uses the COCB1 function block.

	Name	DB Name	Data type	Conversion type	Point (HEX)	Use (bin)
1	Object state (close)	F120V001	USINT	Is close	PDE CO 0800	0011
2	Breaker position open	F120E000		Always off		0011
3	Breaker position close	F120E001		Always on		0011
4	Breaker position faulty	F120E002		Always off		0011
5	Breaker position middle	F120E003		Always off		0011
6	Object state (close change detect)	F120V001	USINT	Change detect	PDE CO 0801	0011
7	Object state (open)	F120V001	USINT	Is open	PDE CO 0802	0011
8	Breaker position open	F120E000		Always on		0011
9	Breaker position close	F120E001		Always off		0011
10	Breaker position faulty	F120E002		Always off		0011
11	Breaker position middle	F120E003		Always off		0011
12	Object state (open change detect)	F120V001	USINT	Change detect	PDE CO 0803	0011
13	Object state (faulty)	F120V001	USINT	Is faulty	PDE CO 0804	0011
14	Breaker position open	F120E000		Always off		0011
15	Breaker position close	F120E001		Always off		0011
16	Breaker position faulty	F120E002		Always on		0011
17	Breaker position middle	F120E003		Always off		0011
18	Object state (faulty change detect)	F120V001	USINT	Change detect	PDE CO 0805	0011

Row 1, 7 and 13 define the signals and the corresponding database objects connected to these points on the Modbus protocol.

Rows 2 to 5 define the events used to update the Modbus data point defined at row 1 and the associated change detect data point (row 6).

Row 6, 12 and 18 define associated momentary change detect points (see Section “Digital inputs” on page 16).

Rows 8 to 11 define the events used to update the Modbus data point defined at row 1 and the associated change detect data point (row 12).

Rows 14 to 17 define the events used to update the Modbus data point defined at row 1 and the associated change detect data point (row 18).

The same information is also accessible from the digital input points of the same point numbers as these coils.

5.4. Analog data - mapping into holding registers (and input registers)

This example uses the MEVO3A and MECU3A function blocks.

	Name	DB Name	Data type	Conversion type	Point (HEX)	Use (bin)
1	Current IL1 in amperes (0.0...20000.0 A)	F200I001	REAL	No operation	PDB HR 0000	1011
2	Voltage U1 or U12 in kV (0.00...999.99 kV)	F204I001	REAL	Scale by 10	PDB HR 0001	1011
3	Input value MEAI (-10000.00000...10000.00000) assumed range for measured signal and scaling -10.000...10.000	F231I001	REAL	Scale by 1000	PDB HR 0002	1011

Row 1 defines the measured phase 1 current as a read-only holding register. The floating-point value is converted into a 16-bit integer without scaling. The resolution is 1A.

The same information is also accessible from the input register of the same point number as this holding register.

Row 2 defines the measured phase 1 voltage as a read-only holding register. The floating point value is converted into a 16-bit integer and scaled by 10. The resolution is then 0,1 kV.

The same information is also accessible from the input register of the same point number as this holding register.

Row 3 defines the measured signal as a read-only holding register. The scaling has to be chosen according to the expected range. For the expected range -10.000...10.000 should be scaled by 1000 and range -10000...10000 should not be scaled.

The same information is also accessible from the input register of the same point number as this holding register.

5.5. Control points: mapping into coils (also into digital inputs for read-only access)

This example uses the COCB1 function block.

	Name	DB Name	Data type	Conversion type	Point (HEX)	Use (bin)
1	Direct open	F120V004	BOOL	None	CTL CO 2000	0001
2	Direct close	F120V005	BOOL	None	CTL CO 2001	0001
3	Select open	F120V006	BOOL	None	CTL CO 2002	0001
4	Select close	F120V007	BOOL	None	CTL CO 2003	0001
5	Execute	F120V010	BOOL	None	CTL CO 2004	0001
6	Cancel	F120V011	BOOL	None	CTL CO 2005	0001

Rows 1 to 6 define write-only data points mapped into coils. In response to a read request, a zero value is returned.

5.6. Mapping of the device clock

	Name	DB Name	Data type	Conversion type	Point (HEX)	Use (bin)
1	Year	DEVCLOCK	TIME_YR	Timer	DDE HR 3000	0101
2	Month	DEVCLOCK	TIME_MON	Timer	DDE HR 3001	0101
3	Day	DEVCLOCK	TIME_DAY	Timer	DDE HR 3002	0101
4	Hour	DEVCLOCK	TIME_HR	Timer	DDE HR 3003	0101
5	Minute	DEVCLOCK	TIME_MIN	Timer	DDE HR 3004	0101
6	Second	DEVCLOCK	TIME_SEC	Timer	DDE HR 3005	0101
7	Tens of milliseconds	DEVCLOCK	TIME_10MS	Timer	DDE HR 3006	0101

Rows 1 to 7 define all fields of the device clock. Values can be read and written, and can be accessed from the application (system) on request.

The same information is also accessible for read-only access from the input registers of the same point numbers as these holding registers.

Time synchronization should be configured to use an external binary input. For more information, see the REF 54_ or RET 54_ Technical Reference Manual. (See “Related documents” on page 5.)

ABB Oy
Distribution Automation
P.O. Box 699
FI-65101 Vaasa
FINLAND
Tel. +358 10 22 11
Fax. +358 10 224 1094
www.abb.com/substationautomation

1MRS755238 EN 07.2005