MBUS User Manual

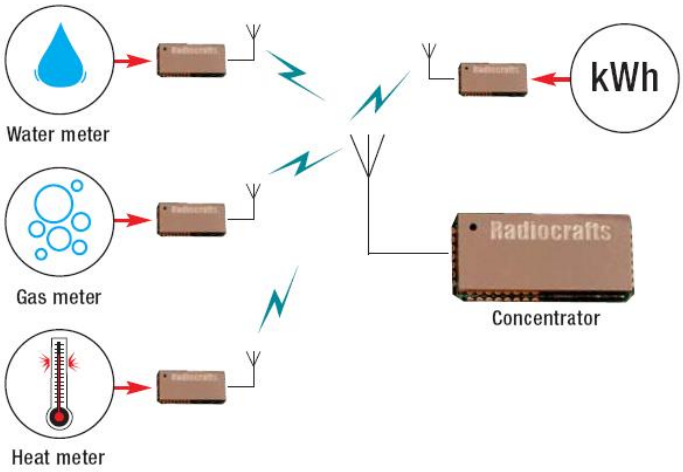

Table of contents

Introduction

This User Manual describes the embedded protocol of the Wireless MBUS Modules from Radiocrafts. The MBUS firmware is available as different feature sets targeting specific applications. The hardware platform is the same for all solutions, and the different feature sets available are listed in the table below. Detailed information on how to use the different feature sets is found in this User Manual. For additional information about the Wireless M-Bus packet structure for NTA 8130 compliance is described in Application Note 011 and is available on request.

Feature List	Feature set	
	RC1180-MBUS1	RC1180-MBUS2
General	Basic wireless M-bus functions	Added features for NTA 8130 compliance
Network role	Master or Slave	Master or Slave
Modes	S1, S2, T1, T2, R2	S1, S2, T1, T2, R2
Encryption	No, must be handled externally	AES according to NTA 8130
Installation mode	No, Must be handled externally	Yes, according to NTA 8130
Filter function	No, receives any MBUS packet. Filtering must be handled externally	Master only receives messages from installed/registered meters
Automatic acknowledge in T2	No, must be handled externally	Yes, according to NTA8131

The command set used to configure the MBUS modules are different for each feature set and an overview is found in the appendixes.

Network Topology

A Wireless M-Bus supported metering system normally consists of a number of heat-, gas-, water and/or electricity meters which reports their meteorological readings to a concentrator. The concentrator acts as the master in the system while the meters are slaves.

The Radiocrafts Wireless M-Bus family of modules RC11xx-MBUSx can be configured to have a role as either master or slave. The slave contains a unique address, and when sending a meter reading this address is added to the wireless message. The message from a slave does not contain any master address (MBUS2, NTA 8130) but the master module within range will receive the message, and based on the slave address it will decode the message and send the data on its serial interface to the host.

In two-way communication modes, the received message can be acknowledged immediately after a slave transmission. The master then returns an addressed message to the slave using the address field originally received from the slave.

Wireless M-Bus Embedded Protocol

Basic functionality
The module offers a buffered packet radio acting as a Wireless M-Bus modem. The module contains a fully embedded protocol supporting EN13757-4:2005 modes:

- Stationary mode S (S1, S1-m, S2)
- Frequent transmit mode T (T1 and T2)
- Frequent receive mode R2

The mode is configurable by the MBUS_MODE parameter.

The required M-Bus mode is configured by setting the module in configuration mode and entering appropriate UART commands. The following modes are supported:

S1/S2-mode:
Set MBUS_MODE = 0
Set PREAMBLE_LENGTH = 0 (for short preamble) or 1 (for long preamble)
The RF channel (channel 11) and data rate (32.768 kchip/s) are set internally in the module according to the S mode, and will override any settings in the RF_CHANNEL and RF_DATA_RATE configuration registers. This setting can also be used for T2 mode slave receive and master transmit.

T1-mode:
Set MBUS_MODE = 1
The RF channel (channel 12), data rate (100 kchip/s) and preamble length are set internally in the module according to the T mode, and will override any settings in the RF_CHANNEL, RF_DATARATE and PREAMBLE_LENGTH configuration registers. This setting can also be used for T2 mode slave transmit and master receive.

T2-mode:
Set MBUS_MODE = 2
Set NETWORK_ROLE = 0 or 1
The RF channel (channel 11 or 12), data rate (32.768 or 100 kchip/s) and preamble length are set internally in the module according to the T2 mode and the selected Network Role, either being a Slave (NETWORK_ROLE = 0) or a Master (NETWORK_ROLE = 1), and change according to receive/transmit. It will override any setting in the RF_CHANNEL configuration register.

R2-mode:
Set RF_CHANNEL = 1-10
Set MBUS_MODE = 4
The data rate (4.8 kchip/s) and preamble length are set internally in the module according to the R mode.

The module supports automatic generation of the Wireless M-Bus frame, i.e.;

- Preamble (header + synchronisation)
- Adding the first block (C-field and address/manufacturing ID)
- CRC
- Postamble

The RF signal is Manchester coded or "3 out of 6" coded for increased signal integrity.

The default M-Bus mode is entered and stored in the modules' non-volatile memory (MBUS_MODE). The M-Bus mode can also be changed using the 'G' command. Using the 'G' command, the value is not stored in non-volatile memory. To do a permanent change, use the 'M' command. The 'G' command should be used for frequent change of mode, to prevent excessive writing to the flash-based non-volatile memory.

The default C-field is entered and stored in the modules' non-volatile memory (CONTROL_FIELD). The C-field can also be changed using the 'F' command. Using the 'F' command, the value is not stored in non-volatile memory. To do permanent change, use the 'M' command.

The default Manufacturer ID and unique meter Address is entered and stored in the modules' non-volatile memory. The destination address (or module address) can also be changed using the 'T' command. Using the 'T' command, the address is not stored in non-volatile memory. To do a permanent change, use the 'M' command. MBUS2 sets the destination address automatically based on the last received message.

The module has an internal buffer and transmits application data as soon as the whole packet is received based on the packet length (first byte of the application frame). The module also has a timeout feature that will empty the input buffer in case of false data packets. Max total payload is 192 bytes, or 255 including the header in the first block.

Sleep mode can be entered via an UART command and wake-up is triggered on UART traffic (one FFh byte). Sleep mode can also be entered automatically after a transmission (configurable by SLEEP_MODE).

The module acts as a buffered packet radio, hence all data to be sent is stored in the module before they are transmitted by the RF circuitry. Likewise, when data is received they are stored in the module before they are sent to the host. This allows the communication controller to add address information, CRC and encryption during transmission, and to do error check and decryption of the received data.

The Module has an UART interface that is used for both Wireless M-Bus packet data and module configuration.

UART Interface for Wireless M-Bus packet handling
The host use the UART Interface to send and receive Wireless MBUS data. The UART packet format can be changed in the configuration mode.

When the Module receives a Wireless M-Bus packet over RF it will send the packet over the UART interface on the TXD Line. When the host MCU wants to transmit a Wireless M-Bus packet over the RF, it must send the packet through the UART Interface on the RXD line.

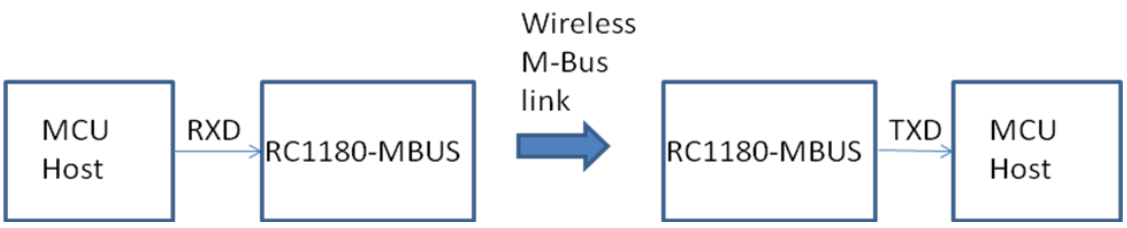

Figure 1: UART interface overview

Frame format for transmitting data
The data frame for the UART RXD pin (input for transmitting a Wireless M-Bus packet) is built like this:

Figure 2: UART interface packet transmission (RXD pin)

L is the length (not including the length byte itself), followed by the application data with the CI byte first. CI is the Control Information byte. The application data typically contains the application header, and data points with VIF and DIF codes. The application data can also be SML of DMLS.

The HEADER and C-field (and adjusted L value) is added to Wireless M-Bus packet automatically by the module before transmitting over RF and both can be changed in configuration mode.

Frame format for receiving data
The data frame for the UART TDX pin (Output for received Wireless M-Bus packets) is built like this:

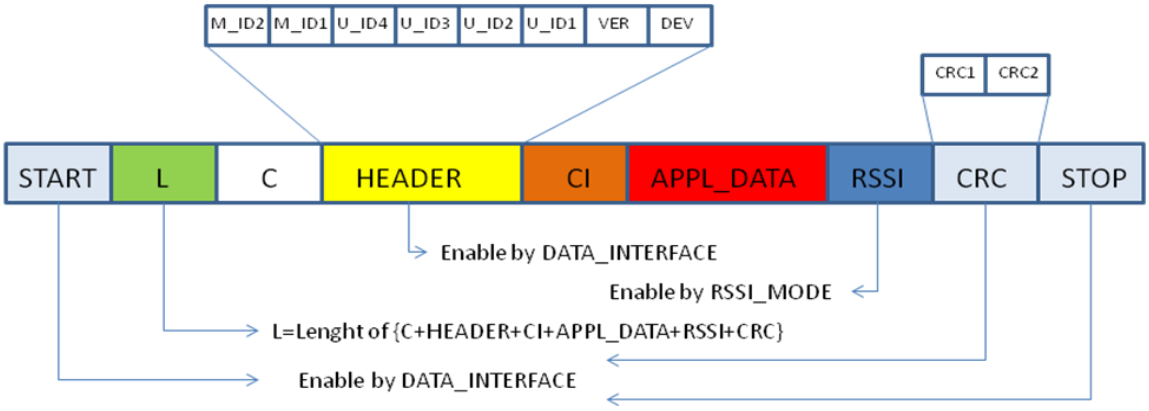

Figure 3: UART interface packet reception (TXD pin)

Data in blue and yellow are optional output parts of the UART message and can be enabled in configuration mode by the DATA_INTERFACE and RSSI_MODE configuration parameters.

L is the length byte and is always present. It does not include itself or the START/STOP bytes, but will include RSSI and CRC if enabled.

When setting DATA_INTERFACE = 1, the received HEADER will not be sent on the UART (typically used on a slave). However, to be able to notify the external application when an Acknowledgement is received (“empty” frame), a special string can be used. By setting DATA_INTERFACE = 3, the two byte string 00:E5h (i.e. L = 0) will be sent on the UART when an empty acknowledge frame is received.

Application data (CI + APPL_DATA) is always present (except when only a HEADER is transmitted).

For host applications using a UART buffer the timing information used for parsing could be lost. In this case a start and stop byte can be used. Setting DATA_INTERFACE = 4 will add a

START byte (68h) and a STOP byte (16h) to the message. This is only used for the module-to-host communication direction (TXD). Setting DATA_INTERFACE = 8 will add a two byte CRC checksum, and DATA_INTERFACE = 0Ch will add START/STOP bytes and CRC. The CRC is sent MSByte first.

The RSSI value is added when RSSI_MODE = 1.

UART Interface for Module Configuration

The configuration of the module can be changed in-circuit from the host during operation, at the time of installation of the equipment, at the manufacturing test, or even as a stand-alone module. The configuration is changed by sending commands on the UART interface after the module is set in configuration mode. The configuration mode is entered by sending 00h to the module, or by asserting the CONFIG pin (set low).

In configuration mode the module will respond by sending a '>' prompt on the TXD pin. This indicates that the module is ready to receive commands. The CONFIG pin (if used) can then be de-asserted. Note that the CONFIG pin must be de-asserted *before* the Exit command ('X') is sent to the module in order to return to normal operation.

After a command is executed, the module responds with the '>' prompt character again, indicating it is ready for a new command. Do not send a new command before the '>' prompt is received. The time required to execute a command can vary depending on the command (see the Timing Information section). There is no '>' prompt after the 'X' exit command.

The parameters that are set by dedicated configuration commands ('C', 'P' etc) take immediate effect after returning to normal operation (IDLE), but will not be stored in non-volatile memory and will be lost in case the supply power is turned off or if the module is reset. These parameters are for example the radio channel and output power.

Permanent changes of parameters can be done by writing to the configuration memory using the memory command 'M'. These are for example *default* radio channel, *default* output power and M-Bus mode, see the Configuration Memory section for details.

The flow diagram bellow illustrates how to use the UART interface to enter configuration mode, change configuration parameters and return to IDLE mode.

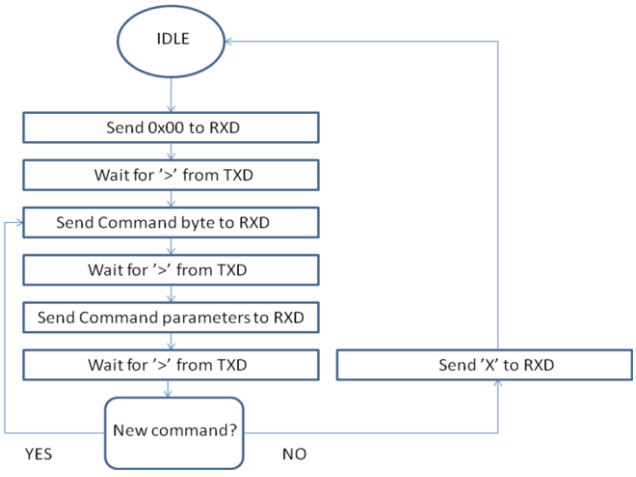

Figure 4: Configuration mode flow diagram

UART Timing Information
A UART byte consist of one start bit, 8 data bits, and one stop bit. In configuration mode a command to prompt reply will looks like this:

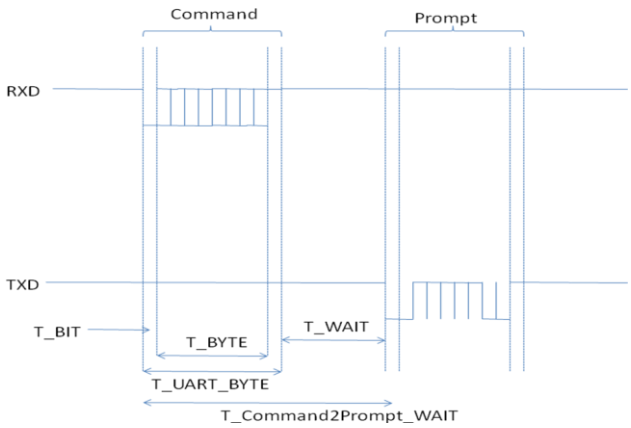

Figure 5: UART Command and prompt

The command-to-prompt wait time (T_Command2Prompt_WAIT) is different from command to command and values are shown in the timing table for each MBUS feature set.

The IDLE state is the normal state where the module both searches for preamble on the RF and wait for a character to be received on the UART. RXD is the state when receiving characters from the host filling up the internal buffer. TX state is when the data is transmitted on the air. RX state is when data is received from the air after preamble detection. TXD is the state where the received data is sent to the host on the UART.

CONFIG is the configuration mode, the state entered by sending 00h or asserting the CONFIG pin and is entered during parameter configuration, while MEMORY CONFIG is the sub-state entered by the ‘M’ command where the non-volatile configuration memory is being programmed. Note the limitation on maximum number of write cycles using the ‘M’ command, see Electrical Specifications.

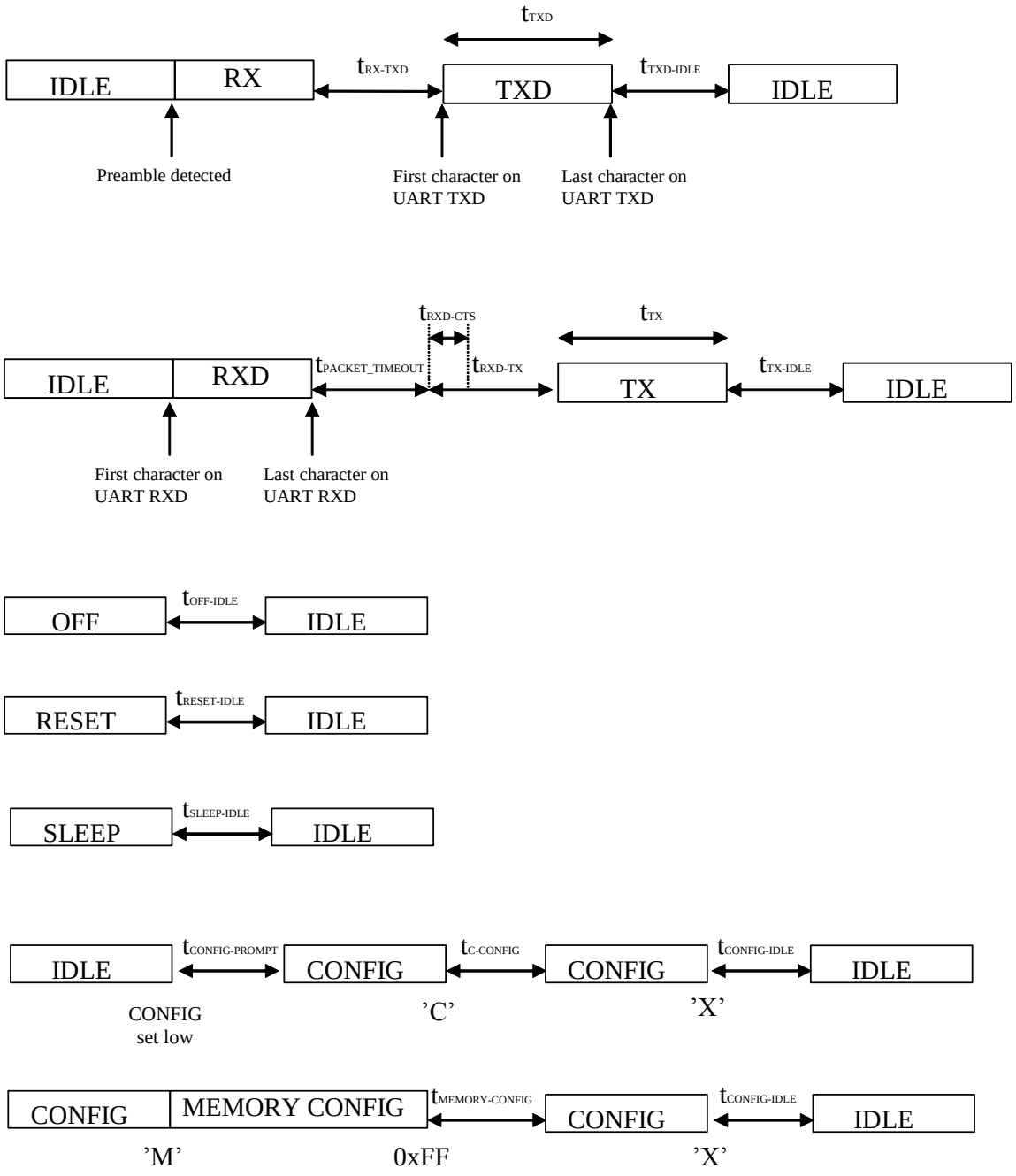

Figure 6: UART timing diagram

Timing values are shown in the timing table for each MBUS feature set.

Power Management
The module can be set in SLEEP mode in order to reduce the power consumption.

The low power SLEEP mode is manually entered by using the SLEEP command 'Z'. It is also possible to configure the module to enter SLEEP automatically after a message has been transmitted (SLEEP_MODE=1). In SLEEP mode the module will not receive or detect incoming data, neither from the host (UART port) nor from the air. The module is awakened from the SLEEP mode by sending a wake-up byte on the UART RXD line. The wake-up byte should be FFh (use a UART Baud rate > 4.8 kBd due to a maximum pulse length requirement). After the module has woken up (see Timing Information) it is ready to receive data on the UART or from the air. The SLEEP command can be used for both Master and Slave.

All configuration settings and RAM values are retained during SLEEP.

If the module is shut completely off (supply power turned off), all configuration settings in non-volatile memory is restored, but values in RAM are overwritten with default settings.

MBUS1 Description

MBUS1 Timing table

The table below shows the timing information for the module when changing between different operating states. Timing symbols are according to figure 5 and 6.

Symbol	Value	Description / Note
t_{RX-TXD}	180 us	Time from last byte is received from the air until first character is sent on the UART
t_{TXD}	Min 590 us	t_{TXD} = # bytes received x 590 us/char (10 bits at 19.2 kBd + 70 us delay per character)
$t_{TXD-IDLE}$	900 us	Time from last character is sent on the UART until module is in IDLE mode (ready for RXD and RX)
$T_{RXD-CTS}$	20 us	Time from last character is received by the UART (including any timeout) until CTS is activated
t_{RXD-TX}	960 us	Time from last character is received by the UART (including any timeout) until the module sends the first byte on the air.
$T_{TX-IDLE}$	960 us	Time from last character is sent on the air until module is in IDLE mode (ready for RXD and RX)
$t_{OFF-IDLE}$	3.2 ms	
$t_{RESET-IDLE}$	3.0 ms	
$t_{SLEEP-IDLE}$	1.3 ms	
$t_{CONFIG-PROMPT}$	60 us	Time from 00h / CONFIG pin is set low until prompt (‘>’)
$T_{C-CONFIG}$	1.1 ms	Delay after channel-byte is sent until prompt (“>”).(For other volatile memory commands there is no delay but immediate prompt)
$T_{G-CONFIG}$	1.1 ms	Delay after new M-Bus mode-byte is sent until prompt (‘>’). (For other volatile memory commands there is no delay but immediate prompt)
T_{WAIT}	1.55 ms (M command) 24 us (all other commands)	Delay from stop bit of the command byte to start bit of the prompt reply. See figure 5 for details.
$t_{MEMORY-CONFIG}$	31 ms	In this period the internal flash (non-volatile memory) is programmed. *Do not reset, turn the module off, or allow any power supply dips in this period as it may cause permanent error in the Flash configuration memory. After the last command parameter byte the host should wait for the ‘>’ prompt before any further action is done to ensure correct re-configuration.*
$T_{CONFIG-IDLE}$	1.1 ms	End of ‘X’ to IDLE
t_{TX}	3.6 ms	TX time for T1 mode when Length=1 on the UART. Preamble, sync, CRC and MBUS address field added internally.

MBUS1 Configuration Commands
A list of commands is shown in the table below. Commands must be sent as ASCII characters or their corresponding binary value. All arguments must be sent as binary values to the module (not as ASCII representation for hex or decimal).

Parameter	Command	Argument in hex (decimal)	Note
Channel	'C' – 0x43	0x01-0x0C (1-10) Apply for R mode only	Data is stored in volatile memory only.
C-field	'F' – 0x46	0x00-0xFF (0-255)	Data is stored in volatile memory only.
M-Bus mode	'G' – 0x47	0x00-0x04 (0-4) 0: S1 1: T1 2: T2 3: Reserved 4: R2	Data is stored in volatile memory only.
Memory configuration	'M' – 0x4D	(Address, Data): see list of parameters below. 0xFF exits memory configuration.	Used to enter memory configuration menu. Parameters changed are stored in non-volatile memory.
Output power	'P' – 0x50	0x01-0x05 (1-5)	Data is stored in volatile memory only.
Quality Indicator	'Q' – 0x51	Returns one byte indicating the signal quality	Based on bit errors in preamble and synch word
Signal Strength (RSSI)	'S' – 0x53	Returns one byte indicating the signal strength of a detected signal or a valid packet.	If a valid packet has been received when in configuration mode, it will return the RSSI of the last received packet.
Destination / module address	'T' – 0x54	8 bytes; MAN_ID2 (Second manufacturer code), MAN_ID1 (First manufacturer code), ID4, ID3, ID2, ID1, VER (Version), DEV (Device Type),	Data is stored in volatile memory only.
Exit command	'X' – 0x58	(none)	Exit to normal operation mode. All changes of parameters take effect.
Sleep mode	'Z' – 0x5A	(none)	Exit sleep mode by sending 0xFF on UART RXD pin
Test mode 0	'0' – 0x30	(none)	List all configuration memory parameters
Test mode 1	'1' – 0x31	(none)	TX carrier
Test mode 2	'2' – 0x32	(none)	TX modulated signal PN9 sequence
Test mode 3	'3' – 0x33	(none)	TX Off, RX mode

Note: ASCII characters are written as 'X', hexadecimal numbers are written like 0x00, and decimal numbers are written like 10 throughout the text. A table of ASCII characters and their respective hex and decimal values are found in the Appendix.

Any invalid command will be ignored and the '>' prompt will be re-sent.

If Test mode 1 or 2 is used, it is important to enter Test mode 3 before exiting the configuration mode ('X') in order to ensure proper operation in normal mode.

Example:
To select RF channel 3, send the follow sequence after asserting the CONFIG line and the '>' prompt is received:

Command	**Hex**	**Response**	**Comment/Note**
Enter	0x00	'>'	Or assert CONFIG pin De-assert CONFIG after '>' prompt
'C'	0x43	'>'	
3	0x03	'>'	Wait for '>' prompt
[A new command could be issued here]			
'X'	0x58	(none)	Module returns to IDLE state

Note that the CONFIG line must be de-asserted after the first '>' prompt was received, but before the 'X' command.

MBUS1 Configuration Memory

The table below shows the complete list of configurable parameters stored in non-volatile memory. These values can be changed using the 'M' command. All addresses and arguments must be sent as binary values to the module (not as ASCII representation for hex or decimal).

Parameter	Description	Address hex	Argument dec	Factory setting hex (dec)	Comment
Radio configuration					
RF_CHANNEL	Default RF channel for R mode only	0x00	1-10	0x01 (1)	See data sheet for channel frequencies. Only used for R mode.
RF_POWER	Default RF output power	0x01	1-5	0x05 (5)	See data sheet for output power levels.
MBUS_MODE	M-Bus mode	0x03	0-4 0: S1 1: T1 2: T2 3: Reserved 4: R	0x01 (1)	Use 'G' command to change value in volatile memory only
SLEEP_MODE	Sleep mode	0x04	0: Disable Sleep 1: Enable Sleep	0x00 (0)	When enabled the module enter Sleep mode after transmission
RSSI_MODE	Append RSSI to received data	0x05	0:Disabled 1: Enabled	0x00 (0)	When enabled the RSSI value is appended to the received data
Radio packet configuration					
PREAMBLE_ LENGTH	Short of long preamble in S mode only	0x0A	0x00 (0): Short 0x01 (1): Long	0x00 (0)	Preamble (header) length, apply for S mode only
Medium access, addressing and network management					
NETWORK_ROLE		0x12	0x00 (0): Slave/Meter 0x01 (1): Master/Concentrator	0x00 (1)	
M_ID1	Manufacturer ID, first byte	0x19	0x00-0xFF (0-255)	0x0C (12)	
M_ID2	Manufacturer ID, second byte	0x1A	0x00-0xFF (0-255)	0xAE (174)	
U_ID1	Unique ID, first byte	0x1B	0x00-0xFF (0-255)	0x12 (18)	
U_ID2	Unique ID, second byte	0x1C	0x00-0xFF (0-255)	0x34 (52)	
U_ID3	Unique ID, third byte	0x1D	0x00-0xFF (0-255)	0x56 (86)	
U_ID4	Unique ID, forth byte	0x1E	0x00-0xFF (0-255)	0x78 (120)	
VER	Version	0x1F	0x00-0xFF (0-255)	0x01 (1)	
DEV	Device	0x20	0x00-0xFF (0-255)	0x07 (7)	
Data and configuration interface, UART Serial Port					
UART_BAUD_RATE	Baud rate	0x30	0x00: Not used 0x01: 2400 0x02: 4800 0x03: 9600 0x04. 14400 0x05: 19200 0x06: 28800 0x07: 38400 0x08: 57600 0x09: 76800 0x0A: 115200 0x0B: 230400	0x05 (5)	BE CAREFUL IF CHANGING AS HOST MAY LOOSE CONTACT WITH MODULE! Does not take effect until module is re-booted / reset.

UART_FLOW_CTRL	UART flow control	0x35	0: None 1:CTS only 3:CTS/RTS 4:RXTX(RS485)	0x00 (0)	
DATA_INTERFACE	Data interface	0x36	0x00: MBUS packet with ID and address 0x01: Application data only 0x02: Reserved 0x03: Application data only with ack (00:3Eh) 0x04: Add start/stop byte 0x08: Add CRC 0x0C: Add start/stop byte and CRC	0x00 (0)	Sets receiver data format. First byte is always packet length (except when using start byte)
CONTROL_FIELD	C-field	0x3B	0x00-0xFF (0-255)	0x44 (68)	Use 'F' command to change value in volatile memory only
Exit from memory configuration		0xFF	No argument should be sent		To exit from command mode the 'X' command must be sent after '>' is received.

To make permanent changes to default values and other parameters, the Memory Configuration command 'M' is used. This command should be followed by pairs of byte being the memory address and the new value to be stored at that address. In order to exit the Memory Configuration mode, the 'address' 0xFF must be sent, but without any data argument. Then wait for the '>' prompt while the internal memory is re-programmed (see Timing Information for typical delay). To completely exit from command mode, the normal exit command 'X' must be sent.

Example:
To change the MAN_ID (at address 0x19 and 0x1A) and set it to (100,200) (0x64, 0xC8), send the following sequence:

Command	**Hex**	**Response**	**Comment/Note**
Enter	0x00	'>'	Or assert CONFIG pin De-assert CONFIG after '>' prompt
'M'	0x4D	'>'	Module ready to receive address
0x19	0x19	(none)	
100	0x64	(none)	
0x1A	0x1A	(none)	
200	0xC8	(none)	
[new address could be sent here]			
[new value could be sent here]			
0xFF	0xFF	'>'	Wait for '>' prompt
'X'	0x58	(none)	Module returns to IDLE state

Test mode 0 ('0' command) can be used to list all parameters stored in non-volatile memory. This command can be used to verify and check the module configuration.

MBUS2 Description

MBUS2 Automatic Acknowledge
The Master must reply with an acknowledge message within 3 ms after a received Access Demand, if further communication shall take place. To meet this timing requirement, the module has built-in automatic acknowledge support. Use the Acknowledge flag (set using the A –command) to indicate which slave shall be acknowledged at the next access. The flag is automatically cleared but can also be cleared manually with the A-command.

MBUS2 Automatic Addressing
When the Master receives a message from a slave, this slave HEADER will be used as the HEADER for the next transmission from the master. If, for example during installation, messages are received from several meters, the last message received will be the default address. This feature can be overridden by using the 'T' command.

MBUS2 Automatic Sleep
The Slave has special support for automatic sleep after data transmission. If automatic SLEEP is enabled (SLEEP_MODE = 1), the module will automatically go to sleep in a configurable time after data transmission, if acknowledge is not received. The receiver timeout is configured by RX_TIMEOUT. The automatic sleep is only done after an Access Demand Install message (C-field is 46h in T1 and 06h in T2), or a regular Access Demand message (C-field is 44h in T1 and 48h in T2), is sent.

MBUS2 Installation and Binding
The module (Master) can be set in Installation Mode using the "I" command. When the module is in Installation Mode it will accept all Access Demand Install messages (C-field is 46h in T1 and 06h in T2).

Slaves can be bound to a Master by registering their addresses in the Address Register. This is done by using the "B" (Bind) command followed by a register number (1-8) and an 8 bytes slave address. Thus, a maximum of 8 meters can be bound to one Master for the MBUS2 feature set (meeting NTA 8130 request for minimum 4 meters).

Note; the host must know which registers are used and which are free at any time.

MBUS2 Encryption
The module supports AES-128 encryption. When a slave is registered into the masters address register, the master host should request a new encryption key from the utility data base. The new key is specific for each slave and related to the slave equipment ID or unique address. The new key should be provided in two versions; plain and encrypted using the slave's default key. That is, the utility needs to keep a register with default keys linked to each meter.

The master host should send the new encrypted key to the slave. And the slave host should configure this new key into the module using the 'K' command, followed by 16 bytes (the encrypted key). The slave module will automatically de-crypt the new key using its default key.

The master host should then send the new (plain) key to the master module using the 'K' command, followed by the register location number, and the 16 bytes (the new key). The register location number must correspond to the address register location for that slave.

The ENCRYPT_FLAG and DECRYPT_FLAG parameters are used to enable / disable the encryption when transmitting and receiving messages. The 8 bit values are interpreted as bit maps corresponding to the 8 address registers, LSB being register 1.

For a message to be encrypted, the encryption flag for the particular slave must be set, and the CI-field and Signature field sent to the module must be according to the standard for encryption to take place. The module will do byte stuffing if required to get a full 16 byte encryption block. Only CI-fields 0x5A, 0x5B and 0x72 allows encryption. The Signature field must be encryption mode 0x04 or 0x05 according to NTA 8130.

When using Signature field 0x05, the application must add the two encryption verification bytes (0x2F) after the header. The Initialization Vector for the encryption is extracted from the long header (for CI-fields 0x5B and 0x72). For the short header (CI-field 0x5A) the Initialization Vector is partly from the MAC header (destination address) and the short application header.

The Access Counter byte in the application header is used by the encryption, and the host application must increment the counter in order to avoid repetitive messages.

For a message to be decrypted the decryption flag for the particular slave must be set, and the CI-field and Signature field must be according to the standard for encryption to take place.

In the slave, only LSB is used as a flag for encryption/decryption.

The ‘D’ and ‘E’ commands are used to set decryption and encryption flags without storing in non-volatile memory. The value following the D and E commands is interpreted as bit maps corresponding to the 8 address registers.

To test the encryption feature you need to have a valid key set for the master and the slave. In addition you need to send a valid UART frame into the module in order for the internal encryption and decryption feature to be activated on this message. The RCTools PC software from Radiocrafts (MBUS_CCT and MBUS_DEMO) can be used to configure key sets and send and receive encrypted messages.

Example of a key set:
Master Key: 0x00 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99 0xAA 0xBB 0xCC 0xDD 0xEE 0xFF
Slave Key: 0x0A 0x90 0xE5 0xB7 0x4D 0x28 0x07 0xA6 0x51 0xF6 0x9A 0xC0 0x89 0x6A 0x09 0xF6
Use factory default for Init vector and Default key in the configuration memory.

Example of UART RXD frames that enable encryption:

Test packet A: No filling byte
Slave TX message: C=6, L=2D, CI=72,
Data=78563412AE07010701002004 2F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFF

Slave TX message: C=6, L=25, CI=5A,
Data=0200200 42F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFF

Slave TX message: C=6, L=2D, CI=5B,
Data=78563412AE07010703002004 2F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFF

Test packet B: Adding filling bytes
Slave TX message: C=6, L=1E, CI=72,
Data=78563412AE07010704001104 2F2FAABBCCDDEEFFAABBCCDDEEFFAABBCC

Slave TX message: C=6, L=17, CI=5A,
Data=0500120 42F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDD

Slave TX message: C=6, L=2E, CI=5B,
Data=78563412AE07010706002104 2F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAA

Test packet C: Adding filling bytes and un-encrypted bytes at the end of the packet.
Slave TX message: C=6, L=22, CI=72,
Data=78563412AE07010707001104 2F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCAAAAAAAA

Slave TX message: C=6, L=1B, CI=5A,
Data=0800120 42F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDDAAAAAAAA

Slave TX message: C=6, L=2B, CI=5B,
Data=78563412AE07010709001A 042F2FAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAABBCCDDEEFFAAAAAAAA

Green: Number of encrypted bytes including the two 0x2F bytes in the start of the encryption part
Read: Second block Header (un-encrypted)
Blue: Signature field. To enable encryption this field must be 04 (NTA3130 encryption) or 05 (NTA8130 encryption harmonized to OMS).
Purple: Encrypted part of UART message. Internal encryption and not encrypted on UART
Black: Un-encrypted part of message

Slave step-by-step guide to send an encrypted message to a Master:
1. Enter configuration mode
2. Send the slave key to the module using the ‘K’ command and ‘Slave Key’ as command parameter.
3. Enable Encryption using the ‘E’ command and ‘0x01’ as command parameter
4. Leave configuration mode using the ‘X’ command
5. Send one of the valid test packets above.

This packet will now be encrypted over the RF link.

Master step-by-step guide to receive an encrypted message from a Slave:
1. Enter configuration mode
2. Install the Slave to this Master using the ‘B’ and ‘Address Register’ as command parameter followed by the slave address. Address register is 1-8.
2. Send the Master key to module using the ‘K’ command and ‘Address Register’ + ‘Master Key’ as command parameters.
3. Enable Decryption using the ‘D’ command and ‘Address Register’ as command parameter.
4. Leave configuration mode using the ‘X’ command

An encrypted packet from the installed slave will now be decrypted before it is provided on the UART TXD.

Note that you also have the option to use the M command to permanently set the Encryption/Decryption flag in the configuration non-volatile memory, instead of using the ‘D’ and ‘E’ commands. Encryption / decryption will only take place when the signature field indicate mode 0x04 or 0x05. If encryption mode 0x00 is used, the message will not be encrypted / decrypted even if the flags are set.

MBUS2 Timing table
The table below shows the timing information for the module when changing between different operating states. Timing symbol is according to figure 5 and 6.

Symbol	Value	Description / Note
t_{RX-TXD}	180 us	Time from last byte is received from the air until first character is sent on the UART
t_{TXD}	Min 590 us	t_{TXD} = # bytes received x 590 us/char (10 bits at 19.2 kBd + 70 us delay per character)
$t_{TXD-IDLE}$	900 us	Time from last character is sent on the UART until module is in IDLE mode (ready for RXD and RX)
$T_{RXD-CTS}$	20 us	Time from last character is received by the UART (including any timeout) until CTS is activated
t_{RXD-TX}	960 us	Time from last character is received by the UART (including any timeout) until the module sends the first byte on the air.
$T_{TX-IDLE}$	960 us	Time from last character is sent on the air until module is in IDLE mode (ready for RXD and RX)
$t_{OFF-IDLE}$	3.2 ms	
$t_{RESET-IDLE}$	3.0 ms	
$t_{SLEEP-IDLE}$	1.3 ms	
$t_{CONFIG-PROMPT}$	60 us	Time from 00h / CONFIG pin is set low until prompt (“>”)
$T_{G-CONFIG}$	1.1 ms	Delay after channel-byte is sent until prompt (“>”).(For other volatile memory commands there is no delay but immediate prompt)
$T_{G-CONFIG}$	1.1 ms	Delay after new M-Bus mode-byte is sent until prompt (“>”).(For other volatile memory commands there is no delay but immediate prompt)
T_{WAIT}	1.55 ms (B, K and M command) 24 us (all other commands)	Delay from stop bit of the command byte to start bit of the prompt reply. See figure 5 for details.
$T_{MEMORY-CONFIG}$	31 ms	In this period the internal flash (non-volatile memory) is programmed. *Do not reset, turn the module off, or allow any power supply dips in this period as it may cause permanent error in the Flash configuration memory. After the last command parameter byte the host should wait for the ‘>’ prompt before any further action is done to ensure correct re-configuration.*
$T_{CONFIG-IDLE}$	1.1 ms	End of ‘X’ to IDLE
t_{TX}	3.6 ms	TX time for T1 mode when Length=1 on the UART. Preamble, sync, CRC and MBUS address field added internally. Depends on M-Bus mode (T, S, R) and L

MBUS2 Configuration Commands

A list of commands is shown in the table below. Commands must be sent as ASCII characters or their corresponding binary value. All arguments must be sent as binary values to the module (not as ASCII representation for hex or decimal).

Parameter	Command	Argument in hex (decimal)	Note
Acknowledge	'A' – 0x41	One byte bitmap, address register 1 is LSB.	Sets bitmap for acknowledge from Master. 0x00 will clear all flags.
Bind	'B' – 0x42	Register number (1-8) followed by 8 byte address (same order as for Destination/module address)	Used to bind slaves to master. Data stored in non-volatile memory.
Channel	'C' – 0x43	0x01-0x0C (1-10) Apply for R mode only	Data is stored in volatile memory only.
Decrypt	'D' – 0x44	One byte bitmap, address register 1 is LSB.	Sets bitmap for decryption of data. 0x00 will clear all flags.
Encrypt	'E' – 0x45	One byte bitmap, address register 1 is LSB.	Sets bitmap for encryption of data. 0x00 will clear all flags.
C-field	'F' – 0x46	0x00-0xFF (0-255)	Data is stored in volatile memory only.
M-Bus mode	'G' – 0x47	0x00-0x04 (0-4) 0: S1 1: T1 2: T2 3: Reserved 4: R2	Data is stored in volatile memory only.
Install	'I' – 0x49	0: Normal operation 1: Install mode 2: Accept all messages	In install mode messages with C-field = 06h and 46h are accepted. Use in Master only.
Key register	'K' – 0x4B	Slave: 16 byte key. Master: Register number (1-8) followed by 16 byte key	Used to set encryption key. Data stored in non-volatile memory.
Memory configuration	'M' – 0x4D	(Address, Data): see list of parameters below. 0xFF exits memory configuration.	Used to enter memory configuration menu. Parameters changed are stored in non-volatile memory.
Output power	'P' – 0x50	0x01-0x05 (1-5)	Data is stored in volatile memory only.
Quality Indicator	'Q' – 0x51	Returns one byte indicating the signal quality of the last received packet	Based on bit errors preamble and synch word
Signal Strength (RSSI)	'S' – 0x53	Returns one byte indicating the signal strength of a detected signal or a valid packet.	If a valid packet has been received when in configuration mode, it will return the RSSI of the last received packet.
Destination / module	'T' – 0x54	8 bytes; M_ID2,	Data is stored in volatile memory only.

address		M_ID1, U_ID4, U_ID3, U_ID2, U_ID1, VER (Version), DEV (Device Type),	
Exit command	'X' – 0x58	(none)	Exit to normal operation mode. All changes of parameters take effect.
Sleep mode	'Z' – 0x5A	(none)	Exit sleep mode by sending 0xFF on UART RXD pin
Test mode 0	'0' – 0x30	(none)	List all configuration memory parameters
Test mode 1	'1' – 0x31	(none)	TX carrier
Test mode 2	'2' – 0x32	(none)	TX modulated signal PN9 sequence
Test mode 3	'3' – 0x33	(none)	TX Off, RX mode

Note: ASCII characters are written as 'X', hexadecimal numbers are written like 0x00, and decimal numbers are written like 10 throughout the text. A table of ASCII characters and their respective hex and decimal values are found in the Appendix.

Any invalid command will be ignored and the '>' prompt will be re-sent.

If Test mode 1 or 2 is used, it is important to enter Test mode 3 before exiting the configuration mode ('X') in order to ensure proper operation in normal mode.

Example:
To select RF channel 3, send the follow sequence after asserting the CONFIG line and the '>' prompt is received:

Command	**Hex**	**Response**	**Comment/Note**
Enter	0x00	'>'	Or assert CONFIG pin De-assert CONFIG after '>' prompt
'C'	0x43	'>'	
3	0x03	'>'	Wait for '>' prompt
[A new command could be issued here]			
'X'	0x58	(none)	Module returns to IDLE state

Note that the CONFIG line must be de-asserted after the first '>' prompt was received, but before the 'X' command.

MBUS2 Configuration Memory

The table below shows the complete list of configurable parameters stored in non-volatile memory. These values can be changed using the 'M' command. All addresses and arguments must be sent as binary values to the module (not as ASCII representation for hex or decimal).

Parameter	Description	Address hex	Argument dec	Factory setting hex (dec)	Comment
Radio configuration					
RF_CHANNEL	Default RF channel for R mode only	0x00	1-10	0x01 (1)	See data sheet for channel frequencies. Only used for R mode.
RF_POWER	Default RF output power	0x01	1-5	0x05 (5)	See data sheet for output power levels.
MBUS_MODE	M-Bus mode	0x03	0-4 0: S1 1: T1 2: T2 3: Reserved 4: R	0x01 (1)	Use 'G' command to change value in volatile memory only
SLEEP_MODE	Sleep mode	0x04	0: Disable Sleep 1: Enable Sleep	0x00 (0)	When enabled the module enter Sleep mode after transmission. Delay set by RX_TIMEOUT
RSSI_MODE	Append RSSI to received data	0x05	0:Disabled 1: Enabled	0x00 (0)	When enabled the RSSI value is appended to the received data
Radio packet configuration					
PREAMBLE_ LENGTH	Short of long preamble in S mode only	0x0A	0x00 (0): Short 0x01 (1): Long	0x00 (0)	Preamble (header) length, apply for S mode only
Medium access, addressing and network management					
NETWORK_ROLE		0x12	0x00 (0): Slave/Meter 0x01 (1): Master/Concentrator	0x00 (0)	
M_ID1	Manufacturer ID, first byte	0x19	0x00-0xFF (0-255)	0x0C (12)	
M_ID2	Manufacturer ID, second byte	0x1A	0x00-0xFF (0-255)	0xAE (174)	
U_ID1	Unique ID, first byte	0x1B	0x00-0xFF (0-255)	0x12 (18)	
U_ID2	Unique ID, second byte	0x1C	0x00-0xFF (0-255)	0x34 (52)	
U_ID3	Unique ID, third byte	0x1D	0x00-0xFF (0-255)	0x56 (86)	
U_ID4	Unique ID, forth byte	0x1E	0x00-0xFF (0-255)	0x78 (120)	
VER	Version	0x1F	0x00-0xFF (0-255)	0x01 (1)	
DEV	Device	0x20	0x00-0xFF (0-255)	0x07 (7)	
Data and configuration interface, UART Serial Port					
UART_BAUD_RATE	Baud rate	0x30	0x00: Not used 0x01: 2400 0x02: 4800 0x03: 9600 0x04. 14400 0x05: 19200 0x06: 28800 0x07: 38400	0x05 (5)	BE CAREFUL IF CHANGING AS HOST MAY LOOSE CONTACT WITH MODULE! Does not take effect until module is re-booted / reset.

			0x08: 57600 0x09: 76800 0x0A: 115200 0x0B: 230400		
UART_FLOW_CTRL	UART flow control	0x35	0: None 1:CTS only 3:CTS/RTS 4:RXTX(RS485)	0x00 (0)	
DATA_INTERFACE	Data interface	0x36	0x00: MBUS packet with ID and address 0x01: Application data only 0x02: Reserved 0x03: Application data only with ack (00:3Eh) 0x04: Add start/stop byte 0x08: Add CRC 0x0C: Add start/stop byte and CRC	0x00 (0)	Sets receiver data format. First byte is always packet length (except when using start byte)
LED_CONTROL		0x3A	0: Disabled 1: Enabled	0x00 (0)	Use to enable LED control for RX and TX mode indication
CONTROL_FIELD	C-field	0x3B	0x00-0xFF (0-255)	0x06 (6)	Use 'F' command to change value in volatile memory only
RX_TIMEOUT		0x3C	0x00-0xFF (0-255)	0x0B (11)	Delay before Sleep mode, n x 0.6 ms
INSTALL_MODE		0x3D	0: Normal mode (accept installed MBUS meters only) 1: Install mode 2: Filter off (accept all MBUS types)	2	
ENCRYPT_FLAG		0x3E		0	Bit mask for encryption, enabled when set
DECRYPT_FLAG		0x3F		0	Bit mask for decryption, enabled when set
DEFAULT_KEY		0x40-0x4F		All 0xFF (255)	
INIT_VECTOR		0x50-0x5F		All 0x00 (0)	
ADDRESS_ID1		0x80-0x87		All 0x00	Address for installed meters.
ADDRESS_ID2		0x88-0x8F		All 0x00	Address for installed meters.
ADDRESS_ID3		0x90-0x97		All 0x00	Address for installed meters.
ADDRESS_ID4		0x98-0x9F		All 0x00	Address for installed meters.
ADDRESS_ID5		0xA0-0xA7		All 0x00	Address for installed meters.
ADDRESS_ID6		0xA8-0xAF		All 0x00	Address for installed meters.
ADDRESS_ID7		0xB0-0xB7		All 0x00	Address for installed meters.
ADDRESS_ID8		0xB8-0xBF		All 0x00	Address for installed meters.
Exit from memory configuration		0xFF	No argument should be sent		To exit from command mode the 'X' command must be sent after '>' is received.

To make permanent changes to default values and other parameters, the Memory Configuration command 'M' is used. This command should be followed by pairs of byte being the memory address and the new value to be stored at that address. In order to exit the Memory Configuration mode, the 'address' 0xFF must be sent, but without any data argument. Then wait for the '>' prompt while the internal memory is re-programmed (See Timing Information for typical delay). To completely exit from command mode, the normal exit command 'X' must be sent.

Example:
To change the MAN_ID (at address 0x19 and 0x1A) and set it to (100,200) (0x64,0xC8), send the following sequence:

Command	**Hex**	**Response**	**Comment/Note**
Enter	0x00	'>'	Or assert CONFIG pin De-assert CONFIG after '>' prompt
'M'	0x4D	'>'	Module ready to receive address
0x19	0x19	(none)	
100	0x64	(none)	
0x1A	0x1A	(none)	
200	0xC8	(none)	
[new address could be sent here]			
[new value could be sent here]			
0xFF	0xFF	'>'	Wait for '>' prompt
'X'	0x58	(none)	Module returns to IDLE state

Test mode 0 ('0' command) can be used to list all parameters stored in non-volatile memory. This command can be used to verify and check the module configuration.

Appendix: MBUS Command list overview

Command list	Feature set	
	RC1180-MBUS1	RC1180-MBUS2
'A' – 0x41	N.A	Acknowledge
'B' – 0x42	N.A	Bind
'C' – 0x43	Channel	Channel
'D' – 0x44	N.A	Decrypt
'E' – 0x45	N.A	Encrypt
'F' – 0x46	C-field	C-field
'G' – 0x47	M-Bus mode	M-Bus mode
'I' – 0x49	N.A	Install
'K' – 0x4B	N.A	Key register
'M' – 0x4D	Memory configuration	Memory configuration
'P' – 0x50	Output power	Output power
'Q' – 0x51	Quality Indicator	Quality Indicator
'S' – 0x53	Signal Strength (RSSI)	Signal Strength (RSSI)
'T' – 0x54	Destination address	Destination address
'X' – 0x58	Exit command	Exit command
'Z' – 0x5A	Sleep mode	Sleep mode
'0' – 0x30	Test mode 0	Test mode 0
'1' – 0x31	Test mode 1	Test mode 1
'2' – 0x32	Test mode 2	Test mode 2
'3' – 0x33	Test mode 3	Test mode 3

Commands in grey are stored in non-volatile memory. The rest of the commands are stored in volatile memory and is lost after a power off or a reset.

Appendix: Configuration Memory Factory Default

Address	MBUS1 factory default Values							
0x00-0x07	0x01	0x05	0x03	0x01	0x00	0x00	0x64	0x00
0x08-0x0F	0x00	0x00	0x00	0xD3	0x91	0xDA	0x80	0x80
0x10-0x17	0x7C	0x00	0x00	0x01	0x00	0x00	0x00	0x00
0x18-0x1F	0x00	0x0C	0xAE	0x12	0x34	0x56	0x78	0x01
0x20-0x27	0x07	0x01	0x01	0x00	0x00	0x00	0x00	0x04
0x28-0x2F	0xFF	0x08	0x00	0x00	0x00	0x00	0x00	0x00
0x30-0x37	0x05	0x08	0x00	0x01	0x05	0x00	0x00	0x01
0x38-0x3F	0x2B	0x00	0x01	0x44	0x00	0x52	0x43	0x31
0x40-0x47	0x31	0x38	0x30	0x2D	0x4D	0x42	0x55	0x53
0x48-0x4F	0x2C	0x31	0x2E	0x30	0x30	0x2C	0x31	0x2E
0x50-0x57	0x31	0x35	0x00	0xFF	0xFF	0xFF	0xFF	0xFF
0x58-0x5F	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x60-0x67	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x68-0x6F	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x70-0x77	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x78-0x7F	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
Address	**MBUS2 factory default Values**							
0x00-0x07	0x01	0x05	0x03	0x02	0x00	0x00	0x64	0x00
0x08-0x0F	0x00	0x00	0x00	0xD3	0x91	0xDA	0x80	0x80
0x10-0x17	0x7C	0x00	0x01	0x01	0x00	0x00	0x00	0x00
0x18-0x1F	0x00	0x0C	0xAE	0x12	0x34	0x56	0x78	0x01
0x20-0x27	0x07	0x01	0x01	0x00	0x00	0x00	0x00	0x04
0x28-0x2F	0xFF	0x08	0x00	0x00	0x00	0x00	0x00	0x00
0x30-0x37	0x05	0x08	0x00	0x01	0x05	0x00	0x00	0x01
0x38-0x3F	0x2B	0x00	0x00	0x06	0x0B	0x02	0x00	0x00
0x40-0x47	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x48-0x4F	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x50-0x57	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x58-0x5F	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x60-0x67	0x00	0x52	0x43	0x31	0x31	0x38	0x30	0x2D
0x68-0x6F	0x4D	0x42	0x55	0x53	0x2C	0x31	0x2E	0x30
0x70-0x77	0x30	0x2C	0x32	0x2E	0x30	0x30	0x00	0xFF
0x78-0x7F	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0x80-0x87	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x88-0x8F	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x90-0x97	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x98-0x9F	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xA0-0xA7	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xA8-0xAF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xB0-0xB7	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xB8-0xBF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xC0-0xC7	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xC8-0xCF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xD0-0xD7	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xD8-0xDF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xE0-0xE7	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xE8-0xEF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xF0-0xF7	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF
0xF8-0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Grey: Reserved (do not change).
Blue: Reserved for Module part number and version information (do not change).

Appendix: ASCII Table

HEX	DEC	CHR	CTRL
0	0	NUL	^@
1	1	SOH	^A
2	2	STX	^B
3	3	ETX	^C
4	4	EOT	^D
5	5	ENQ	^E
6	6	ACK	^F
7	7	BEL	^G
8	8	BS	^H
9	9	HT	^I
0A	10	LF	^J
0B	11	VT	^K
0C	12	FF	^L
0D	13	CR	^M
0E	14	SO	^N
0F	15	SI	^O
10	16	DLE	^P
11	17	DC1	^Q
12	18	DC2	^R
13	19	DC3	^S
14	20	DC4	^T
15	21	NAK	^U
16	22	SYN	^V
17	23	ETB	^W
18	24	CAN	^X
19	25	EM	^Y
1A	26	SUB	^Z
1B	27	ESC	
1C	28	FS	
1D	29	GS	
1E	30	RS	
1F	31	US	
20	32	SP	
21	33	!	
22	34	"	
23	35	#	
24	36	$	
25	37	%	
26	38	&	
27	39	'	
28	40	(	
29	41	)	
2A	42	*	
2B	43	+	
2C	44	,	
2D	45	–	
2E	46	.	
2F	47	/	
30	48	0	
31	49	1	
32	50	2	
33	51	3	
34	52	4	
35	53	5	
36	54	6	
37	55	7	
38	56	8	
39	57	9	
3A	58	:	
3B	59	;	
3C	60	<	
3D	61	=	
3E	62	>	
3F	63	?	

HEX	DEC	CHR
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	^
5F	95	_
60	96	`
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	|
7D	125	}
7E	126	~
7F	127	DEL

Document Revision History

Document Revision	Changes
1.0	First release
1.10	Detailed UART interface and Encryption example included. Minor changes and corrections
1.11	Timing and figure 3 corrections. Auto sleep included. Minor corrections in text.

Disclaimer

Radiocrafts AS believes the information contained herein is correct and accurate at the time of this printing. However, Radiocrafts AS reserves the right to make changes to this product without notice. Radiocrafts AS does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the Radiocrafts website or by contacting Radiocrafts directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the Radiocrafts website. Customers are encouraged to check regularly for the most recent updates on products and support tools.

Trademarks

RC232™ is a trademark of Radiocrafts AS. The RC232™ Embedded RF Protocol is used in a range of products from Radiocrafts. The protocol handles host communication, data buffering, error check, addressing and broadcasting. It supports point-to-point, point-to-multipoint and peer-to-peer network topologies.

All other trademarks, registered trademarks and product names are the sole property of their respective owners.

Life Support Policy

This Radiocrafts product is not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Radiocrafts AS customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Radiocrafts AS for any damages resulting from any improper use or sale.

Contact Information

Web site: www.radiocrafts.com

Address:
Radiocrafts AS
Sandakerveien 64
NO-0484 OSLO
NORWAY

Tel: +47 4000 5195
Fax: +47 22 71 29 15
E-mails: radiocrafts@radiocrafts.com
sales@radiocrafts.com
support@radiocrafts.com