Radiocrafts
Embedded Wireless Solutions

RC232™ User Manual

RC12xx
RC11xx(HP)-RC232
RC2500(HP)-RC232
RC17XX(HP)-RC232

Table of Contents

Table of Tables

Table of Figures

Abbreviations

Abbreviation	Description
RSSI	Received Signal Strength Indicator

1 Introduction

1.1 Quick Start

The RC232™ Module has a UART interface that is used for both packet data and module configuration. Use the UART format with settings (19200, 8, 1, N, no flow control).

How do I configure the RF power, RF channel or any other parameter?

To change configurable parameters:

1. Assert the CONFIG-pin by pulling it low. This will take the module into configuration mode. More about configuration mode is found in chapter 2.1
2. Use special commands to access the configuration registers and test modes. All the commands for the configuration are found in chapter 2.2
3. Exit from configuration mode by sending the 'X' command.
4. Start using the module as normal in packet mode. In some cases you need to reset the module for the changes to take effect. Details about sending and receiving packet data is found in chapter 2.3.

How do I transmit data?

Send your data to the RXD pin on the module when the module is in normal mode (not CONFIG mode)

How do I receive data?

Any received RF data packet addressed to the receiving module will be output on the TXD pin when the module is in normal mode (not during SLEEP or CONFIG modes)

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 Differencies between RC12xx, RC11xx, RC25xx and RC17xx series

Table 1 – RC232 Feature Sets Overview

Feature List	Feature Set			
	RC12xx	RC11xx(HP)	RC2500(HP)	RC17xx(HP)
Radio	Sub 1GHz, narrowband	Sub 1GHz	2.4GHz	Sub 1GHz, narrowband
Data buffer length *)	128	1024	1024	128
4-byte addressing option	No	Yes	Yes	No
Encryption	No	No	No	Yes
LED Control	No	Yes	Yes	Yes
Serial number	No	No	No	Yes

*) The maximum packet length is depending on both the data buffer length, addressing mode, and whether CRC is enabled or not.

An overview of the command set used to configure the modules are found in chapter 2.2 and the commands are further described in Appendix A .

1.3 Optional custom specific version

Custom variants of the RC232 protocol to meet certain requrements may be available. Please contact Radiocrafts.

2 Basic Functionality

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

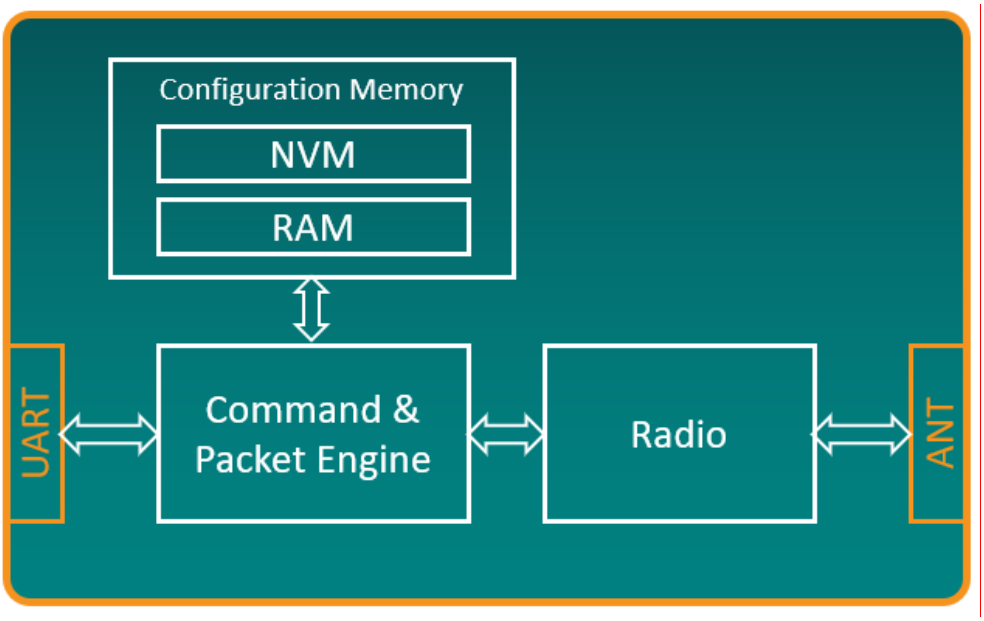


Figure 1 – Block diagram

The host use the UART Interface to send and receive data. The UART packet content, like the addition of RSSI, CRC and start/stop bytes, can be changed in the configuration mode.

When the module receives a packet with correct destination address over RF it will send the packet over the UART interface on the TXD line. When the host MCU wants to transmit a packet over the RF, it must send the packet through the UART interface on the RXD line.

Figure 2 – RC232 point to point Link

The addressing scheme provide point to opint and point to multipoint (broadcast) options.

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 by asserting the CONFIG pin (set low).

In configuration mode the module will respond by sending a '>' prompt on the TXD pin. This indicates that the module is ready to receive commands. The CONFIG pin (if used) can then be de-asserted.

***Note!* 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.

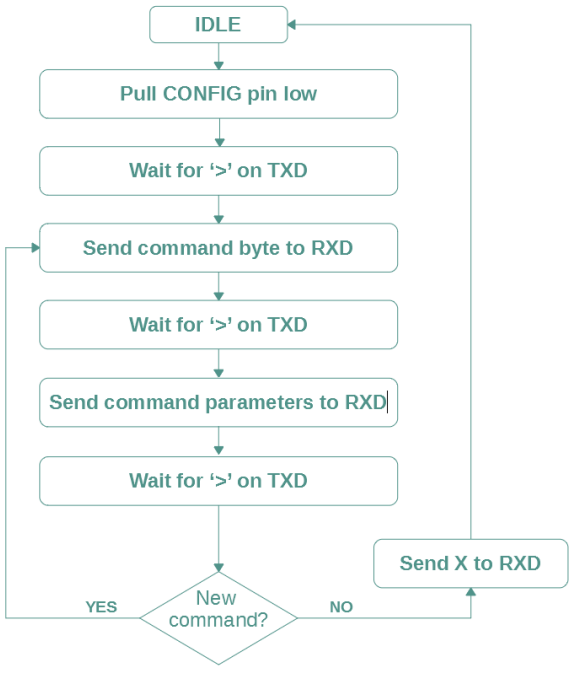


Figure 3 – Configuration mode flow diagram

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 details for each command is found in Appendix A .***

Table 2 – Configuration Commands overview

Command			RC12xx	RC11xx	RC25xx	RC17xx	Stored in NVM (Flash)	Stored in RAM
ASCII	hex	Parameter						
'C'	0x43	Channel	✓	✓	✓	✓	✗	✓
'M'	0x4D	Memory configuration	✓	✓	✓	✓	✓	✗
'P'	0x50	Output power	✓	✓	✓	✓	✗	✓
'S'	0x53	Signal Strength (RSSI)	✓	✓	✓	✓	✗	✓
'T'	0x54	Destination address	✓	✓	✓	✓	✗	✓
'U'	0x55	Temperature monitoring	✓	✓	✓	✓	✗	✓
'V'	0x56	Voltage monitoring	✓	✓	✓	✓	✗	✓
'X'	0x58	Exit command	✓	✓	✓	✓	✗	✓
'Y'	0x59	Memory Read one byte	✓	✓	✓	✓	✗	✓
'Z'	0x5A	Sleep mode	✓	✓	✓	✓	✗	✓
'0'	0x30	List Configuration Memory	✓	✓	✓	✓	✗	✓
'1'	0x31	Test mode 1	✓	✓	✓	✓	✗	✓
'2'	0x32	Test mode 2	✓	✓	✓	✓	✗	✓
'3'	0x33	Test mode 3	✓	✓	✓	✓	✗	✓
'4'	0x34	Test mode 4	✓	✓	✓	✓	✗	✓
'@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. It is possible to configure different send criterias for the data stream:

- PACKET_LENGTH will initiate the radio to send when the specified number of bytes are received.
- PACKET_END_CHARACTER will initiate the radio to send when end character is received.
- PACKET_TIMEOUT will initiate the radio to send the received data after a given timeout. The timeout is started when the first byte is recieved on RXD.

Likewise, when data is received they are stored in the module before they are sent to the host. This allows the communication controller to add address information, CRC and encryption during transmission, and to do error check and decryption of the received data.

The RC232™ protocol has a built in error detection based on a 16 bit CRC. The error detection can be enabled with CRC_MODE in the configuration memory. If a received packet contains an invalid checksum, it will be discarded and not sent to the host.

The module has an internal buffer of 128 or 1024 bytes that determines the maximum length of the RF packet (over-the-air). The maximum UART packet length is determined by this limit, but also by addressing mode and CRC enabled/disabled.

Table 3 – Maximum UART packet length (default setting highlighted)

No addressing	2byte addressing	4byte addressing	CRC enabled	RC12xx RC17xx(HP)	RC11xx(HP) RC25xx(HP)
✓	✗	✗	✗	127	1023
✓	✗	✗	✓	125	1021
✗	✓	✗	✗	125	1021
✗	✓	✗	✓	**123**	**1019**
✗	✗	✓	✗	N.A.	1015
✗	✗	✓	✓	N.A.	1013

The PACKET_LENGTH configuration parameter is default set to 128 (0x80).

2.4 UART Flow control

Normally no flow control (handshake) is used. Any microcontroller with hardware or software UART or UART converter communicating with the module in its configurable UART speed can be used to communicate with the module.

Optionally the CTS and/or RTS/RXTX can be used for hardware flow control:

CTS pin – Clear to send: The low-asserted CTS pin provides flow control for the module. When CTS is asserted (low), serial data can be sent to the module for RF transmission. If the module is busy, like during RF data transmission or reception, the CTS pin will be de-asserted (high) to stop any data transfer to the module.

RTS pin – Ready to send: When RTS is asserted (low) the host allow data to be sent from the module to the host. The host can stop the module from sending data by de-asserting (high) the RTS signal. Note that if the module has data waiting in the receive buffer, it will not be able to receive or transmit further data until the RTS has been asserted and the data in the buffer is transferred to the host.

RXTX pin – RS485 driver control: RXTX is low when the module can receive data on RXD. RXTX is high when the module is transmitting data on TXD and additionally 5 ms for the module to turn from TXD to IDLE mode (see Timing Information in the module data sheet). The RXTX pin is normally connected to the /RE and DE pins on the RS485 driver circuit.

The configuration of the flow control for the UART interface is done by changing UART_FLOW_CTRL in the non-volatile configuration memory.

Note! The module CTS is set up during the first stop bit sent from to module when the buffer is full, and the host should then halt further character transmissions to prevent character loss. If the host cannot detect the CTS quickly enough during hardware handshake, it should be configured for two stop bits.

2.5 Network Topology and Addressing Modes

The RC232 devices from Radiocraft supports both point to point (addressed) and point to multipoint (broadcast) communication. The configuration parameters used for this are:

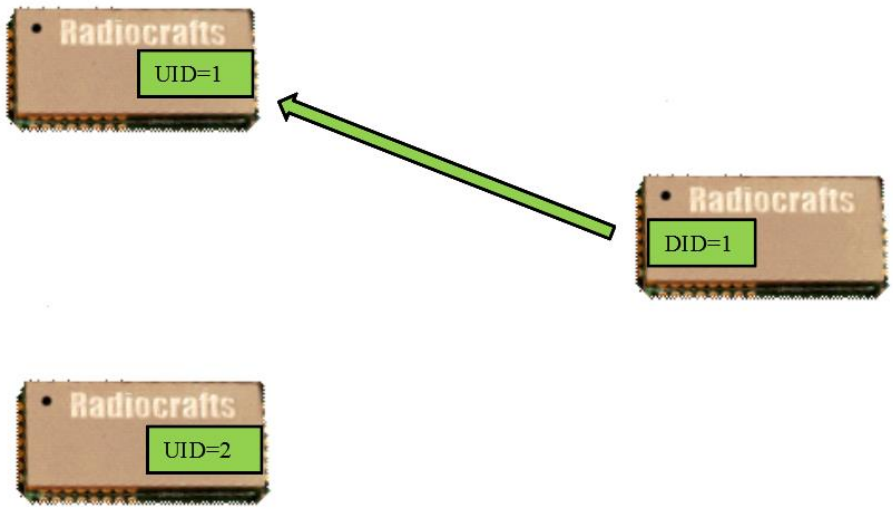
SYSTEM_ID (SID)	All the nodes in one system should have the same SYSTEM_ID. If ADDRESS_MODE = 0x02, SID is 1 byte and SID1 is used. If ADDRESS_MODE = 0x08, SID is 4 byte and SID[4:1] is used.
UNIQUE_ID (UID)	Each device within a system must have it's own UID. If ADDRESS_MODE = 0x02, UID is 1 byte and UID1 is used. If ADDRESS_MODE = 0x08, UID is 4 byte and UID[4:1] is used.
DESTINATION_ID (DID)	This is the Unique ID (UID) of the module you want to transfer to. Set to 0xFF if you want to broadcast to all nodes within the system. If ADDRESS_MODE = 0x02, DID is 1 byte and DID1 is used. If ADDRESS_MODE = 0x08, DID is 4 byte and DID[4:1] is used.
BROADCAST_ID (BID)	This is default set to 255 (0xFF). All devices within a broadcast group must have the same BID.
ADDRESS_MODE	The ADDRESS_MODE parameter controls how the system uses the SID, UID, DID and BID to control the communcation: ADDRESS_MODE = 0x00 means addressing is off. All modules that «hear» the message will output it. ADDRESS_MODE = 0x02 means 1 byte addressing. Only receivers that have SID and UID (or SID + BID) matching the DID in the transferred message will output it. ADDRESS_MODE = 0x08 means 4 byte addressing. Otherwise similar to ADDRESS_MODE = 0x02.

For addressing to work properly make sure:

- All nodes have addressing enabled (ADDRESS_MODE)
- All nodes within the system have the same SYSTEM_ID
- All nodes within the system have the same BROADCAST_ID
- Each node within the system has one unique UNIQUE_ID

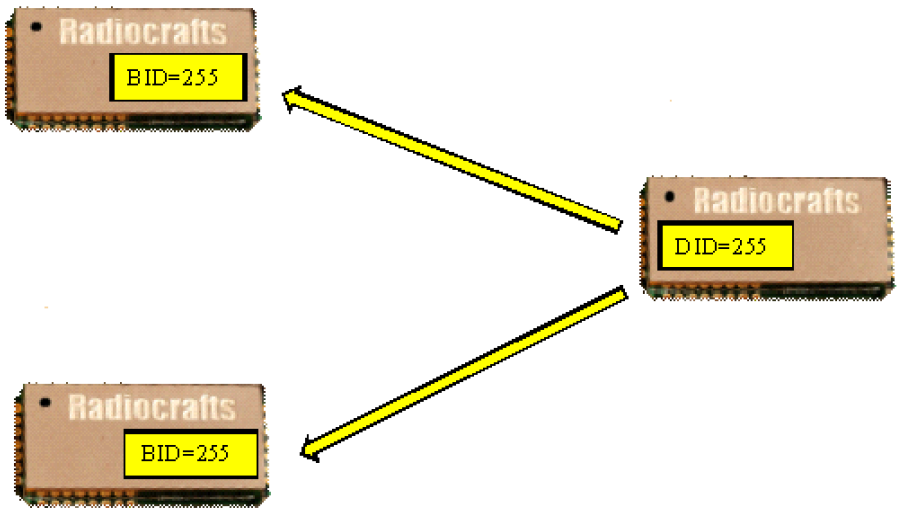
Point to point

To send a packet to a specific node, set the destination address to the specific node's UNIQUE_ID.



Broadcast mode

If a broadcast is to be made, set the destination address to the BROADCAST_ID. By default the BROADCAST_ID is 0xFF (decimal 255), but this can be changed in the configuration. Remember that the BROADCAST_ID cannot be the same as any UNIQUE_ID.



3 Timing

The timing information varies between module families and can be found in their respective datasheets.

4 Power Management

There are different ways to minimize the total current consumption and thus extend the battery lifetime for battery operated devices. What determines the battery lifetime is determined by several factors, for example how often you send packages. But it is also crucial to be able to set the module in a state where it draws very little current when you are not sending. It is also important that the low power mode retains settings so that you spend a minimum time “on” every time you wake up the module to send a packet.

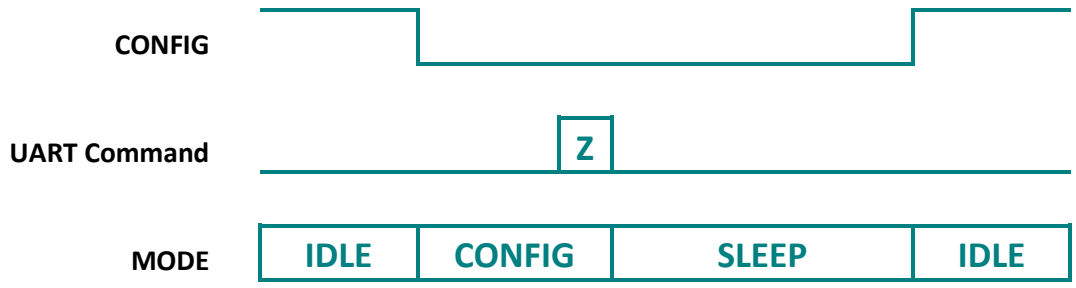
The Radiocrafts modules offer several configuration options to allow the module to be “off” for an extensive part of the time:

- SLEEP_MODE = 0x00, using Config-pin and Z-command to control SLEEP mode.
- SLEEP_MODE = 0x02, using RTS/SLEEP pin to control SLEEP mode.
- Turn off module completely (not recommended) except for RC12xx.

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

4.1 SLEEP_MODE = 0x00

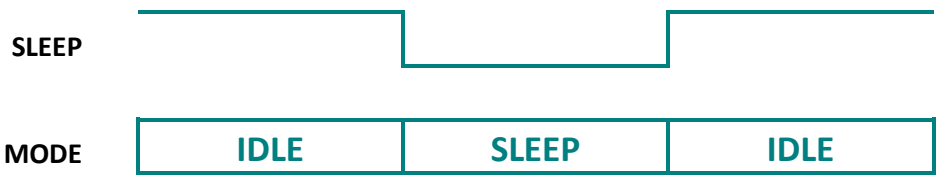
If SLEEP_MODE = 0x00, you can enter the low current SLEEP mode by sending the Z-command while in config mode and still keeping the CONFIG pin low. Exit SLEEP by releasing the config pin as shown in the diagram below.



Note! During SLEEP the CONFIG pin does not have any internal pull-up, so the CONFIG pin must be driven high in order to wake the module. Any activity on the RXD pin will make the module wake up, but immediately return to Sleep as long as CONFIG is kept low. Such activity on the RXD pin should be avoided in order to reduce current consumption. All configuration settings and RAM values are retained during SLEEP.

4.2 SLEEP_MODE = 0x02

With SLEEP_MODE = 0x02 you can enter and exit the low current SLEEP mode by pulling the SLEEP pin low like the diagram below shows. SLEEP_MODE = 0x02 must not be used together with UART flow control.



Note! During Sleep the SLEEP pin does not have any internal pull-up, so the RTS/SLEEP pin must be driven high in order to wake up the module. Any activity on the RXD or CONFIG pins will not make the module wake up. Activity on module pin 16 and 17 can wake the module and must be avoided (do not connect). CONFIG must be

high when awakening the module to avoid setting the module directly in configuration mode. All configuration settings and RAM values are retained during SLEEP.

4.3 Power cycling

All configuration settings in nonvolatile memory is restored, but values in RAM are overwritten with default settings.

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

For RC12xx – please see the datasheet.

5 Encryption and decryption (RC17xxHP only)

The RC232 protocol can encrypt data before transmission, and decrypt at reception. The CRC is used to verify correct decryption, so the 2 byte CRC (CRC_MODE = 2) should be enabled when using encryption. Also, both the ENCRYPT_FLAG and DECRYPT_FLAG must be set for a successful link.

The encryption method used is AES-128-CTR mode. Using the counter mode means that any packet length can be encrypted without the use of filler bytes. Both the Initialization Vector (Initial Counter Block) and the Encryption Key are fixed. AES uses a symmetric key. That is, the same Encryption Key is used for both encryption and decryption. Also the Initialization Vector must be the same. These are stored as INIT_VECTOR and DEFAULT_KEY, both 16 bytes long.

5.1 Enabling Encryption and decryption

You need to set both ENCRYPT_FLAG and DECRYPT_FLAG to have a successfull link.

Appendix A Configuration command details

<table>
<tr><th colspan="2">Command</th><th rowspan="2">Details</th><th rowspan="2">Comments</th></tr>
<tr><th>ASCII</th><th>hex</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>See datasheet for details.</td><td>Prompt (>)</td></tr></table>
See datasheet for channel frequencies</td><td>Caution! 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>‘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 4 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>Caution! The M-command is writing to NVM/flash, and this operation must not be interrupted by for example a reset or powercycle. Interrupting the NVM update may leave the configuration memory corrupted.

Caution! The NVM also has a limited number of writecycles (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>‘P’</td><td>0x50</td><td>Output Power
This command is used to change the volatile paramter 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>
</table>

<table>
<tr><th colspan="2">Command</th><th rowspan="2">Details</th><th rowspan="2">Comments</th></tr>
<tr><th>ASCII</th><th>hex</th></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 provide 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.</td></tr>
<tr><td>‘T’</td><td>0x54</td><td>Destination Address
This command is used to set the destination address. It will set 1 byte or 4 bytes depending on ADDRESS_MODE configuration parameter. It will output a prompt character after each byte sent in, so if sending in 4 bytes, you must wait for prompt between each byte.

<table><tr><th>Argument</th><th>Options</th><th>Returns</th></tr><tr><td>DID</td><td></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 rowspan="2">Details</th><th rowspan="2">Comments</th></tr>
<tr><th>ASCII</th><th>hex</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>‘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 4 for overview of configuration memory</td><td>1 byte value for given address +Prompt (>)</td></tr></table></td><td></td></tr>
<tr><td>‘Z’</td><td>0x5A</td><td>Sleep
This command is used to exit configuration mode and enter the low power SLEEP state when SLEEP_MODE = 0x00. CONFIG pin must be held LOW while issuing this command. To exit sleep mode, release CONFIG pin.

<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>
</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>‘@RC’</td><td>0x40
0x52
0x43</td><td>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.</td><td>Caution!
The @RC-command is writing to NVM/flash, and this operation must not be interrupted by for example a reset or powercycle. Interrupting the NVM update may leave the configuration memory corrupted.

Caution! The NVM also has a limited number of writecycles (10000x), and we therefore strongly recommend that you do not use the @RC-command “in field”, only when configuring and testing your module as development or as part of your production test.</td></tr>
<tr><td>@’RR’</td><td>0x40
0x52
0x52</td><td>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.</td><td></td></tr>
<tr><td>‘0’</td><td>0x30</td><td>Test Mode 0 – List configuration Memory
This command will list all the contents of the non-volatile configuration memory.

Argument | Options | Returns
(none) | (none) | Configuration memory +Prompt (>)</td><td></td></tr>
<tr><td>‘1’</td><td>0x31</td><td>Test mode 1 – TX Carrier

Argument | Options | Returns
(none) | (none) | Prompt (>)</td><td>Note! Before using test mode 1 you need to send a normal package so that</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>‘2’</td><td>0x32</td><td>TX modulated signal, PN9 sequence

Argument | Options | Returns
(none) | (none) | Prompt (>)</td><td>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.</td></tr>
<tr><td>‘3’</td><td>0x33</td><td>RX mode, TX off

Argument | Options | Returns
(none) | (none) | Prompt (>)</td><td></td></tr>
<tr><td>‘4’</td><td>0x34</td><td>IDLE. Radio OFF.

Argument | Options | Returns
(none) | (none) | Prompt (>)</td><td></td></tr>
</table>

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 4 – Configuration Memory

Parameter	Addr (hex)	RC12xx	RC11xx(HP)	RC2500(HP)	RC17xx(HP)	Argument range	Comment	Default/ Factory Setting
Radio Configuration								
RF_CHANNEL	0x00	✓	✓	✓	✓	See Data sheet	Default RF channel for R mode only See data sheet for channel frequencies.	See Data sheet
RF_POWER	0x01	✓	✓	✓	✓	See Data sheet	Default RF output power See data sheet for output power levels.	See Data sheet
RF_DATA_RATE	0x02	✓	✓	✓	✓	See Data sheet	RF data rate See Data sheet for details.	See Data sheet
SLEEP_MODE	0x04	✗	✓	✓	✓	0x00 or 0x02	Configure how to enter SLEEP mode **0** 0x00 Use Z-command from CONFIG to enter and exit SLEEP **2** 0x02 Use RTS/SLEEP pin to enter and exit SLEEP	0x00
RSSI_MODE	0x05	✓	✓	✓	✓	0x00 - 0x01	Append RSSI to received data. When enabled the RSSI value is appended to the received data **0** 0x00 Disabled **1** 0x01 Enabled	
PACKET_LENGTH_H	0x0E	✗	✓	✓	✗		Max. packet length. When buffer is full, modem will transmit data. When the max packet lenght is 1023 bytes, PACKET_LENGTH = PACKET_LENGHT_H:PACKET_LENGHT_L. When PACKET_LENGTH_H is not available, the maximum packet length is 127.	0x00
PACKET_LENGHT PACKET_LENGTH_L	0x0F	✓	✓	✓	✓			0x80

Parameter	Addr (hex)	RC12xx	RC11xx(HP)	RC2500(HP)	RC17xx(HP)	Argument range	Comment	Default/ Factory Setting
PACKET_TIMEOUT	0x10	✓	✓	✓	✓	0x00 - 0xFE	Time before modem timeout and transmit buffered data. **0** 0x00 Disabled **1** 0x01 32ms **2** 0x02 48 ms **3** 0x03 64 ms **124** 0x7C 2s (default) **249** 0xF9 4s None means packet timeout is disabled (not 0 s). Use packet length or end character instead. Timeout value is (PACKET_TIMOEOUT x 16 ms) + 0/16 ms min/max 0xFE (254) is max, giving 4.080 sec.	0x7C
PACKET_END_ CHARACTER	0x11	✓	✓	✓	✓		Packet end character **0** 0x00 None (default) **10** 0x0A LF **13** 0x0D CR **90** 0x5A 'Z'	0x00
MAC_MODE	0x13	✓	✗	✗	✗		 **0** 0x00 Transparent **1** 0x01 Buffered Transparent means using RXEN and TXEN	0x01

Parameter	Addr (hex)	RC12xx	RC11xx(HP)	RC2500(HP)	RC17xx(HP)	Argument range	Comment	Default/ Factory Setting
ADDRESS_MODE	0x14	✓	✓	✓	✓		Set adressing mode. Using addressing adds the SID and DID bytes to the radio packet. **0** 0x00 No addressing **2** 0x02 One byte addressing for SID, UID, DID and Broadcasting (default) **8** 0x08 4 byte adressing for SID, UID, DID and Broadcasting ***Note! This option is not available in RC12xx and RC17xx*** **ADDRESS_MODE=2** UID=UID1 SID=SID1 DID=DID1 BID=BID **ADDRESS_MODE=8** UID=UID4:UID3:UID2:UID1 SID=SID4:SID3:SID2:SID1 DID=DID4:DID3:DID2:DID1 BID=BID:BID:BID:BID	
CRC_MODE	0x15	✓	✓	✓	✓		CRC_MODE **0** 0x00 None **2** 0x02 CRC16 (default)	
UID1	0x19	✓	✓	✓	✓	0x00 - 0xFE	Unique ID 1	0x01
UID2	0x1B	✗	✓	✓	✗		Unique ID 2	0x01
UID3	0x1D	✗	✓	✓	✗		Unique ID 3	0x01
UID4	0x1F	✗	✓	✓	✗		Unique ID 4	0x01
SID1	0x1A	✓	✓	✓	✓	0x00 - 0xFE	System ID 1	0x01
SID2	0x1C	✗	✓	✓	✗		System ID 2	0x01
SID3	0x1E	✗	✓	✓	✗		System ID 3	0x01
SID4	0x20	✗	✓	✓	✗		System ID 4	0x01
DID1	0x21	✓	✓	✓	✓	0x00 - 0xFF	Destination ID 1	0x01
DID2	0x22	✗	✓	✓	✗		Destination ID 2	0x01
DID3	0x23	✗	✓	✓	✗		Destination ID 3	0x01
DID4	0x24	✗	✓	✓	✗		Destination ID 4	0x01
BID	0x28	✓	✓	✓	✓	0xFF	Broadcast Address	0xFF

Parameter	Addr (hex)	RC12xx	RC11xx(HP)	RC2500(HP)	RC17xx(HP)	Argument range	Comment	Default/ Factory Setting
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 Not used** 1 0x01 600 2400 2 0x02 1200 4800 3 0x03 42400 9600 4 0x04 4800 14400 5 0x05 9600 19200 6 0x06 19200 28800 7 0x07 N.A. 38400 8 0x08 N.A. 57600 9 0x09 N.A. 76800 10 0x0A N.A. 115200 11 0x0B N.A. 230400	0x05 RC12xx: 0x06
UART_FLOW_ CONTROL	0x35	✓	✓	✓	✓		UART flow control **0** 0x00 None **1** 0x01 CTS only **3** 0x03 CTS/RTS **4** 0x04 RXTX (RS485)	0x00
DATA_INTERFACE	0x36	✓	✗	✗	✗		When transparent mode is selected for MAC_MODE use synchronous interface **0** 0x00 UART using RXD and TXD **1** 0x01 Synchronous interface (SDA, SCL) using RXEN and TXEN for direction control.	
LED_CONTROL	0x3A	✗	✓	✓	✓	0x00 - 0x01	Use to enable LED0/LED1 for RX/TX packet indication. **0** 0x00 Disabled **1** 0x01 RX/TX indicator	0x00
ENCRYPT_FLAG	0x3E	✗	✗	✗	✓		Default setting for encryption, enabled when set **0** 0x00 Disabled **1** 0x01 Enabled	

<table>
<tr><th>Parameter</th><th>Addr (hex)</th><th>RC12xx</th><th>RC11xx(HP)</th><th>RC2500(HP)</th><th>RC17xx(HP)</th><th>Argument range</th><th>Comment</th><th>Default/ Factory Setting</th></tr>
<tr><td>DECRYPT_FLAG</td><td>0x3F</td><td>✗</td><td>✗</td><td>✗</td><td>✓</td><td></td><td>Default setting for decryption, enabled when set

0 0x00 Disabled
1 0x01 Enabled</td><td></td></tr>
<tr><td>DEFAULT_KEY</td><td>0x40-0x4F</td><td>✗</td><td>✗</td><td>✗</td><td>✓</td><td></td><td>Used for Encryption and decryption. 16 bytes.</td><td>All 0xFF</td></tr>
<tr><td>INIT_VECTOR</td><td>0x50-0x5F</td><td>✗</td><td>✗</td><td>✗</td><td>✓</td><td></td><td>Used for Encryption and decryption. 16 bytes.</td><td>All 0x00</td></tr>
<tr><td>PART_NUMBER</td><td>See table in comments</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td></td><td>Name of modele, reflecting both hardware and firmware variant.

<table><tr><th>Module series</th><th>Start</th><th>Stop</th></tr><tr><td>RC12xx</td><td>0x3C</td><td>0x3F</td></tr><tr><td>RC11xx/RC11xxHP</td><td>0x3C</td><td>0x3F / 0x41</td></tr><tr><td>RC2500/RC2500HP</td><td>0x3C</td><td>0x3F / 0x41</td></tr><tr><td>RC17xx/RC17xxHP</td><td>0x89</td><td>0x94 / 0x96</td></tr></table></td><td>RC11xxRC232
RC17xxHP-RC232
...and so on.</td></tr>
<tr><td>HW_REV_NO</td><td>See table in comments</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td></td><td>Minimum required HW version to run this FW
Format: X.YZ
Addr 0x6E: X
Addr 0x6F: .
Addr 0x70: Y
Addr 0x71: Z

Example (Minimum HW= 3.00):
<table><tr><td>Addr</td><td>0x6E</td><td>0x6F</td><td>0x70</td><td>0x71</td></tr><tr><td>Hex</td><td>0x33</td><td>0x2E</td><td>0x30</td><td>0x30</td></tr><tr><td>ASCII</td><td>3</td><td>.</td><td>0</td><td>0</td></tr></table>
<table><tr><th>Module series</th><th>Start</th><th>Stop</th></tr><tr><td>RC12xx</td><td>0x43</td><td>0x46</td></tr><tr><td>RC11xx/RC11xxHP</td><td>0x49 / 0x4B</td><td>0x4C / 0x4E</td></tr><tr><td>RC2500/RC2500HP</td><td>0x49 / 0x4B</td><td>0x4C / 0x4E</td></tr><tr><td>RC17xx/RC17xxHP</td><td>0x96 / 0x98</td><td>0x99 / 0x9B</td></tr></table>
Note! Actual HW revision marked on the module label may be higher than the revision indicated here.</td><td></td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Addr (hex)</th><th>RC12xx</th><th>RC11xx(HP)</th><th>RC2500(HP)</th><th>RC17xx(HP)</th><th>Argument range</th><th>Comment</th><th>Default/ Factory Setting</th></tr>
<tr><td>FW_REV_NO</td><td>See table in comments</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td></td><td>FW version
Format: X.YZ
Addr 0x73: X
Addr 0x74: .
Addr 0x75: Y
Addr 0x76: Z

Example (FW= 1.34):
<table><tr><td>Addr</td><td>0x73</td><td>0x74</td><td>0x75</td><td>0x76</td></tr><tr><td>Hex</td><td>0x31</td><td>0x2E</td><td>0x33</td><td>0x34</td></tr><tr><td>ASCII</td><td>1</td><td>.</td><td>3</td><td>4</td></tr></table>
<table><tr><td>Module series</td><td>Start</td><td>Stop</td></tr><tr><td>RC12xx</td><td>0x48</td><td>0x4B</td></tr><tr><td>RC11xx/RC11xxHP</td><td>0x4E / 0x50</td><td>0x51 / 0x53</td></tr><tr><td>RC2500/RC2500HP</td><td>0x4E / 0x50</td><td>0x51 / 0x53</td></tr><tr><td>RC17xx/RC17xxHP</td><td>0x9B / 0x9D</td><td>0x9E / 0xA0</td></tr></table></td><td></td></tr>
<tr><td>SERIAL_NUMBER</td><td>0xA9 – 0xB0</td><td>✗</td><td>✗</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></td></tr>
</table>

Appendix C Configuration Memory Defaults

RC12xx, HW 2.60, FW 2.34								
0x00	36	05	03	00	02	01	64	00
0x08	00	00	08	D3	91	DA	80	80
0x10	7C	00	02	01	02	02	00	00
0x18	00	01	01	0A	0A	0A	0A	01
0x20	01	01	01	00	00	00	00	04
0x28	FF	08	00	00	00	00	00	00
0x30	06	08	00	01	05	00	00	01
0x38	2B	00	01	00	52	43	31	32
0x40	(XX)	(XX)	2C	32	2E	36	30	2C
0x48	32	2E	33	34	41	20	20	20
0x50	00	FF	FF	FF	FF	FF	FF	FF
0x58	FF	FF	FF	FF	FF	FF	FF	FF
0x60	FF	FF	FF	FF	FF	FF	FF	FF
0x68	FF	FF	FF	FF	FF	FF	FF	FF
0x70	FF	FF	FF	FF	FF	FF	FF	FF
0x78	FF	FF	FF	FF	FF	FF	FF	FF
0x80	7D	8F	8B	8E	83	8D	49	87

PART_NUMBER
HW_REV_NO
FW_REV_NO

		RC1210: 0x0A RC1230: 0x11 RC1240: 0x36 RC1244: 0x54 RC1250: 0x31 RC1280: 0x29 RC1290: 0x1A
		RC1290: 0x04
		RC1230: 0x02 RC1250: 0x02
		RC1244: 0x03 0x08 0x0E 0x80 0xF0

RC11xx-RC232, HW 2.00, FW 1.34								
0x00	04	05	03	00	00	00	64	00
0x08	05	3C	00	D3	91	DA	00	80
0x10	7C	00	02	01	02	02	00	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	08	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	01
0x38	2B	00	00	00	52	43	31	31
0x40	(XX)	(XX)	2D	52	43	32	33	32
0x48	2C	32	2E	30	30	2C	31	2E
0x50	33	34	20	20	00	FF	FF	FF
0x58	FF	FF	FF	FF	FF	FF	FF	FF
0x60	FF	FF	FF	FF	FF	FF	FF	FF
0x68	FF	FF	FF	FF	FF	FF	FF	FF
0x70	FF	FF	FF	FF	FF	FF	FF	FF
0x78	FF	FF	FF	FF	FF	FF	FF	FF

PART_NUMBER
HW_REV_NO
FW_REV_NO

		RC1190: 0x01
		RC1190: 0x03

RC11xxHP-RC232, HW, 2.00, FW 1.34								
0x00	0D	05	03	01	00	00	64	00
0x08	05	3C	00	D3	91	DA	00	80
0x10	7C	00	02	01	02	02	00	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	08	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	01
0x38	2B	00	00	00	52	43	31	31
0x40	(XX)	(XX)	48	50	2D	52	43	32
0x48	33	32	2C	32	2E	30	30	2C
0x50	31	2E	33	34	00	FF	FF	FF
0x58	FF	FF	FF	FF	FF	FF	FF	FF
0x60	FF	FF	FF	FF	FF	FF	FF	FF
0x68	FF	FF	FF	FF	FF	FF	FF	FF
0x70	FF	FF	FF	FF	FF	FF	FF	FF
0x78	FF	FF	FF	FF	FF	FF	FF	FF

PART_NUMBER
HW_REV_NO
FW_REV_NO

		RC1170HP: 0x04 RC1190HP: 0x1A

RC2500-RC232, HW 1.00, FW 1.34								
0x00	28	05	03	00	00	00	64	00
0x08	05	3C	00	D3	91	DA	00	80
0x10	7C	00	02	01	02	02	00	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	08	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	01
0x38	2B	00	00	00	52	43	32	35
0x40	30	30	2D	52	43	32	33	32
0x48	2C	31	2E	30	30	2C	31	2E
0x50	33	34	20	20	00	FF	FF	FF
0x58	FF	FF	FF	FF	FF	FF	FF	FF
0x60	FF	FF	FF	FF	FF	FF	FF	FF
0x68	FF	FF	FF	FF	FF	FF	FF	FF
0x70	FF	FF	FF	FF	FF	FF	FF	FF
0x78	FF	FF	FF	FF	FF	FF	FF	FF

PART_NUMBER
HW_REV_NO
FW_REV_NO

RC2500HP-RC232, HW, 1.00, FW 1.34								
0x00	28	05	03	01	00	00	64	00
0x08	05	3C	00	D3	91	DA	00	80
0x10	7C	00	02	01	02	02	00	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	08	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	01
0x38	2B	00	00	00	52	43	32	35
0x40	30	30	48	50	2D	52	43	32
0x48	33	32	2C	31	2E	30	30	2C
0x50	31	2E	33	34	00	FF	FF	FF
0x58	FF	FF	FF	FF	FF	FF	FF	FF
0x60	FF	FF	FF	FF	FF	FF	FF	FF
0x68	FF	FF	FF	FF	FF	FF	FF	FF
0x70	FF	FF	FF	FF	FF	FF	FF	FF
0x78	FF	FF	FF	FF	FF	FF	FF	FF

PART_NUMBER
HW_REV_NO
FW_REV_NO

RC1701HP-RC232, FW 1.07								
0x00	01	05	04	11	00	00	00	00
0x08	05	3C	00	D3	91	DA	80	80
0x10	7C	00	01	01	02	02	17	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	00	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	00
0x38	00	00	00	06	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	06	07	10	10	10	10
0x70	0E	0E	0E	0E	10	10	10	10
0x78	01	00	00	00	09	9A	8C	9A
0x80	3A	B0	A0	B0	30	2F	2E	2C
0x88	00	52	43	31	37	(XX)	(XX)	48
0x90	50	2D	52	43	32	33	32	2C
0x98	31	2E	31	30	2C	31	2E	30
0xA0	37	20	20	20	20	20	20	20
0xA8	20	00	00	00	00	00	00	00
0xB0	00	FF	FF	FF	FF	FF	FF	FF
0xB8	FF	FF	FF	FF	FF	FF	FF	FF
0xC0	FF	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

PART_NUMBER
HW_REV_NO
FW_REV_NO
SERIAL_NUMBER

RC17x0HP-RC232, FW 1.07								
0x00	01	05	04	11	00	00	00	00
0x08	05	3C	00	D3	91	DA	80	80
0x10	7C	00	01	01	02	02	17	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	00	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	00
0x38	00	00	00	06	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	06	07	10	10	10	10
0x70	0E	0E	0E	0E	10	10	10	10
0x78	01	00	00	00	09	9A	8C	9A
0x80	3A	B0	A0	B0	30	2F	2E	2C
0x88	00	52	43	31	37	(XX)	30	48
0x90	50	2D	52	43	32	33	32	2C
0x98	32	2E	30	30	2C	31	2E	30
0xA0	37	20	20	20	20	20	20	20
0xA8	20	00	00	00	00	00	00	00
0xB0	00	FF	FF	FF	FF	FF	FF	FF
0xB8	FF	FF	FF	FF	FF	FF	FF	FF
0xC0	FF	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

PART_NUMBER

HW_REV_NO

FW_REV_NO

SERIAL_NUMBER

RC1780HP: 0x3D

RC1740-RC232, FW 1.07								
0x00	01	05	04	11	00	00	00	00
0x08	05	3C	00	D3	91	DA	80	80
0x10	7C	00	01	01	02	02	17	00
0x18	00	01	01	01	01	01	01	01
0x20	01	01	01	01	01	00	00	04
0x28	FF	00	00	00	00	00	00	00
0x30	05	08	00	01	05	00	00	00
0x38	00	00	00	06	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	06	07	10	10	10	10
0x70	0E	0E	0E	0E	10	10	10	10
0x78	01	00	00	00	09	9A	8C	9A
0x80	3A	B0	A0	B0	30	2F	2E	2C
0x88	00	52	43	31	37	34	30	2D
0x90	52	43	32	33	32	2C	31	2E
0x98	30	30	2C	31	2E	30	37	20
0xA0	20	20	20	20	20	20	20	20
0xA8	20	00	00	00	00	00	00	00
0xB0	00	FF	FF	FF	FF	FF	FF	FF
0xB8	FF	FF	FF	FF	FF	FF	FF	FF
0xC0	FF	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

PART_NUMBER
HW_REV_NO
FW_REV_NO
SERIAL_NUMBER

Document Revision History

Document Revision	Changes
1.0	First release
2.0	Rewritten to new company standard.
2.01	Removed timing information to avoid redundancy, as this information is present in datasheets

Disclaimer

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

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

Trademarks

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

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

Life Support Policy

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

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/