

RAMP Wireless Module

RM024

Version 3.5

REVISION HISTORY

| Version | Date | Notes | Approver |
|---------|--------------|--|----------------|
| 1.0 | | Initial Version | Chris Downey |
| 2.1 | | Added firmware changes, updated the name of the removed old references to LT2510 part numbers, added new information on cyclic sleep and Antenna Switch Override. Added a table under Max Power and a table for the Set Max Power command. | Chris Downey |
| 2.2 | | Minor grammatical fixes. | Chris Downey |
| 2.3 | 27 June 2013 | PWM output data was corrected to a 39.3846 μ S period vs. 315.077, as was stated previously. | Chris Downey |
| 2.4 | 12 July 2013 | Minor edits, removed Firmware History and references irrelevant to RM024. | Chris Downey |
| 2.5 | 10 Oct 2013 | Corrected Antenna Select Override information error. Changed 0x59 to 0x5B | Chris Downey |
| 3.0 | 10 Dec 2013 | Separated Hardware Integration Guide (HIG) from User Guide information (created two separate documents). Add Related Documents section. | Sue White |
| 3.1 | 13 Jan 2014 | Added information on FW v2.0, deprecated 50mW radio versions and added 10mW radio versions | Chris Downey |
| 3.2 | 14 Sept 2015 | Removed deprecated links in Related Documents and Files | Jonathan Kaye |
| 3.3 | 16 Mar 2016 | Updated to the latest template | Sue White |
| 3.4 | 31 Jan 2017 | Added Status Request command | Jennifer Gibbs |
| 3.5 | 18 July 2017 | Fixed header of RSSI Control (0x68) settings table and RSSI control table. | Jennifer Gibbs |

CONTENTS

| | |
|---|----|
| RAMP Modules | 4 |
| Overview | 4 |
| Key Features | 4 |
| Theory of Operation | 5 |
| Server/Client Architecture | 5 |
| Adjustable RF Data Rate | 5 |
| Modes of Operation | 6 |
| Serial Interface Baud Rate | 7 |
| Interface Timeout/RF Packet Size | 8 |
| Flow Control | 9 |
| Force 9600 | 10 |
| Radio Configurations | 11 |
| EEPROM Parameters | 19 |
| Configuring the RM024 | 29 |
| AT Commands | 29 |
| Utility Commands | 31 |
| Status Commands | 33 |
| Command Descriptions | 35 |
| Serial Firmware Upgrades | 42 |
| Overview | 42 |
| Upgrading Via Windows OEM Configuration Utility | 42 |
| Upgrading FW Commands | 43 |
| Command Descriptions | 43 |
| Process to Manually Upgrade RM024 | 45 |
| API Operation | 46 |
| API Send Data Complete | 46 |
| API Receive Packet | 46 |
| API Transmit Packet | 47 |
| Related Documents and Files | 48 |

RAMP MODULES

Laird RAMP (Range Amplified MultiPoint) modules are designed to provide robust wireless communications for any number of applications requiring a wireless transport for serial data. RAMP modules feature a Frequency Hopping Spread Spectrum (FHSS) protocol for excellent interference and multipath immunity. RAMP modules server/client architecture allows for more than 16 million clients to be addressed and communicating within the network.

Overview

The RM024 RAMP module is based on Laird LT2510 core technology, enhanced with a new RF front end for improved sleep, improved link budget, and a switchable antenna output. The RM024 is available in two versions, one with 125 mW maximum conducted output power which is approved for North American and similar markets, and one with 10 mW maximum conducted output power which is approved for European and similar markets. These modules are identical except for output power, transmit power consumption, and the number of RF channels available. Differences between the two versions, where applicable, are denoted based on part number.

This document contains information about the hardware and software interface between a Laird RM024 transceiver and an OEM host. Information includes the theory of operation, specifications, interface definitions, and mechanical drawings.

Note: Unless mentioned specifically by name, the RM024 modules are referred to as “radio” or “transceiver”. Individual naming is used to differentiate product specific features. The host (PC/microcontroller/any device to which the RM024 module is connected) are referred to as “OEM host” or “host.”

Key Features

- Forward Error Correction option for longer range
- Retries and acknowledgements
- Configurable network parameters
- Multiple generic I/O
- 280 kbps or 500 kbps RF data stream
- Idle current draw of 9.5 mA, sleep current of 38 uA
- Software selectable interface baud rates from 1200 bps to 460.8 kbps
- Upgradable FW through serial port
- Low cost, low power, and small size ideal for high volume, portable, and battery powered applications
- All modules are qualified for Industrial temperatures (-40°C to 85°C)
- Advanced configuration available using AT commands
- Easy to use Configuration and Test Utility software
- Switchable antenna output, either integrated antenna or external antenna through U.FL

THEORY OF OPERATION

Server/Client Architecture

The RM024 utilizes server-client network architecture to synchronize the frequency hopping. Each network must have one radio configured as a server and all other radios configured as clients. When a radio is configured as a server, it transmits a beacon containing timing and identification information at the beginning of each hop. The beacon is never visible to the OEM host. Upon boot, radios configured as clients enter receive mode where they are scanning the available frequencies listening for a beacon from a server in their network. When a client detects the server’s beacon, the client synchronizes its frequency hopping to that of the server and transition the In Range pin Low. When the server and the client are synchronized they can begin transferring data.

Each network consists of only one server. Multiple networks can exist in the same area, provided the networks are configured on different channels. The RM024 utilizes an intelligent Frequency Hopping algorithm which ensures minimal interference between networks. The possible interference between collocated networks is given by the following equation:

$$\text{Maximum number of interfering bins} = \# \text{ of collocated servers} - 1$$

For example, with ten collocated networks, there is up to nine bins every hop cycle that are occupied by more than one network at the same time. Although two or more networks might occupy the same hop bin at the same time, there is truly only interference if two or more radios from alternate networks are trying to transmit on the same bin at the same time in the same coverage area.

Adjustable RF Data Rate

The RM024’s RF data rate can be adjusted to provide a trade-off between throughput and range.

Table 1: RM024 RF Data Rate

| Product Model | RF Profile | RF Data Rate | Number of Hops | Receiver Sensitivity | Throughput ¹ |
|-----------------------------------|------------|--------------|----------------|----------------------|-------------------------|
| All RM024 | 0x00 | 500 kbps | 43 | -94 dBm | 250 kbps |
| 125 mW versions (RM024-X125) only | 0x01 | 280 kbps | 79 | -95 dBm | 120 kbps |
| All RM024 | 0x03 | 280 kbps | 43 | -95 dBm | 120 kbps |

1. Throughput is ideal, one direction, with no retransmissions. All practical RF applications should allow for retransmission of data due to interference or less than ideal RF conditions.

Deciding which RF data rate to choose depends on the individual application. The fast RF data rate delivers much faster throughput, but has much less range. In addition, because the lower data rate solution uses more hops, it is better situated for collocated networks.

A rule of thumb for RF systems is every six dB of gain doubles the effective distance. The four dB increase of receive sensitivity for the lower data rate solution means it is able to transmit almost 60% further than the higher data rate solution.

Modes of Operation

The RM024 has three different types of interface modes:

- [Transparent Mode](#)
- [API Mode](#)
- [Command Mode](#)

The first two modes are used to transmit data across the RF and the third mode is used to configure the radio.

Transparent Mode

When operating in transparent mode, the RM024 can act as a direct serial cable replacement in which RF data is forwarded over the serial interface and vice versa. In transparent mode, the radio needs to be programmed with the MAC address of the desired recipient. The destination address can be programmed permanently or on-the-fly.

When transparent mode is used, data is stored in the RX buffer until one of the following occurs:

- The RF packet size is reached (EEPROM address 0x5A)
- An Interface timeout occurs (EEPROM address 0x58)

All parameters can be configured by entering Command Mode using either AT commands or by toggling the Command/Data pin low on the transceiver.

Transparent mode is the default radio operation mode.

API Mode

API mode is an alternative to the default transparent operation of the RM024 and provides dynamic packet routing and packet accounting abilities to the OEM host without requiring extensive programming by the OEM host. API mode utilizes specific frame-based packet formats, specifying various vital parameters used to control radio settings and packet routing on a packet-by-packet basis. The API features can be used in any combination that suits the OEM's application specific needs.

The RM024 has three API functions:

- [Send Data Complete](#)
- [Receive API](#)
- [Transmit API](#)

For additional details and examples, please refer to the [API Operation](#) section of this guide.

Command Mode

Command mode is used to configure and poll for status of the transceiver. Command mode can be entered by issuing the Enter AT Command string or by setting the CMD/Data pin low. Details of using command mode to configure the RM024 are detailed in [Configuring the RM024](#).

Serial Interface Baud Rate

In order for the OEM host and a transceiver to communicate over the serial interface, they must have the same serial data rate. This value determines the baud rate used for communicating over the serial interface to a transceiver. For a baud rate to be valid, the calculated baud rate must be within $\pm 3\%$ of the OEM host baud rate.

Table 2: Baud Rate/Interference Rate

| Desired Baud Rate | Baud (0X42) | Minimum Interface Timeout ¹ (0X58) |
|---------------------|-------------|---|
| 230400 | 0x0A | 0x02 |
| 115200 ² | 0x09 | 0x02 |
| 57600 | 0x08 | 0x02 |
| 38400 | 0x07 | 0x02 |
| 28000 | 0x06 | 0x03 |
| 19200 | 0x05 | 0x05 |
| 14400 | 0x04 | 0x07 |
| 9600 | 0x03 | 0x10 |
| 4800 | 0x02 | 0x15 |
| 2400 | 0x01 | 0x2A |
| 1200 | 0x00 | 0x53 |
| Non-standard | 0xE3 | Use equation below |

- Interface timeout = 200 μ s per increment, the EEPROM address 0x58 is ignored if Auto Config is enabled. To use a non-standard Interface timeout, disable Auto Config.
- Default baud rate.

For baud rates other than those shown in [Table 2](#), the following equations can be used:

$$\text{Baud Rate} = \frac{(256 + \text{BAUD_M} * (2^{\text{BAUD_E}}) * \text{FREQUENCY}}{2^{28}}$$

Where:

FREQUENCY = 26 MHz

BAUD_M = EEPROM Address 0x43

BAUD_E = EEPROM Address 0x44

$$\text{Minimum Interface Timeout} = \frac{100000}{\text{Baud Rate}}$$

Tips:

- The RM024 supports a majority of standard as well as non-standard baud rates. To select a standard baud rate, use the value shown for EEPROM address 0x42 in [Table 2](#). To enable a non-standard baud rate, program EEPROM address 0x42 (Custom Baud Enable) to 0xE3 and then use the equation above to solve for BAUD_M and BAUD_E.
- Adjusting the serial interface baud rate does not affect the RF data rate.
- Radio can accept serial combinations (number of bits, parity, number of stop bits) of 8-N-1, 7-N-2, 7-1-1, by default. Modes of 8-1-1, 8-N-2, 7-1-2 are acceptable with 9-bit mode enabled.

Interface Timeout/RF Packet Size

Interface Timeout

Interface timeout specifies a maximum byte gap between consecutive bytes. When that byte gap is exceeded, the bytes in the transmit buffer are processed as a complete packet. Interface timeout (EEPROM address 0x58), in conjunction with the RF packet size, determines when a buffer of data is sent out over the RF as a complete RF packet, based on whichever condition occurs first. Interface timeout is adjustable in 200 us increments and should be equal to or greater than two full bytes times. The minimum interface timeout is 0x02.

The radio uses the default interface timeout for a given baud rate if Auto Config is enabled, despite what is written in the interface timeout address. To use a non-standard interface timeout, the OEM needs to disable Auto Config.

RF Packet Size

RF packet size is used in conjunction with interface timeout to determine when to delineate incoming data as an entire packet based on whichever condition is met first. When the transceiver receives the number of bytes specified by RF packet size (EEPROM address 0x5A) without experiencing a byte gap equal to interface timeout, that block of data is processed as a complete packet. Every packet the transceiver sends over the RF contains extra header bytes not counted in the RF packet size. Therefore, it is much more efficient to send a few large packets than to send many short packets. The maximum RF packet size is 239 bytes, or 0xEF, at 500 kbps RF data rate and 96 bytes, or 0x60, at 280 kbps RF data rate.

The RF packet size in address 0x5A will not be used if Auto Config (Address 0x56, bit 0) is enabled. The default for the RF data rate will be used instead. The RF packet size should not be set to less than 0x07, to ensure AT commands can still be issued.

RF packet size is also used by the radio to determine the number of data slots per hop. In order to efficiently transmit data across the RF the radio automatically adds more data slots to the hop to correspond to a smaller RF packet size. The number of slots available is reduced when FEC mode is enabled. The number of slots per hop is given in the table below.

Table 3: Number of Slots per Hop

| RF Data Rate | FEC Mode | RF Packet Size | Number of Data Slots |
|--------------|----------|----------------|----------------------|
| 280 kbps | Disabled | 0x01 – 0x09 | 4 slots |
| 280 kbps | Disabled | 0x0A – 0x25 | 3 slots |
| 280 kbps | Disabled | 0x26 – 0x60 | 2 slots |
| 500 kbps | Disabled | 0x01 – 0x0C | 6 slots |
| 500 kbps | Disabled | 0x0D – 0x25 | 5 slots |
| 500 kbps | Disabled | 0x26 – 0x47 | 4 slots |
| 500 kbps | Disabled | 0x48 – 0x7D | 3 slots |
| 500 kbps | Disabled | 0x7E – 0xEF | 2 slots |
| 280 kbps | Enabled | 0x01 – 0x06 | 2 slots |
| 280 kbps | Enabled | 0x07 – 0x4A | 1 slot |
| 500 kbps | Enabled | 0x01 – 0x02 | 4 slots |
| 500 kbps | Enabled | 0x03 – 0x1A | 3 slots |
| 500 kbps | Enabled | 0x1B – 0x4B | 2 slots |
| 500 kbps | Enabled | 0x4C – 0xE2 | 1 slot |

Tips:

- The more slots per hop, the less likely that retries will occur on a new frequency which may reduce the effectiveness of the module as a Frequency Hopping radio.
- Idle current consumption increases as more slots are added.
- You must use the same number of slots for every radio on the network.
- Full duplex only reserves the first slot for the server. If there are six slots, the first slot is reserved for the server to transmit and the remainder is shared by the clients. If there is only one slot per hop available, the Server and Client will alternate transmissions with the Server transmitting in one slot/hop and the client transmitting on the next hop.
- RF Packet Size should not be set to less than 0x07 or the Enter AT Command string will not be accepted. If the RF Packet Size is less than 0x07, the Enter AT Command string can still be issued in Force 9600 mode.

RS-485 Data Enable

The timing of the DE-RE pin varies depending on the selected interface baud rate. The values to set are:

- 485_Delay_H: Address 0x49
- 485_Delay_M: Address 0x4A
- 485_Delay_L: Address 0x4B

To set them, use the following equation (round the result up):

- Address 0x49 and 0x4A: $485H/M = 8.125 \text{ MHz} / (81 * \text{Baud_Rate})$, quotient only
- Address 0x4B: $485L = (8.125\text{MHz} / \text{Baud_Rate}) \bmod 81$

So for 19,200 you should calculate 00 05 12.

Flow Control

Although flow control is not required for transceiver operation, it is recommended to achieve optimum system performance and to avoid overrunning the RM024's serial buffers. The RM024 uses separate buffers for incoming and outgoing data.

RXD Data Buffer and CTS

As data is sent from the OEM host to the radio over the serial interface, it is stored in the RM024's buffer until the radio is ready to transmit the data packet. The radio waits to transmit the data until one of the following conditions occur (whichever occurs first):

- The RF packet size is reached (EEPROM address 0x5A)
- An interface timeout occurs (EEPROM address 0x58)

The data continues to be stored in the buffer until the radio receives an RF Acknowledgement (ACK) from the receiving radio (addressed mode), or all transmit retries/broadcast attempts are used. Once an ACK has been received or all retries/attempts are exhausted, the current data packet is removed from the buffer and the radio begins processing the next data packet in the buffer.

To prevent the radio's RXD buffer from being overrun, we strongly recommend that the OEM host monitors the radio's CTS output. When the number of bytes in the RXD buffer reaches the value specified by CTS_ON (EEPROM address 0x5C - 0x5D), the radio de-asserts (high) CTS to signal to the OEM host to stop sending data over the serial interface. CTS is re-asserted after the number of bytes in the RXD buffer is reduced to the value specified by CTS_OFF (EEPROM addresses 0x5E- 0x5F); signaling to the OEM host that it may resume sending data to the transceiver.

Note: We recommend that the OEM host stop all data transmission to the radio while CTS is de-asserted (high), otherwise potential data loss may occur.

TXD Data Buffer and RTS

As data to be forwarded to the OEM Host accumulates, it is stored in the RM024's outgoing buffer until the radio is ready to begin sending the data to the OEM Host. Once the data packet has been sent to the Host over the serial interface, it will be removed from the buffer and the radio will begin processing the next data packet in the buffer. With RTS Mode disabled, the transceiver will send any data to the OEM Host as soon as it has data to send. However, some OEM Hosts are not able to accept data from the transceiver all of the time. With RTS Mode Enabled, the OEM Host can prevent the transceiver from sending it data by de-asserting RTS (High), causing the transceiver to store the data in its buffer. Upon asserting RTS up to two additional bytes can be received over the serial interface before the flow is stopped. Once RTS is re-asserted (Low), the transceiver will continue sending data to the OEM Host, beginning with any data stored in its buffer.

Note: Leaving RTS de-asserted for too long can cause data loss once the radio's TXD buffer reaches capacity.

Note: RTS is disabled in Force 9600 mode in Firmware 2.0 and above.

Tip:

Can I implement a design using just TXD, RXD, and GND (three-wire interface)?

Yes. However, it is strongly recommended that your hardware monitor the CTS pin of the radio. CTS is taken High by the radio when its interface buffer is getting full. Your hardware should stop sending at this point to avoid a buffer overrun (and subsequent loss of data). You can perform a successful design without monitoring CTS. However, you need to take into account the amount of latency the radio adds to the system, any additional latency caused by retries, how often you send data, non-delivery network timeouts, and interface data rate. Laird Technologies can assist in determining whether CTS is required for your application.

Force 9600

Force 9600 mode is a recovery mode in which the radio ignores specific EEPROM configurations. Force 9600 mode is primarily used to recover a radio when the configuration is not known.

Enabling Force 9600

Force 9600 is triggered by the Force 9600 pin (pin 12 for SMT modules and pin 9 for pluggable modules). When you pull the Force 9600 pin logic low and apply power or reset the module, the transceiver's serial interface is forced to a 9600, 8-N-1 rate. Additional RTS Enable is disabled and the RF Packet Size in EEPROM is ignored. The Radio's receiver is disabled in Force 9600 mode, so it is unable to receive beacons or packets.

Note: Because this mode disables some modes of operation, it should not be permanently pulled Low during normal operation.

Radio Configurations

Antenna Switch (EEPROM 0xC1, bit 5)

Selects either integrated chip antenna or U.FL connector for external antenna.

Note: On RM024 –C units with no integrated antenna, the RF switch is still active and it is possible, though not advised to switch to the integrated antenna option, even though there is no antenna connected. RF performance in this configuration would be degraded. See [Antenna Select Override](#) for additional options.

Antenna Select Override (EEPROM 0x5B)

Disables the antenna switch on –C products causing the firmware to ignore the setting in Antenna Switch and use the U.FL port automatically.

Note: Product ID's containing an "M" (RM024-S125-M-01, RM024-P125-M-01, RM024-S50-M-01 and RM024-P50-M-01) have both antennas installed (chip antenna and u.FL). However, products containing a "C" (RM024-S125-C-01, RM024-P125-C-01, RM024-S50-C-01 and RM024-P50-C-01) only have the u.FL installed. Therefore, selecting chip antenna on a "C" product results in no RF link. This feature does not work in FW v1.3-0 on 50 mW radios (RM024-x50-C-01).

Auto Channel (EEPROM 0x56, bit 3)

To allow for more flexible network configurations, Auto Channel can be enabled in clients to allow them to automatically synchronize with the first server they detect, regardless of channel number.

Note: A client with Auto Channel only synchronizes with a server that has a matching System ID.

Auto Config (EEPROM 0x56 bit 0)

The optimal settings for interface timeout and RF packet size vary according to the selected RF profile and interface baud rate. Enabling Auto Config bypasses the value for these variables stored in EEPROM and uses predetermined values that have been optimized for the given mode. When Auto Config is disabled, these values must be programmed in the transceiver EEPROM.

Auto Destination (EEPROM 0x56, bit 4)

To simplify EEPROM programming, Auto Destination can be enabled in the radio which allows the radio to automatically set its destination to the address of the radio from which it last received a successful transmission from (beacon or data packet).

Auto Destination on Beacons Only (Address 0x56, bit 7)

When Auto Destination is enabled, the client radio addresses itself to the source of any received packet, including beacons from the server and any addressed or broadcast packets it receives. For point to multipoint networks where the client is intended to only communicate back to the server, this could cause the client to inadvertently become addressed to another client. By enabling Auto Destination on Beacons Only, the client only addresses itself upon reception of beacons, therefore it only addresses itself to the server. Auto Destination on Beacons Only is only functional when Auto Destination is also enabled.

Auto System ID (EEPROM 45, bit 4)

When enabled, Auto System ID allows a client to attach to any server on the same RF channel, regardless of the System ID on the server or the client.

Beacon Skip (EEPROM 0x6F)

When set, the transceiver will send (server) or listen (client) for a beacon on hops spaced by the Beacon Skip number. On a client, once the Beacon Skip count is reached the client will listen every hop until it successfully hears a beacon. It will then wait a number of hops specified by the Beacon Refresh before listening again.

Enabling this will allow the transceiver to conserve power by disabling its RF circuitry during the beacon time. Enabling this on the server causes substantially longer sync times on the clients.

Broadcast (EEPROM 0xC1, bit 7)

In Broadcast mode, the transceiver transmits the packet to all transceivers with the same Channel Number and System ID settings. There is no RF acknowledgement sent from the recipient(s) back to the transmitter, therefore the packet is sent out the number of times specified by Broadcast Attempts.

Broadcast Attempts (EEPROM 0x4D)

When transmitting broadcast packets, the RF packet is broadcast out to all eligible receivers on the network. Broadcast Attempts is used to increase the odds of successful delivery to the intended receivers. Transparent to the OEM host, the transmitter sends the RF packet to the receivers. If a receiver detects a packet error, it throws out the packet. This continues until the transmitter exhausts all of its attempts. Once the receiver successfully receives the packet, it sends the packet to the OEM host. It throws out any duplicates caused by further broadcast attempts. The received packet is only sent to the OEM host if it is received free of errors. Because broadcast packets have no RF acknowledgement, each packet is transmitted the number of times specified by Broadcast Attempts. This makes for inefficient use of the available bandwidth; therefore, it is recommended that Broadcast Attempts be set as low as possible and that broadcast packets be limited in use.

Note: Setting to 0 is equal to 256.

Cyclic Sleep (EEPROM 0x61, bit 0)

Causes the radio to sleep for a programmable period of time and wake for a programmable period of time. The radio can be awakened from sleep before its sleep cycle completes using the Force 9600 pin. Additionally, the wake time is an inactivity counter. Therefore, the device stays awake indefinitely as long as the device continues sending packets over the RF interface.

Destination Address (EEPROM 0x79-0x75)

The Destination Address is simply the MAC (IEEE) address of the intended receiver on the network. In Addressed mode, the RF packet is sent out to the intended receiver designated by the destination address. Only the four LSBs (Least Significant Bytes) of the destination address are actually used for packet delivery. This field is ignored if Broadcast mode, Auto Destination, or Transmit API is enabled.

Disable Status Bin (EEPROM 0xC1, bit 4)

When set, disables the reception on the status slot of the bin. The result is that the bin analyzer and remote I/O functionality is disabled on the radio with the benefit of saving approximately 1 mA average current consumption.

Discard Framing Error Packets (EEPROM 0x57, bit 7)

When set, the radio checks for a framing error in the UART buffer before processing incoming data. If an error is detected on any of the bytes in the buffer, the entire buffer is discarded.

Forward Error Correct (See RF Profile for configuration information)

Specific RF Profiles are reserved to enable Forward Error Correction (FEC). Forward Error Correction can be used to decrease the packet error rate in the presence of bursty errors over the air. The RM024 uses convolutional coding and interleaving to allow the receiver to recover from small bit errors. When enabled, FEC will cause the radio to transmit additional bits of data over the air to allow for error recovery. When FEC is enabled the maximum RF Packet size is decreased, the number of data slots may be reduced and the throughput of the radio can be reduced as much as half of non-FEC mode. FEC is most useful when near the receiver sensitivity limit, but due to the nature of the shared nature of the 2.4 GHz ISM bands, can provide improvements in packet error rate even when the signal strength is strong. Though FEC does not increase the receiver sensitivity or affect the link budget, in real-world range tests, FEC enabled had as much as a 3dBm equivalent improvement.

FEC is enabled by selecting one of the following RF Profiles.

- ODH: 500kbps (1.5MHz steps) + FEC, 43 hops, 2.4GHz 500kbps RF
- 0EH: 280kbps (900kHz steps) + FEC, 79 hops, 2.4GHz (FCC Only)
- 0FH: 280kbps (900kHz steps) + FEC, 43 hops, 2.4GHz (FCC Only)
- 10H: 280kbps (1.5MHz steps) + FEC, 43 hops, 2.4GHz 280kbps RF

Full Duplex (EEPROM 0x56, bit 1)

In Half Duplex mode, the transceiver sends a packet out over the RF immediately. This can cause packets sent at the same time by a server and a client to collide with each other over the RF. To prevent this, Full Duplex mode can be enabled. This mode reserves a transmit “slot” for the server. If the server does not have any data to transmit, clients are permitted to transmit during that time. If the server does have data to send, clients will not be permitted to transmit during that slot. Likewise, the server will not be able to transmit during a client slot. Though the RF hardware is still technically half duplex, it makes the transceiver seem full duplex. This can cause overall throughputs to be cut in half.

Note: All transceivers on the same network must have the same setting for Full Duplex.

Hop Packet Delineation (EEPROM 0x57, bit 6)

When enabled, in addition to using RF packet size and interface timeout as criteria for processing incoming data, the radio also delineates packets up to once per hop once a minimum of six characters has been received over the serial port.

Legacy RSSI (EEPROM 0x45, bit 2)

RSSI (Received Signal Strength Indicator) is a measure of how well the receiving radio is able to hear the transmitting radio. By default, RSSI is reported in 8-bit 2's complement hexadecimal format; therefore, values range from 0x80 - 0x7F. Many preceding products have, instead, reported RSSI in the range of 0x00 - 0xFF. Legacy RSSI causes 0x80 to be added to the RSSI result prior to reporting it to the host.

Max Power (EEPROM 0x63)

On 50mW and 125mW RM024 radios, the transceiver has an adjustable RF output power. Power can be adjusted dynamically to optimize communications reliability and conserve power. Each increment represents a 3 dBm 50% decrease in power. The radios have a maximum input RF level of 0 dBm. When operated very close together at full power, the radio's receiver can saturate and no transmissions are possible. If the distance between the transmitter and receiver is very short (generally less than 2 ft or 0.6 m with 2.5 dBi antennas), the maximum power should be reduced.

On 10mW RM024 radios, the Max Power setting will have no effect on the output power, all four power settings are fixed at 10mW.

Mode (Server/Client) (EEPROM 0x41)

The server controls the frequency hop timing by sending out regular beacons (transparent to the transceiver host) which contain system timing information. This timing information synchronizes the client radio frequency hopping to the server. Each network should consist of only one server.

Nine Bit Mode (EEPROM 0x57, bit 1)

With Nine Bit mode disabled, the transceiver communicates over the asynchronous serial interface in 8-N-1 format (8 data bits, No parity, 1 stop bit). Some systems require a parity or 9th data bit. Enabling Nine Bit mode causes the transceiver to communicate using 8-1-1 format (8 data bits, 1 parity bit, 1 stop bit). In this mode, the transceiver does not validate the parity bit but simply transmits it over the RF. This is useful as some systems use the ninth bit as an extra data bit and not just a parity bit. However, because the ninth bit is transmitted over the RF, enabling Nine Bit mode cuts the transceiver interface buffer size by 1/9 and reduces the RF bandwidth by the same ratio.

Random Backoff (EEPROM 0xC3)

The transceivers utilize a retry protocol with Random Backoff and a programmable back-off seed. Therefore, in the event an acknowledgement is not received, the transceiver backs off and retries the packet. For example, when two transceivers collide with each other (transmitting packets at the same time), each transceiver chooses a random number of packet times that it will wait before retrying the packet. Ideally, they each choose a different number and are successful in the next transmission. A good rule of thumb is to set Random Backoff to a number slightly larger than the maximum number of transceivers that would be expected to be transmitting at the same time. When set to transmit broadcast packets, where there is no acknowledgment available, the Random Backoff value is used for all subsequent attempts.

Range Refresh (EEPROM 0x3D)

Range refresh specifies the maximum amount of time a transceiver reports In Range without having heard a server's beacon. It is adjustable in hop periods. Do not set to 0.

Remote I/O Mode (Address 0x57, bit 3)

Remote I/O mode allows GPIOs on two radios to be joined together so their states are reflected on the other radio. Enabling Remote I/O mode allows the local radio to transmit its GPIO states whenever there is a change. The states are transmitted to the radio specified by the Destination Address (or as a broadcast if Broadcast mode is enabled). State information is only transmitted when there is a change on one of the enabled Digital Inputs. The states are retransmitted up to the number of specified Utility Retries (Address 0x4E). Any changes to the Digital Inputs that occur while a utility retransmission is occurring are not transmitted unless the change persists until all utility retries have been sent or an acknowledge was received. Therefore, this feature should

only be used for slow-moving changes that occur less than the time it takes to expend all retries. Remote I/O is disabled when the Force 9600 pin is set at boot.

Remote I/O control lines occur in pairs, with the Digital Input on the local radio driving a Digital Output on the remote radio and vice-versa. This makes Remote I/O useful for both point-to-point and point-to-multipoint networks. Multipoint-to-point networks do not benefit from using a single pair of lines as the central point isn't able to tell where the line change was sourced. Multiple control lines are available though, so up to four pairs of lines can be used simultaneously. Likewise, analog inputs can be used (Address 0x57, bit 4) as the input (with a PWM output on the remote radio), though analog states are only transmitted when a utility packet is sent, which are only triggered by the change of a Digital Input. Threshold settings are not available on analog Inputs.

Output lines are initialized at boot according to Remote I/O Status (Address 0xC9-0xCA) for the digital lines and PWM_Init (Address 0xC8) for the PWM output.

Which control lines are used in Remote I/O is set by the Remote I/O Control bit field (Address 0x60). Note that TxD/RxD is one pair of Remote I/O lines available. If this pair is used, the module does not respond to commands and is not able to transmit or receive serial data. If this pair is enabled, Force 9600 must be low at boot to disable Remote I/O if serial communications are desired.

Table 4: Remote I/O Control bit fields (Address 0x60)

| Address 0x60, Bit | Input | Output |
|--------------------------|--|--------|
| Bit 0 set | GIO_4 | GIO_0 |
| Bit 1 set | GIO_8 ¹ | GIO_1 |
| Bit 2 set | GIO_7 | GIO_3 |
| Bit 3 set | CMD/Data | GIO_2 |
| Bit 4 set | RTS | CTS |
| Bit 5 set | RXD | TXD |
| Bit 6 clear, Bit 7 clear | All I/O are Outputs | |
| Bit 6 set, Bit 7 clear | All I/O are Inputs | |
| Bit 7 set | Inputs and outputs are as specified in table | |

1. GIO_8 (Pin 18) on board revisions 0050-00203 Rev 0 and 0050-00196 rev 2 (and below) is internally not connected. This pin is unavailable as a GPIO on these boards.

Tips:

- When using GIO_7/GIO_3 Pairs, the input/output will be digital unless Remote Analog Enable bit is set (Address 0x57, bit 4) in which case the input is Analog and the output is PWM.
- TXD and RXD are not available for UART serial data when used as in Remote I/O. Force 9600 must be Low on boot to disable Remote I/O Mode and issue commands.
- When not using pairs (bit 7 clear), one radio should have all I/O as inputs and the other radio or radios should have all I/O as output.
- Remote I/O Mode must be enabled on both the local and remote radio and the Remote I/O Control Bit must be set for the same pair on both radios.
- All I/O state information for all lines is transmitted when any update is triggered. Thus, on the receiving radio any enabled output pins will be updated, regardless of whether those pins were enabled on the transmitting radio.

RF Channel Number (EEPROM 0x40)

This product uses FHSS (Frequency Hopping Spread Spectrum) protocol in which the transceiver communicates using frequency “bins” spaced throughout the frequency band. Therefore, RF Channel Number specifies a unique pseudo-random hopping sequence.

RF Profile (EEPROM 0x54)

RF Profile can be adjusted to provide a trade-off between throughput and range. Deciding which RF profile to choose depends on the individual application. Selecting a higher RF baud rate provides increased RF bandwidth. However, selecting the lower RF baud rate provides significantly improved range. Selecting fewer hops provides a shorter sync time, whereas more hops provides better interference and collocated system immunity. Forward Error Correction (FEC) is also enabled by selecting an appropriate RF Profile. FEC will further increase the range of the radio and allow for less packet errors in the presence of bursty RF interference. FEC will affect the reduce maximum RF Packet Size available, reduce the number of data slots per hop and the reduce the overall throughput of the radio.

RSSI

Received Signal Strength Indicator (RSSI) is available to the OEM through a number of means. AT commands such as Get Last RSSI and Bin Analyzer report RSSI, API Packets for Receive API and Send Data Complete report RSSI, and one of three pins can be configured to provide a PWM output representing the RSSI. By default, AT commands and API packets represent RSSI in an 8-bit hexadecimal 2’s complement range. Legacy RSSI (detailed above) can be enabled to provide the RSSI in a non 2’s complement form from 0x00 (very weak signal) to 0xFF (very strong signal). The control commands for PWM output utilize a Legacy RSSI format from 0x00 to 0xFF.

The RSSI values reported can be converted to a decibel value with the following formulas:

- For Non-Legacy values where the RSSI is reported in 8-bit 2’s complement hexadecimal ranging from 0x80 to 0x7F, use the following to calculate the RSSI_dBm. For these calculations, convert the reported hexadecimal value directly to decimal notation, ignoring the 2’s complement conversion:
 - If this value is greater than or equal to 128, then:
 - $RSSI_dBm = (RSSI_Dec - 256)/2 - RSSI_Offset$
 - If this value is less than 128, then:
 - $RSSI_dBm = (RSSI_Dec)/2 - RSSI_Offset$
- For Legacy RSSI the equation is:
 - $RSSI_dBm = (RSSI_Dec - 128)/2 - RSSI_Offset$
 - RSSI_Dec is the reported value represented in Decimal notation
 - RSSI_Offset = 82

Reported RSSI values are meant as estimates and have an accuracy of +/- 2 dBm. The RSSI reported by various commands has an effective range of -25 dBm to -95 dBm. Outside of this range, the accuracy is not maintained.

RSSI_Control (EEPROM 0x68)

RSSI Control is a bit field used to control the output of the RSSI PWM output and what messages the radio reports on.

Note: If Disable Hop Frame is disabled (so as to report Hop Frame), it is output on GO_0 (pin 1 of SMT module); therefore, the PWM output should not be set to output to that pin or conflicting signals will be sent on that output pin.

Table 5: RSSI Control

| Address 0x68, Bit | Description |
|-------------------|-----------------------------------|
| bit 0 set | Update RSSI on Beacons |
| bit 1 set | Update RSSI on Addressed Packets |
| bit 2 set | Update RSSI on Broadcast Packets |
| bit 3 set | Update RSSI on Unintended Packets |
| bit 4 set | Invert RSSI Output |
| bit 5 set | Moving Average on RSSI Pin |
| bit 6 set | RSSI Pin Location 0 |
| bit 7 set | RSSI Pin Location 1 |

RSSI_Lag (EEPROM 0x67)

Controls a filter on the PWM output to smooth out the changes made to the PWM signal.

- Setting the value to a very low number results in very quick changing output.
- Setting the value to a higher number results in a slower varying PWM output.
- Setting the value to 0x00 results in an instantaneous RSSI.

Because RSSI is measured per hop and the radio can hop over 43 or 79 hops, instantaneous RSSI may move too quick to be of use as a signal strength indicator. The default value is 0x40 and should be sufficient for most applications. It should be set to a value of less than 0x80.

RSSI_Lag affects the PWM Output according to the following equations:

- $Cumulative_Lag = Cumulative_Lag + (RSSI_Current - Old_RSSI_Avg)$
- $New_RSSI_Avg = Old_RSSI_Avg + (Cumulative_Lag \bmod EE_Lag)$

Cumulative_Lag is then stored in memory until the next time RSSI is calculated.

If $(Cumulative_Lag \bmod EE_Lag) > 0$, then $Cumulative_Lag = \text{remainder of } (Cumulative_Lag \bmod EE_Lag)$

RSSI Output to PWM

A moving RSSI average can be written to the PWM Output as a signal strength indicator. The output pin to use, the threshold range for the RSSI and the RSSI Type reported can all be configured through EEPROM addresses.

The PWM output has a 39.3846 μ s period. The duty cycle is set by the RSSI value recorded by the transceiver and the RSSI Threshold High and RSSI Threshold Low values.

RSSI Threshold High (EEPROM 0x65)

The upper limit of the recorded RSSI reading. RSSI Values reported above this value (strong signals) report a 100% duty cycle on the PWM output.

RSSI Threshold Low (EEPROM 0x66)

The lower limit of the recorded RSSI reading. RSSI Values reported below this value (weak signals) report a 0% duty cycle on the PWM output.

To calculate the thresholds, use the following equation:

- $RSSI_Dec = (RSSI_dBm + 82) * 2 + 128$

Then convert this from decimal to hexadecimal notation.

Sleep Indicator (EEPROM 0x45, bit 6)

When enabled, GIO_1 toggles low during sleep and high when the module is awake.

Sleep Timer High/Low (EEPROM 0xCD-0xCE)

Two bytes to indicate the length of time to sleep in seconds.

Wake Count (EEPROM 0xCF)

Time in number of hops (13.19 ms each) to stay awake during cyclic sleep. This counter is an inactivity counter, therefore the counter is reset as long as the device continues to transmit packets over the RF interface.

Note: Once the Wake Counter has expired, the radio waits for a slot of inactivity (meaning that no RF packet is being received/transmitting, the serial port is idle, and the radio is not in AT Command mode). Once all of these conditions are met, the radio enters its sleep cycle. To prevent the radio from entering its sleep cycle or to force it out of its sleep cycle, the 9600 baud pin can be held low.

Sniff Permit (EEPROM 0x45, bit 0)

Sniff Permit allows a radio to receive a data packet from another radio on the network regardless of the destination MAC address in the packet. This allows an OEM to create a sniffer for all network traffic. Sniff Permit must be enabled on the transmitting radio to grant its permission to be heard. Sniff Report and Sniff Permit must be enabled on the sniffer radio to cause it to send sniffed packets out the serial port.

System ID (EEPROM 0x76)

System ID is similar to a password character or network number and makes network eavesdropping more difficult. A receiving transceiver will not go in range of or communicate with another transceiver on a different System ID. System ID can be ignored on a client by enabling Auto System ID.

Transmit Retries (EEPROM 0x4C)

When transmitting addressed packets, the RF packet is sent out to the receiver designated by its destination address. Transmit Retries is used to increase the odds of successful delivery to the intended receiver. Transparent to the OEM host, the transmitter sends the RF packet to the intended receiver. If the receiver receives the packet free of errors, it sends the transmitter an acknowledgement. If the transmitter does not receive this acknowledgement, it assumes the packet was never received and retries the packet. This continues until the packet is successfully received or the transmitter exhausts all of its retries. The received packet is only sent to the OEM host if and when it is received free of errors.

Note: Setting to 0 is equal to 256.

Unicast Only (EEPROM 0xC1, bit 3)

To prohibit transceivers from receiving broadcast packets, Unicast Only can be enabled. Unicast Only restricts the transceiver to only receive addressed packets.

Vendor ID

The Vendor ID, like the System ID, can be used to uniquely identify a network. Radios with the Vendor ID set, only communicate with other radios with the same set Vendor ID.

The Vendor ID is a protected EEPROM parameter and its value cannot be read. It can only be written once. OEMs should be aware that improperly setting the Vendor ID can cause communication issues. Setting the Vendor ID to an unknown setting effectively renders the radio unable to communicate in a network.

Note: The Vendor ID is a one-time write parameter; it cannot be read.

9600 Boot Option (EEPROM 0x57, bit 0)

When enabled, 9600 Boot Option causes the 9600 pin to be ignored on cold boot (power-up) and brown-out conditions. Therefore, the 9600 pin is only observed on warm boots (reset pin toggled). This can be helpful so that brown-out conditions don't cause the baud rate to change if the 9600 pin happens to be low at the time. When 9600 Boot Option is disabled, the 9600 pin is used for warm and cold boots as well as brown-out conditions.

EEPROM PARAMETERS

The RM024 utilizes a server-client network architecture to synchronize the frequency hopping. Each network must have one radio configured as a server and all other radios configured as clients. When a radio is configured as a server, it transmits a beacon at the beginning of each hop. Radios configured as clients default to a receive mode where they are scanning the available frequencies listening for a beacon from a server in their network. When a client detects the server's beacon, the client synchronizes to it and transitions the In Range pin low. When the server and the client are synchronized, they can begin transferring data.

Table 6: EEPROM Parameters

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|------------------------|----------------|----------------|---|---------|--|
| Product ID | 0x00 | 0x23 | | | Product identifier string, includes revision information for software and hardware. |
| Range Refresh | 0x3D | 1 | 0x01-0xFF | 0x48 | Specifies the maximum amount of time a transceiver reports In Range without having heard a server's beacon. Equal to hop period * value, do not set to 0x00. |
| Channel Number | 0x40 | 1 | 79 Hops: 0x00 – 0x4D, 43 Hops: 0x00 – 0x29 | 0x00 | Selects a unique hopping sequence in order to demarcate collocated networks. |
| Mode: Server/Client | 0x41 | 1 | 0x01: Server 0x02: Client | 0x02 | Sets the mode type. Each network has one and only one server and any number of clients. The server is responsible for transmitting beacons, which are used by the clients to locate and synchronize their hopping to that of the server. |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|--------------------|----------------|----------------|--------------------|---------|--|
| Baud Rate | 0x42 | 1 | 0x00-0x0A, 0xE3 | 0x09 | Baud Rate, see Serial Interface section for details. Default represents 115,200 kbps. Setting this address to 0xE3 allows the user to set a custom baud rate with the Baud_M and Baud_E registers. |
| Baud_M | 0x43 | 1 | 0x00-0xFF | 0x00 | Baud_M is used for setting custom baud rate, see Serial Interface Baud Rate section for more details. |
| Baud_E | 0x44 | 1 | 0x00-0xFF | 0x02 | Baud_E is used for setting custom baud rate, see Serial Interface Baud Rate section for more details. |
| Control 0 | 0x45 | 1 | Bit Adjustable | 0x88 | Settings are: bit-7: Reserved. Do not modify bit-6: Sleep Indicator: GIO_1 0 = Disable Sleep Indicator 1 = Enable Sleep Indicator bit-5: Reserved. Do not modify bit-4: Auto System ID 0 = Disable Auto System ID 1 = Enable Auto System ID bit-3: Command/Data Receive Disable 0 = Disable CMD/Data RX Disable 1 = Enable CMD/Data RX Disable (radio accumulates received RF packets until the CMD/Data pin goes high, at which time it forwards all stored packets to its host). bit-2: Legacy RSSI 0 = Disable Legacy RSSI 1 = Enable Legacy RSSI bit-1: Sniff Report 0 = Discard sniffed packets 1 = Report sniffed packets bit-0: Sniff Permit 0 = Disable Sniff Permit 1 = Enable Sniff Permit |
| Transmit Retries | 0x4C | 1 | 0x01-0xFF | 0x03 | Maximum number of times a packet is retransmitted when Addressed packets are being sent. Note: A setting of 0x00 will select 256 retries. |
| Broadcast Attempts | 0x4D | 1 | 0x01-0xFF | 0x03 | Number of times each packet is transmitted when Broadcast packets are being sent. Note: A setting of 0x00 selects 256 attempts. |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description | | | | | | | | | | | | | | | | | | | | | |
|-----------------|----------------|---|-----------------|------------|--|--------------|----------------|---------|------|----------|---|------|----------|--|------|----------|--|------|----------|---|------|----------|--|------|----------|---|
| Utility Retries | 0x4E | 1 | 0x00-0xFF | 0x03 | Used by Remote I/O mode to increase the odds of delivery of a Remote I/O line update. Sets the number of retries or attempts used to transmit a Remote I/O line status change. | | | | | | | | | | | | | | | | | | | | | |
| RF Profile | 0x54 | 1 | See Description | RF Profile | <table border="1"> <thead> <tr> <th>RF Data Rate</th> <th>Number of Hops</th> <th>Details</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>500 Kbps</td> <td>43 Valid for international use. Can be selected on any RM024 product</td> </tr> <tr> <td>0x01</td> <td>280 Kbps</td> <td>79 For FCC Markets only. This is the default settings for the RM024-x125-x models</td> </tr> <tr> <td>0x03</td> <td>280 Kbps</td> <td>43 This is the default setting for RM024-x50-x models</td> </tr> <tr> <td>0x0D</td> <td>500 kbps</td> <td>43 FEC Enabled. Valid for international use, can be selected by any RM024 product.</td> </tr> <tr> <td>0x0E</td> <td>280 kbps</td> <td>79 FEC Enabled. For FCC Markets only.</td> </tr> <tr> <td>0x10</td> <td>280 kbps</td> <td>43 FEC Enabled. Valid for international use, can be selected by any RM024 product.</td> </tr> </tbody> </table> <p>Superseded RF Data Rate (addresses 0x51 – 0x53) in FW v1.3 and beyond.</p> | RF Data Rate | Number of Hops | Details | 0x00 | 500 Kbps | 43 Valid for international use. Can be selected on any RM024 product | 0x01 | 280 Kbps | 79 For FCC Markets only. This is the default settings for the RM024-x125-x models | 0x03 | 280 Kbps | 43 This is the default setting for RM024-x50-x models | 0x0D | 500 kbps | 43 FEC Enabled. Valid for international use, can be selected by any RM024 product. | 0x0E | 280 kbps | 79 FEC Enabled. For FCC Markets only. | 0x10 | 280 kbps | 43 FEC Enabled. Valid for international use, can be selected by any RM024 product. |
| RF Data Rate | Number of Hops | Details | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x00 | 500 Kbps | 43 Valid for international use. Can be selected on any RM024 product | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x01 | 280 Kbps | 79 For FCC Markets only. This is the default settings for the RM024-x125-x models | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x03 | 280 Kbps | 43 This is the default setting for RM024-x50-x models | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0D | 500 kbps | 43 FEC Enabled. Valid for international use, can be selected by any RM024 product. | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0E | 280 kbps | 79 FEC Enabled. For FCC Markets only. | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x10 | 280 kbps | 43 FEC Enabled. Valid for international use, can be selected by any RM024 product. | | | | | | | | | | | | | | | | | | | | | | | | |
| Control 1 | 0x56 | 1 | 0x01-0xFF | 0x61 | <p>Settings are:</p> <ul style="list-style-type: none"> bit-7: Auto Destination on Beacons only (only functional when Auto Destination is also enabled) <ul style="list-style-type: none"> 0 = Disable - Radio sets destination based on any received packet 1 = Enable - Radio sets destination based only on the beacon from the server bit-6: Disable Hop Frame <ul style="list-style-type: none"> 0 = Disable (Turns on Hop Frame Pin) 1 = Enable (Turns off Hop Frame Pin) bit-5: Reserved bit-4: Auto Destination <ul style="list-style-type: none"> 0 = Use Destination Address 1 = Use Auto Destination bit-3: Client Auto Channel <ul style="list-style-type: none"> 0 = Disable Auto Channel 1 = Enable Auto Channel bit-2: RTS <ul style="list-style-type: none"> 0 = Disable RTS handshaking 1 = Enable RTS handshaking | | | | | | | | | | | | | | | | | | | | | |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|-------------------|----------------|----------------|-----------|---------|--|
| | | | | | bit-1: Duplex 0 = Half Duplex 1 = Full Duplex bit-0: Auto Config 0 = Auto Config Disabled 1 = Auto Config Enabled |
| Control 2 | 0x57 | 1 | 0x00-0xFF | 0x01 | Settings are: bit-7: Discard Framing Errors 0 = Framing error register is ignored 1 = Upon detecting a framing error, the entire packet is discarded bit-6: Hop Packet Delineation 0 = Disabled 1 = UART data is transmitted once per hop as soon as a minimum of six bytes have been accumulated, regardless of whether Interface Timeout or RF Packet Size has been reached. bit-5: Override 485 timing 0 = 485 DE-RE timing is set by the radio automatically 1 = 485 DE-RE timing is set by value in EEPROM bit-4: Remote Analog Enable 0 = Disable Remote Analog 1 = Enable Remote Analog bit-3: Remote I/O Mode 0 = Disable Remote I/O Mode 1 = Enable Remote I/O Mode bit-2: RS-485 Data Enable 0 = Disable RS-485 Data Enable 1 = Enable RS-485 Data Enable bit-1: Nine Bit Mode 0 = Disable Nine Bit Mode 1 = Enable Nine Bit Mode bit-0: 9600 Boot Option 0 = Disable 9600 Boot Option 1 = Enable 9600 Boot Option |
| Interface Timeout | 0x58 | 1 | 0x02-0xFF | 0x03 | Specifies a byte gap timeout, used in conjunction with RF packet size to determine when a packet coming over the interface is complete (200 μ s per increment). This is only used when Auto Config is disabled. |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|-------------------------|----------------|----------------|-----------------|---------|---|
| Antenna Select Override | 0x5B | 1 | 0x00-0xFF | 0xFF | When enabled with 0xE3, prevents radios with a –C Product Identifier from using the Antenna Switch option. On these models the radio automatically selects the U.fl port. |
| RF Packet Size | 0x5A | 1 | See Description | 0x60 | Used in conjunction with Interface Timeout; specifies the maximum size of an RF packet. Value in address is only used when Auto Config is disabled. RF Packet Size should not be less than 0x06 to ensure AT Commands can be issued. The maximum value is 0x60 for 280 kbps RF Data Rate and 0xEF for 500 kbps RF Data Rate. Note: Must be set to a minimum of 6 in order to send the Enter AT command. |
| CTS On | 0x5C | 2 | 0x0000 - 0x1FFF | 0x01C0 | If the transceiver buffer fills up and more bytes are sent to it before the buffer can be emptied, data loss occurs. The transceiver prevents this loss by deasserting CTS High as the buffer fills up and asserting CTS Low as the buffer is emptied. CTS should be monitored by the Host device and data flow to the radio should be stopped when CTS is high. CTS is deasserted (High) when the transmit buffer contains this many characters or more. |
| CTS Off | 0x5E | 2 | 0x0000 - 0x01FE | 0x0180 | If the transceiver buffer fills up and more bytes are sent to it before the buffer can be emptied, data loss occurs. The transceiver prevents this loss by deasserting CTS High as the buffer fills up and asserting CTS Low as the buffer is emptied. CTS should be monitored by the Host device and data flow to the radio should be stopped when CTS is High. Once the CTS is deasserted, CTS is reasserted (Low) when the transmit buffer contains this many or less characters. |
| Remote I/O Control | 0x60 | 1 | 0x00-0xFF | 0x00 | Settings are: bit-7: Use Pairs 0 = Disable pairs and allows radio I/O already set in bits 5-0 to be all input or all output 1 = Enable pairs with bits 5-0 set bit-6: All Inputs |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|---------------|----------------|----------------|-----------|---------|---|
| | | | | | <p>0 = All configured I/O are output (if Bit 7 is clear)</p> <p>1 = All configured I/O are inputs (if Bit 7 is clear)</p> <p>bit-5: Enable RXD/TXD Pair</p> <p>0 = Disable RXD/TXD Pair for Remote I/O Mode</p> <p>1 = Enable RXD/TXD Pair for Remote I/O Mode (disables serial data)</p> <p>bit-4: Enable RTS/CTS Pair</p> <p>0 = Disable RTS/CTS Pair for Remote I/O Mode</p> <p>1 = Enable RTS/CTS Pair for Remote I/O Mode (disables RTS)</p> <p>bit-3: Enable CMD/Data –GIO_2 Pair</p> <p>0 = Disable CMD/Data –GIO_2 Pair for Remote I/O Mode</p> <p>1 = Enable CMD/Data –GIO_2 Pair for Remote I/O Mode (May affect DE-RE)</p> <p>bit-2: Enable GIO_7/GIO_3 Pair</p> <p>0 = Disable GIO_7/GIO_3 Pair for Remote I/O Mode</p> <p>1 = Enable GIO_7/GIO_3 Pair for Remote I/O Mode (I/O Type depends on Remote Analog Enable)</p> <p>bit-1: Enable GIO_8/GIO_1 Pair1</p> <p>0 = Disable GIO_8/GIO_1 Pair for Remote I/O Mode</p> <p>bit-0: Enable GIO_4/GIO_0 Pair</p> <p>0 = Disable GIO_4/GIO_0 Pair for Remote I/O Mode</p> <p>1 = Enable GIO_4/GIO_0 Pair for Remote I/O Mode (may affect Hop Frame output)</p> <p>1. Pin 18 (GIO_8) on board revisions 0050-00203 Rev 0 and 0050-00196 rev 2 (and below) is internally not connected. This pin is unavailable as a GPIO on these boards.</p> |
| Sleep Control | 0x61 | 1 | 0x00-0xFF | 0x00 | <p>Used to modify sleep settings</p> <p>Settings are:</p> <p>bit-7: Reserved. Do not modify</p> <p>bit-6: Reserved. Do not modify</p> <p>bit-5: Reserved. Do not modify</p> <p>bit-4: Reserved. Do not modify</p> <p>bit-3: Reserved. Do not modify</p> |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description | | | | | | | | | | |
|----------------------|----------------------|----------------|-----------|---------|---|--------------|-------------|----------------------|---------------------|----------------------|----------------------|----------------------|----------------------|----------------------|---------------------|
| | | | | | bit-2: Reserved. Do not modify bit-1: Reserved. Do not modify bit-0: Cyclic Sleep 0 = Disables Cyclic Sleep 1 = Enables Cyclic Sleep | | | | | | | | | | |
| Max Power | 0x63 | 1 | 0x00-0x03 | 0x00 | Used to increase/decrease output power: <table border="1"> <thead> <tr> <th>RM024-x125-x</th> <th>RM024-x50-x</th> </tr> </thead> <tbody> <tr> <td>0x00: 21 dBm typical</td> <td>0x00: 17dBm typical</td> </tr> <tr> <td>0x01: 17 dBm typical</td> <td>0x01: 14 dBm typical</td> </tr> <tr> <td>0x02: 14 dBm typical</td> <td>0x02: 11 dBm typical</td> </tr> <tr> <td>0x03: 11 dBm typical</td> <td>0x03: 8 dBm typical</td> </tr> </tbody> </table> <p>Note: The transceivers are shipped at maximum allowable power.</p> | RM024-x125-x | RM024-x50-x | 0x00: 21 dBm typical | 0x00: 17dBm typical | 0x01: 17 dBm typical | 0x01: 14 dBm typical | 0x02: 14 dBm typical | 0x02: 11 dBm typical | 0x03: 11 dBm typical | 0x03: 8 dBm typical |
| RM024-x125-x | RM024-x50-x | | | | | | | | | | | | | | |
| 0x00: 21 dBm typical | 0x00: 17dBm typical | | | | | | | | | | | | | | |
| 0x01: 17 dBm typical | 0x01: 14 dBm typical | | | | | | | | | | | | | | |
| 0x02: 14 dBm typical | 0x02: 11 dBm typical | | | | | | | | | | | | | | |
| 0x03: 11 dBm typical | 0x03: 8 dBm typical | | | | | | | | | | | | | | |
| RSSI Threshold High | High 0x65 | 1 | 0x00-0xFF | 0xFF | Sets the High threshold for RSSI. If the RSSI is above RSSI Threshold High, then 0xFF (100% Duty Cycle) is written to the PWM Output. *See RSSI Output on PWM for more details | | | | | | | | | | |
| RSSI Threshold Low | 0x66 | 1 | 0x00-0xFF | 0x50 | Sets the Low threshold for RSSI. If the RSSI is below RSSI Threshold Low, then 0x00 (0% Duty Cycle) is written to the PWM Output. *See RSSI Output on PWM for more details | | | | | | | | | | |
| RSSI Lag | 0x67 | 1 | 0x00-0xFF | 0x40 | Constant controlling the rate of change of the PWM Output. | | | | | | | | | | |
| RSSI Control | 0x68 | 1 | 0x00-0xFF | 0xF1 | Settings are: bit-7: PWM Output Port1 bit-6: PWM Output Port1 bit-5: Use Average RSSI 0 = Ignore RSSI_Lag and report instantaneous RSSI 1 = Uses RSSI_Lag and reports moving average of RSSI bit-4: Invert Report 0 = PWM Outputs higher duty cycle for a stronger signal 1 = PWM Outputs lower duty cycle for a stronger signal bit-3: Unintended Report 0 = Disable Unintended report 1 = Reports RSSI on packets not intended for this transceiver bit-2: Broadcast Report 0 = Disable Broadcast Report 1 = Reports RSSI on Broadcast packets received | | | | | | | | | | |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description | | | | | | | | | | | | | | | | | | | | |
|-------------------------|----------------|----------------|---------------|---------|---|-------|-------|---------|---------------|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|----|
| | | | | | <p>bit-1: Addressed Report 0 = Disables Addressed Report 1 = Reports RSSI on packets specifically addressed to this transceiver</p> <p>bit-0: Beacon Report 0 = Disables Beacon Report 1 = Reports RSSI on the beacon from the server</p> <p>1. PWM Output Port. The OEM can control which ports are used for PWM Output according to the following table:</p> <table border="1"> <thead> <tr> <th>Bit 6</th> <th>Bit 7</th> <th>SMT Pin</th> <th>Pluggable Pin</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>6</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>7</td> </tr> <tr> <td>1</td> <td>0</td> <td>5</td> <td>19</td> </tr> <tr> <td>1</td> <td>1</td> <td>5</td> <td>19</td> </tr> </tbody> </table> | Bit 6 | Bit 7 | SMT Pin | Pluggable Pin | 0 | 0 | 2 | 6 | 0 | 1 | 1 | 7 | 1 | 0 | 5 | 19 | 1 | 1 | 5 | 19 |
| Bit 6 | Bit 7 | SMT Pin | Pluggable Pin | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 2 | 6 | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 7 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 5 | 19 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 5 | 19 | | | | | | | | | | | | | | | | | | | | | | |
| Beacon Skip | 0x6F | 1 | 0x00-0xFF | 0x00 | <p>When set, the transceiver sends (server) or listens (client) for a beacon on hops spaced by the Beacon Skip number. On a client, once the Beacon Skip count is reached, the client listens to every hop until it successfully hears a beacon. It then waits a number of hops specified by the Beacon Skip before listening again.</p> <p>Enabling this allows the transceiver to conserve power by disabling its RF circuitry during the beacon time. Enabling this on the server causes substantially longer sync times on the clients.</p> | | | | | | | | | | | | | | | | | | | | |
| Destination MAC Address | 0x70 | 6 | 0x00-0xFF | | Specifies destination for RF packets. Only the last four LSBs are used for RF communication. | | | | | | | | | | | | | | | | | | | | |
| System ID | 0x76 | 1 | 0x00-0xFF | 0x01 | Similar to network password. Radios must have the same System ID to communicate with each other. | | | | | | | | | | | | | | | | | | | | |
| MAC Address | 0x80 | 6 | | | Factory set unique MAC Address of radio. Only the last four LSBs are used for RF Communication. | | | | | | | | | | | | | | | | | | | | |
| Part Numbers | 0x90 | 16 | | | Factory set part number for the unit. | | | | | | | | | | | | | | | | | | | | |
| User Memory | 0xA0 | 16 | 0x00-0xFF | 0xFF | This memory is reserved host memory and is never used by the radio. The host is free to use this memory as desired and it is only modified when instructed to do so by the host. | | | | | | | | | | | | | | | | | | | | |
| API Control | 0xC1 | 1 | 0x01- | 0x00 | Settings are: | | | | | | | | | | | | | | | | | | | | |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|----------------|----------------|----------------|-----------|---------|---|
| | | | 0xFF | | bit-7: Broadcast Mode 0 = Disable 1 = Enable bit-6: In range High on Wake bit-5: Antenna Select 0 = Select Integrated Chip Antenna 1 = Select U.FL bit-4: Disable Status Bin 0 = Use Status Bin 1 = Ignore Status Bin bit-3: Unicast Only 0 = Disable 1 = Enable bit-2: Send Data Complete API 0 = Disable 1 = Enable bit-1: Transmit API 0 = Disable 1 = Enable bit-0: Receive API 0 = Disable 1 = Enable |
| Random Backoff | 0xC3 | 1 | 0x00-0xFF | 0x00 | The transceivers utilize a Carrier Sense Multiple Access (CSMA) protocol with Random Backoff and a programmable back-off seed. Therefore, in the event of a collision, the transceiver backs off and retries the packet. Specifically, when two transceivers collide with each other (transmitting packets at the same time), each transceiver chooses a random number of packet times that it will wait before retrying the packet. Ideally, they will each choose a different number and will be successful in the next transmission. A good rule of thumb is to set Random Backoff to a number slightly larger than the maximum number of transceivers that would be expected to be transmitting at the same time. Settings: 1 packet time: 0x00 1-2 packet times: 0x01 1-4 packet times: 0x03 1-8 packet times: 0x07 1-16 packet times: 0x0F 1-32 packet times: 0x1F |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|----------------------------|----------------|----------------|-----------|---------|---|
| | | | | | 1-64 packet times: 0x3F 1-128 packet times: 0x7F 1-256 packet times: 0xFF |
| PWM Initialize | 0xC8 | 1 | 0x00-0xFF | 0xFF | Initializes PWM Output (GIO_3) at boot to this value. |
| Remote I/O Initialize High | 0xC9 | 1 | 0x00-0xFF | 0xFF | Settings are: bit-7: Reserved. Do not modify bit-6: Reserved. Do not modify bit 5: RXD Initialize 0 = RXD is initialized Low at boot 1 = RXD is initialized High at boot bit 4: RTS Initialize 0 = RTS is initialized Low at boot 1 = RTS is initialized High at boot bit 3: CMD/Data Initialize 0 = CMD/Data is initialized Low at boot 1 = CMD/Data is initialized High at boot bit 2: GIO_7 Initialize 0 = GIO_7 is initialized Low at boot 1 = GIO_7 is initialized High at boot bit 1: GIO_8 Initialize1 0 = GIO_8 is initialized Low at boot 1 = GIO_8 is initialized High at boot bit 0: GIO_4 Initialize 0 = GIO_4 is initialized Low at boot 1 = GIO_4 is initialized High at boot 1. Pin 18 (GIO_8) on board revisions 0050-00203 Rev 0 and 0050-00196 rev 2 (and below) is internally not connected. This pin is unavailable as a GPIO on these boards. |
| Remote I/O Initialize Low | 0xCA | 1 | 0x00-0xFF | 0xFF | Settings are: bit-7: Reserved. Do not modify bit-6: Reserved. Do not modify bit 5: TXD Initialize 0 = TXD is initialized Low at boot 1 = TXD is initialized High at boot bit 4: CTS Initialize 0 = CTS is initialized Low at boot 1 = CTS is initialized High at boot bit 3: GIO_2 Initialize 0 = GIO_2 is initialized Low at boot 1 = GIO_2 is initialized High at boot bit 2: GIO_3 Initialize 0 = GIO_3 is initialized Low at boot |

| Parameters | EEPROM Address | Length (Bytes) | Range | Default | Description |
|-----------------|----------------|----------------|-----------|---------|---|
| | | | | | 1 = GIO_3 is initialized High at boot bit 1: GIO_1 Initialize 0 = GIO_1 is initialized Low at boot 1 = GIO_1 is initialized High at boot bit 0: GIO_0 Initialize 0 = GIO_0 is initialized Low at boot 1 = GIO_0 is initialized High at boot |
| Sleep Time High | 0xCD | 1 | 0x00-0xFF | 0xFF | Most Significant Byte of Sleep Time |
| Sleep Time Low | 0xCE | 1 | 0x00-0xFF | 0xFF | Least Significant Byte of Sleep Time, when combined with Sleep Time High, this is the total time to sleep in seconds (in hexadecimal) |
| Wake Count | 0xCF | 1 | 0x00-0xFF | 0xFF | Time in number of hops (13.19 ms each) to stay awake. This counter is an inactivity counter, therefore the device stays awake indefinitely as long as the device continues to send or receive packets. |
| D.O.B. | 0xE0 | 4 | | | Provides factory calibration and test date. |

CONFIGURING THE RM024

The RM024 can be configured using AT Configuration Commands. These commands can be issued only in Command mode. Command mode can be entered by setting the CMD/Data pin of a transceiver low or by issuing the Enter AT Command.

AT Commands

There are four types of configuration supported by the RM024:

- On-the-Fly commands for dynamic reprogramming of running memory
- EEPROM commands for making persistent changes to EEPROM
- Utility commands for dealing with Command mode
- Status commands for querying the radio for information

When in Command mode, the RF is still active and incoming RF packets are queued in the radio until Command mode is exited. If in Command mode via the CMD/Data pin and Command/Data RX Disable has been disabled, incoming RF packets are not queued and are not sent to the host immediately.

AT Commands sent to the RM024 must still adhere to the interface guidelines specified by the Interface Timeout and RF Packet Size. For this reason, standard terminal emulators, such as HyperTerminal, cannot be used to configure the RM024. Manually typing AT Commands causes the Interface Timeout to be reached between characters and the command is discarded if already in Command Mode or transmitted if not. AT commands should be issued either with the Laird Technologies OEM Configuration Utility or sent directly from a microcontroller or other host. When an invalid command is sent, the radio discards the data and no response is sent to the OEM Host. The following *Config and Diagnostic Commands* table (Table 7) shows a quick summary of the basic configuration and diagnostic commands available on the RM024. For detailed command information, please refer to the [Command Descriptions](#) immediately following the Quick Reference Table.

On-the-Fly Control Commands

The RM024 transceiver contains memory that holds many of the parameters that control the transceiver operation. Using the On-the-Fly command set allows many of these parameters to be viewed and changed during system operation. Because the memory these commands affect is dynamic, when the transceiver is reset, these parameters revert back to the settings stored in the EEPROM.

EEPROM Configuration Commands

Two commands are available to read and write the EEPROM of the radio. These commands are powerful as they can control the entire configuration of the radio. They should be used with caution as overwriting reserved areas of memory can adversely affect the operation of the radio. The radio must be reset for any changes to the EEPROM to take effect.

Utility Commands

Utility commands are used to enter and exit AT Command mode and to reset the radio.

Status Commands

Status commands are used to query the radio for information. Status commands can be used to poll GPIOs, ADCs, or to retrieve information about the state of the network. Status commands do not affect the operation of the transceiver aside from being in Command mode.

Table 7: Config and Diagnostic Commands

| Command Name | Command (all bytes in hex) | Return (all bytes in hex) |
|---|---|------------------------------------|
| UTILITY COMMANDS | | |
| Enter AT Command Mode | <0x41> <0x54> <0x2B> <0x2B> <0x2B> <0x0D> | <0xCC> <0x43> <0x4F> <0x4D> |
| Exit AT Command Mode | <0xCC> <0x41> <0x54> <0x4F> <0x0D> | <0xCC> <0x44> <0x41> <0x54> |
| Enter Sleep | <0xCC> <0x86> <Mode> <Res> <Timer_H> <Timer_L> | None |
| Soft Reset | <0xCC> <0xFF> | None |
| Restore Factory Settings and Soft Reset | <0xCC> <0xFF> <0xDF> | None |
| STATUS COMMANDS | | |
| Status Request | <0xCC> <0x00> <0x00> | <0xCC> <Firmware Version> <Status> |
| Check Status Registry | <0xCC> <0x00> <0x01> | <0xCC> <Response[32bits]> |
| Bin Analyzer | <0xCC> <0x8F> <Data> <Optional NumRuns> | <0xCC> <Data> |
| Read Temperature | <0xCC> <0xA4> | <0xCC> <Temperature> |
| ON-THE-FLY COMMANDS | | |
| Change Channel | <0xCC> <0x02> <Channel> | <0xCC> <Channel> |
| Change Server/Client | <0xCC> <0x03> <Data> | <0xCC> <Firmware Version> <Status> |
| Set Broadcast Mode | <0xCC> <0x08> <Data> | <0xCC> <Data> |
| Read IRAM | <0xCC> <0x0A> <Location> | <0xCC> <Value> |

| Command Name | Command (all bytes in hex) | Return (all bytes in hex) |
|---------------------------|--|--|
| Write IRAM | <0xCC> <0x0B> <Location> <Value> | <0xCC> <Location> <Value> |
| Write Destination Address | <0xCC> <0x10> <Dest Address last three bytes> | <0xCC> <Dest Address> |
| Read Destination Address | <0xCC> <0x11> | <0xCC> <Dest Address> |
| Auto Destination/Channel | <0xCC> <0x15> <Data> | <0xCC> <Data> |
| Read API Control | <0xCC> <0x16> | <0xCC> <API Control> |
| Write API Control | <0xCC> <0x17> <API Control> | <0xCC> <API Control> |
| Read Digital Input | <0xCC> <0x20> | <0xCC> <Data> |
| Read ADC | <0xCC> <0x21> <Data> | <0xCC> <ADC Hi> <ADC Lo> |
| Get Last RSSI | <0xCC> <0x22> | <0xCC> <RSSI> |
| Write Digital Outputs | <0xCC> <0x23> <Data> | <0xCC> <Data> |
| Write PWM | <0xCC> <0x24> <Data> | <0xCC> <Data> |
| Set Power Control | <0xCC> <0x25> <Power> | <0xCC> <Power> |
| Antenna Switch Command | <0xCC> <0x26> <Port Select> | <0xCC> <0x26> <Port Select> |
| EEPROM COMMANDS | | |
| EEPROM Byte Read | <0xCC> <0xC0> <Start> <Length> | <0xCC> <Start> <Length> <Data [n-0]> |
| EEPROM Byte Write | <0xCC> <0xC1> <Start> <Length> <Data> | <Start> <Length> <Last Byte Written> |
| Write Flash | <0xCC><0xC4><Start_H><Start_L> <Length_H><Length_L><Data> | <0xCC><0xC4><Result> <Start_H><Start_L> |
| Decrypt New Image | <0xCC><0xC5> | <0xCC><0xC5><Data> |
| Erase Flash | <0xCC><0xC6> | <0xCC><0xC6> |
| Read Flash | <0xCC><0xC9><Start_H><Start_L> <Length_H><Length_L> | <0xCC><0xC9><Result><Start_H> <Start_L><Length_H> <Length_L><Data> |
| Set Vendor ID | <0xCC> <0xF2> <0x06> <Vid_H><Vid_L> | <0xCC> <0xF2> <0x06> <Data> |

Utility Commands

Enter AT Command Mode

Prior to sending this command, the OEM Host must ensure that the transceiver's serial RX buffer is empty. This can be accomplished by framing the command with rest periods equal to the Interface Timeout or greater (depending on radio configuration). If the buffer is not empty, the radio might concatenate the string to existing data in the buffer and it will be sent over the RF.

| | |
|----------------------------------|---|
| Command: | <0x41> <0x54> <0x2B> <0x2B> <0x2B> <0x0D> |
| Number of Bytes Returned: | 4 |
| Response: | <0xCC> <0x43> <0x4F> <0x4D> |

Exit AT Command Mode

The OEM Host should send this command to exit AT Command mode and resume normal operation.

| | |
|----------------------------------|------------------------------------|
| Command: | <0xCC> <0x41> <0x54> <0x4F> <0x0D> |
| Number of Bytes Returned: | 4 |
| Response: | <0xCC> <0x44> <0x41> <0x54> |

Enter Deep Sleep

The OEM host issues this command to put the module into a sleep state to minimize current draw. While sleeping, the processor has all interfaces disabled (including RF and serial). Pin states are maintained while in sleep. A reset of the radio (soft or hard) cancels any current sleep mode. Two sleep modes are supported in the module: PM2 and PM3.

- PM2:** The module can either be awakened by the Sleep Interrupt pin (Force 9600) going Low or the sleep timer expiring. When a server enters PM2, it stops sending beacons until it awakes. Upon awakening from PM2, a client calculates which frequency the server should be located on (based on the duration of the sleep period) and attempts to re-sync with the server on that frequency. As such, a client is not permitted to transmit data following PM2 until receiving a beacon from its server. Enabling In Range High on Wake permits the host to monitor the In Range pin to determine when the client has received a beacon from its server.

Note: PM2 only maintains sync for two minutes. See the Firmware Version 1.3-0 Release Notes.

- PM3:** The module can only be awakened by the Sleep Interrupt pin (Force 9600). The sleep timer is not active in PM3 and the bytes controlling the timer (RES, Timer_H, Timer_L) are disregarded and can be omitted from the command as well.

Sleep Time is calculated according to the following formula:

$$\text{Sleep Time} = ((\text{RES} * \text{TIMER (H\&L)}) / 32.768 \text{ khz})$$

Sleep modes must last for a minimum of 11.08 ms

| | |
|----------------------------------|---|
| Command: | <0xCC> <0x86> <Mode> <RES> <Timer_H> <Timer_L> |
| Number of Bytes Returned: | None |
| Response: | None |
| Parameter Range: | <Mode> = 0x02 = PM2 0x03 = PM3 <RES> = 0x00 = 20 0x01 = 21 0x02 = 22 0x03 = 23 <Timer_H/TIMER_L> = 0x0001 - 0xFFFF |

Reset

The OEM host issues this command to perform a soft reset of the transceiver. Any transceiver settings modified by CC commands revert to the values stored in the EEPROM.

| | |
|----------------------------------|---------------|
| Command: | <0xCC> <0xFF> |
| Number of Bytes Returned: | None |
| Response: | None |

Restore Factory Defaults

This command restores the original factory EEPROM settings to the module. The command also issues a reset as well as restoring the EEPROM. The default EEPROM is active following the reset.

| | |
|----------------------------------|----------------------|
| Command: | <0xCC> <0xFF> <0xDF> |
| Number of Bytes Returned: | 0 |
| Response: | None |

Status Commands

Status Request

The OEM Host issues this command to request the firmware and link status of the transceiver.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x00> <0x00> |
| Number of Bytes Returned: | 3 |
| Response: | <0xCC> <Firmware Version> <Status> |
| Parameter Range: | <Firmware> = Radio Firmware Version <Status> = 0x02: Server 0x03: Client in Range 0x01: Client not in Range |

Check Status Registry

Status Registry contains error codes which may be useful for the OEM to note.

Note: Sleep crystal is only checked on boot.

Wake Event is written every time the module wakes from sleep.

Boot Error is written on boot.

Calibration Event indicates that the module is performing a calibration of the external crystal used to track the length of sleep during a Sleep Timer sleep mode. If Sleep Calibration is enabled in the EEPROM, this bit will always be set. The sleep timer will constantly be calibrated.

| | |
|----------------------------------|---------------------------|
| Command: | <0xCC> <0x00> <0x01> |
| Number of Bytes Returned: | 5 |
| Response: | <0xCC> <Response[32bits]> |

| | |
|-------------------------|--|
| Parameter Range: | <p><Response> = bit 0: Sleep Crystal check- If set Sleep timing crystal failed to Start bit 1: Wake Event-If set, the radio was awakened from sleep with the Sleep Interrupt (Force 9600) pin bit 2: Boot Error-If set, last reset was due to brown-out or power-on reset. If not set, then reset was due to uP_Reset pin or Soft Reset command. bit 3: Calibration Event- If set, then the sleep timer is performing a calibration.</p> |
|-------------------------|--|

Check Firmware Status

The Check Firmware Status command is used to verify proper loading of the firmware. Since the firmware must be loaded via multiple images, this command verifies that all pages have been upgraded. Should an error be encountered, the first page encountered that is out-of-date is reported in the response.

| | |
|-------------------------|---|
| Command: | <0xCC><0x00><0x02> |
| Response: | <0xCC><0x02><Error><Page> |
| Parameter Range: | <p><Error> = 0x00: success, 0x09: firmware not complete <Page> = 0x00: no error, 0x10-0x1E: corresponds to first page in memory that needs to be upgraded</p> |

Bin Analyzer

The Bin Analyzer is a powerful command for understanding the link conditions between two radios over the entire frequency hopping spectrum. The Bin Analyzer causes the local radio to send an RSSI request packet to the radio specified in the Destination Address field. The remote radio responds with RSSI information and this is then streamed to the OEM through the Serial UART. This sequence occurs once per hop.

Due to the random frequency hopping sequence of the radios, it is not possible to associate a specific bin # with a specific frequency, though the OEM can use the Bin Analyzer response to identify any possible interferers and to provide a quantitative analysis of the total number of good versus bad bins.

The Bin Analyzer command streams data results back for each hop until the command is turned off or the number of runs is met. The Bin Analyzer command must be issued from Command mode, but it continues to stream results back even after the OEM has exited Command mode. The Bin Analyzer packet is sent as part of the RF packet header and does not affect the throughput of data between two radios. See the [RSSI](#) section for equations to convert RSSI to a dBm value.

| | |
|----------------------------------|-----------------------------------|
| Command: | <0xCC> <0x8F> <Control> <NumRuns> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <0x01> |
| Bing Response Stream: | <0xCC> <Bin#> <RSSI_1> <RSSI_2> |

| | | |
|-------------------------|--------------|--|
| Parameter Range: | <Control> = | 0x00= Turn Bin Analyzer Off 0x01= Turn Bin Analyzer On |
| | <NumRuns> = | 0x00= Continuous 0x01- 0xFF= Number of runs [bins] |
| | <Bin#> = | Bin# from 0 to max bin number according to the selected RF Profile |
| | <RSSI_1> = | How strongly the remote radio heard the local radio's bin request. |
| | <RSSI_2> = | How strongly the local radio heard the remote radio's response |
| | Note: | RSSI 1 and RSSI 2 range from 0x80 to 0x7F. |

Command Descriptions

Read Temperature

The OEM host issues this command to read the onboard temperature sensor.

Note: The temperature sensor is uncalibrated and has a tolerance of $\pm 3^{\circ}\text{C}$. For calibration instructions, contact Laird technical support.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0xA4> |
| Number of Bytes Returned: | 2 |
| Response: | 0xCC <Temp.> |
| Parameter Range: | <Temp.> = Temperature (Celsius) in two's complement format |

Change Channel

The OEM host issues this command to set a new RF channel number.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x02> <Channel> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <Channel> |
| Parameter Range: | For 43-channel hop mode, range is 0x00 to 0x29. For 79-channel hop mode, range is 0x00 to 0x4D. <Channel> = RF Channel to switch to, in the range 0x00 to 0x29 |

Change Server/Client

The OEM host issues this command to set the module as a server or a client on-the-fly.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x03><Data> |
| Number of Bytes Returned: | 3 |
| Response: | <0xCC> <Firmware Version><Status> |
| Parameter Range: | <Data> 0x02 = Server 0x03 = Client <Firmware> = Radio Firmware version <Status> = 0x02: Server = 0x03: Clients In Range = 0x01: Client not In Range |

Set Broadcast Mode

The host issues this command to set the addressing mode in the radio. If both Broadcast mode and Auto Destination are enabled, the radio sends the RF packets in Broadcast mode.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x08> <Data> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <Data> |
| Parameter Range: | <Data> = 0x00: Disable Broadcast Mode 0x01: Enable Broadcast Mode |

Read IRAM

This command is used to view the IRAM contents. The IRAM contains the active (in memory) copy of the EEPROM parameters. Some of these fields are available for reading and writing to provide on-the-fly instantaneous setting of parameters. Parameters changed via the IRAM commands take effect immediately and remain in effect until the radio is reset. IRAM changes do not persist after a reboot.

Restrictions on the specific IRAM parameters are the same as for the EEPROM parameters.

Available locations in IRAM are:

| Name | Address |
|---------------------------|---------|
| Range Refresh | 0x3D |
| RF Channel | 0x41 |
| Interface Timeout | 0x58 |
| RF Packet Size | 0x5A |
| CTS_On_H | 0x5C |
| CTS_On_L | 0x5D |
| CTS_Off_H | 0x5E |
| CTS_Off_L | 0x5F |
| Max Power | 0x63 |
| Destination MAC Address 3 | 0x72 |
| Destination MAC Address 2 | 0x73 |
| Destination MAC Address 1 | 0x74 |
| Destination MAC Address 0 | 0x75 |
| System ID | 0x76 |

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x0A> <Location> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <Value> |
| Parameter Range: | <Location>: [Dependant of parameter] <Value>: 0x00-0xFF |

Write IRAM

This command is used to change the IRAM contents. The IRAM contains the active (in memory) copy of the EEPROM parameters. Some of these fields are available for reading and writing to provide on-the-fly instantaneous setting of parameters. Parameters changed via the IRAM commands take effect immediately and remain in effect until the radio is reset. IRAM changes do not persist after a reboot.

Restrictions on the specific IRAM parameters are the same as for the EEPROM parameters.

Available locations in IRAM are:

| Name | Address |
|---------------------------|---------|
| Range Refresh | 0x3D |
| RF Channel | 0x41 |
| Interface Timeout | 0x58 |
| RF Packet Size | 0x5A |
| CTS_On_H | 0x5C |
| CTS_On_L | 0x5D |
| CTS_Off_H | 0x5E |
| CTS_Off_L | 0x5F |
| Max Power | 0x63 |
| Destination MAC Address 3 | 0x72 |
| Destination MAC Address 2 | 0x73 |
| Destination MAC Address 1 | 0x74 |
| Destination MAC Address 0 | 0x75 |
| System ID | 0x76 |

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x0B> <Location> <Value> |
| Number of Bytes Returned: | 3 |
| Response: | <0xCC> <Location> <Value> |
| Parameter Range: | <Location>: [Dependant of parameter] <Value>: 0x00-0xFF |

Write Destination Address

The OEM host issues this command to the transceiver to change the destination address.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x10> <MAC1> <MAC2> <MAC3> |
| Number of Bytes Returned: | 4 |
| Response: | <0xCC> <MAC1> <MAC2i> <MAC3> |
| Parameter Range: | 0x00 - 0xFF corresponding to the 3 LSBs of the destination MAC Address. MAC3 is the LSB. |

Destination Address

The OEM host issues this command to the transceiver to read the destination address.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x11> |
| Number of Bytes Returned: | 4 |
| Response: | <0xCC> <MAC1> <MAC2> <MAC3> |
| Parameter Range: | 0x00 - 0xFF corresponding to the 3 LSBs of the destination MAC Address. MAC3 is the LSB. |

Auto Destination / Auto Channel

The host issues this command to change the Auto Destination setting. When issuing this command, the Auto Destination setting is only changed if the corresponding enable bit is set. Otherwise, the command performs a read of Auto Destination. If both Broadcast mode and Auto Destination are enabled, the radio sends the RF packets in Broadcast mode.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x15> <Data> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <Auto Dest> |
| Parameter Range: | <Auto Dest> = bit 7: Ignored bit 6: Ignored bit 5: Enable Modification of Auto Channel bit 4: Enable Modification of Auto Destination bit 3: Ignored bit 2: Ignored bit 1: Auto Channel bit 0: Auto Destination |

Read API Control

The OEM host issues this command to read the API Control byte.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0x16> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <API Control> |
| Parameter Range: | <Auto Dest> = bits 7-3: 0 bit-2: Send Data Complete bit-1: Transmit API bit-0: Receive API |

Write API Control

The OEM host issues this command to write the API Control byte to enable or disable the API features.

| | |
|----------------------------------|-----------------------------|
| Command: | <0xCC> <0x17> <API Control> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <API Control> |

| | |
|-------------------------|---|
| Parameter Range: | <Auto Dest> = bits 7-3: 0 bit-2: Send Data Complete bit-1: Transmit API bit-0: Receive API |
|-------------------------|---|

Read Digital Inputs

The OEM host issues this command to read the state of both digital output lines.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0x20> |
| Number of Bytes Returned: | 2 |
| Response: | 0xCC <Digital In> |
| Parameter Range: | <Digital In> = bit-1: GIO_8 = bit-0: GIO_4 |

Read ADC

The OEM host issues this command to read the analog to digital converters at up to 12-bit resolution. Higher resolutions can cause slower responses from the command. The time required for a conversion is: $T_{conv} = (\text{decimation rate} + 16) * 0.23 \mu\text{s}$.

In most applications this is used to measure the input voltage (to detect reduced battery power) with $V_{cc}/3$, the temperature sensor or the Analog input pin. For the most accurate results the 1.25 V internal reference should be chosen, though this limits the OEM to a maximum A/D Input of 1.25 V. $V_{cc}/3$ and the temperature sensor readings should use the internal 1.25 V reference as those voltages will never exceed 1.25 V.

The ADC result is represented in a two's complement form. The result is the difference between ground and the selected channel and is a value between -2048 and 2047 with 2047 representing the maximum value where the ADC result equals the reference voltage and -2048 equals the negative of the reference voltage. The ADC cannot measure a voltage higher than the reference voltage.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0x21> <Data> |
| Number of Bytes Returned: | 3 |
| Response: | 0xCC <Hi ADC> <Lo ADC> |
| Parameter Range: | <Data bits 7-6> = <Reference Voltage> 00: Internal 1.25V reference 10: Voltage on Vcc pin <Data bits 5-4> = <Resolution> 00: 64 decimation rate (7 bits resolution) 01: 128 decimation rate (9 bits resolution) 10: 256 decimation rate (10 bits resolution) 11: 512 decimation rate (12 bits resolution) <Data bits 3-0> = <Channel> 0000: AD/In (GIO_7) 1100: GND 1101: Positive Voltage Reference |

1110: Temperature Sensor
 1111: Vcc/3
 <Hi ADC> = MSB or requested 12-bit ADC value
 <Lo ADC> = LSB of requested 12-bit ADC value

Get Last RSSI

This command is used to report signal strength (RSSI) information from the last RF packet the radio received. This command reports the signal strength of any received RF packet including the server beacon, data packets, and RF packets that are not intended for this radio (e.g., packets with a different Destination MAC Address than this radio).

| | |
|----------------------------------|---------------------|
| Command: | <0xCC> <0x22> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <RSSI> |
| Parameter Range: | <RSSI>: 0x80 – 0x7F |

Write Digital Outputs

The OEM host issues this command to write both digital output lines to particular states. These states are persistent unless overwritten by another enabled function (such as Hop Frame).

| | |
|----------------------------------|--|
| Command: | <0xCC> <0x23><Digital Out> |
| Number of Bytes Returned: | 2 |
| Response: | 0xCC <Digital Out> |
| Parameter Range: | <Digital Out> = bit-1: GIO_1 = bit-0: GIO_0 |

Write PWM Output

The PWM output is a square wave with a 39.3846 μ S period. The PWM ratio is the ratio of the high pulse time to the low pulse time. A value of 0x00 outputs a continuous low signal. A ratio of 0xFF outputs a continuous high signal. A ratio of 0x80 puts out a repeating pulse of 19.6923 μ S high and 19.6923 μ S Low.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0x24> <PWM Ratio> |
| Number of Bytes Returned: | 2 |
| Response: | 0xCC <0x24> |
| Parameter Range: | <PWM Ratio> = 0x00 -0xFF, the ratio of the high pulse versus the Low pulse for a single period. |

Set Max Power

The OEM host issues this command to adjust the maximum output power.

Max Power settings are typical and can vary by 2 dBm. The maximum output power never exceeds +18 dBm for RM024-x50-x or +21dBm for RM024-x125-x.

| | |
|----------------------------------|------------------------|
| Command: | <0xCC> <0x25><Max Pwr> |
| Number of Bytes Returned: | 2 |
| Response: | 0xCC <Max Pwr> |

| | | |
|-------------------------|----------------------|----------------------|
| Parameter Range: | <Max Power> | |
| | RM024-x125-x | RM024-x50-x |
| | 0x00: 21 dBm typical | 0x00: 17dBm typical |
| | 0x01: 17 dBm typical | 0x01: 14 dBm typical |
| | 0x02: 14 dBm typical | 0x02: 11 dBm typical |
| | 0x03: 11 dBm typical | 0x03: 8 dBm typical |

Antenna Select Command

The OEM host issues this command to select the antenna port.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0x26><Port Select> |
| Number of Bytes Returned: | 3 |
| Response: | 0xCC <0x26> <Port Select> |
| Parameter Range: | <Port Select> 0x00 (Integrated Chip Antenna) 0x01 (U.FL Port) |

EEPROM Byte Read

Upon receiving this command, a transceiver responds with the desired data from the EEPROM addresses requested by the OEM host.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0xC0> <Start> <Length> |
| Number of Bytes Returned: | 4+ |
| Response: | <0xCC> <Start> <Length> <Data> |
| Parameter Range: | <Start> = EEPROM address to begin reading at <Length> = Length of data to be read <Data> = Requested data |

EEPROM Byte Write

Upon receiving this command, a transceiver writes the data byte to the specified address but does not echo it back to the OEM host until the EEPROM write cycle is complete.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0xC1> <Start> <Length> <Data> |
| Number of Bytes Returned: | 3 |
| Response: | <Start> <Length> <Last byte> |
| Parameter Range: | <Start> = EEPROM address to begin writing at <Length> = Length of data to be written <Data> = Data to be written <Last byte> = Value of last byte written |

Vendor ID

This command is used to set the Vendor ID in the radios. The Vendor ID is a one-time writable parameter which is not able to be read back for security purposes. Radios with the Vendor ID set only communicate to other radios that have Vendor ID set to the same value. Once the Vendor ID is set it cannot be changed. Setting this value incorrectly results in a radio which is not able to communicate with other radios on the network. Setting the Vendor ID is not necessary for the radios to operate and care should be taken when using it. To see if the Vendor ID has been set the OEM can issue the command <0xCC> <0xF2> <0x06> <0xFF> <0xFF>. If the response is 0xFF 0xFF then VID has not been set. If the response is 0x00 0x00 then the VID has been set. Issuing the command <0xCC> <0xF2> <0x06> <0xFF> <0xFF> will not set the Vendor ID.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0xF2> <0x06> <Vid_H> <Vid_L> |
| Number of Bytes Returned: | 2 |
| Response: | <0xCC> <0xF2> <0x06> <Value> |
| Parameter Range: | <Vid_H> = 0x00-0xFF <Vid_L> = 0x00-0xFF <Value> = If Vid is set the response will be 0x00 0x00. If Vid is not set the response will be 0xFF 0xFF |

SERIAL FIRMWARE UPGRADES

Overview

The active firmware running on the module may be upgraded via the serial interface. The Windows OEM Configuration Utility v5.07 and above can perform this operation for modules with access to a PC, but for embedded applications it may be necessary for the OEM to program the FW upgrade through a microcontroller. Upgrade of the FW over the air is not supported on the RM024. The firmware upgrade procedure will not affect the radio EEPROM settings except to update the version number.

Upgrading Via Windows OEM Configuration Utility

Upgrading through the OEM Configuration Utility is a straight-forward process. The PC Settings tab has an option when the RM024 is the selected product to perform a Radio Firmware Update. This section of the OEM software can be used to upgrade the FW of the radio connected to Port 1 of the software.

To perform a firmware upgrade, follow these steps (illustrated in [Figure 1](#)):

1. Connect the Development Kit or other interface board with the RM024 to the PC.
2. Select **Find Ports**, then select the appropriate Windows Comm Port from the drop-down menu.

Note: If the appropriate port doesn't appear, click Add Ports to add the port manually, then repeat step 2.

3. Set the correct settings for the radio in Port 1. By default this is 115,200-8-N-1 with Hardware Handshaking enabled.
4. Verify that the Port Status in Port 1 is Open.
5. In the Radio Firmware Upgrade box, select **Firmware Upgrade**.
6. In the Open File Window, select the appropriate RM024 FW binaries.
7. When the download is successful, a success window appears and the new firmware is now active.

- 8. Send the <0xCC><0x00><0x02> command to the radio from the Command page to verify that all pages have been upgraded. If this command reports an error, locate the corresponding binary image and repeat this process with that file.

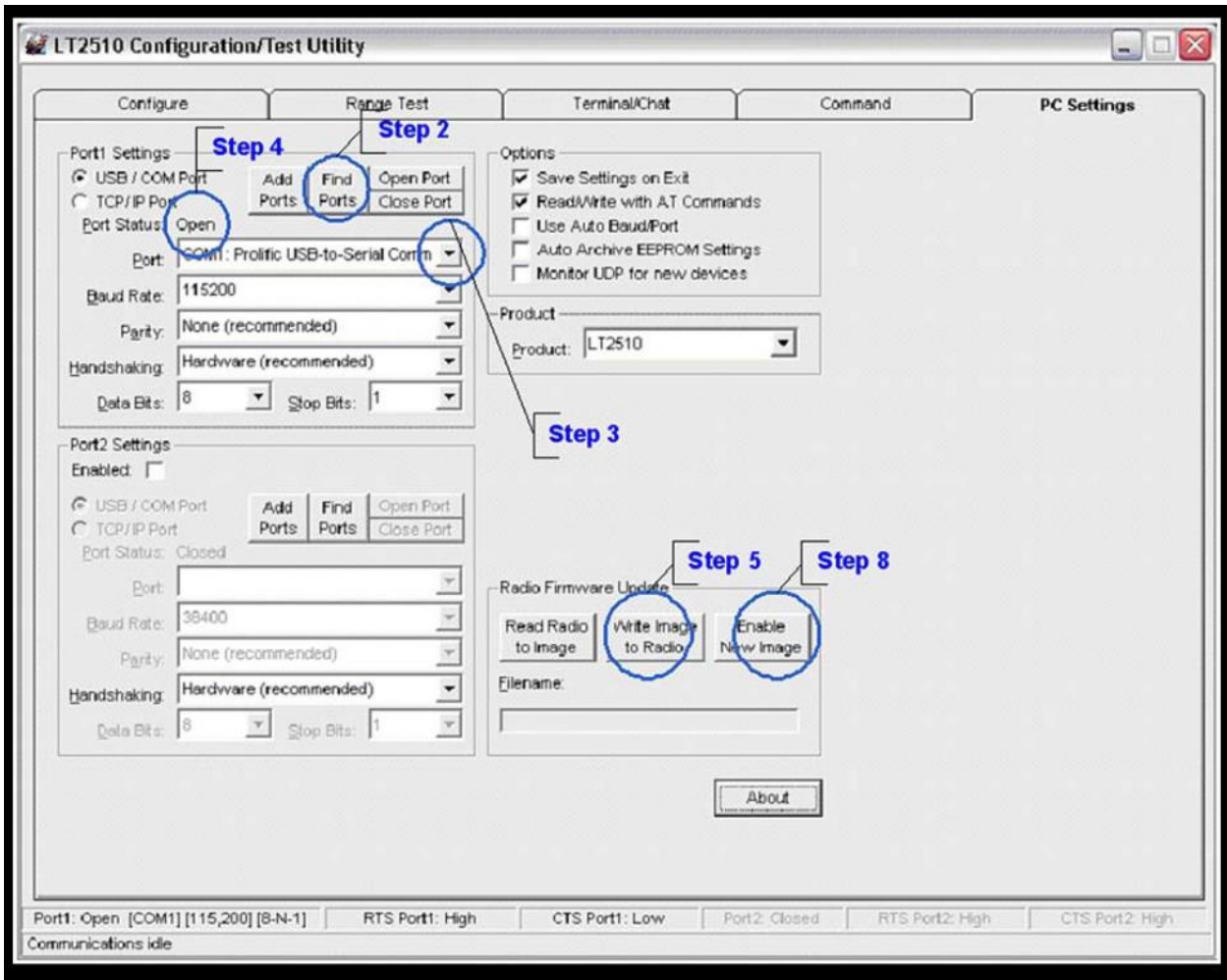


Figure 1: Upgrading firmware in Laird Configuration Utility

Upgrading FW Commands

These commands are used to upgrade the FW on the RM024; these commands are only valid on FW version 1.3 and above.

Command Descriptions

Write Flash

This command is used to write an encrypted binary file to the reserve flash. The OEM should first issue the Erase Flash command before writing a new image to the flash. Writes can only be done in increments less than 256 bytes. The binary file should start being written at address 0x00 and incremented from there.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0xC4> <StartAddress[1,0]> <Length[1,0]> <BytesToWrite> |
| Number of Bytes Returned: | 5 |
| Response: | 0xCC> <0xC4> <Result> <StartAddress[1,0]> |
| Parameter Range: | <StartAddress[1,0]>: 0x00 0x00 – 0x3B 0xFF <Length[1,0]>: 0x00 0x01 – 0x00 0xFF <BytesToWrite>: The actual data being written <Result> 0x00 No Error 0x03 Command Timed Out 0x06 Bounds Exceeded (Start Address + Length is outside the image download area) 0x04 A valid image already exists, erase that Flash before overwriting. |

Decrypt New Message

This command is used to decrypt the binary file that was previously downloaded to the radio. The next reboot of the module causes this decrypted image to be loaded into the memory. Once the file is decrypted, it cannot be read from the module.

| | |
|----------------------------------|--|
| Command: | <0xCC> <0xC5> |
| Number of Bytes Returned: | 3 |
| Response: | <0xCC> <0xC5> <Data> |
| Parameter Range: | <Data> 0x00 No Error 0x01 Error in File integrity, erase Flash and retry download 0x02 Not enough free memory. Reset the module and try again. 0x04 Image Already Decrypted |

Erase New Image from Flash

It is not necessary to issue this command. The radio automatically erases the new image upon loading it into active memory. However, if the wrong image is loaded, the image can be erased. It is not possible to overwrite a downloaded image without erasing it or activating it. This command erases the memory between addresses 0x0000 and 0x7FF. Memory between addresses 0x800 and 0x3BFF is erased automatically by the first write exceeding to address 0x800 or higher. Because this memory contains active code, the radio now disconnects from the network until the upgrade process is completed.

| | |
|----------------------------------|---------------|
| Command: | <0xCC> <0xC6> |
| Number of Bytes Returned: | 2 |
| Response: | |
| Parameter Range: | <0xCC> <0xC6> |

Read Flash

This command is used to read the encrypted image of the flash that resides on the radio module. If the image has already been decrypted, then this command reports an error.

Note: The length of this command is dictated by the dynamic memory heap. If the heap is mostly free, then a length of up to 700 bytes could be accepted. However, if the heap is full, the command returns with an error or possibly no response at all if the heap is completely full.

| | |
|----------------------------------|---|
| Command: | <0xCC> <0xC9> <StartAddress[1,0]><Length[1,0]> |
| Number of Bytes Returned: | 5 |
| Response: | <0xCC> <0xC9> <Result> <StartAddress[1,0]> <Data> |
| Parameter Range: | <StartAddress[1,0]>: 0x00 0x00 – 0x3A 0xFF <Length[1,0]>: 0x00 0x00 – 0x02BC <BytesToWrite>: The actual data being written <Result> 0x00 No Error 0x03 Command Timed Out 0x06 Bounds Exceeded (Start Address + Length is outside the image download area) 0x04 Image Already Decrypted 0x02 Not Enough Free Memory, try the command again with a shorter Length <Data>: Contents of Flash |

Process to Manually Upgrade RM024

By using the above AT commands, it is possible to upgrade a RM024 over the UART from a microcontroller or other device. To do this, follow these basic steps:

1. Make sure the OEM host has a copy of an encrypted binary firmware upgrade files.
2. Erase the Existing flash on the RM024
 Command: 0xCC 0xC6
 Response: 0xCC 0xC6
3. Write binary images to flash. Note that the firmware is loaded in multiple images. The primary image (the file denoted as “[00]”) **must** be sent first. The first write to memory location 0x800 or above incurs a 300 ms delay while the memory between addresses 0x800 and 0x3BFF is erased. Because this memory contains active code, the radio disconnects from the network until the upgrade process is completed.
 - While not end of binary file:
 - Command: 0xCC 0xC4 0x00 0x00 0x00 0x80 <80 bytes of data>
 - Response: 0xCC 0xC4 0x00 0x00 0x00
 - Check EEPROM Write:
 - Command 0xCC 0xC9 0x00 0x00 0x00 0x80
 - Response: 0xCC 0xC9 0x00 0x00 0x00 0x80 <80 bytes of data>
 - Verify Read response is the same data that was written
 - Increment start position
 - Verify length
 - Repeat steps a-f until entire image has been loaded

4. Decrypt the FW image.
Command: 0xCC 0xC5
Response: 0xCC 0xC5 0x00
5. Reset the radio to begin using the new FW image.
Command: 0xCC 0xFF
6. Repeat steps 2-5 for each binary file.

API OPERATION

API Operation is a powerful alternative to the default transparent operation of the RM024 and provides dynamic packet accounting abilities to the OEM host without requiring extensive programming by the OEM host. API operation utilizes specific packet formats. The API features can be used in any combination that suits the OEM's specific needs and can be different between radios operating on the same network.

API Send Data Complete

API Send Data Complete can be used as a software acknowledgement indicator. When a radio sends an addressed packet, it looks for a received acknowledgement (transparent to the OEM host). If an acknowledgement is not received, the packet is retransmitted until one is received or all retries have been exhausted.

For applications where data loss is not an option, the OEM host may wish to monitor the acknowledgement process using the API Send Data Complete. If an acknowledgement is not received (failure), the OEM host can resend the packet to the transceiver. Because the Send Data Complete is reliant upon a successful RF acknowledgement from the receiving radio, a false positive is not possible but a false negative may occur. That is, the receiving radio may have received the packet successfully, but the acknowledgement was not received at the transmitting radio. For broadcast packets, the Send Data Complete always reports success.

API Send Data Complete is enabled when bit-2 of the API Control (Address 0xC1) byte is enabled. The transceiver sends the OEM host the data shown in [Figure 2](#) upon receiving an RF acknowledge or exhausting all attempts.

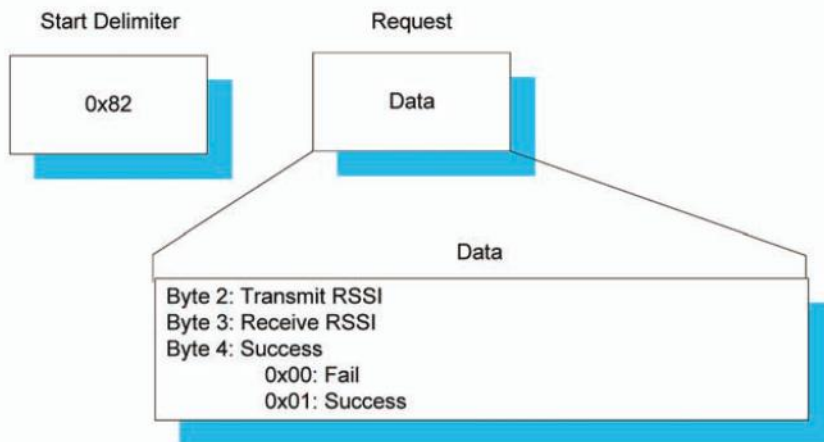


Figure 2: Send Data Complete Packet Format

API Receive Packet

By default, the source MAC is not included in the received data string sent to the OEM Host. For applications where multiple radios are sending data, it may be necessary to determine the origin of a specific data packet.

When API Receive Packet is enabled, all packets received by the transceiver include the MAC address of the source radio as well as an RSSI indicator which can be used to determine the link quality between the two.

API Receive Packet is enabled when bit-0 of the API Control (Address 0xC1) byte is enabled. Upon receiving a RF packet, the radio sends its OEM host the data as shown in [Figure 3](#).

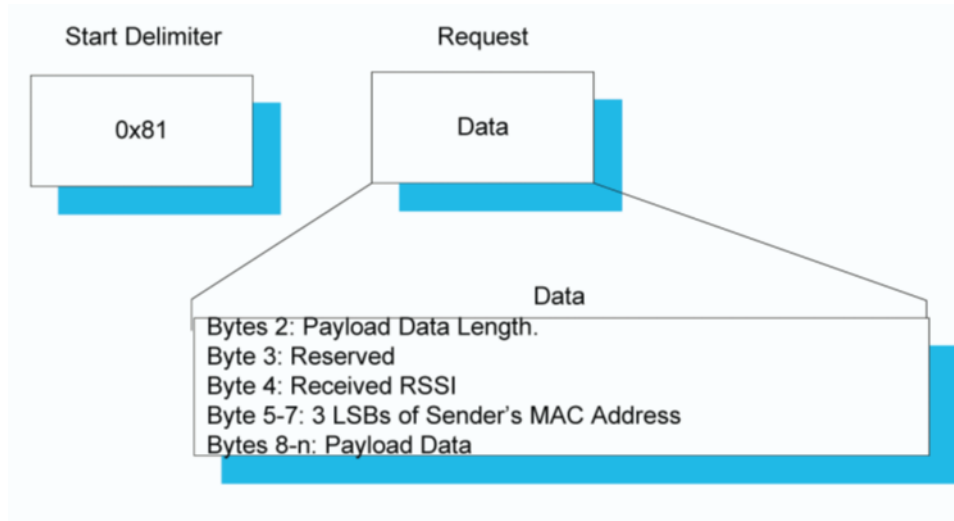


Figure 3: Receive API Packet Format

API Transmit Packet

API Transmit Packet is a powerful API mode that allows the OEM host to send data to a single or multiple (via Broadcast) transceivers on a packet-by-packet basis. This can be useful for many applications including polling and mesh networks.

API Transmit Packet is enabled when bit-1 of the API Control (address 0xC1) byte is enabled. The OEM should pre-pend each packet of data with the following header information ([Figure 4](#)).

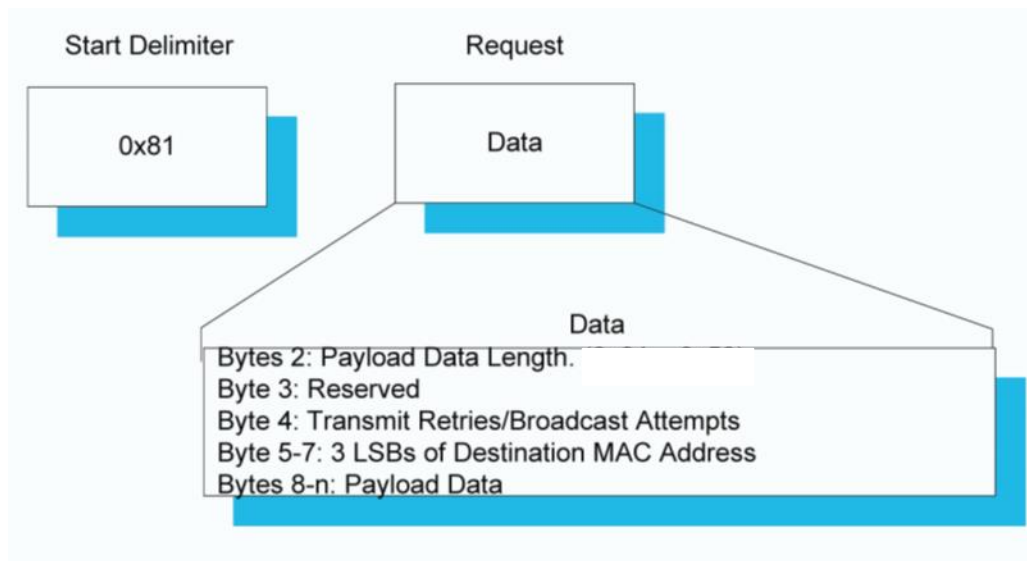


Figure 4: Transmit API Packet Format

Note: Payload Data Length is dependent on the RF Packet Size and RF Profile selected as per [Table 3](#) on page 8. Auto Config selects the maximum packet size for either FEC or non-FEC profiles.

Note: Setting the destination address of 0xFF 0xFF 0xFF broadcasts the packet to all available transceivers in the network.

RELATED DOCUMENTS AND FILES

The following additional RM024 technical documents are also available from the [RM024 product page](#) under the Documentation tab:

- Product Brief
- Hardware Integration Guide
- Application Note – Transitioning from LT2510 to RM024
- DVK – RM024 Hardware User Guide
- Firmware Release Notes (version1.3-0)

The following downloads are also available from the Software Downloads tab of the [RM024 product page](#):

- Configuration Utility
- USB Drivers
- Firmware Files (version 1.3-0)
- C Library Examples

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Laird Connectivity:](#)

[RM024-S10-C-30](#)