Radiocrafts
Embedded Wireless Solutions

MBUS User Manual

RC11xx-MBUS3
RC1701HP-MBUS4

Important note:
This User Manual contains patented technology. See page 6 for more information.

Table of Contents

Table of Tables

Table of Figures

Table of Examples

Abbreviations

Abbreviation	Description
RSSI	Received Signal Strength Indicator
NTA 8130	Dutch companion standard
EN13757-4 (2013)	Wireless M-Bus Standard
OMS	Open Metering System, companion standard
NVM	Non-Volatile Memory
LBT	Listen Before Talk
SML	Short Message Standard, please see EN13757-4 for details
DLMS	Please see EN13757-4 for details
VIF	Value Information Field, please see EN13757-4 for details
DIF	Data Information Field, please see EN13757-4 for details

1 Introduction

1.1 Quick Start

The Module has a UART interface for both Wireless M-Bus packet data and module configuration. Use the UART format with settings (19200, 8, 1, N, no flow control).

How do I configure the M-Bus mode, RF channel or any other parameter?

To change configurable parameters:

1. Send one byte to the module with the value 0x00 or assert the CONFIG-pin. This will take the module into configuration mode. More about configuration mode is in chapter 2.1
2. Use special commands to access the configuration registers and test modes. All the commands for the configuration are in chapter 2.2
3. Exit from configuration mode by sending the 'X' command.
4. Start using the module as normal in packet mode. You may need to reset the module for the changes to take effect. Details about sending and receiving packet data is in chapter 2.3.

How do I transmit data?

Send your data to the RXD pin on the module. The first byte of the message must contain the message length sent over UART. The maximum length byte is 0xF6. The link layer header is automatically added. The module will transmit the data when the whole packet is received. See page 11 for details about the frame format for transmitting data. Details about sending and receiving packet data is found in chapter 2.3.

How do I receive data?

Any received RF data packet with correct Wireless M-Bus format and check sums will be output on the TXD pin. Optionally the meter address (first M-Bus block) is added to the data string. The RSSI value (received signal strength) can optionally be appended to the message. Also, CRC and start/stop bytes can be added. See page 11 for details about the frame format for receiving data. Details about sending and receiving packet data is found in chapter 2.3.

What about the antenna?

In most cases a simple quarter wavelength wire or a PCB track will do. Connect a piece of wire to the RF pin with length corresponding to the quarter of a wavelength. For space limited products, contact Radiocrafts and we will recommend the best antenna solution for your application.

Radiocrafts also offer an Application Note on tuning a 169 MHz antenna in AN025, found here: https://radiocrafts.com/resources/application-notes/

1.2 Differences between RC11xx and RC17xx series

This User Manual describes the embedded protocol of the Wireless M-Bus modules from Radiocrafts. The MBUS firmware is available as different feature sets targeting specific applications. The hardware has the same size and pin-out for all frequency versions, and the different feature sets available are listed in the table below. The feature sets and the embedded functions are independent of the frequency, so this User Manual is valid for all versions RC11xx(HP)-MBUS3 and RC1701HP-MBUS4. Detailed information on how to use the different feature sets is found in this User Manual. Additional information about the Wireless M-Bus packet structure for NTA 8130 compliance is described in Application Note 011 and is available on request.

Table 1 – MBUS Feature Sets Overview

	Feature Set	
Feature List	**RC11xx-MBUS (MBUS3™)**	**RC17xx-MBUS (MBUS4™)**
General	General Wireless M-Bus modem, with OMS feature set.	Wireless M-Bus mode N at 169 MHz
Network role	Master, Slave or Repeater	Master, Slave or Repeater
Modes	C1, S1, S2, T1, T2	N1, N2
Encryption	AES mode 4 and 5, and ELL encryption mode 1	AES mode 4 and 5, and ELL encryption mode 1
Installation mode	Yes, according to OMS	Yes
Number of installed meters	Up to 64	256 internally, unlimited externally (>1000 meters per concentrator)
Filter function	Master only receives messages from installed/registered meters (optional)	Master only receives messages from installed/registered meters (optional)
Two-way communication	Yes, according to OMS	Yes, for N2 mode
Automatic message generation from Master	Yes, patented automatic message generation. According to OMS. Supporting two-way slaves. Standard response or a predefined message from mailboxes or templates.	Yes, patented auto message generation. Special support for handling of > 1000 meters (Slaves).

The command set used to configure the MBUS modules are described in Appendix A .

1.3 Optional custom specific version

As an option to the standard feature sets, a full Wireless M-Bus application layer can be integrated in the module based on customer specification. In this case all the application layer protocol and timing will be handled internally by the module. Radiocrafts offer multiple such variants:

- MBUS with Pulse Counter
  - RC1140/60/70/80-MPC1
  - RC1701HP-MPC1
- MBUS Sensor Module
  - RC1180-MSM
  - RC1701HP-MSM
- Wize protocol
  - RC1701HP-WIZE

1.4 Important note on patented technology

Some of the technical solutions described in this User Manual are based on patented technology. In particular the methods used in the MBUS3 and MBUS4 to meet the T2 and N2 timing requirements for a master, using an address register, a flag register, an encryption key register combined with an auto-message generator for standard messages and its combination with a mailbox with pre-generated messages or templates, and a given message priority, depending on incoming messages, are subject to patenting.

Any infringements of patents and IP rights held by Radiocrafts will be prosecuted to the fullest extent.

2 Basic Functionality

The figure below shows a simplified block diagram of the module:

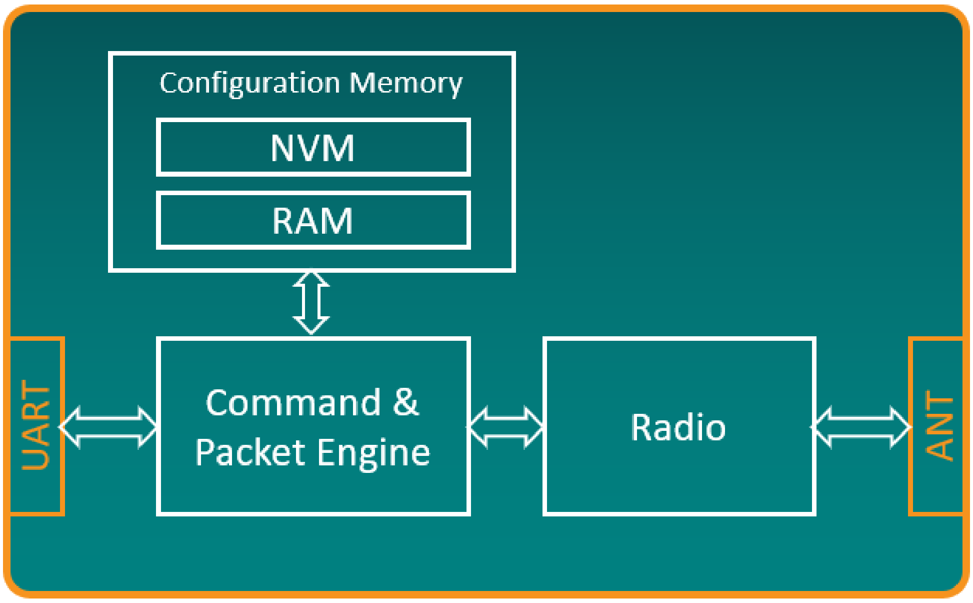

Figure 1 – Block diagram

The host use the UART Interface to send and receive Wireless M-Bus data. The UART packet content, like the addition of RSSI, CRC and start/stop bytes, can be changed in the configuration mode. The UART packet always starts with a Length byte (unless it is a one-byte command, or the module is in 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.

Note, while receiving data over the air, the module will not accept data on the UART. It is therefore recommended to use UART HW flow control (CTS) to avoid losing data or the module receiving a wrong length byte.

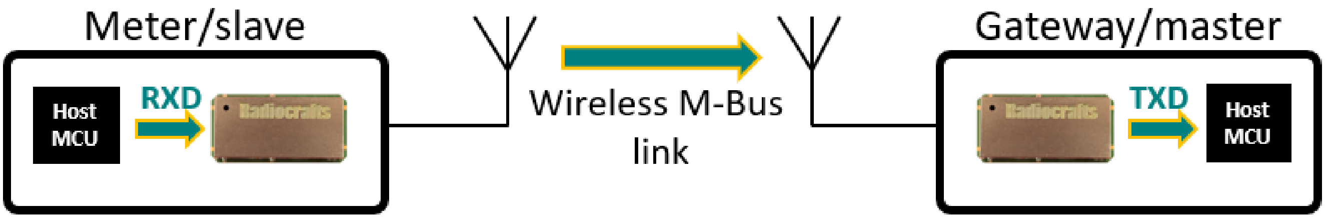

Figure 2 – Wireless M-Bus Link

2.1 UART Interface – 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 or at the manufacturing test. 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 0x00 to the module, or by asserting the CONFIG pin (set low). In this case the 0x00 is not interpreted a Length byte, but a single byte command to enter configuration mode.

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! 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' and so on) 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 Appendix A for details.

Figure 3 illustrates how to use the UART interface to enter configuration mode, change configuration parameters and return to IDLE mode.

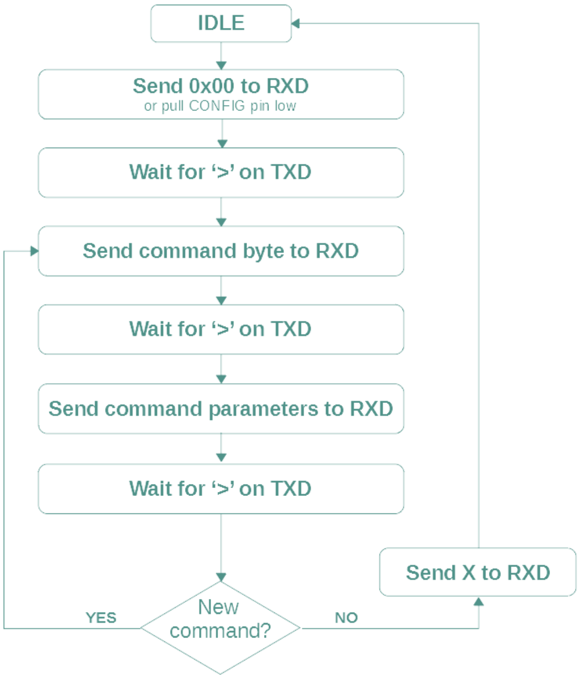

Figure 3 – Configuration mode flow diagram

2.2 UART Interface –Configuration Commands

The UART interface is connected to a command and packet engine. Table 2 shows an overview of the available commands when the module is in configuration mode. ***The detail for each command is found in Appendix A .***

Table 2 – Configuration Commands overview

Command						
ASCII	**hex**	**Parameter**	**MBUS3**	**MBUS4**	**Stored in NVM (Flash)**	**Stored in RAM**
'A'	0x41	Auto-message flags	✓	✓	✗	✓
'B'	0x42	Bind	✓	✓	✓	✗
'C'	0x43	Channel	✓	✓	✗	✓
'E'	0x45	Encrypt	✓	✓	✗	✓
'F'	0x46	C-field	✓	✓	✗	✓
'G'	0x47	M-Bus mode	✓	✓	✗	✓
'I'	0x49	Install	✓	✓	✗	✓
'K'	0x4B	Key register	✓	✓	✓	✗
'L'	0x4C	List binding	✓	✓	✓	✗
'M'	0x4D	Memory configuration	✓	✓	✓	✗
'N'	0x4E	Access Number	✓	✓	✗	✓
'O'	0x4F	Read Auto-message flag register	✓	✓	✗	✓
'P'	0x50	Output power	✓	✓	✗	✓
'Q'	0x51	Quality Indicator	✓	✓	✗	✓
'R'	0x52	Read mailbox	✓	✓	✗	✓
'S'	0x53	Signal Strength (RSSI)	✓	✓	✗	✓
'T'	0x54	Destination address	✓	✓	✗	✓
'U'	0x55	Temperature monitoring	✓	✓	✗	✓
'V'	0x56	Voltage monitoring	✓	✓	✗	✓
'W'	0x57	Write to mailbox	✓	✓	✗	✓
'X'	0x58	Exit command	✓	✓	✗	✓
'Y'	0x59	Memory Read one byte	✓	✓	✗	✓
'Z'	0x5A	Sleep mode	✓	✓	✗	✓
'a'	0x61	Auto-message flags for 4 slaves in RAM	✗	✓	✗	✓
'b'	0x62	Bind for 4 slaves in RAM	✗	✓	✗	✓
'k'	0x6B	Key register for 4 slaves in RAM	✗	✓	✗	✓
'l'	0x6C	List binding for 4 slaves in RAM	✗	✓	✗	✓
'o'	0x6F	Read Auto-message flag register for 4 slaves in RAM	✗	✓	✗	✓
's'	0x73	Continuous Signal Strength (RSSI)	✗	✓	✗	✓
'0'	0x30	List Configuration Memory	✓	✓	✗	✓

Command ASCII	hex	Parameter	MBUS3	MBUS4	Stored in NVM (Flash)	Stored in RAM
‘1’	0x31	Test mode 1	✓	✓	✗	✓
‘2’	0x32	Test mode 2	✓	✓	✗	✓
‘3’	0x33	Test mode 3	✓	✓	✗	✓
‘4’	0x34	Test mode 4	✓	✓	✗	✓
‘7’	0x37	Test mode 7	✗	✓	✗	✓
‘8’	0x38	Test mode 8	✗	✓	✗	✓
‘@RC’	0x40 0x52 0x43	Reset Configuration to factory defaults	✓	✓	✗	✓
‘@RR’	0x40 0x52 0x52	Reset Radio	✓	✓	✗	✓

2.3 UART Interface – Send and receive messages

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.

In normal data mode, the first byte sent to the module is interpreted as the Length field. The module will then read that number of bytes from the UART, add packet information according to module configuration and transmit. The maximum length of the message sent to the module is 0xF6, the minimum length is 0x01. The maximum length corresponds to 255 bytes including the link layer as transmitted on the air.

Note, the module is “deaf” while receiving data over the air. If the module is configured for bi-directional use (not only as transmitter), but care should also be taken not to send data to the module while busy by using the UART hardware handshake (if enabled, CTS is high during radio reception). The host may halt transfer data from the module by pulling RTS high (if enabled).

The Length byte values > 0xF6 have been given some special meaning in the RC modules and are referred to as single byte “soft commands” since they are commands that are sent when the module is in normal data mode and not in command and configuration mode. As mentioned above, 0x00 is also an exception used to enter command and configuration mode.

Table 3 – Soft commands.

Length byte / soft command	MBUS3	MBUS4	Set
0x00	✓	✓	Set the module in command and configuration mode
0x01-0xF6	✓	✓	Legal length values
0xF7-0xF9	✓	✓	Illegal length values. Reserved for future use. Module will return to IDLE.
0xFA	✗	✓	Override LBT
0xFB	✗	✓	No response message to be sent, module will go to IDLE mode
0xFC	✗	✓	“Key challenge” transfer from host to module

Length byte / soft command	MBUS3	MBUS4	Set
0xFD	✗	✓	Set module in IDLE mode, enable RF receiver (use after SLEEP)
0xFE	✓	✓	Module will send “empty” message (link layer only, no application layer)
0xFF	✓	✓	Set module in IDLE mode, disable RF receiver (UART only)

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:

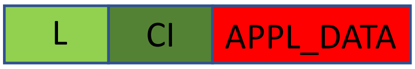

Figure 4 – UART interface packet transmission (RXD pin)

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 (transport layer), and data points with VIF and DIF codes. The application data can also be SML or DMLS or manufacturer specific.

An Extended Link Layer (ELL) can be added before the application data using this structure:

L + $\mathrm{CI}_{\mathrm{ELL}}$ + ELL + $\mathrm{CI}_{\mathrm{APL}}$ + APPL_DATA

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

To transmit only a HEADER without Application data (CI+APPL_DATA) a L=0xFE can be sent to the module UART without additional bytes.

Frame format when receiving data

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

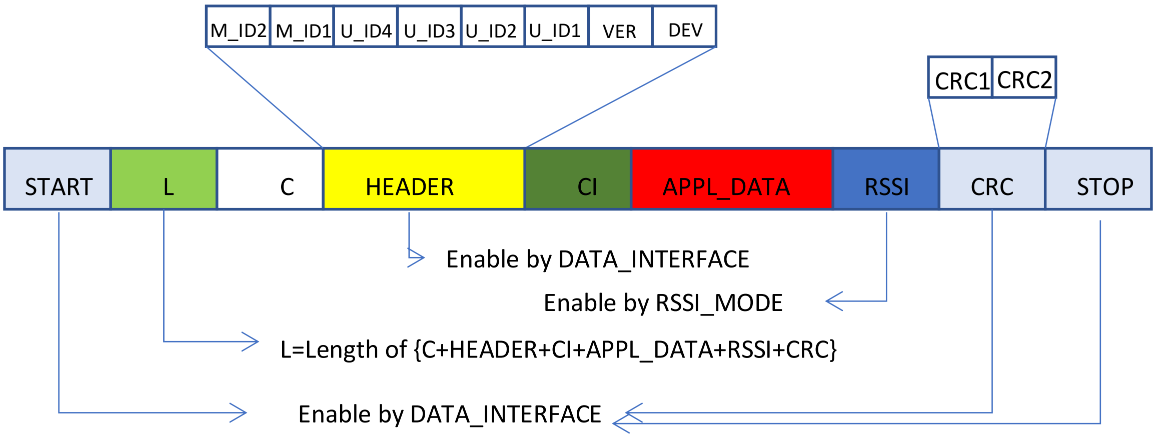

Figure 5 – UART interface packet reception (TXD pin)

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. CRC is calculated from length byte including RSSI (if enabled).

Note! The length bytes itself must be reduced by 2 before calculating CRC on host. The length byte was shorter at calculation in module as the CRC was not added at time of calculation.

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 received).

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 appended when RSSI_MODE = 1.

2.4 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. In the standard the Master is referred to as "Other."

The Radiocrafts Wireless M-Bus family of modules 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 but the Master module within range will receive the message and based on the Slave address (if the Slave is installed and the Master is configured for filtering), it will decode the message and send the data on its serial interface (TXD-pin).

In two-way communication modes, the battery-operated meter (slave) will keep the receiver "on" for a brief time. During this time slot the master can acknowledge the received message in order to open the communication channel (NTA 8130) or send a command (OMS) and thereby start a communication sequence.

MBUS3 (OMS) and MBUS4 also allows for a one-way (unidirectional) repeater. The repeater will re-transmit all messages from slaves within range. Modules with MBUS3 and MBUS4 feature sets can be configured as a repeater.

MBUS3 has since its original release been extended to support the new C-mode (Compact mode), in addition to OMS functionality. A unique feature of MBUS3 is that T mode and C mode messages can be received in parallel with the same configuration.

For battery operated devices the slave is always initiating the communication, and the master must then transmit (if need) within a brief time window (2-3 ms in the T mode). After one such "ping-pong" sequence, the slave will have a pause (enter sleep mode) for 2-5 seconds, before it again does a new transmission allowing the master to do another transmission. This means that all messages to be sent from the master must be ready and transmitted within a very short time. The MBUS3 Auto-message generator and Mailbox features make this task easy and doable (patented implementation).

MBUS3 supports 64 slaves registered in a master.

MBUS3 also support reception of mode C. Mode C messages can be received in combination with mode T. The module also automatically receives and distinguish between Frame Format A and B. The current implementation supports C1 mode (unidirectional) as specified in the EN 13757-4 (2013). C2 mode timing is currently not supported. C mode meters can be installed inside the Master module, optionally with encryption keys, mixed with T mode meters.

The module supports the new Extended Link Layer (ELL), its new AES-128-CTR mode of encryption, and the new compact frame formats, all according to EN13757-4 (2013). The new ELL is available for all modes.

The MBUS4 supports the mode N (169 MHz) variant of the Wireless M-Bus standard as specified in the EN 13757-4 (2013). The implementation supports both N1 (unidirectional communication) and N2 (bi-directional communication). That is, two-way communications with transmission and reception by both Slave and Master is available, including the two-way timing for sleeping Slaves.

The MBUS4 functionality is like the MBUS3 functionality, with the addition of the Master supporting 256 meters (Slaves) internally and > 1000 Slaves registered externally (in the host).

Like the MBUS3, the MBUS4 Master can be configured to receive all messages or only messages from installed meters. Messages to/from installed meters can be encrypted/decrypted.

The MBUS4 receives and decodes mode N with both Frame Format A and B automatically in real-time. The Frame Format used for transmission of messages is set by the “PREAMBLE_LENGTH” parameter, as described below.

The number of channels available in mode N were increased in the 2018 revision of EN13757-4. This revision also added a new data rate, 6.4 kbps. The channel and data rate settings to support this are shown below.

The MBUS4 support both Application Layer (i.e., Transport Layer) encryption and the new Link Layer Encryption, using the Extended Link Layer, as specified in EN13757-4 (2013). The module accepts all CI-fields and will automatically use the correct encryption scheme. Encryption / decryption is enabled in the Flag Register. The default value of the Flag Register encryption / decryption flags is set using the ENCRYPT_FLAG / DECRYPT_FLAG configuration parameters as for MBUS3. When ELL is used and encryption is enabled, the module also automatically adds the correct Payload CRC. The Payload CRC can therefore be set to 0x00 by the host in this case.

2.5 *MBUS Basic functionality*

The module offers a buffered packet radio acting as a Wireless M-Bus modem. The module contains a fully embedded protocol supporting the following modes:

- Stationary mode S (S1, S1-m, S2)
- Frequent transmit mode T (T1 and T2)
- Frequent receive mode R2
- C1 mode
- Narrowband mode N (N1 and N2)

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:

Table 4 – M-Bus mode overview

M-Bus Mode	Set	Comments

S1/S2
short preamble
	MBUS_MODE = 0
	PREAMBLE_LENGTH = 0
		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.

S1/S2
long preamble
	MBUS_MODE = 0
	PREAMBLE_LENGTH = 1	

T1	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 slave	MBUS_MODE = 2
	NETWORK_ROLE = 0
		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.

T2 master	MBUS_MODE = 2
	NETWORK_ROLE = 1	

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

C1	MBUS_MODE = 9	
C2	MBUS_MODE = 8	
C1+T2	MBUS_MODE = 10	The RF channel, data rate and preamble length are set internally in the module
C1+T1	MBUS_MODE = 11	
N1	MBUS_MODE = 17	
N2	MBUS_MODE = 16	

The module supports automatic generation of the Wireless M-Bus frame, which is:

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

The RF signal is Manchester coded or "3 out of 6" coded for increased signal integrity (S and T modes).

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.

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 (volatile memory).

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.

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. The default timeout is 2 seconds. The maximum total payload is 246 bytes, giving 255 bytes when including the header in the first block.

3 Timing

This chapter describes the function critical timing between different states and for some commands/operations in configuration mode.

3.1 States overview

The figure below shows the different states of the module. Table 5 explains the details of each state.

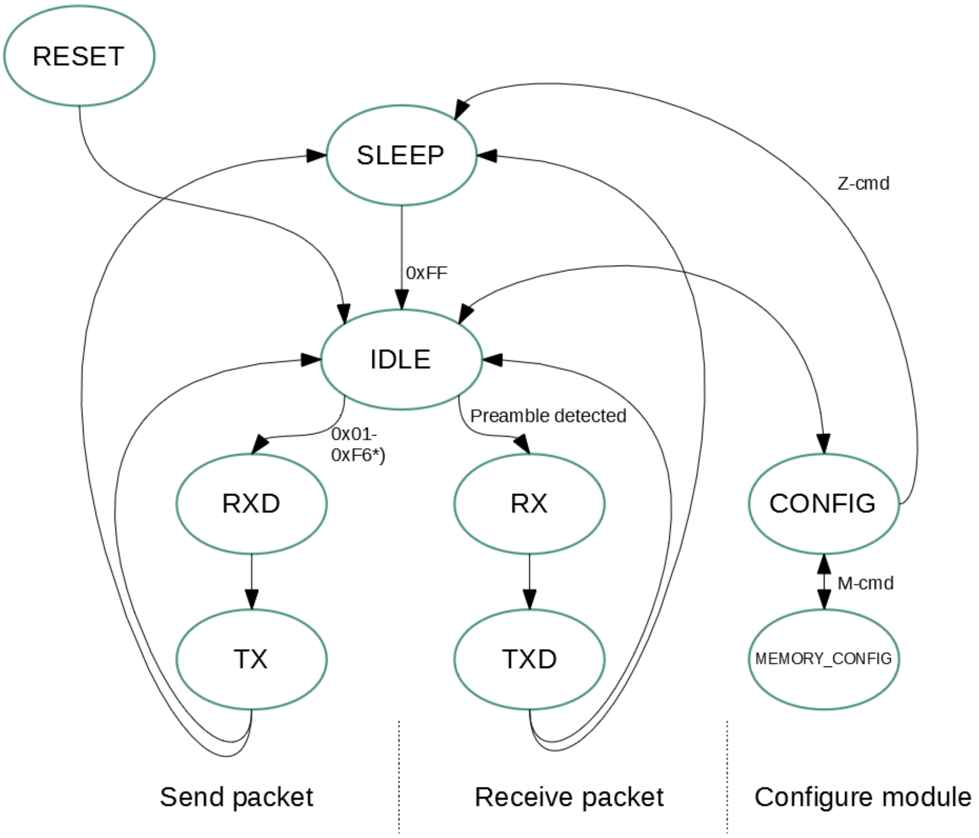

Figure 6 – State diagram. State transitions for auto sleep is not shown.

Table 5 – State names and descriptions

State	Description
RESET	Module is reset. All configurations not stored in NVM are lost.
SLEEP	The module is in a low power consumption mode and needs to be woken up by sending 0xFF on UART RXD to enter IDLE state.
IDLE	This is the normal state where the module both searches for preamble on RF (if enabled) and wait for a character to be received on the UART.
RXD	The state when receiving characters on UART from the host, filling up the internal buffer.
TX	When the data is transmitted on the air.
RX	When data is received from the air after preamble detection.
TXD	The state where the received data is sent to the host on the UART.
CONFIG	CONFIG is the configuration mode, the state entered by sending 0x00 or asserting the CONFIG pin and is entered during parameter configuration.
MEMORY_CONFIG	MEMORY CONFIG is the sub-state of CONFIG entered by the 'M' command where the non-volatile configuration memory is being programmed.

3.2 Timing – from RESET and SLEEP to IDLE

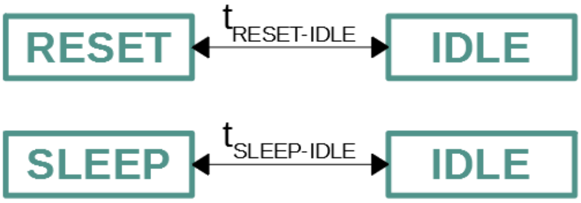

Figure 7 – Timing definitions, from RESET or SLEEP to IDLE

Table 6 – Timing numbers, from RESET or SLEEP to IDLE

Symbol	MBUS3	MBUS4	Description / Note
$t_{RESET\text{-}IDLE}$	6.6 ms	9.1 ms	
$t_{SLEEP\text{-}IDLE}$	1.6 ms	3.2 ms	For UART data rates up to 4.8 kBd the sequence 0xFF : 0x00 can be sent without delay (from Sleep to Config). For higher UART baud rates, add 2 ms delay before setting Config mode

3.3 Timing – in and out of CONFIG mode

For t_{CONFIG_PROMPT} it is recommended to

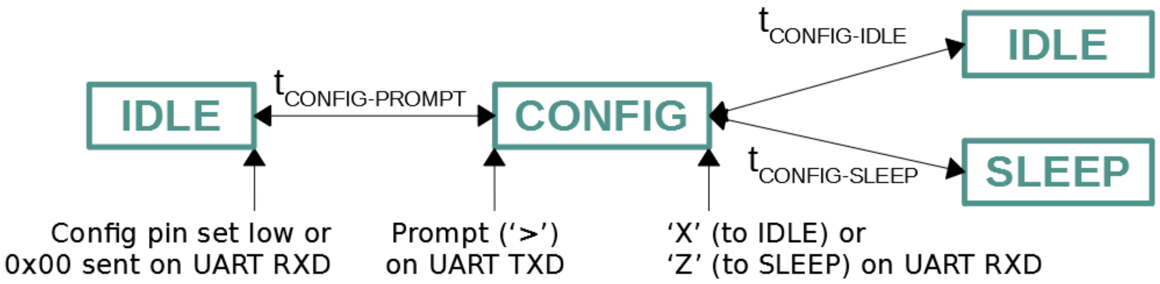

Figure 8 – Timing definitions, in and out of CONFIG

Table 7 – Timing numbers, in and out of CONFIG

Symbol	MBUS3	MBUS4	Description / Note
$t_{CONFIG\text{-}PROMPT}$	Wait for prompt		Time from 0x00 on UART RXD or CONFIG pin is set low until prompt (">") is received in UART TXD
$t_{CONFIG\text{-}IDLE}$	146us	1.2ms	Time from end of 'X' character on UART RXD to IDLE
$t_{CONFIG\text{-}SLEEP}$	~150us		Time from end of 'Z' character on UART RXD to IDLE

3.4 Timing – Programming Non-Volatile Memory (NVM)

When writing to NVM the operation must not be interrupted by reset or power off. This can cause the memory to be corrupted. Therefore, it is important to wait for the prompt on TXD after issuing such commands.

M-command, @RC-command

Both the M-command and the @RC-command are programming the configuration area of the NVM.

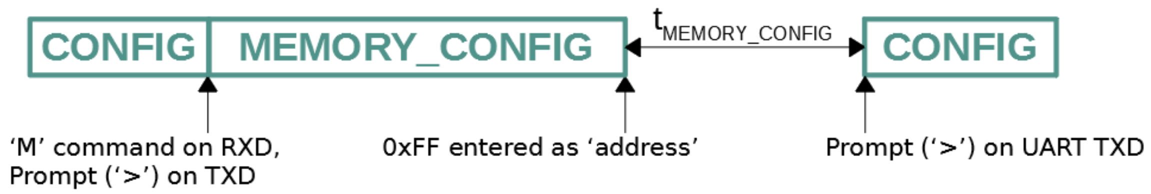
CONFIG
MEMORY_CONFIG
t_MEMORY_CONFIG
CONFIG
'M' command on RXD, Prompt ('>') on TXD
0xFF entered as 'address'
Prompt ('>') on UART TXD

Figure 9 – Timing definitions, in and out of MEMORY_CONFIG

Table 8 – Timing numbers, in and out of MEMORY_CONFIG

Symbol	MBUS3	MBUS4	Description / Note
$t_{MEMORY-CONFIG}$	31 ms	31ms	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.***

B, K – command

These commands are reading back, changing, and writing a larger portion of the NVM and thus uses longer time than the M-command. They too return the prompt character, and it is important to wait for it.

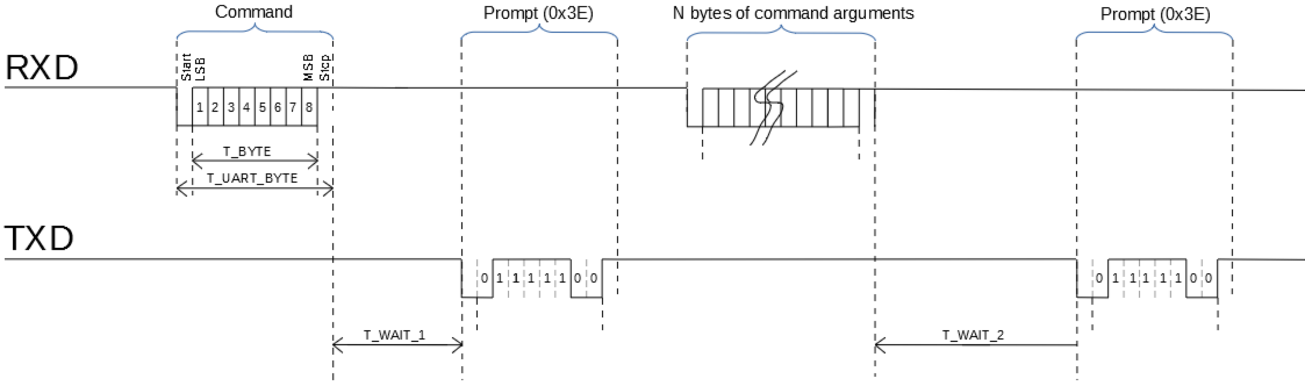

Figure 10 – Timing definitions, commands with arguments but no response data.

Table 9 – Timing numbers, configuration commands with arguments but no response data

Command	Parameter	MBUS3	MBUS4	Comment
Any	T_{BYTE}			Varies with UART data rate
Any	T_{UART_BYTE}			Varies with UART data rate
B, K	T_{WAIT_1}	3.9 ms / 7.8ms	~50us	
B, K	T_{WAIT_2}	45 ms	55 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.***

3.5 LED indicator

There are two LED driver outputs, LED0 and LED1, see Data Sheet for pinout. These can be configured to indicate the current status of the module and is configured by LED_CONTROL.

Table 10 - LED indicators configuration

LED_CONTROL	LED0	LED1	Comment
0x01	RX	TX	The LEDs are indicating ongoing radio packet reception (RX), and packet transmission (TX). The RX indicator will turn on after the preamble, synch and Length byte is received (Note, the packet may still be discarded due to CRC error or wrong address). The TX indicator is on during the whole transmission of the packet starting with pre-amble.
0x02	RF RX	UART RX	The LEDs are indicating when the radio is listening (RF RX), and when the UART is listening for data (UART RX).
0x03	RX TXD	TX	The LEDs are indicating when an accepted radio packet is being transferred to the host over UART (RX TXD), and packet transmission (TX). The RX TXD indicator will turn on after the received packet has been verified and the address is accepted. The TX indicator is on during the whole transmission of the packet starting with pre-amble (same as for setting 0x01).

4 Power Management

There are diverse ways to minimize the total current consumption and thus extend the battery lifetime for battery operated devices. What determines the battery lifetime is how much time the module spends in the modes where current consumption is "high."

The Radiocrafts modules offers several configuration options to allow the module to be "off" for an extensive part of the time:

- Manual SLEEP function (‘Z’-command from config mode)
- Automatic SLEEP functions (SLEEP_MODE configuration parameter)
- IDLE mode without listening to RF.
- Turn off module completely (not recommended)

The details of these methods are described in the following subchapters.

4.1 Manual SLEEP function

The low power SLEEP mode is manually entered by using the SLEEP command ‘Z’ after the module is set in configuration mode.

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 the wake-up byte 0xFF on the UART RXD line (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.

4.2 MBUSx Automatic Sleep

It is also possible to configure the module to enter SLEEP automatically after a message has been transmitted (SLEEP_MODE=1). With this setup the module has to enter TX-mode (transmit a message) after power-on before entering SLEEP mode first time.

The Slave has special support for automatic sleep after data transmission. If automatic SLEEP after TX is enabled (SLEEP_MODE = 1), the module will automatically go to sleep after data transmission, if a message is not received. The automatic sleep is done after any message transmitted. The slave host application is responsible for waking up the module for a re-transmission in 2-5 seconds if a communication sequence was started, but no reply received from the master.

If automatic SLEEP after RX (and TX) is enabled (SLEEP_MODE = 3), the module will automatically go to sleep after data reception and the message is sent to the host over the UART. This feature should normally only be used for a Slave.

4.3 IDLE mode without listening to RF

A ‘0xFF’ byte will wake up a sleeping Slave into a low power idle mode (IDLE). The Slave will only accept UART input (RXD UART) in this IDLE mode, not RF, to save current. The current saving depends on UART rate and gives less current consumption for low UART rates (but data transfer takes longer time) Figure 11 illustrates current vs time for a slave configured for auto sleep at 2.4 kBaud UART rate when receiving acknowledge from Master after the first transmission.

An additional ‘0xFF’ byte in IDLE mode will force the Slave into RX listening mode to enable RF reception before transmitting.

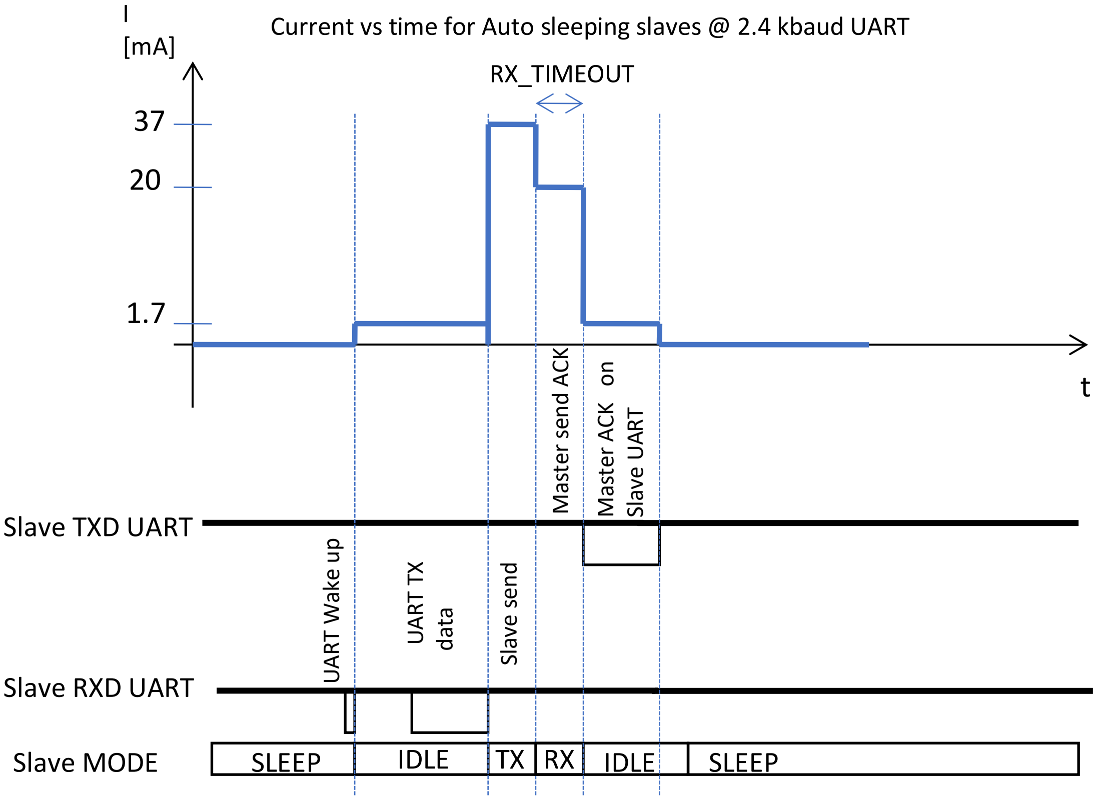

Figure 11 – Current vs Time for Auto Sleeping Slaves

4.4 Power cycling

It is not recommended to turn off the module as an alternative to using SLEEP mode. The combination of extremely low SLEEP current and fast start-up gives a lower overall current consumption.

5 Installation and Binding

In order to use encryption or two-way communication (or both), slaves must be installed at the gateway. The installation feature can also be used to filter out messages from the slaves of interest. When INSTALL_MODE = 0x00 (Filter on), only messages from the installed slaves will be output on UART.

5.1 Available address registers, key registers, and flag registers

To bind a slave to a master, the slave's address must be stored in an address register on the master. MBUS3 has 64 such address registers, numbered 1-64. MBUS4 has 254 address registers, 4 of which are in RAM only.

MBUS3 has the same number of address registers, key registers and flag registers, and an installed slave must use the same number for address, key, and flags.

MBUS4 has 254 address registers, 254 key registers and 132 flag registers. However, MBUS4 has additional commands and features so that slave addresses and keys can be stored in the host MCU. It also has 4 RAM-based address/key/flag numbers that can be used to set up two-way communication to any slave. You can read more about this in chapter 7.9. The table below shows an overview of address register numbers, key, and flag registers.

#	Address register	Key register	Flag register
1 64	MBUS3 & MBUS4 (NVM)	MBUS3 & MBUS4 (NVM)	MBUS3 & MBUS4 (NVM)
65 128	MBUS4 (NVM)	MBUS4 (NVM)	MBUS4 (NVM)
129 132	MBUS4 (RAM)	MBUS4 (RAM)	MBUS4 (RAM)
133 254	MBUS4 (NVM)	MBUS4 (NVM)	

5.2 The installation and binding process

The installation and binding process can be divided as such:

- Gateway is set into INSTALL_MODE = 0x01. This can be done by using the 'I'-command. In this mode all incoming messages of type SND-IR will be output on UART TXD. Other message types will be ignored.
- Host MCU evaluates SND-IR message and extracts address.
- Host MCU has control over which installation addresses that are available and decides on which Installation address to install the slave.
- The host MCU sends a Bind command ('B') to Radiocrafts module with installation address and slave address.
- The host MCU sets INSTALL_MODE = 0x00 (Filter on).
- The host MCU sets up two-way communication so that next time the same slave sends SND-IR message it will be replied automatically with CNF-IR (optional)

A typical sequence for installing a slave is shown in Figure 12

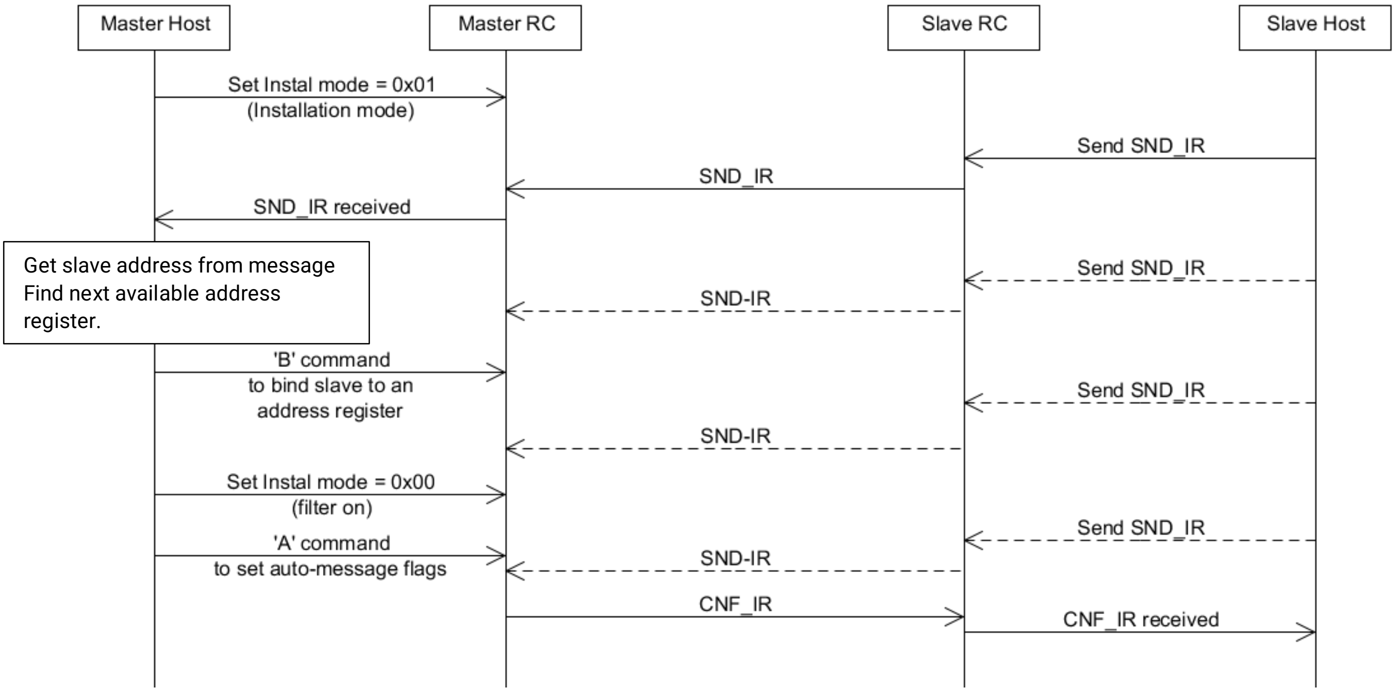

Figure 12 – Installation and binding process

5.3 Examples

Example 1 – Setting Install mode

Example

This example shows how to enter Install mode, the first step of the installation and binding process. Before sending the I-command you must set the module in configuration mode and wait for prompt (>).

Command	Hex	Response	Comment/Note
Enter	0x00	‘>’	Enter CONFIG mode. Wait for ‘>’ prompt. *Alternative*: Assert CONFIG pin. De-assert CONFIG after ‘>’ prompt
‘I’	0x49	‘>’	I-command entered. Wait for ‘>’ prompt.
1	0x01	‘>’	Command argument. Wait for ‘>’ prompt.
(A new command can be issued here)			
‘X’	0x58	(none)	Exit CONFIG mode. Module returns to IDLE state

Example 2 – Binding a slave to a master

Example

Before you run the code below:

- Set module in configuration mode
- Set installation mode (‘I’ - command).

In this example a slave with the following address will be bound to address register #2.

MAN_ID1	MAN_ID2	ADDR_ID1	ADDR_ID2	ADDR_ID3	ADDR_ID4	ADDR_VER	ADDR_DEV
0xAB	0xCD	0x12	0x34	0x56	0x78	0x01	0x04

Command	Hex	Response	Comment/Note
‘B’	0x42	>	B-command entered. Wait for ‘>’ prompt
2	0x02	(none)	Bind to address register #2
	0x78 0x56 0x34 0x12		ADDR_ID (4 bytes) Least significant byte first
	0xCD 0xAB		MAN_ID Least significant byte first
	0x01		ADDR_VER
	0x04	>	ADDR_DEV
(A new command can be issued here)			
‘X’	0x58	(none)	Exit CONFIG mode. Module returns to IDLE state

6 Encryption and decryption

The module supports AES-128 encryption for both extended link layer (ELL) encryption and transport layer (TPL) encryption.

6.1 *Key exchange*

When a slave is produced, it is normally programmed with a unique ID and a corresponding unique key. These ID-key pairs must be kept in a central database.

When a slave requests to be installed at a gateway, the gateway host MCU must request the encryption key from the central utility data base. This is done over a secure connection. The host MCU then programs the key for the slave into the same register location number as is used for installing the slave ID. The ‘K’-command is used for this.

The slave must have its own key installed at register location number 0x01.

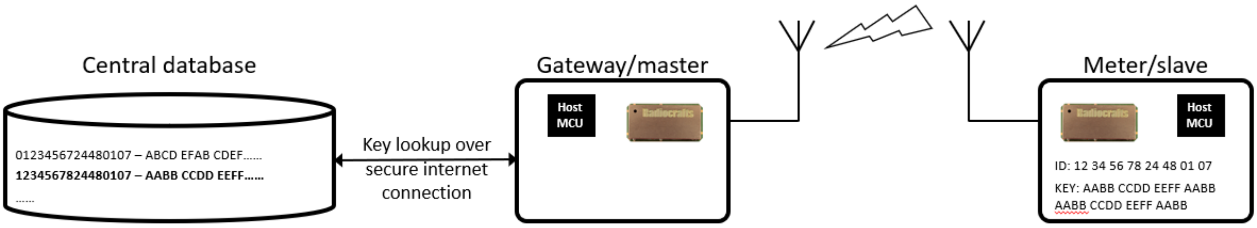
Central database
0123456724480107 – ABCD EFAB CDEF......
1234567824480107 – AABB CCDD EEFF......
......
Key lookup over secure internet connection
Gateway/master
Host MCU
Radiocrafts
Meter/slave
Radiocrafts
Host MCU
ID: 12 34 56 78 24 48 01 07
KEY: AABB CCDD EEFF AABB AABB CCDD EEFF AABB

Figure 13 – Encryption key lookup

6.2 *Enabling Encryption and decryption*

When the key is registered on the master side, the slave and master can start using encryption.

Two conditions need to be met for encryption and decryption to take place:

1. Encrypt or decrypt flag (or both) needs to be set in the flag register for each slave that requires encryption. For the gateway, the flag register corresponding to that particular slave needs to be used. On the slave side register number 1 must be used.
2. Encryption needs to be defined in the packet header when sending the packet to the RC1xxx-MBUSx module. Here there are two options, and how encryption is defined is slightly different for the two options.
    - Link Layer Encryption
    - Transport Layer Encryption

6.3 *The flag registers*

There are two configuration parameters in NVM named ENCRYPT and DECRYPT. These will set the encrypt/decrypt flags in the flag registers for all register location numbers when the device is booted. The flag registers are in RAM and are also referred to as auto-message flag registers. They can be edited with the ‘A’-command.

6.4 *Link Layer Encryption*

If you want to use Link Layer encryption, you need to use the Extended Link Layer (ELL).

If both Link Layer encryption and Transport Layer Encryption is defined – Link Layer encryption will be used.

The ELL encryption mode 1, uses the counter mode (AES-128-CTR) and does not need any padding of data.

For ELL encryption the module will automatically add or check the Payload CRC if the ENCRYPT_FLAG or DECRYPT_FLAG parameters are set with &0x02.

6.5 Transport Layer Encryption

Transport Layer Encryption can be used for all message types with short or long header. It cannot be used for messages with «no header». Security modes 4 and 5 are supported.

If both Link Layer encryption and Transport Layer Encryption is defined – Link Layer encryption will be used.

The TPL encryption uses block mode (AES-128-CBC) and need padding to 16-byte blocks. The Configuration Word must set encryption mode 0x04 or 0x05. If encryption mode is set to 0x00 the message will not be encrypted even if flags are set. When using 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, 0x60, 0x64, 0x6C, 0x6D, 0x72, 0x7C, 0x7E, 0x80 and 0x8B). For the short header (CI-field fields 0x5A, 0x61, 0x65, 0x7A, 0x7D, 0x7F and 0x8A) the Initialization Vector is partly from the link layer header (destination address) and the short application header.

The Access Counter byte in the application header is used by the encryption, and the counter must be incremented in order to avoid repetitive messages. When using the mailbox template, the counter is automatically incremented by the module (see description of two-level buffering).

6.6 Mailbox pre-encryption

The ‘E’ command can be used to encrypt a message in the Mailbox before it is sent. This is done to save time and needs to be done in T2-mode if the message to be sent back via the mailbox is more than one block.

6.7 Examples

Example 3 – Message with Link Layer encryption

Example

Given that the conditions below are met, the packet shown below can be sent over the air and will be encrypted.

- Slave is registered on master
- Slave has its own key in key register 1
- Master has the slave's key at the register where slave is installed
- Encrypt/decrypt flags are set in flag registers

L	C	LL_HEADER								ELL			APP DATA - NO TPL_HEAD										
		MID2	MID1	UID4	UID3	UID2	UID1	VER	DEV	CI	CC	ACC	CI	DIF	VIF	DATA	DATA	DIF	VIF	DATA	DATA	RSSI	CRC

	Field	Hex	Comment/Note
🟥	L	0xyy	Length
🟧	C	0x44	C-field (0x44 = SND-NR). Added automatically. ***Set value in configuration mode during initialization***
🟨	MAN_ID2 MAN_ID1	0xAB 0xCD	Manufacturer ID. MSB first. Added automatically. ***Set value in configuration mode during initialization***.
🟨	U_ID4 U_ID3 U_ID2 U_ID1	0x12 0x34 0x56 0x78	Unique ID. MSB first. Added automatically. ***Set value in configuration mode during initialization***.
🟨	VER	0x01	Version. Added automatically. ***Set value in configuration mode during initialization***.
🟨	DEV	0x04	Device Type (0x04 = Heat meter). Added automatically ***Set value in configuration mode during initialization.***
🟩	CI		CI field for the Extended Link Layer
🟩	CC		
🟩	ACC		
🟦	CI	0x78	CI field for the Application Layer. No header.
🟦	DIF	0x01	User data DIF field
🟦	VIF	0x02	User data VIF field
🟦	DATA	0x03	User data
🟦	DATA	0x04	User data
🟦	DIF	0x05	User data DIF field
🟦	VIF	0x06	User data VIF field
🟦	DATA	0x07	User data
🟦	DATA	0x08	User data
🟦	RSSI	-	Optional. Controlled by RSSI_MODE
🟦	CRC	-	Optional. Controlled by DATA_INTERFACE

Based on the above this translates to the following packet when sent over UART
0x14 0xAB 0xCD 0x12 0x34 0x56 0x78 0x01 0x04 0xyy 0xyy 0xyy 0xaa 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08

Example 4 – Message with Transport Layer encryption

Example

Given that the conditions below are met, the packet shown below can be sent over the air and will be encrypted.

- Slave is registered on master at slave address #1
- Slave has its own key in key register #1
- Master has the slave key at the register where slave is installed (#1)
- Encrypt/decrypt flags are set in flag registers

L	C	LL_HEADER								TPL_LONG_HEADER													APPLICATION DATA																		
		MID1	MID2	UID1	UID2	UID3	UID4	VER	DEV	CI	UID4	UID3	UID2	UID1	MID2	MID1	VER	DEV	AN	STAT	CF2	CF1	PAD1	PAD1	CI	DIF	VIF	DATA	DATA	DIF	VIF	DATA	DATA	PAD2	PAD2	PAD2	PAD2	PAD2	PAD2	RSSI	CRC

	Field	Hex	Comment/Note
🟥	L	0x26	Length of MBUS packet.
🟧	C	0x44	C-field (0x44 = SND-NR). Added automatically. ***Set value in configuration mode during initialization***
🟨	MAN_ID1 MAN_ID2	0xAB 0xCD	Manufacturer ID. MSB first. Added automatically. ***Set value in configuration mode during initialization***.
🟨	U_ID1 U_ID2 U_ID3 U_ID4	0x12 0x34 0x56 0x78	Unique ID. MSB first. Added automatically. ***Set value in configuration mode during initialization***.
🟨	VER	0x01	Version. Added automatically. ***Set value in configuration mode during initialization***.
🟨	DEV	0x04	Device Type (0x04 = Heat meter). Added automatically ***Set value in configuration mode during initialization.***
🟩	CI	0x72	CI field (0x72 = Long header).
🟩	UID4 UID3 UID2 UID1	0x78 0x56 0x34 0x12	Unique ID. LSB first. ***Not*** added automatically.
🟩	MID2 MID1	0xCD 0xAB	Manufacturer ID. LSB first. ***Not*** added automatically.
🟩	VER	0x01	Version. ***Not*** added automatically.
🟩	DEV	0x04	Device Type. ***Not*** added automatically.
🟩	AN	0x01	Access Number
🟩	STAT	0x00	Status
🟩	CF2	0x10	Configuration Word 2
🟩	CF1	0xC5	Configuration Word 1

Example (continued)

	Field	Hex	Comment/Note
🟦	PAD1	0x2F 0x2F	Encryption Verification Bytes. Mandatory
🟦	DIF	0x01	User data DIF field
🟦	VIF	0x02	User data VIF field
🟦	DATA	0x03	User data
🟦	DATA	0x04	User data
🟦	DIF	0x05	User data DIF field
🟦	VIF	0x06	User data VIF field
🟦	DATA	0x07	User data
🟦	DATA	0x08	User data
🟦	PAD2	0x2F 0x2F 0x2F 0x2F 0x2F 0x2F	Padding to fill block (16 bytes)
■	RSSI	-	Optional. Controlled by RSSI_MODE
■	CRC	-	Optional. Controlled by DATA_INTERFACE

Based on the above this translates to the following packet when sent over UART:
'0x1D 0x72 0x78 0x56 0x34 0x12 0xCD 0xAB 0x01 0x04 0x01 0x00 0x10 0xC5 0x2F 0x2F 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x2F 0x2F 0x2F 0x2F 0x2F 0x2F' *)

*) Packet length sent over UART is not identical to packet length over-the-air for the MBUS packet. The difference is calculated within the module

7 Two-way communication

In T-mode the master must reply with a new message within 3 ms after a received message, if further communication shall take place. To meet this timing requirement, the module has multiple built-in options. These include:

- One mailbox flag register. This is found in configuration memory.
  To alter the MAILBOX flags you need to use the M-command
- One Auto-message flag register for each NVM-registered slave (1-64 in MBUS3, 1-128 in MBUS4) or internal RAM-registered slave (MBUS4 only)
  Auto-message flags are set by the 'A'-command (or 'a'-command for MBUS4 slaves 129-132), and can be read back with the 'O'-command ('o'-command)
- 15 mailboxes, each 64 byte long. Mailboxes can be combined.
  Mailboxes can be written to by the 'W'-command and read back using the 'R'-command.

The RC1xxx-MBUSx modules supports 4 types of two-way communication:

- Automatically reply with a standard message
- Automatically reply with a predefined message from mailbox
- Automatically reply with a template from mailbox
- Two Level Auto Message for handling a sequence of commands

The options that come into play, and the different modes of two-way communication are discussed in the following subchapters.

7.1 Mailbox register

The MAILBOX configuration register is used to set certain features of the Auto-message Generator:

7	6	5	4	3	2	1	0
RR		TLAM	ACSM	ACMB	MBSM	DFC	A

Where:

Acronym	Description	Default Value
RR	Reserved for future use	**00**
TLAM	**Two Level Auto Message Enable** If TLAM is set, the master checks the slave reply Access Number and use two level auto-message handling. When the two-level message handler is used the auto clearing flags should not be set (ACSM=0, ACMB=0). The auto-message flags/mailbox will be cleared as a part of the two-level auto-message handler after a correct response is received from the Slave.	**0**
ACSM	**Auto Clear Standard Message** If the ACSM or ACMB bit is set, the auto-message flag will be automatically cleared after a transmission.	**1**
ACMB	**Auto Clear Mailbox** If the ACSM or ACMB bit is set, the auto-message flag will be automatically cleared after a transmission.	**0**

Acronym	Description	Default Value
MBSM	**Mailbox Special Messages** 0: Mailbox will respond to any message type (C-field). 1: Auto-message function will check incoming C-field and only reply if message is SND-NR, RSP-UD, or ACK.	**1**
DFC	**Check Data Flow Control bit in the C-field** If this bit is set, the DFC bit of the incoming message (in the C-field) will be checked before an auto-message is sent. Otherwise, it will be ignored, and the message is sent regardless of the incoming message.	**1**
A	**Check Accessibility bit** If the A bit is set, the Accessibility of the incoming message (in the Transport Layer or Application Header Configuration Word) will be checked before an auto-message is sent, sending only to meters that signals they are accessible (otherwise it will be ignored, and the message sent regardless of the incoming message).	**1**

7.2 Flag register / Auto-message flag register

Auto-message flags (set using the A –command) are used to indicate which slave shall be replied at the next access, what type of message that will be sent back, and if it is sent from the mailbox or is a standard message. The module supports two schemes of auto message replies; single level or two level.

If several auto-message flags are set, the most significant bit flag has the highest priority. The Mailbox has priority above the standard messages.

Single level scheme

In the single level scheme (MAILBOX.TLAM = 0) the message is sent from the master without regard to the previous message or Access Number of the incoming message.

By default, settings MAILBOX.ACSM and MAILBOX.ACMB are set and then the auto-message flag is automatically cleared when a reply is sent.

For each address register there is a corresponding flag register. In single level scheme this flag register is two bytes:

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
SSSSSSSS								E	D	FCB	R	MB			

Where:

SSSSSSSS	Standard Message. See Table 11 for details.
E	Encryption flag
D	Decryption flag
FCB	Frame Count Bit (to be used in the next transmission)
R	Reserved
MB	Mailbox number (1-15). 0000 = No mailbox

The E and D flags can be set by default for all register positions using the ENCRYPT and DECRYPT configuration parameters.

Two level scheme

A Frequent Access Cycle (FAC) between a Slave and a Master consist of several Master messages depending on the slave message and its Access Number in the case of lost messages. A two-level Auto-message handler can be enabled (MAILBOX.TLAM = 1) to streamline the communication with the meter in this case. Using two level Auto-messages the Master can be configured to check replies from the Slave and respond differently from the Mailbox or standard message depending on the Slave packet Access Number.

If two level Auto-message handler is used (MAILBOX.TLAM=1) an additional byte is added to the auto-message flag register and gives:

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SSSSSSSS								E	D	FCB	R	MB				**ANV**	**SMV**	**MBV**	**R**	**MB2**			

Where in addition:

ANV	**Access Number Valid** The ANV (Access Number Valid) flag is defined to tell if the Access Number should be checked by the Master before reply. (Strictly, the master transmission is not a "reply", as the Master is the primary station in unbalanced communication. But we use the term "reply" because the Slave defines the timing by its reception window). For the first Message in the Frequent Access Cycle (FAC) the ANV flag is not set, as the first reply is in response to the first initiating access (e.g., SND-NR) from the Slave.
SMV	**Standard Message Valid** The SMV and MBV flags store information about the last sent message by the Master, if it was from a Mailbox or if it was a Standard message, and type of standard message. Normally these flags shall not be altered by the user. If MBV was not set it means that a Standard Message was sent. SMV=1 means the message sent was a response to RSP-UD and msb of 0x38 in SSSSSSSS will be cleared. SMV=0 means the message sent was a response to RSP-UD or ACK and msb of 0x07 in SSSSSSSS will be cleared.
MBV	**Mailbox Valid** See SMV description.
R	Reserved for future use
MB2	Mailbox level 2 (0000-1111)

Only the LSByte nibble (MB2) should normally be altered by the 'A'-command. ANV, SMV and MBV are used internally by the module to keep track of the message sequence. These flags should be read and 'or-ed' when writing a new MB2 setting.

The flags are read by using the 'O'-command (letter "O," or lowercase 'o'). If two level Auto-message handler is used (MAILBOX.TLAM=1), the 3-byte flag register is returned, and an additional byte containing the last Access Number.

The flags are set by using the 'A'-command.

7.3 Mailboxes

With default settings mailbox messages will only be sent in reply to certain incoming message types. However, the module can also be configured (the MBSM parameter) to use the mailbox for any incoming message. The MAILBOX configuration parameter is used to set auto-clearing of flags for standard messages and the mailbox.

There are up to 15 mailboxes that can be used to store “pre-cooked” messages. Each mailbox is 64 bytes, except mailbox number 15 which is 128 bytes. Mailboxes can be combined to support messages up to 255 bytes (less the header). When using more than 64 bytes for one message, the following mailbox cannot be used. That is, if four mailboxes of 255 bytes are to be used, they should be addressed as number 1, 5, 9 and 13.

Mailboxes can be written and read using the ‘W’-command, and the ‘R’-command.

The Mailbox number should be 1-15. There is no restriction on the C-field value but do note that OMS specify only a set of allowable C-fields. The Length byte shall be the number of bytes following, not including the Length byte itself.

Long mailbox messages must be pre-encrypted in order to meet the time constraints in T mode. This is done by using the E-command. If the message holds a valid address, the Configuration Word is set for encryption, and the Encryption flag is disabled, the message will be encrypted and re-stored in the Mailbox.

Caution!! The binding (‘B’-command) and Encryption Key entry (‘K’-command) will erase the Mailbox (due to memory constraints in the module).
Note! If the message (including C and Length) is more than 64 bytes, the following mailbox cannot be used.
Note! If the Encryption flag is not disabled, the message will be encrypted (again!) when transmitted.

7.4 Automatically reply with a standard message

The table below show what slave messages and replies that can be set up using standard message autoreply. Other replies need to be set up using mailbox.

Table 11 – Standard message autoreply codes

Slave		Master		
C	Name	CI	Name	Auto message register, part: SSSSSSSS
0x46	SND-IR	0x06	CNF-IR	0b1000 0000 / 0x80
0x48	ACC-DMD	0x00	ACK	0b0100 0000 / 0x40
0x44	SND-NR	-	No reply	0b0000 0000 / 0x00
		0x5A 0x7A*)	REQ-UD1	0b0010 0000 / 0x20
		0x5B 0x7B*)	REQ-UD2	0b0001 0000 / 0x10
		-	Reserved	0b0000 1000 / 0x08
0x08 0x28	RSP-UD	-	No reply	0b0000 0000 / 0x00
		0x5A 0x7A*)	REQ-UD1	0b0000 0100 / 0x04
		0x5B 0x7B*)	REQ-UD2	0b0000 0010 / 0x02
		-	SND-NKE	0b0000 0001 / 0x01
0x00 0x20	ACK	-	No reply	0b0000 0000 / 0x00
		0x5A 0x7A*)	REQ-UD1	0b0000 0100 / 0x04

		0x5B 0x7B*)	REQ-UD2	0b0000 0010 / 0x02
		-	SND-NKE	0b0000 0001 / 0x01

) Note! The module will automatically set the Frame Count Bit in the C-field depending on the last transmitted FCB bit (as stored in the Flag Register). The FCB bit is automatically alternated.

The Status byte will automatically be set to the incoming packet RSSI value,
Access Number will be set automatically, depending on the incoming message. Before a transmission, the Access Number can be set to a new value using the N-command. A reply to SND-IR and ACC-DMD use the incoming Access Number for the reply.

7.5 Automatically reply with message from a mailbox

The Status byte will automatically be set to the incoming packet RSSI value. Normally a Standard Message is sent as an answer to SND-IR and ACC-DMD using the same AN, but even if an answer is sent from the Mailbox, the AN must be the same as the incoming message. For all other cases, the Master will set a new AN. Using the single level Auto-message handler (TLAM = 0) the Access Number is always incremented with no regard to the incoming message.

7.6 Automatically reply with template from a mailbox

If the message in the mailbox has an Application Layer Address = 0, Access Number = 0 and Status = 0, the message is called a *template*. When using a template, the blank fields will be added on-the-fly when responding to the slave. This makes it possible to point at the same template message for many meters.

The RSSI of the last received packet will automatically be added in the Status byte by the module if set to zero. The Access Number (AN) will automatically be increased for every new message transmission if set to zero in the mailbox.

Note! If the Template message requires encryption (as determined by the signature, and the Encryption enable flag is set), the time to encrypt the message on-the-fly might violate the 2-3 ms response time in T-mode. Templates with more than one block to be encrypted can only be used for S mode (up to 50 ms response time). In this case the pre-encrypted mailbox message must be used.

7.7 Two-level autoreply

Using the two-level Auto-message handler (TLAM=1) the Master AN will not be incremented if the message is repeated (because the last Master reply was not received by the Slave, the Master AN will be used again). For a new message, the Master AN is incremented automatically.

The RSSI of the last received packet will automatically be added in the Status byte by the module if set to zero.

7.8 Automatic Addressing

When the Master receives a message from a slave, this slave's address will be used as the Application Layer Address for the next transmission from the master when using the auto-message feature.

When messages are sent from the UART buffer, the Application Layer Address used is determined by the host.

Note! The Link Layer Address (in the MAC header) is always the masters own address as stored in the configuration memory. The Application Layer Address can be changed by using the 'T' command (volatile memory), or permanently changed using the 'M' command (non-volatile memory).

7.9 MBUS4: Key Challenge Function to support infinite number of installed slaves

The N2 timing includes a response time of 100 ms (fast), 1100/2100 ms (slow) or 5000 ms (extended, which is default) after the Slave message is received. This timing is handled by the Master module. Also, a sleeping Slave will wake up and listen for any incoming message after this response time. The response time is the default «extended» time unless the Extended Link Layer (ELL) request fast or slow.

Due to the long range of the N mode, a large number of meters are expected to be handled by one single master. Hence the MBUS4 Master is designed to support more than 1000 meters. Several mechanisms are used to achieve this (patented):

- 128 slaves can be registered in non-volatile memory Address and Key Registers, with corresponding Flag Register supporting auto-message generation
- 4 slaves can be registered in volatile memory Address and Key Registers (no write cycle limitations), with corresponding Flag register supporting auto-message generation
- 122 slaves can be registered in non-volatile memory Address and Key Registers, but without corresponding Flag register.
- An "infinite" number of slaves can be supported through a special protocol between the module and the host. In this case, the host must store the Addresses and Keys.

The auto-message generator can be used for the 128 slaves in non-volatile memory, and the 4 slaves in volatile memory, by using the Flag Register. The 4 slaves in volatile memory can be registered by "b" and "k" command and can be listed using the "l" command. The Flag Register can be access by "a" and "o" command (they are also mapped as Flag Registers 129-132 using the "A" and "O" command. Note, non-volatile registers 129-132 should not be used.

The 122 slaves (registers 133 to 254) in non-volatile memory without corresponding Flag Register are using ENCRYPT_FLAG and DECRYPT_FLAG in configuration memory for enabling encryption and decryption, respectively. Incoming messages from these Slaves can be decrypted using the registered key if the decrypt flag is set. But note that the auto-message generation of standard or mailbox messages are not possible for these slaves. However, using the special protocol to the host, it is still possible to respond to these slaves with the correct response time as handled by the module.

An "infinite" number of slaves can be supported in the host, only limited by the memory and processing power of the host controller, interacting with the module over a special protocol. The master module is still handling the response timing and message encryption. The same protocol can be used for the 122 slaves without flag registers.

When a slave message is received, the master module will search for the slave address in the Address register. If the slave was registered, the message will be decrypted before sent to the host (if decryption was enabled in the individual Flag register or by the DECRYPT_FLAG). If the slave address was not found, the (still encrypted) message is sent to the host. The host may now send a Key ("Key challenge" using 0xFC) to the module, and the module will decrypt the message using this Key and send the decrypted message to the host. Further, the host may now send a new message to the module. This message will be encrypted using the previously transferred Key challenge, or the Key already registered in the module (as for the 122 slaves). If a Key challenge was not sent, a Key may be sent to the module after the message was transferred. The fast/slow timing of the response is handled by the module. If no response is to be sent, the host may terminate the response cycle by sending the 0xFB command.

The “Key challenge,” or the Key following the message, is sent to the module using the 0xFC command (instead of length byte) followed by the 16 Key bytes.

Using this special protocol, the timing is important. When the Slave expect a fast response, they Key challenge, new message (and following key), must be transferred to the module within 90 ms. If a slow response is used, within 1090/2090 ms. If the extended response is used (default), 4990 ms. It is recommended to use a UART Baud rate of minimum 115 kBd to meet these timing requirements.

If a message is sent to the module during the response time cycle that does not match the last incoming message address, the response time cycle is terminated, and the message is sent as a normal message without further time delay.

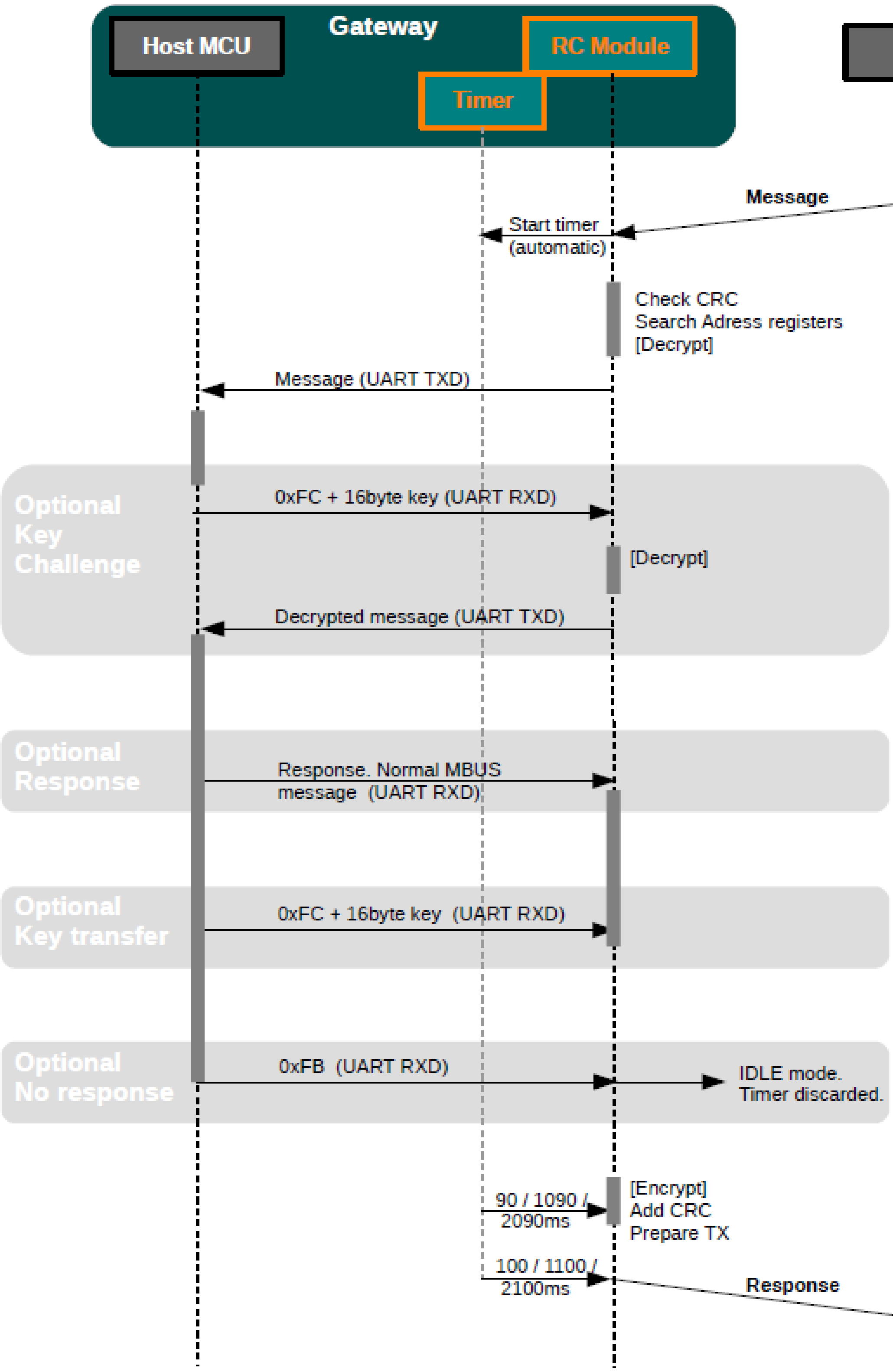

Figure 14 - Key challenge diagram

7.10 Examples

Example 5 – Write to mailbox

Example

This example shows how to write a SND-UD2 message to mailbox #1 with the W-command. Before sending the W-command you must set the module in configuration mode and wait for prompt (>).

Command	Hex	Response	Comment/Note
Enter	0x00	'>'	Enter CONFIG mode.
			Wait for '>' prompt.
			Alternative: Assert CONFIG pin.
			De-assert CONFIG after '>' prompt
'W'	0x57	'>'	W-command entered.
			Wait for '>' prompt.
1	0x01	(none)	Write to mailbox #1
	0x43		C-field SND-UD2
	0x1D		Length excluding C-field and Length byte
	0x6D		CI-field
	0x04 0x03		Address
	0x02 0x01		Address
	0xAE 0x0C		Address
	0x01 0x07		Address
	0x7D 0x00		
	0x10 0xC5		16 bytes to be encrypted, encryption
	0x2F 0x2F		method 5
	0x01 0x32		1 block (16 bytes) user data, padded with
	0x01 0x00		0x2F.
	0x00 0x00		
	0x00 0x00		
	0x00 0x00		
	0x2F 0x2F		
	0x2F 0x2F	'>'	
			Wait for '>' prompt.
(A new command can be issued here)			
'X'	0x58	(none)	Exit CONFIG mode.
			Module returns to IDLE state

Example 6 – Read mailbox

Example

This example shows how to read back from mailbox #1 with the R-command.
Before sending the R-command you must set the module in configuration mode and wait for prompt (>).

Command	Hex	Response	Comment/Note
Enter	0x00	'>'	Enter CONFIG mode. Wait for '>' prompt. *Alternative*: Assert CONFIG pin. De-assert CONFIG after '>' prompt
'R'	0x52	'>'	R-command entered. Wait for '>' prompt.
1	0x01	Contents + '>'	Command argument. Wait for '>' prompt.
(A new command can be issued here)			
'X'	0x58	(none)	Exit CONFIG mode. Module returns to IDLE state

Example 7 – Encrypt mailbox #1

Example

To encrypt the contents of mailbox #1, make sure that destination address in message matches a message that is bound to the master. Also make sure that encryption method and number of bytes to be encrypted is set in the message itself.

Command	Hex	Response	Comment/Note
Enter	0x00	'>'	Enter CONFIG mode. Wait for '>' prompt. *Alternative*: Assert CONFIG pin. De-assert CONFIG after '>' prompt
'E'	0x45	'>'	E-command entered. Wait for '>' prompt.
1	0x01	'>'	Mailbox #1 is being encrypted. Wait for '>' prompt.
(A new command can be issued here)			
'X'	0x58	(none)	Exit CONFIG mode. Module returns to IDLE state

8 Repeater Mode

The module can be configured as a Repeater by setting NETWORK_ROLE = 2. Repeater mode is supported for S, T and C mode (MBUS3), and N mode (MBUS4).

The Repeater can be configured for repeating both C and T mode messages by setting MBUS_MODE = 0x0B (C+T1). The message will be retransmitted with the same mode as originally received.

The Repeater operates as a stand-alone unit after power on, no installation is required. The Repeater should be mains powered as its receiver is working continuously. The Repeater can handle up to 15 messages simultaneously, each with a random delay (5-25s) before the retransmission.

The Repeater will repeat all SND-NR and SND-IR messages within reach after a random delay of 5-25 seconds, with the Hop bit set to 1. If it is an SND-IR message, the repeater will in addition generate an SND-NKE message after 2-5 seconds to indicate it is within range to support an installation tool. The SND-NKE message contains the RSSI level for the received SND-IR message.

The Repeater supports repetitions using the ELL or the TPL (short and long).

The Repeater generates a SND-NR "management message" every 240 minutes. This message indicates that the Repeater is alive and working.

The Repeater can be set in Installation mode by activating (setting low) the Install button input (pin 25, Development Board S5) for < 1 second. The LED driver output (pin 29, Development Board D1, red LED) will blink rapidly when in Installation mode. In Installation mode the module transmits SND-IR every 30-60 seconds. After 80 transmissions it returns to normal mode.

The installation mode can also be turned off by activating the Install button input again. The LED output will then stop toggling.

All pending messages are cleared when entering Installation mode, but the module also works as a normal repeater when in Installation mode.

In MBUS3 it is possible to choose to filter messages to be repeated by manufacturer identification code (MAN_ID). This is done by setting the configuration parameter REPEATER_MAN_ID[2:1]. The filtering function is off by default (REPEATER_MAN_ID[2:1] = 0x00 0x00).

Note; when the module has been configured as Repeater, a software watchdog will be running. In order to enter Configuration mode without the watchdog being trigged, the CONFIG pin must be activated (kept low) while the RESET pin is trigged. This will force the module to boot in Configuration mode. After the prompt ('>') has been received, the CONFIG pin can be de-asserted.

9 MBUS4 Extra Features

9.1 Antenna tuning feature

The embedded antenna tuning feature inside the MBUS4 module will make the antenna tuning for 169MHz in a customer specific design easy. See White Papers and Application Note on antenna tuning at Radiocrafts web page.

https://radiocrafts.com/uploads/WP008_Antenna_Selection_Guide.pdf

https://radiocrafts.com/uploads/AN025_Tuning_the_Antenna_with_Antenna_Tuning_Feature.pdf

9.2 Italian CIG extension of Wireless M-Bus

The Italian “CIG Interchangeability Task Force” has published UNI/TS 11291-11-4, Gas measurement systems – Hourly based gas metering systems, Part 11-4, Communication profile PM1. This is a companion standard, on top of EN 13757-4 mode N and EN 13757-3 but using DLMS / COSEM as application layer. It contains some additional requirements to EN13757-4, such as Listen Before Talk (LBT), always using Slow response time, and output power control in fine steps. This is supported by MBUS4 and can be enabled through configuration memory parameters.

To enable the output power control in fine steps, set PA_TABLE_EXTENDED > 0. This is the default value and can be set from 1 to 19 (or 20 in case of VHP variant). It is possible to change the output power by using the ‘P’ command. The output power changes approximately 3 dB for each step. See Data Sheet for details. If PA_TABLE_EXTENDED = 0, the module is backward compatible using the 5 steps in RF_POWER.

The LBT feature is enabled by setting LBT_ENABLE = 0x01. The following parameters configure the LBT: LBT_RSSI_THRESHOLD, LBT_MAX_ATTEMPT, LBT_BO_PERIOD, LBT_BO_FLAT, LBT_MAX_DELAY. These parameters should be set according to CIG recommendations.

The host controller may send 0xFA to override LBT for the following message. It applies only for one message and is automatically cleared after this one transmission.

If the LBT algorithm fails, a command/result message is sent from the module on UART. The result message is 0x01 for failure. After successful TX, the 0x00 is sent. This result message is sent always when LBT is enabled, even if it is overridden.

To always use the Slow response time (2.1s / 1.1s depending on data rate) is set by CAT1_ENABLE = 0x02. It can be combined with Category 1 receiver (see below) by setting 0x03.

***Note!* CIG only use Frame Format B, so PREAMBLE_LENGTH must be set to 0x02.**

9.3 Category 1 receiver

In some cases, it is advisable to trade off some sensitivity for better selectivity and blocking properties. In noisy environments the radio communication range is not limited by thermal noise, but interference from other strong transmitters. In this case it is possible to configure the module for Category 1 receiver settings giving increased blocking properties as specified in EN 300220. To enable this feature, set CAT1_ENABLE = 0x01. It can be combined with enforced Slow response time setting 0x03.

9.4 Wize Protocol

The Wize protocol is offered as a separate module: RC1701HP-WIZE, with separate User Manual and Data Sheet. The Wize Alliance was founded by GRDF, Suez and Sagemcom, and has specified the Wize protocol, a companion standard based on Wireless M-Bus mode N (Narrowband, 169 MHz). It is used the GRDF for gas meter reading and by Suez water meter reading. It can also be used for other Smart City and sensor reading applications.

Appendix A Configuration command details

Command						
ASCII	hex	Details	Argument	Options	Returns	Comments
‘A’	0x41	Auto-message Flags / Flag registers.	Register #	1-64 (MBUS3) 1-128 (MBUS4)	(none)	The auto-message flags are typically used for setting up automatic two-way communication. Please see chapter 7 for details. Readback functions: • ‘O’-command • ‘o’-command (RAM)
			Flags	2 or 3 bytes of flags. See chapter 7 for details on the flags	Prompt (>)	
‘a’	0x61	Auto-message Flags / Flag registers. RAM based installation addresses.	Register #	129-132 (MBUS4)	(none)	
			Flags	2 or 3 bytes of flags. See chapter 7 for details on the flags	Prompt (>)	
‘B’	0x42	Bind	Register #	1-64 (MBUS3) 1-128 + 133-254 (MBUS4)	(none)	This command is used on the master side to bind slaves to this master as a part of the Installation and binding process, described in chapter 5. The host MCU must keep overview of which registers are available and not, so that entries for already installed slaves are not unintentionally overwritten. Readback functions: • ‘L’-command • ‘l’-command (RAM)
			M_ID2 M_ID1	0x00-0x7F 0x00-0xFF	(none)	
			U_ID4 U_ID3 U_ID2 U_ID1	0x00-0xFF 0x00-0xFF 0x00-0xFF 0x00-0xFF	(none)	
			VER	0x00-0xFF	(none)	
			DEV	Please refer to MBUS standard for defined device types.	Prompt (>)	
‘b’	0x62	Bind. RAM based installation addresses.	Register #	129-132 (MBUS4)	(none)	
			Address	8 bytes slave address. See Bind for details.	Prompt (>)	

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>'C'</td><td>0x43</td><td>Channel Number
This command changes the configuration memory parameter RF_CHANNEL
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Channel</td><td>RC11xx-MBUS3: 1-10
RC17xx-MBUS4: 1-41</td><td>Prompt (>)</td></tr></table>See datasheet for channel frequencies</td><td><b><i>Caution!</i></b> The response time of this command, from command is sent to prompt is received, is set by the radio and is therefore longer than for most other commands. See Chapter 3 for details on timing.</td></tr>
<tr><td>'E'</td><td>0x45</td><td>Encrypt
This command is used to encrypt the contents of a mailbox before sending it, in order to meet the timing requirements for T2 mode.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Mailbox #</td><td>1-16</td><td>Prompt (>)</td></tr></table></td><td><b><i>Note!</i></b> The command uses address and encryption information from the message in the mailbox to look up the correct key and encryption method. Therefore, the message in the mailbox must be a properly formed MBUS message.</td></tr>
<tr><td>'F'</td><td>0x46</td><td><b>C</b>-field
This command changes the volatile parameter C_FIELD in RAM. The C_FIELD config parameter is including in outgoing messages and must not be included in the message sent over UART.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>C-field</td><td>0-255</td><td>Prompt (>)</td></tr></table></td><td>There is no restriction on the C-field value for this command, but do note that OMS specify only a set of allowable C-fields

<b><i>Note!</i></b> Using the 'F' command, the value is not stored in non-volatile memory. To do a permanent change, use the 'M' command.</td></tr>
<tr><td>'G'</td><td>0x47</td><td>M-Bus Mode
This is used to change the M-Bus Mode.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>MBUS mode</td><td><b><u>RC11xx-MBUS3:</u></b>
0x00: S2
0x01: T1
0x02: T2
0x03: S1
0x04: R
0x08: C2
0x09: C1
0x0A: T2+C
0x0B: T1+C
<b><u>RC17xx-MBUS4:</u></b>
0x10: N2 mode (default)
0x11: N1 mode (slave)</td><td>Prompt (>)</td></tr></table></td><td><b><i>Note!</i></b> Using the 'G' command, the value is not stored in non-volatile memory. To do a permanent change, use the 'M' command.</td></tr>
</table>

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>‘I’</td><td>0x49</td><td>Install
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Install mode</td><td>0x00: Filtered operation
0x01: Installation
0x02: Accept all</td><td>Prompt (>)</td></tr>
</table></td><td>This command is used on the master as part of the Installation and binding process described in Chapter 5.</td></tr>
<tr><td>‘K’</td><td>0x4B</td><td>Key
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Register #</td><td>1-64 (MBUS3)
1-128 + 133-254 (MBUS4)</td><td>(none)</td></tr>
<tr><td>Key</td><td>16-byte key</td><td>Prompt (>)</td></tr>
</table></td><td rowspan="2">Unique key for slave corresponding to address register position.
It is not possible to read back keys.</td></tr>
<tr><td>‘k’</td><td>0x6B</td><td>Key. RAM based installation addresses.
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Register #</td><td>129-132 (MBUS4)</td><td>(none)</td></tr>
<tr><td>Key</td><td>16-byte key</td><td>Prompt (>)</td></tr>
</table></td></tr>
<tr><td>‘L’</td><td>0x4C</td><td>List Binding.
This command returns the address of the slave stored at the given register number.
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Register #</td><td>1-64 (MBUS3)
1-128 + 133-254 (MBUS4)</td><td>8-byte address to slave installed at this register number.
+ Prompt (>)</td></tr>
</table></td><td>This command is used as a part of the Installation and binding process, described in chapter 5.
Setter functions:
• ‘B’-command
• ‘b’-command (RAM)</td></tr>
</table>

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>‘l’</td><td>0x6C</td><td>List Binding. RAM based installation addresses.
This command returns the address of the slave stored at the given register number.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Register #</td><td>129-132 (MBUS4)</td><td>8-byte address to slave installed at this register number.
+ Prompt (>)</td></tr></table></td><td></td></tr>
<tr><td>‘M’</td><td>0x4D</td><td>Memory Configuration
This command is used to change configuration parameters in NVM.
When you have sent the M-command you enter memory configuration mode. Multiple pairs of Address + Data can be sent before you send the 0xFF command to exit Memory configuration mode.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Address</td><td>See Table 12 for overview of configuration parameters available.</td><td>(none)</td></tr><tr><td>Data</td><td>Configuration Parameter value</td><td>(none)</td></tr><tr><td>Address</td><td>(optional)</td><td>(none)</td></tr><tr><td>Data</td><td>(optional)</td><td>(none)</td></tr><tr><td>Exit</td><td>0xFF</td><td>Prompt (>)</td></tr></table></td><td><b>Caution!</b> The M-command is writing to NVM/flash, and this operation must not be interrupted by for example a reset or power cycle. Interrupting the NVM update may leave the configuration memory corrupted.
<b>Caution!</b> The NVM also has a limited number of write cycles (10000x), and we therefore strongly recommend that you do not use the M-command “in field”, only when configuring and testing your module as part of your production test.</td></tr>
<tr><td>‘N’</td><td>0x4E</td><td>Access Number
This command sets new/specific access number.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Access #</td><td>0x00-0xFF</td><td>Prompt (>)</td></tr></table></td><td>Note! Access number is normally auto incremented.</td></tr>
</table>

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>‘O’</td><td>0x4F</td><td>Read Auto-message Flags
This command returns the auto-message flags for a given register number.
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Register #</td><td>1-64 (MBUS3)
1-128 (MBUS4)</td><td>2 or 3 bytes *) of flags
+ Prompt (>)</td></tr>
</table>
*) 3 bytes if MAILBOX.TLAM bit is set</td><td rowspan="2">The auto-message flags are typically used for setting up automatic two-way communication. Please see chapter 7 for details.
Setter functions:
• ‘A’-command
• ‘a’-command (RAM)</td></tr>
<tr><td>‘o’</td><td>0x6F</td><td>Read Auto-message Flags. RAM based installation addresses.
This command returns the auto-message flags for a given register number.
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Register #</td><td>129-132 (MBUS4)</td><td>2 or 3 bytes *) of flags
+ Prompt (>)</td></tr>
</table>
*) 3 bytes if MAILBOX.TLAM bit is set</td></tr>
<tr><td>‘P’</td><td>0x50</td><td>Output Power
This command changes the volatile parameter RF_POWER in RAM.
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>Power Level</td><td>1-5
See datasheet for output power levels.</td><td>Prompt (>)</td></tr>
</table></td><td></td></tr>
<tr><td>‘Q’</td><td>0x51</td><td>Quality Indicator
This command returns one byte indicating the signal quality of the last received packet. It is based on a soft decision signal quality of the demodulator
<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>(none)</td><td>(none)</td><td>1 byte quality number
+ Prompt (>)</td></tr>
</table></td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>‘R’</td><td>0x52</td><td>Read Mailbox
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Mailbox #</td><td>1-16</td><td>Mailbox content
+ Prompt (>)</td></tr></table></td><td>The mailboxes are typically used for setting up automatic two-way communication. Please see chapter 7 for details.

Setter function:
• ‘W’-command</td></tr>
<tr><td>‘S’</td><td>0x53</td><td>Signal Strength (RSSI)
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>(none)</td><td>(none)</td><td>1 byte quality number
+ Prompt (>)</td></tr></table>
The module provides a digital Received Signal Strength Indicator (RSSI) through the ‘S’ command or attached to the received messages (RSSI_MODE configuration parameter). The RSSI value appended to a received message is the signal strength of that received packet. The RSSI value is an 8-bit character (one byte) indicating the current input signal strength or the signal strength of the received message. The signal strength can be used as an indication of fading margin, or as a carrier sense signal to avoid collisions.

The RSSI value increases with increased input signal strength in 0.5 dB steps. Input signal strength is given by (typ.):

P = - RSSI / 2 [dBm]</td><td>The instantaneous RSSI can be read using the ‘S’ command. This reading has a delay due to the settling time of the receiver.

<b>MBUS4 only:</b>
For continuously monitoring RSSI, a faster way is to use
• ‘3’ (Test mode 3) to set RX mode
• ‘s’ (small caps) to read the RSSI multiple times
• ‘4’ (Test mode 4) to exit RX mode</td></tr>
<tr><td>‘s’</td><td>0x73</td><td>Signal Strength (RSSI), continuous
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>(none)</td><td>(none)</td><td>1 byte quality number
+Prompt (>)</td></tr></table></td><td><b>MBUS 4 only</b>
See description of ‘S’ command.</td></tr>
</table>

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>‘T’</td><td>0x54</td><td>Destination Address

This command is used to set the destination address (8 byte).

<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>M_ID2
M_ID1</td><td>0x00-0xFF
0x00-0xFF</td><td>(none)</td></tr>
<tr><td>U_ID4
U_ID3
U_ID2
U_ID1</td><td>0x00-0xFF
0x00-0xFF
0x00-0xFF
0x00-0xFF</td><td>(none)</td></tr>
<tr><td>VER</td><td>0x00-0xFF</td><td>(none)</td></tr>
<tr><td>DEV</td><td>Please refer to MBUS standard for defined device types.</td><td>Prompt (>)</td></tr>
</table>
</td><td></td></tr>
<tr><td>‘U’</td><td>0x56</td><td>Temperature monitoring

<table>
<tr><th>Argument</th><th>Options</th><th>Returns</th></tr>
<tr><td>(none)</td><td>(none)</td><td>1 byte temperature + Prompt (>)</td></tr>
</table>

The module provides readings of a digital temperature monitoring sensor (TEMP) through the ‘U’ command. The module returns an 8-bit character (one byte) indicating the current temperature in degrees Celsius (°C) followed immediately by a second character which is the prompt (‘>’). The TEMP value increases with increased temperature in 1 °C steps and has an accuracy of +/- 2 °C. The temperature is given by:

T = TEMP (dec) – 128 [°C]
(example: TEMP=0x98 equals +24 °C)</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Command</th><th></th><th></th></tr>
<tr><th>ASCII</th><th>hex</th><th>Details</th><th>Comments</th></tr>
<tr><td>‘V’</td><td>0x56</td><td>Voltage monitoring
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>(none)</td><td>(none)</td><td>1 byte temperature + Prompt (>)</td></tr></table>
The module provides readings of an internal power supply voltage monitoring sensor (VCC) through the ‘V’ command. The module returns an 8-bit character (one byte) indicating the current power supply voltage level followed immediately by a second character which is the prompt (‘>’). The command can be useful for battery power monitoring.
The VCC value increases with increased supply voltage in 30 mV/step. The power supply voltage is given by:
V = VCC (dec)*0.030 [V]
(example: VCC=0x68 equals 3.12 V)</td><td></td></tr>
<tr><td>‘W’</td><td>0x57</td><td>Write to Mailbox
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Mailbox #</td><td>1-16</td><td>Prompt (>)</td></tr><tr><td>Message</td><td>Message in accordance with standard</td><td>Prompt (>)</td></tr></table>
The format for writing to the mailbox is:
Mailbox number, C-field, Length, CI-field, followed by the rest of the message</td><td>The mailboxes are typically used for setting up automatic two-way communication. Please see chapter 7 for details.
Readback function:
• ‘R’-command</td></tr>
<tr><td>‘X’</td><td>0x58</td><td>Exit
This command exits configuration mode and sets module back in idle mode, ready to send/receive data packets.
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>(none)</td><td>(none)</td><td>(none)</td></tr></table></td><td></td></tr>
<tr><td>‘Y’</td><td>0x59</td><td>Memory Read One Byte
<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>Address</td><td>See Table 12 for overview of configuration memory</td><td>1 byte value for given address +Prompt (>)</td></tr></table></td><td></td></tr>
</table>

Command			
ASCII	hex	Details	Comments
'Z'	0x5A	Sleep This command is used to exit configuration mode and at the same time set module in sleep mode. To exit sleep mode (go to IDLE again) – send 0xFF on UART RXD. **Argument** **Options** **Returns** (none) (none) (none)	
'@RC'	0x40 0x52 0x43	Reset Configuration Memory to factory defaults This command will reset the configuration memory to the factory defaults. **Argument** **Options** **Returns** (none) (none) Prompt (>) At the same time as you run this command you must also assert the Config pin on the module. This is an extra safety to avoid unintended resets to factory default. It can be disabled by setting CONFIG_INTERFACE = 1 in the configuration memory. Serial number is retained.	**Caution!** The @RC-command is writing to NVM/flash, and this operation must not be interrupted by a reset or power cycle. Interrupting the NVM update may leave the configuration memory corrupted. **Caution!** The NVM also has a limited number of write cycles (10000x), and we therefore strongly recommend that you do not use the @RC-command "in field", only when configuring and evaluating your module in development or as part of your production test.
@'RR'	0x40 0x52 0x52	Reset This command will reset the module. It has the same effect as pulling the reset pin low. **Argument** **Options** **Returns** (none) (none) (none) At the same time as you run this command you must also assert the config pin on the module. This is an extra safety to avoid unintended resets to factory default. It can be disabled by setting CONFIG_INTERFACE = 1 in the configuration memory.	

Command						
ASCII	hex	Details	Argument	Options	Returns	Comments
‘0’	0x30	Test Mode 0 – List configuration Memory This command will list all the contents of the non-volatile configuration memory.	(none)	(none)	Configuration memory +Prompt (>)	
‘1’	0x31	Test mode 1 – TX Carrier	(none)	(none)	Prompt (>)	**Note!** Before using test mode 1 you need to send a normal package so that the TX frequency is calibrated. **Note!!** It is important to enter Test mode 3 before exiting the configuration mode (‘X’) in order to ensure proper operation in normal mode. **Note!!!** Test mode 4 must be used between Test modes 1 and 2.
‘2’	0x32	TX modulated signal, PN9 sequence	(none)	(none)	Prompt (>)	
‘3’	0x33	RX mode, TX off	(none)	(none)	Prompt (>)	
‘4’	0x34	IDLE	(none)	(none)	Prompt (>)	
‘7’	0x37	Antenna Tuning This test mode is an antenna tuning function. It will scan frequency range +/-2 MHz around the centre frequency. 5 seconds between transmissions and 200 kHz separation between test frequencies.	(none)	(none)	Prompt (>)	**MBUS 4 only** The test modes ‘7’ and ‘8’ enable frequency sweep and pulsed transmissions and can be used to tune and optimize an antenna. See AN025:Tuning the Antenna with Antenna Tuning Feature for the detailed procedure.

Command			
ASCII	hex	Details	Comments
'8'		Pulsed transmission This command will send 100ms packet for every 5 seconds. It is used to verify antenna. **Argument** / **Options** / **Returns** (none) / (none) / Prompt (>)	

Appendix B 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).

Table 12 – Configuration Memory

Parameter	Addr (hex)	MBUS3	MBUS4	Argument range	Comment	Default/ Factory Setting
					Radio Configuration	
RF_CHANNEL	0x00	✓	✓	MBUS3: 1-10 MBUS4: 1-41	Default RF channel. See data sheet for channel frequencies. Only used for R mode (MBUS3) and N mode (MBUS4).	MBUS3: 0x01 MBUS4: 0x03
RF_POWER	0x01	✓	✓	1-5	Default RF output power See data sheet for output power levels.	RC1180HP: 0x03 Others: 0x05
DATA_RATE	0x02	✗	✓		RF data rate **MBUS3** / **MBUS4** **1** 0x01 - 2.4kbps **2** 0x02 - 4.8kbps **3** 0x03 - NA **4** 0x04 - 19.2kbps **5** 0x05 - 6.4kbps	MBUS4: 0x01
MBUS_MODE	0x03	✓	✓		M-Bus mode. Use ‘G’ command to change value in volatile memory only. R mode not supported in OMS Mode 10 and 11 allows reception of both modes. **MBUS3** / **MBUS4** **0** 0x00 S2 - **1** 0x01 T1 - **2** 0x02 T2 - **3** 0x03 S1 - **4** 0x04 R - **8** 0x08 C2 **9** 0x09 C1 **10** 0x0A T2+C - **11** 0x0B T1+C - **16** 0x10 - N2 **17** 0x11 - N1	**MBUS3:** RC1170: 0x00 RC1170HP: 0x00 Others: 0x01 **MBUS4:** RC17xx: 0x10

Parameter	Addr (hex)	MBUS3	MBUS4	Argument range	Comment	Default/ Factory Setting
SLEEP_MODE	0x04	✓	✓		Sleep mode. When enabled the module enter Sleep mode after transmission (or reception). Delay set by RX_TIMEOUT If enabled with sleep timeout, the module goes directly to Sleep after a Reset, and to Sleep after TIMEOUT when wakeup from Sleep or exit Config mode. **0** 0x00 Disable sleep **1** 0x01 Enable sleep after TX **2** 0x02 Reserved **3** 0x03 Enable sleep after TX and RX **5** 0x05 Enable sleep after TX + sleep timeout **7** 0x07 Enable sleep after TX and RX + sleep timeout	0x00
RSSI_MODE	0x05	✓	✓		Append RSSI to received data. When enabled the RSSI value is appended to the received data **0** 0x00 Disabled **1** 0x01 Enabled	0x00
PA_TABLE_EXTENDED	0x06	✗	✓		0 Disables. 1-20 is on. See Data Sheet for output power levels. **0** 0x00 Extended table disabled. RF_POWER apply. **1-20** 0x01-0x14 Default power step	0x00
CAT1_ENABLE	0x07	✗	✓		**0** 0x00 Disabled **1** 0x01 Category 1 enabled **2** 0x02 Slow response time enabled **3** 0x03 Cat 1 and Slow response time	0x00
Radio Packet configuration						
PREAMBLE_LENGTH	0x0A	✓	✓		MBUS3: Short of long preamble in S mode only. MBUS4: Transmit frame format A or B **MBUS3 S-mode** / **MBUS3 Other** / **MBUS4** **0** 0x00 Short - FFA **1** 0x01 Long - N.A. **2** 0x02 - - FFB	0x00

Parameter	Addr (hex)	MBUS3	MBUS4	Argument range	Comment	Default/ Factory Setting
TIMEOUT	0x10	✓	✓	0x01-0xFE	Time before modem UART buffer is cleared without transmitting if Buffer size < Length byte (first byte). IF SLEEP_MODE=3 the same timeout is used to auto sleep slaves. Timeout(ms) = 16 * (TIMEOUT+1) Some example values: **0** 0x00 Not allowed **1** 0x01 32ms **2** 0x02 48ms **3** 0x03 64ms **124** 0x7C 2s **249** 0xF9 4s **254** 0xFE 4,064s **255** 0xFF Not allowed Modems transmit without timeout when Buffer size = length byte.	0x7C
Medium access, addressing and network management						
NETWORK_ROLE	0x12	✓	✓		0x00 (0): Slave/Meter 0x01 (1): Master/Concentrator 0x02 (2): Repeater	0x01
MAILBOX	0x16	✓	✓		Auto-message and mailbox configuration 0x20: Two Level Auto Message Enable 0x10: Auto Clear standard message flag 0x08: Auto Clear mailbox flag 0x04: Use mailbox only for special messages* 0x02: Check DFC bit before transmission 0x01: Check Accessibility before transmission *Special messages are: SND-NR, RSP-UD and ACK	
LBT_ENABLE	0x18	✗	✓		**0** 0x00 Disabled **1** 0x01 Enabled	0x00
M_ID1	0x19	✓	✓	0x00-0xFF	Manufacturer ID, first byte	0x0C
M_ID2	0x1A	✓	✓	0x00-0xFF	Manufacturer ID, second byte	0xAE
U_ID1	0x1B	✓	✓	0x00-0xFF	Unique ID, first byte	0x12
U_ID2	0x1C	✓	✓	0x00-0xFF	Unique ID, second byte	0x34
U_ID3	0x1D	✓	✓	0x00-0xFF	Unique ID, third byte	0x56
U_ID4	0x1E	✓	✓	0x00-0xFF	Unique ID, fourth byte	0x78
VER	0x1F	✓	✓	0x00-0xFF	Version	0x01
DEV	0x20	✓	✓	0x00-0xFF	Device	0x07
REPEATER_MAN_ID1	0x21	✓	✓	0x00-0xFF	MBUS3 only. Repeater filter manufacturer ID, first byte. Filter is off when both REPEATER_MAN_ID1 and REPEATER_MAN_ID2 = 0x00.	0x00
REPEATER_MAN_ID2	0x21	✓	✓	0x00-0xFF		0x00
LBT_RSSI_ THREASHOLD	0x2A	✗	✓	0x32-0x6E (50-110)	LBT RSSI threshold value measured in -dBm	0x50 (80)
LBT_MAX_ATTEMPT	0x2B	✗	✓	0x03-0x08		0x05

Parameter	Addr (hex)	MBUS3	MBUS4	Argument range	Comment	Default/ Factory Setting
LBT_BO_PERIOD	0x2C	✗	✓		In 10ms steps	0x04 Do not change
LBT_BO_FLAT	0x2D	✗	✓	0x01-0x08		0x03
LBT_MAX_DELAY	0x2E	✗	✓		In 10ms steps	0x4B (75) Do not change
Data and configuration interface, UART Serial port						
UART_BAUD_RATE	0x30	✓	✓		Baud rate. ***BE CAREFUL IF CHANGING AS HOST MAY LOOSE CONTACT WITH MODULE!*** Does not take effect until module is re-booted / reset. **0** 0x00 Not used **1** 0x01 2400 **2** 0x02 4800 **3** 0x03 9600 **4** 0x04 14400 **5** 0x05 19200 **6** 0x06 28800 **7** 0x07 38400 **8** 0x08 57600 **9** 0x09 76800 **10** 0x0A 115200 **11** 0x0B 230400	0x05
UART_FLOW_CTRL	0x35	✓	✓		UART flow control **0** 0x00 None **1** 0x01 CTS only **3** 0x03 CTS/RTS **4** 0x04 RXTX (RS485)	0x00
DATA_INTERFACE	0x36	✓	✓		Data interface Sets receiver data format. First byte is always packet length (except when using start byte) **0** 0x00 MBUS packet with ID and address **1** 0x01 Application data only **2** 0x02 Reserved **3** 0x03 Application data only with ack (0x00:0x3E) **4** 0x04 Add start/stop byte **8** 0x08 Add CRC **12** 0x0C Add start/stop byte and CRC	0x00

Parameter	Addr (hex)	MBUS3	MBUS4	Argument range	Comment	Default/ Factory Setting
CONFIG_INTERFACE	0x37	✓	✓		Configuration interface As a special protection the @ commands require the CONFIG pin to be asserted, but this can be disabled. **0** 0x00 Config pin required for @ commands **1** 0x01 Config pin NOT required for @ commands	0x00
LED_CONTROL	0x3A	✓	✓		Use to enable LED0 / LED1 for indication of different states. See section 3.5 for details. **0** 0x00 Disabled **1** 0x01 RX / TX indicator **2** 0x02 RF RX / UART RX indicator **3** 0x03 RX TXD received message accepted and sent on UART / TX indicator	0x00
CONTROL_FIELD	0x3B	✓	✓	0x00-0xFF	C-field Use 'F'-command to change value in volatile memory only	0x06
RX_TIMEOUT	0x3C	✓	✓	0x00-0xFF	Delay before Sleep mode, n x 0.6 ms Minimum 3 ms (6) in T mode, minimum 50 ms (86) in S mode. 0x08 (8) = 4.8 ms 0x56 (86) = 50 ms	0x08 Do not change
INSTALL_MODE	0x3D	✓	✓		**0** 0x00 Normal mode (accept installed MBUS meters only) **1** 0x01 Install mode **2** 0x02 Filter off (accept all MBUS types)	0x02
ENCRYPT_FLAG	0x3E	✓	✓		Default setting for encryption, enabled when set **0** 0x00 Disabled **1** 0x01 Enabled **3** 0x03 Enabled and add payload CRC	0x00
DECRYPT_FLAG	0x3F	✓	✓		Default setting for decryption, enabled when set **0** 0x00 Disabled **1** 0x01 Enabled **3** 0x03 Enabled and check payload CRC	0x00
PART_NUMBER	See appendix C	✓	✓			RC11xx(HP)-MBUS3/ RC17xxHP-MBUS4

<table>
<tr><th>Parameter</th><th>Addr (hex)</th><th>MBUS3</th><th>MBUS4</th><th>Argument range</th><th>Comment</th><th>Default/ Factory Setting</th></tr>
<tr><td>HW_REV_NO</td><td>See appendix C</td><td>✓</td><td>✓</td><td>0x00-0x09</td><td>Minimum required HW version to run this FW
Format: X.YZ
Addr 0x6E: X
Addr 0x6F: .
Addr 0x70: Y
Addr 0x71: Z

<i>Example RC11xx-MBUS3 (Minimum HW= 1.00):</i>
<table>
<tr><td><b>Addr</b></td><td>0x6E</td><td>0x6F</td><td>0x70</td><td>0x71</td></tr>
<tr><td><b>Hex</b></td><td>0x31</td><td>0x2E</td><td>0x30</td><td>0x30</td></tr>
<tr><td><b>ASCII</b></td><td>1</td><td>.</td><td>0</td><td>0</td></tr>
</table></td><td></td></tr>
<tr><td>FW_REV_NO</td><td>See appendix C</td><td>✓</td><td>✓</td><td>0x00-0x09</td><td>FW version
Format: X.YZ
Addr 0x73: X
Addr 0x74: .
Addr 0x75: Y
Addr 0x76: Z

<i>Example RC11xx-MBUS3 (FW= 3.16):</i>
<table>
<tr><td><b>Addr</b></td><td>0x73</td><td>0x74</td><td>0x75</td><td>0x76</td></tr>
<tr><td><b>Hex</b></td><td>0x33</td><td>0x2E</td><td>0x31</td><td>0x36</td></tr>
<tr><td><b>ASCII</b></td><td>3</td><td>.</td><td>1</td><td>6</td></tr>
</table></td><td></td></tr>
<tr><td>SERIAL_NUMBER</td><td>See appendix C</td><td>✓</td><td>✓</td><td></td><td>8 bytes reserved for serial number for traceability. The number is programmed by Radiocrafts during test.</td><td>0xXX</td></tr>
</table>

Appendix C Configuration Memory Defaults

RC11xx-MBUS3 default values

0x00	01	05*)	03	01**)	00	00	09	00
0x08	05	3C	00	D3	91	DA	80	80
0x10	7C	00	01	01	00	00	17	00
0x18	00	48	24	12	34	56	78	01
0x20	07	00	00	00	00	00	00	04
0x28	FF	08	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	00
0x38	00	00	00	06	08	02	00	00
0x40	FF	FF	FF	FF	FF	FF	FF	FF
0x48	FF	FF	FF	FF	FF	FF	FF	FF
0x50	00	00	00	00	00	00	00	00
0x58	00	00	00	00	00	00	00	00
0x60	00	52	43	31	31	38	30	2D
0x68	4D	42	55	53	33	2C	32	2E
0x70	30	30	2C	33	2E	31	37	00
0x78	00	00	00	00	00	00	00	00
0x80	FF	FF	FF	FF	FF	FF	FF	FF
0x88	FF	FF	FF	FF	FF	FF	FF	FF
0x90	FF	FF	FF	FF	FF	FF	FF	FF
0x98	FF	FF	FF	FF	FF	FF	FF	FF
0xA0	FF	FF	FF	FF	FF	FF	FF	FF
0xA8	FF	FF	FF	FF	FF	FF	FF	FF
0xB0	FF	FF	FF	FF	FF	FF	FF	FF
0xB8	FF	FF	FF	FF	FF	FF	FF	FF
0xC0	E0	EA	FA	07	84	84	E0	EA
0xC8	FA	47	84	C4	FF	FF	FF	FF
0xD0	FF	FF	FF	FF	FF	FF	FF	FF
0xD8	FF	FF	FF	FF	FF	FF	FF	FF
0xE0	FF	FF	FF	FF	FF	FF	FF	FF
0xE8	FF	FF	FF	FF	FF	FF	FF	FF
0xF0	FF	FF	FF	FF	FF	FF	FF	FF
0xF8	FF	FF	FF	FF	FF	FF	FF	FF

*)	RC1180HP has reduced output power as default (0x03)
**)	RC1170 and RC1170HP uses MBUS_MODE = 0x00 (S2) as default
##	Area used for device name, minimum HW version and current FW version. Value varies between modules.
##	Area used for serial number

RC1701HP-MBUS4 default values (1.08)								
0x00	03	05	01	10	00	00	00	00
0x08	05	3C	00	D3	91	DA	80	80
0x10	7C	00	01	01	00	00	17	00
0x18	00	48	24	12	34	56	78	01
0x20	07	00	00	00	00	00	00	04
0x28	FF	00	50	05	04	03	4B	00
0x30	05	08	00	01	05	00	00	00
0x38	00	00	00	44	00	02	00	00
0x40	FF	FF	FF	FF	FF	FF	FF	FF
0x48	FF	FF	FF	FF	FF	FF	FF	FF
0x50	00	00	00	00	00	00	00	00
0x58	00	00	00	00	00	00	00	00
0x60	93	93	93	93	73	03	5E	03
0x68	0D	07	07	07	1D	1D	1D	1D
0x70	20	20	20	20	10	10	10	10
0x78	0C	0C	0C	0C	10	10	10	10
0x80	01	00	00	00	09	98	98	98
0x88	2A	B0	B0	B0	15	15	15	15
0x90	E9	E9	E9	E9	43	43	43	43
0x98	00	52	43	31	37	30	31	48
0xA0	50	2D	4D	42	55	53	34	2C
0xA8	31	2E	31	30	2C	31	2E	30
0xB0	38	20	20	20	20	20	20	20
0xB8	20	00	00	00	00	00	00	00
0xC0	00	FF	FF	FF	FF	FF	FF	FF
0xC8	FF	FF	FF	FF	FF	FF	FF	FF
0xD0	FF	FF	FF	FF	FF	FF	FF	FF
0xD8	FF	FF	FF	FF	FF	FF	FF	FF
0xE0	FF	FF	FF	FF	FF	FF	FF	FF
0xE8	FF	FF	FF	FF	FF	FF	FF	FF
0xF0	FF	FF	FF	FF	FF	FF	FF	FF
0xF8	FF	FF	FF	FF	FF	FF	FF	FF

##	Area used for device name, minimum HW version and current FW version
##	Area used for serial number

Appendix D Additional Examples

Example 8 – Changing RF Channel

Example

To select RF channel 4, send the follow sequence after asserting the CONFIG line and the '>' prompt is received:

Command	Hex	Response	Comment/Note
Enter	0x00	'>'	Enter CONFIG mode. Wait for '>' prompt. *Alternative*: Assert CONFIG pin. De-assert CONFIG after '>' prompt
'C'	0x43	'>'	C-command entered. Wait for '>' prompt.
4	0x04	'>'	Command argument(s). Wait for '>' prompt.
(A new command can be issued here)			
'X'	0x58	(none)	Exit CONFIG mode. Module returns to IDLE state

Example 9 – Changing MAN_ID

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	'>'	Enter CONFIG mode. Wait for '>' prompt. *Alternative*: Assert CONFIG pin, then de-assert CONFIG pin after '>' prompt.
'M'	0x4D	'>'	M-command entered. Wait for '>' prompt. Module ready to receive address.
0x19	0x19	(none)	Address
100	0x64	(none)	Value
0x1A	0x1A	(none)	Address
200	0xC8	(none)	Value
(New address and value pairs can be sent here)			
0xFF	0xFF	'>'	End M-command Wait for '>' prompt.
(A new configuration command can be issued here)			
'X'	0x58	(none)	Exit CONFIG mode. 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.

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.
1.20	Added MBUS3
1.21	Clarified order of address bytes for binding.
1.22	MBUS3 updates before official release.
1.30	MBUS2 new features for FW 2.20: Auto sleep functionality update for slaves. 8 Byte serial number reservations in configuration memory. 1 byte frequency tolerance calibration value in configuration memory.
1.31	- Changed terminology from RC1180-MBUSx to MBUSx - Included quick start chapter (and removed it from data sheet 2.20) - RSSI reading included (and removed it from data sheet rev 2.20) - Optional custom specific version chapter included for MBUS2 - SLEEP_MODE, TIMEOUT and LED CONTROL update for MBUS2
1.40	MBUS3 new features for FW 3.09: 8 Byte serial number reservations in configuration memory. 1 byte frequency tolerance calibration value in configuration memory. -SLEEP_MODE, TIMEOUT and LED CONTROL update
1.50	- Correction of SERIAL_NUMBER location - Correction PART_NUMBER, HW_REV_NO and FW_REV_NO location - info about only HEADER packets sending (L=0xFE) included - Auto sleep info for MBUS3 Included. - U, V and Y command info included - Appendix 1&2 update - Two level auto-message handler info included (New Feature FW 3.11)
1.60	- MBUS4 Description added - MBUS3 Description for C1 mode added
1.71	Additional information and corrections for MBUS3 and MBUS4 versions
1.72	Updated RF_CHANNEL frequencies
1.73	-Updated MBUS4 description with CIG features; extended power table, LBT and Category 1 receiver - Minor corrections
1.75	Added description of new features in FW 1.05 (MBUS4) More channels (11-41), new data rate (5) and new test modes ('7' and '8') added for MBUS4
2.01	Rewritten to new company standard.
2.02	Updated information to do with Automatic Standard Message generation
2.03	Clarified LED_CONTROL functionality and updated according to latest implementation
2.04	Corrected settings for C+T1 mode = 11 = 0x0B, and C + T2 mode = 10 = 0x0A. Added information on using the Repeater in C + T1 mode, and how to enter Configuration mode while acting as a Repeater (Section 8). Added notes on the use of CTS and RTS.
2.05	Corrected reference, grammar, and spelling errors
2.06	Updated Configuration Memory for MBUS4

Disclaimer

Radiocrafts AS believes the information contained herein is correct and accurate at the time of this printing. However, Radio crafts 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 availabl e 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 syste m 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 indemnif y Radiocrafts AS for any damages resulting from any improper use or sale.

Radiocrafts Support:

Knowledge base:	https://radiocrafts.com/knowledge-base/
Application notes library:	https://radiocrafts.com/resources/application-notes/
Whitepapers:	https://radiocrafts.com/resources/articles-white-papers/
Technology overview:	https://radiocrafts.com/technologies/
RF Wireless Expert Training:	https://radiocrafts.com/resources/rf-wireless-expert-training/

Contact Radiocrafts

Sales requests: https://radiocrafts.com/contact/