

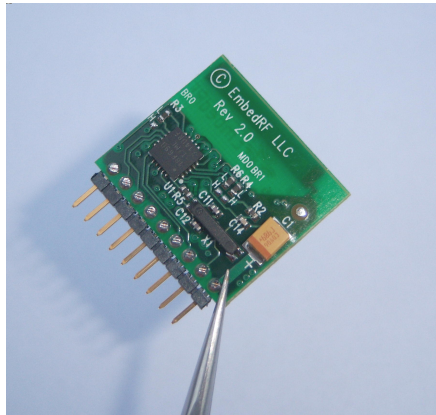


EmbedRF System Specification  
Manufacturer Part Number: EmbedRF

---

## General Description

EmbedRF LLC has developed a transceiver called EmbedRF. This module may be used within a number of applications requiring an off-the-shelf, low-power, turn-key wireless solution. EmbedRF modules can be easily configured using a standard serial interface to be a periodic transceiver for applications such as datalogging or an on-demand transceiver for applications such as key-fobs or remote control applications. This document provides an overview of the specifications and implementation of the EmbedRF.



---

## Features

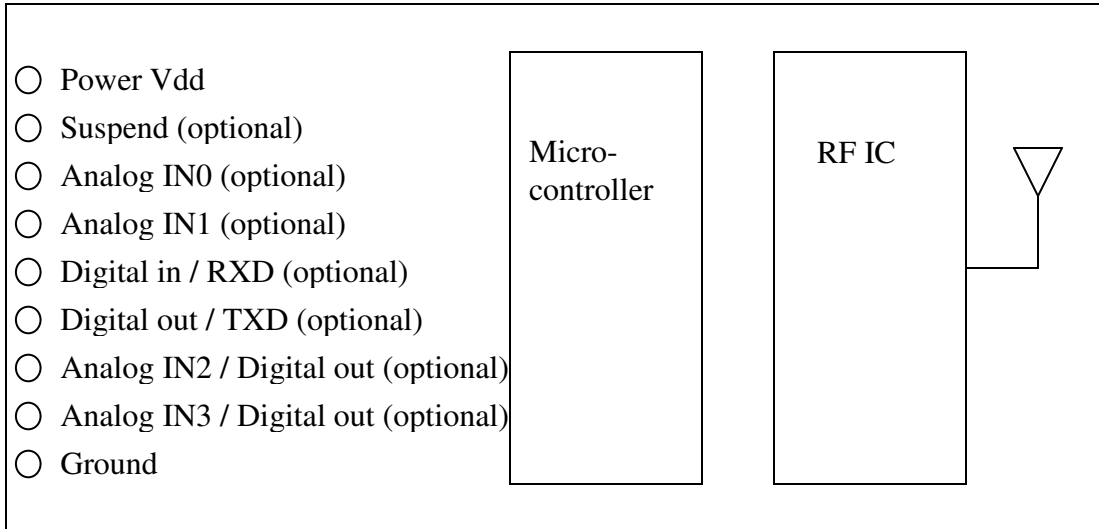
- Configurable periodic transceiver (0.25 to 12.75 second intervals)
- 4 Analog Channels for Sensor Applications
- Operated from 400Mhz - 928Mhz (typical use 433 / 868 / 915 Mhz)
- Range of 50 feet typical (longer with alternative antenna)
- Bidirectional transmit/receive
- Point-to-point and point-to-multipoint network configurable
- Intelligent search capability using nearest transmitter criteria
- Standard serial UART interface
- Notification on transmit / receive
- Receive and transmit data is buffered
- Small size
- Output power up to 5 dBm
- CRC RF packet error checking
- Digital RSSI signal
- Security using 2<sup>24</sup> bit ID and 2<sup>24</sup> bit network ID
- Zero data latency
- PC / USB interface available
- 2.0 to 3.6V operations
- -40 to 85 °C (no trimming)
- Flexible design licensing – seamlessly integrate our Gerbers into your PCB
- Design for lowest power battery operation.
- Simplest, lowest cost turnkey wireless solution

1	System Overview .....	4
1.1	Block Diagram and module mechanicals .....	4
1.2	Network implementation .....	5
1.2.1	Point to Point wireless connectivity.....	5
1.2.2	Point to Point wireless connectivity.....	5
1.2.3	Point to multi-point wireless connectivity .....	5
2	EmbedRF Standard Module Feature Description .....	6
2.1	Transmission Interval.....	6
2.2	Analog Inputs and Digital Outputs .....	6
2.3	Transmission Frequency .....	6
2.4	Transmission Range.....	6
2.5	Bi-directional Communication.....	6
2.6	Point-to-point and Multi-point-to-point Network Options .....	6
2.7	Intelligent Search using “Proximity Search” or “Find First Search”.....	7
2.8	Standard Serial UART Interface .....	7
2.9	Notification on Transmit/Receive.....	7
2.10	Receive and Transmit Data is Buffered .....	7
2.11	Output Power up to 5 dBm .....	7
2.12	RF data and CRC .....	7
2.13	Digital Receive Signal Strength Indicator (RSSI) .....	7
2.14	Node device ID .....	8
2.15	Network ID .....	8
2.16	Zero Data Latency.....	8
2.17	PC / USB Interface Available .....	8
2.18	Electrical .....	8
3	Uart control of EmbedRF.....	10
3.1	Uart Baud Rate Setting .....	10
3.2	Setting Wireless Data Packet using Uart .....	10
3.3	Analog / Digital Control Command.....	10
3.4	EmbedRF UART Commands .....	11
3.4.1	Command general format .....	11
3.4.2	Get Firmware Version Command.....	11
3.4.3	Get Transmission Interval Command.....	12
3.4.4	Set Transmission Interval Command.....	12
3.4.5	Set Transmit / Receive Mode.....	13
3.4.6	Get Transmit Mode .....	13
3.4.7	Get Data Received by EmbedRF .....	14
3.4.8	Send Data Immediately.....	15
3.4.9	Set Data to Transmit and Optionally Transmit Immediately.....	15
3.4.10	Set Transmit Power Level.....	16
3.4.11	Get Transmit Power Level.....	16
3.4.12	Set Device ID .....	17
3.4.13	Get Device ID .....	18
3.4.14	Set Analog / Digital Settings.....	18
3.4.15	Get Analog / Digital Settings.....	19

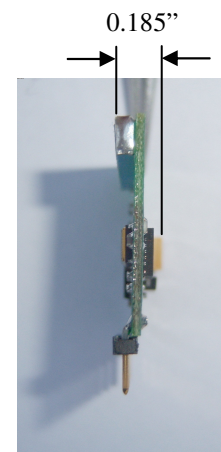
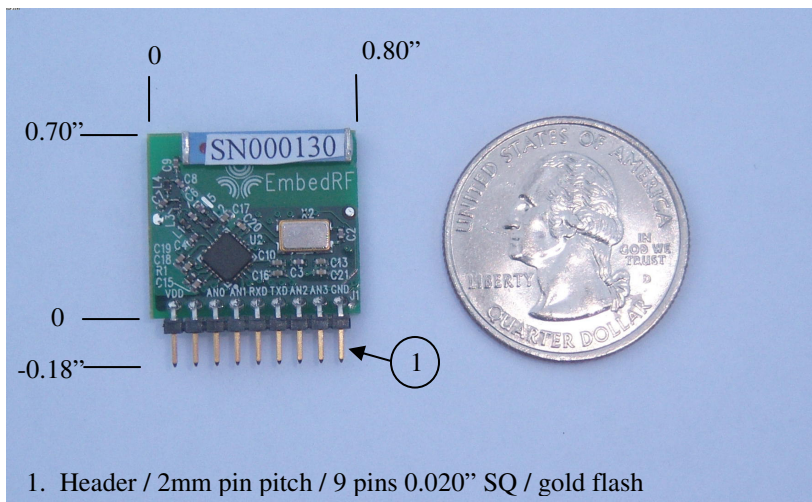
3.4.16	Find Transmitter.....	19
3.4.17	Set Network ID .....	20
3.4.18	Set Notify Mode.....	21
3.4.19	Set Device ID to Pseudo-Random Number .....	21
3.4.20	Set Network Mode .....	22
3.4.21	Get Network Mode .....	22
4	Factory Configurations .....	23

# 1 System Overview

## 1.1 Block Diagram and module mechanicals



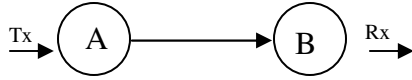
Dimensions (including antenna)	0.800" x 0.700"
Antenna	Chip antenna – on board
Board thickness	0.031"
Layers	4-copper layer
Standard design rules	6mil space / 6 mil trace; via size is 12 mils
Weight	1.4 grams



## 1.2 Network implementation

The EmbedRF is designed for the following network applications

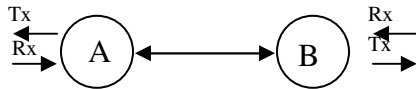
### 1.2.1 Point to Point wireless connectivity



Node A → stand alone analog input sensor or digital asynchronous data communication

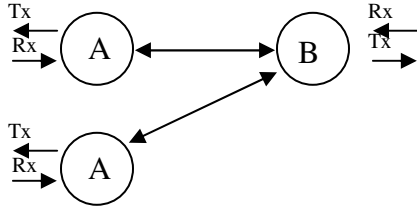
Node B → Digital asynchronous data receiver

### 1.2.2 Point to Point wireless connectivity



Nodes are re-configurable to be transmitter, receiver, or bidirectional transceiver.

### 1.2.3 Point to multi-point wireless connectivity



## **2 EmbedRF Standard Module Feature Description**

### ***2.1 Transmission Interval***

The EmbedRF is capable of transmitting data at intervals of 0.25 seconds up to 12.75 seconds in intervals of 50 milliseconds. This interval is configurable using a simple 2-wire serial UART interface.

### ***2.2 Analog Inputs and Digital Outputs***

The EmbedRF has 4 analog inputs. EmbedRF may be configured to periodically read and store A/D values on these channels and transmit them wirelessly. Two of the analog inputs may alternatively be configured as digital outputs for controlling external equipment that may be required for using the analog readings. See sections 3.4.14 and 3.4.15 for more information on configuring the analog and digital lines on the EmbedRF device.

### ***2.3 Transmission Frequency***

The EmbedRF is capable of transmitting at RF frequencies from 400Mhz to 928Mhz (component values and antenna specification need to match); initial release will operate within the ISM band (US); 902Mhz to 928Mhz. See Chipcon CC1100 datasheet for details.

### ***2.4 Transmission Range***

The EmbedRF is capable of transmitting data reliably in any physical orientation from 0 to an estimated 50 feet; using the small on-board antennas. The range can be extended to an estimated 100+ feet LOS using alternate antennas. Further information on setting the EmbedRF transmission range can be found under section 3.4.3 and 3.4.4.

### ***2.5 Bi-directional Communication***

An EmbedRF module may be configured to be either a transmitter, a receiver, or both a transmitter and receiver. More information on configuring bidirectional communication on the EmbedRF device can be found under section 3.4.5. See our application note entitled “Bidirectional wireless communication using EmbedRF”.

### ***2.6 Point-to-point and Multi-point-to-point Network Options***

EmbedRF can be configured to work within either a point-to-point or a multi-point to point network environment. When an EmbedRF module is configured for multi-point-to-point network functionality, it communicates with any other EmbedRF module that shares its network ID number. When an EmbedRF module is configured for point-to-point functionality, it only communicates with an EmbedRF module with the same network ID and a unique device ID. More information on configuring the network options for EmbedRF can be found under section 3.4.20 and 3.4.21. Also, please see our application note entitled “Setting up an Analog Sensor Network using EmbedRF”.

## **2.7 Intelligent Search using “Proximity Search” or “Find First Search”**

An EmbedRF module that is configured to be a receiver is capable of “locking-in” to the nearest EmbedRF transmitter within its reception vicinity (“Proximity Search”) or to the first EmbedRF transmitter from which it receives a packet (“Find First Search”). The “Nearest Transmitter Criteria” is possible because of the “receive signal strength indicator” (RSSI) within EmbedRF. The RSSI is larger for EmbedRF transmitters that are closer in proximity to the EmbedRF receiver. For information on how to execute a proximity search, please see our application note entitled “Proximity search using EmbedRF” as well as section 3.4.16.

## **2.8 Standard Serial UART Interface**

The EmbedRF device may be controlled by a host microcontroller using a simple 2-wire serial UART interface. The commands used to control and configure the EmbedRF device are specified in section 3.4 of this datasheet. Alternatively, the EmbedRF device may be configured in conjunction with the EmbedRF USB interface board which connects to a desktop PC and may be used with the EmbedRF DesktopPro software. Please see the document “Getting Started with EmbedRF DesktopPro” application note for more information.

## **2.9 Notification on Transmit/Receive**

The EmbedRF can be configured to notify a host microcontroller whenever it has transmitted or received data. The notification is done through the UART interface and is described under section 3.4.18.

## **2.10 Receive and Transmit Data is Buffered**

The 10-byte transmit and receive data packet is buffered within the Embed device. This enables host microcontrollers to asynchronously poll for data received and also to asynchronously update a data packet that will be transmitted at the next period. Data received by an EmbedRF device is maintained within the Receive buffer indefinitely until the next data packet arrives or power is removed from the device.

## **2.11 Output Power up to 5 dBm**

The EmbedRF device has 4 power settings that may be configured using the standard serial UART interface. The 4 power levels are 5 dBm, 0 dBm, -5 dBm, and -10 dBm. The specification for controlling the power is described in section 3.4.10 and 3.4.11.

## **2.12 RF data and CRC**

The EmbedRF Board transmits at 100 KBaud. The on-air (RF) bit rate is hard-coded in the firmware and fixed by the proprietary RF protocol. The RF packet integrity is guaranteed via CRC.

## **2.13 Digital Receive Signal Strength Indicator (RSSI)**

The EmbedRF modules provide a Digital RSSI that indicates relative distances of a transmitter from a receiver. This RSSI may be used by a host microcontroller connected to an EmbedRF receiver to determine the relative distance of the EmbedRF receiver to

different EmbedRF transmitters. The RSSI may be requested from an EmbedRF receiver using the “Get Data Received by EmbedRF” command described under section 3.4.7.

### **2.14 Node device ID**

All EmbedRF transmitters have a  $2^{24}$  bit transmit ID and a  $2^{24}$  bit receive ID. These IDs can be assigned by either a host microcontroller or using the USB Interface Board and the EmbedRF DesktopPro software. The transmit ID uniquely identifies an EmbedRF board configured to transmit data. The receive ID specifies which ID an EmbedRF board is specifically listening to receive data from. Alternatively, an EmbedRF board configured as a receiver can search for any transmitter EmbedRF board that has the same network ID. Two methods are available for searching for an EmbedRF transmitter as described above in section 2.7. The method for configuring the device ID is described in section 3.4.12 and 3.4.13

### **2.15 Network ID**

All EmbedRF systems have a 24 bit network ID. This ID is write-only for security purposes. This ID allows identical hardware systems to be used by various manufacturers without cross talk interference. This ID gives each manufacture a unique PAN (personal area network) with security since this ID can not be altered. One “public” network is available to all; ID = 0xFFFFFFFF. Please see section 3.4.17 to learn how to set the Network ID on an EmbedRF device.

### **2.16 Zero Data Latency**

The EmbedRF modules may be used in applications whereby a host microcontroller can periodically update the data within an EmbedRF transmitter so that this new data may be transmitted to an EmbedRF receiver. The EmbedRF transmitter may be updated with the new data immediately prior to a data transmission such that there is zero data latency between when the transmitter was updated with the new data and when the new data was transmitted.

### **2.17 PC / USB Interface Available**

EmbedRF may be configured and controlled by the EmbedRF DesktopPro software that may be downloaded free from the EmbedRF.com web site. This software has many features including implementation of all serial UART commands described in section 3.4 as well as rudimentary datalogging capability.

### **2.18 Electrical**

Sym	Notes:	Characteristic	Min	Typical	max
Vbat		Supply Voltage	2.0V		3.6V
BAUD	Host	Asynchronous Digital comm.	1.2K		19.2K
Tacq		Acquisition time (see note 1)	0	4x	6x
Idd		Supply Current		See table2	
		Operating temperature	-40C		85C



		Data latency (see note 3)		5.6mS	
--	--	---------------------------	--	-------	--

**Table 1: Electrical characteristics of the EmbedRF board.**

Sym	Notes:	Characteristic	Min	Typical	max
Idd		Transmitter (see note 2) at -10dBm $I_{dd} (uA) = (X + 52.7) / X$	5uA		212uA
		Transmitter (see note 2) at -5dBm $I_{dd} (uA) = (X + 55.7) / X$	5.5uA		224uA
		Transmitter (see note 2) at 0dBm $I_{dd} (uA) = (X + 63.5) / X$	6uA		255uA
		Transmitter (see note 2) at +5dBm $I_{dd} (uA) = (X + 71.3) / X$	6.6uA		286uA
Idd		Receiver (see note 2) $I_{dd} (uA) = (X + 105) / X$	9.3uA		421uA
Idd		Sleep		0.3uA	2.0uA
Iacq		During acquisition (see note 1)		2.1mA	
Idd		Tx/Rx and Rx/Tx (see note 4)		Rx + Tx	

**Table 2: operating current of the EmbedRF board at 915MHz**

**Notes:**

1. Typical time for the Receiver to acquire a transmitter is 4 times the periodicity at a given configuration. The acquisition time is required once per power cycle or telemetry range.
2. The average operating current of the module is dependant on the periodicity setting. The minimum and maximum values shown are for 12.75s and 0.25S respectively. To calculate the typical average current at any periodicity follow the equation; where  $x$  = periodicity time in seconds.
3. Data latency is measured from the end of the last byte transmitted to the beginning of the first byte received.
4. The typical average current in Tx/Rx and Rx/Tx is the sum of the individual modes listed in table 2. There are some cases where minor increased Rx times are expected.

### 3 Uart control of EmbedRF

The EmbedRF is capable of communicating with any host microcontroller via the standard serial UART. Messages are sent using single or multiple command bytes. The commands are summarized in section 3.4.

#### 3.1 Uart Baud Rate Setting

Baud rates are selected based on the table below:

Jumper Settings [BR1 / BR0]	Baud Rate
0 / 0	19200
0 / 1	9600
1 / 0	2400
1 / 1	1200

Table 2: Baud rate jumper settings for EmbedRF board.

#### 3.2 Setting Wireless Data Packet using Uart

The EmbedRF wirelessly transmits and receives 17 bytes of data. The first 6 bytes are not accessible except to the EmbedRF operating system as they consist of the network ID and the device transmit ID. Ten of these bytes may be explicitly set using the UART interface. The EmbedRF may also be configured (0x41 command) to record A/D readings from the transceiver and place these values into the data packet for wireless transmission. The data bytes that are accessible to the user are shown in the table below.

Byte Number	Description
0	A/D 0 MSByte or Custom
1	A/D 0 LSByte or Custom
2	A/D 1 MSByte or Custom
3	A/D 1 LSByte or Custom
4	A/D 2 MSByte or Custom
5	A/D 2 LSByte or Custom
6	A/D 3 MSByte or Custom
7	A/D 3 LSByte or Custom
8	Custom
9	Custom
10	Counter byte

Table 3: Wireless data packet configurable by the user.

The counter byte is incremented each time a new RF packet is transmitted by an EmbedRF configured in transmit mode.

#### 3.3 Analog / Digital Control Command

There are 4 A/D inputs that may be utilized by the EmbedRF. Each of the 4 A/D channels may be set to “Active” mode. In active mode, the A/D readings are inserted

into the RF data packet at the positions indicated by Table 3. If an A/D input is selected as active, then custom data may not be inserted into the wireless data packet at the byte positions for the active A/D channel. For example, if A/D 0 is set to be active, then bytes 0 and 1 in the wireless data packet may not be set to have custom values. Conversely, if an A/D is specified as inactive (ie. A/D readings are not stored for that channel), then the bytes associated with that A/D may be set to custom values. Two of the A/D channels (A/D channels 2 and 3) may be configured as digital outputs instead of analog inputs. They may also be set either high or low once they have been set to a digital output. See section 3.4.14 and 3.4.15 for an example of how to configure the analog and digital settings of the EmbedRF boards.

### **3.4 EmbedRF UART Commands**

The EmbedRF board may be controlled using a standard UART interface. The complete list of commands that may be used to read and write the settings for an EmbedRF board are given below. Alternatively, you can use the EmbedRF DesktopPro software to view the commands and the replies from the EmbedRF modules. The commands and replies are provided in the lower part of the main window of the EmbedRF DesktopPro window.

#### **3.4.1 Command general format**

All commands one byte in length. Several commands have additional *command data bytes* following the first byte of the command. All replies are one byte. Several replies have additional *reply data bytes* following the first reply byte. The command and reply data bytes are given in the format XX YY etc. where the XX and YY represent different data bytes.

#### **3.4.2 Get Firmware Version Command**

This command requests the EmbedRF version number.

##### Command

0x56

##### Reply

0x76 XX YY

XX = major version #

YY = minor version #

##### Example

The following command requests the version. The reply is version 1.0

Command: 0x56

Reply: 0x76 0x01 0x00

### 3.4.3 Get Transmission Interval Command

This command requests the interval that an EmbedRF module is either transmitting data or expecting to receive data. The transmission interval is sent as a reply data byte where each count represents 0.05 seconds. The transmission range can be set from 0.25 seconds (reply data byte=5) to 12.75 seconds (reply data byte = 255) in 0.05 second increments.

#### Command

0x48

#### Reply

0x68 XX

XX = data byte representing transmission interval. Acceptable value is between 5 and 255.

#### Example

Command to request transmission interval. Reply is that the device is in 0.25 second transmission interval mode.

Command: 0x48

Reply: 0x68 0x05

### 3.4.4 Set Transmission Interval Command

This command sets the interval that an EmbedRF module transmits or is expecting to receive data. The transmission interval is sent as a command data byte where each count represents 0.05 seconds. The transmission interval can be set from 0.25 seconds (command data byte=5) to 12.75 seconds (reply data byte = 255) in 0.05 second increments.

#### Command

0x49 XX

#### Reply

0x69 XX

XX = data byte representing transmission interval. Acceptable value is between 10 and 255.

#### Example

Command to set transmission interval to 0.25 seconds. Reply is that the device was set to 0.25 second transmission interval mode.

Command: 0x49 0x05

Reply: 0x69 0x05

### 3.4.5 Set Transmit / Receive Mode

This command configures the EmbedRF to be a transmitter, a receiver, or a bidirectional transceiver. For bidirectional communication, the EmbedRF device must be set to either “transmit then receive”, or to “receive and then transmit”.

#### Command

0x54 XX

XX = configuration byte

0 = OFF – NO TRANSMIT or RECEIVE

1 = Receive only

2 = Transmit only

3 = Receive then transmit

4 = Transmit then receive

#### Reply

0x74 XX

XX = configuration byte as described above.

#### Example

This example puts the device into transmit then receive mode.

Command: 0x54 0x04

Reply: 0x74 0x04

### 3.4.6 Get Transmit Mode

This command gets the mode that the EmbedRF is currently set to. An EmbedRF can be configured to be a transmitter, a receiver, or a bidirectional transceiver. For bidirectional communication, the EmbedRF device must be set to either “transmit then receive”, or to “receive and then transmit”.

#### Command

0x58

#### Reply

0x78 XX

XX = configuration byte

0 = OFF – NO TRANSMIT or RECEIVE

1 = Receive only

2 = Transmit only

3 = Receive then transmit

4 = Transmit then receive

### Example

This example gets the transmit mode. The reply is that the EmbedRF is in “receive then transmit mode”.

Command: 0x58

Reply: 0x78 0x03

### **3.4.7 Get Data Received by EmbedRF**

This command requests an EmbedRF device configured as a receiver to send the data most recently received from another EmbedRF configured as a transmitter. The first 3 bytes of this data packet are the ID of the EmbedRF transmitter that sent the data. Also included in the data packet is the receive signal strength indicator (RSSI) of the received signal. The RSSI is a number between 0 and 255 that indicates how close the transmitter is to the receiver. A larger RSSI indicates that the transmitter is closer in proximity to the receiver. The last data byte in the data packet is an RF Packet Count. The RF Packet Count is incremented by the EmbedRF transmitter every time a packet is transmitted and it is transmitted to the EmbedRF receiver. The RF Packet Count can be used to monitor whether a packet has been missed by an EmbedRF receiver.

### Command

0x47

### Reply

0x67 LL MM NN OO PP QQ RR SS TT UU VV WW XX YY ZZ

LL: Most significant byte of the device transmit ID that sent this data

MM: Middle byte of the device transmit ID that sent the data

NN: Least significant byte of the device transmit ID that sent the data

OO: Byte 0

PP: Byte 1

QQ: Byte 2

RR: Byte 3

SS: Byte 4

TT: Byte 5

UU: Byte 6

VV: Byte 7

WW: Byte 8

XX: Byte 9

YY: RSSI

ZZ: Packet counter

### Example

This example gets the last packet received by an EmbedRF. The bytes received are the numbers 1-10. The data was received from a transmit device ID of 0x0000001. The RSSI is 225 and the Packet count is 12.

Command: 0x47

Reply: 0x67 0x00 0x00 0x01 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x0a 0xE1  
0x0c

### 3.4.8 Send Data Immediately

This command forces the EmbedRF module to wirelessly transmit whatever data is in its transmit buffer.

#### Command

0x43 XX

XX: data byte indicating whether the EmbedRF should attempt to receive a data packet immediately after transmitting the data packet (XX=0x01) or just transmit (XX=0x00)

#### Reply

0x63

#### Example

This example forces the EmbedRF to transmit data immediately and then receive a data packet. The reply notifies that data is sent.

Command: 0x43 0x01

Reply: 0x63

### 3.4.9 Set Data to Transmit and Optionally Transmit Immediately

This command sets the data buffer which the EmbedRF shall transmit on the next regularly scheduled transmission interval. The last byte in the message can be used to force the EmbedRF module to transmit its data packet immediately. If the last byte is 0, then the module will not transmit immediately after receipt of the command. If the last byte is 1, then the module will transmit immediately.

#### Command

0x53 LL MM NN OO PP QQ RR SS TT UU VV

LL: Byte 0

MM: Byte 1

NN: Byte 2

OO: Byte 3

PP: Byte 4

QQ: Byte 5

RR: Byte 6

SS: Byte 7

TT: Byte 8

UU: Byte 9

VV: Data byte indicating what to do after setting the data packet. The EmbedRF may either do nothing (UU=0x00), transmit immediately after setting the data bytes (UU=0x01), or transmit and then receive a data packet (UU=0x02)

Reply

0x73

Example

This example sets the data in the transmit buffer to the bytes values 1 to 10 and forces the EmbedRF to transmit and then receive a data packet immediately. The reply is an acknowledgement.

Command: 0x53 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 x0a 0x02

Reply: 0x73

### **3.4.10 Set Transmit Power Level**

This command sets the EmbedRF power level to one of 4 power levels.

Command

0x50 XX

XX: Data byte representing power level

0x00 = -10 dbm

0x01 = -5 dbm

0x02 = 0 dbm

0x03 = 5 dbm

Reply

0x70 XX

Example

This example sets the EmbedRF power level to -5 dbm. The reply acknowledges the setting.

Command: 0x50 0x01

Reply: 0x70 0x01

### **3.4.11 Get Transmit Power Level**

This command gets the EmbedRF power level.

Command

0x4f

Reply



0x6f XX

XX: Data byte representing power level

0x00 = -10 dbm

0x01 = -5 dbm

0x02 = 0 dbm

0x03 = 5 dbm

#### Example

This example gets the EmbedRF power level. The reply states that the power level is set to 0 dbm.

Command: 0x4f

Reply: 0x6f 0x02

### **3.4.12 Set Device ID**

This command sets either the transmit or receive ID for the EmbedRF. When in point-to-point network mode, the EmbedRF will only communicate with the programmed transmit / receive ID, respectively. Data bytes are sent most-significant byte first. The device ID can range from 0 to 16,777,215. Immediately following the 3-byte ID should be a 1-byte specifier which indicates whether you are setting the transmit ID (0x01) or the receive ID (0x00).

#### Command

0x55 WW XX YY ZZ

WW = MSByte of device ID

XX = Middle byte of device ID

YY = LSByte of device ID

ZZ = specifier indicating whether transmit (ZZ=0x01) or receive (ZZ = 0x00) is being set

#### Reply

0x75 WW XX YY

WW = MSByte of device ID

XX = Middle byte of device ID

YY = LSByte of device ID

#### Example

This example sets the transmit ID of the EmbedRF to the number 66051.

Command: 0x55 0x01 0x02 0x03 0x01

Reply: 0x75 0x01 0x02 0x03

### 3.4.13 Get Device ID

This command gets the 3-byte transmit or receive ID for the EmbedRF depending on the data byte sent following the command byte. Data bytes are sent MSByte first followed by the specifier byte. The specifier byte indicates that the transmit ID (0x01) or the receive ID (0x00) is being requested.

#### Command

0x57 XX

XX = specifier indicating whether transmit (XX=0x01) or receive (XX = 0x00) is being requested

#### Reply

0x77 AA BB CC

AA = MSByte of device ID

BB = Middle byte of device ID

CC = LSByte of device ID

#### Example

This example gets the transmit ID of the EmbedRF. The reply is that the transmit ID is the number 66051.

Command: 0x57 0x01

Reply: 0x77 0x01 0x02 0x03

### 3.4.14 Set Analog / Digital Settings

This command configures how the EmbedRF acquires and stores analog data how the EmbedRF digital outputs are configured. There are 4 analog / digital channels on the EmbedRF - A/D 0, A/D 1, A/D 2, and A/D 3. A/D 0 and A/D 1 can be configured to be analog inputs only. A/D 2 and A/D 3 can be configured to be either analog inputs or digital outputs. If A/D 2 or A/D 3 is set to be a digital output, these channels can be set to either logic 1 or logic 0. One data byte within the command sets whether an A/D channel is disabled, analog, digital logic 1, or digital logic 0.

Configuration byte

0x00 : Disabled

0x01 : Analog active

0x02 : Digital active logic 1

0x03 : Digital active logic 0

#### Command

0x41 WW XX YY ZZ

WW: Configuration byte as defined above for A/D 0

XX: Configuration byte as defined above for A/D 1

YY: Configuration byte as defined above for A/D 2

ZZ: Configuration byte as defined above for A/D 3

Reply

0x61 WW XX YY ZZ

Example

This example sets A/D 0 to analog input, A/D 1 to disabled, A/2 to digital output logic 1 and A/D 3 to digital output logic 0.

Command 0x41 0x01 0x00 0x003 0x02

Reply: 0x61 0x01 0x00 0x03 0x02

### 3.4.15 Get Analog / Digital Settings

This command retrieves the EmbedRF analog settings. The analog settings are as defined in the previous section.

Command

0x42

Reply

0x62 WW XX YY ZZ

WW: Configuration byte as defined above for A/D 0

XX: Configuration byte as defined above for A/D 1

YY: Configuration byte as defined above for A/D 2

ZZ: Configuration byte as defined above for A/D 3

Example

This example gets the A/D settings. The reply states A/D 0 is set to analog input, A/D 1 to disabled, A/3 to digital output logic 1 and A/D 3 to digital output logic 0.

Command 0x42

Reply: 0x62 0x01 0x00 0x03 0x02

### 3.4.16 Find Transmitter

This command clears the EmbedRF receive ID and puts the EmbedRF into find mode during which it searches for an EmbedRF transmitter that has an equivalent network ID.

There are two search modes that can be set using the data byte which follows the command – Find First Transmitter mode and Proximity Find mode.

Find First Transmitter mode enables an EmbedRF configured as a receiver to find the first transmitter that matches the network ID (data byte=0x00). All subsequent transmitters are ignored. The Proximity Find mode enables the EmbedRF receiver to wait 10 times the current transmit interval and during this time, it listens to all EmbedRF transmitters with an identical network ID. The EmbedRF transmitter that is selected at

the end of the search period is the EmbedRF board which is closest in proximity to the EmbedRF receiver board based on the largest RSSI. All other EmbedRF transmitters are ignored by the EmbedRF receiver.

Once the EmbedRF board finds a valid transmitter using either Find First or Proximity Find, it responds with the command byte 0x66 followed by the 3-byte ID it is now set to receive.

#### Command

0x46 XX

XX: Data byte indicates whether a Proximity Find (XX=0x01) or a Find First (XX=0x00) should be done

#### Reply

0x66 XX YY ZZ

XX: Most-significant byte of ID detected

YY: Middle byte of ID detected

ZZ: Least-significant byte of ID detected

#### Example

This example puts the EmbedRF into a proximity find. The reply is that ID 0x000001 was found, which means ID 0x000001 was either the only transmitter in the vicinity or else it was the closest in proximity to the EmbedRF receiver.

Command 0x46 0x01

Reply: 0x66 0x00 0x00 0x01

### **3.4.17 Set Network ID**

This command sets the 3-byte network ID for the EmbedRF. For two EmbedRF devices to communicate, they must share the same network ID. Network IDs are write-only. For security purposes, they cannot be read. Network ID 0x000000 is considered an open network available for all EmbedRF users. Data bytes are sent MSByte first.

#### Command

0x4d XX YY ZZ

XX: Most-significant byte of network ID

YY: Middle byte of network ID

ZZ: Least-significant byte of network ID

#### Reply

0x6d XX YY ZZ

XX: Most-significant byte of network ID

YY: Middle byte of network ID

ZZ: Least-significant byte of network ID

### Example

This example sets the network ID to 66051.

Command 0x4d 0x01 0x02 0x03

Reply: 0x6d 0x01 0x02 0x03

### **3.4.18 Set Notify Mode**

This command is used to set notification mode on or off. If notification mode is on, then an EmbedRF will a notification message on the UART line when a data transmission or reception has occurred. This mode is useful if a host microcontroller needs to know immediately when a new data packet has arrived at or been transmitted by an EmbedRF device. If an EmbedRF is configured to be a transmitter, the byte 0x73 is sent on the UART immediately after a data packet is sent. If an EmbedRF is configured to be a receiver, the reply described in section 3.4.7 is sent on the UART.

#### Command

0x4e XX

XX: data byte indicating whether notification is on (0x01) or off (0x00)

#### Reply

0x6e XX

XX: data byte indicating whether notification is on (0x01) or off (0x00)

### Example

This example sets notification ON.

Command 0x4e 0x01

Reply: 0x6e 0x01

### **3.4.19 Set Device ID to Pseudo-Random Number**

This command may be used to set the device transmit ID or the device receive ID to a random number generated by the EmbedRF device. The internal clock on the EmbedRF PCB is used to calculate the random number, thereby making it pseudo random depending on when a user sends the command.

#### Command

0x44 XX

XX: data byte indicating whether the device receive ID (0x00) or the transmit ID (0x01) is being set to a random number.

#### Reply

0x64 XX YY ZZ

XX: Most-significant byte of new random device ID

YY: Middle byte of new random device ID

ZZ: Least-significant byte of new random device ID

#### Example

This example sets the device transmit ID to a random number. The random number was assigned by the EmbedRF board to be 66051

Command 0x44 0x01

Reply: 0x64 0x01 0x02 0x03

### **3.4.20 Set Network Mode**

The EmbedRF device can be set to either point-to-point or point-to-multipoint mode.

In point-to-point mode, the EmbedRF device will only communicate with one other EmbedRF device. The two devices that are communicating must have identical network IDs. The EmbedRF device configured to receive data, must have the same receive device ID as the EmbedRF device that is configured to transmit data.

Typically, only EmbedRF devices configured to be receivers should be set to be in point-to-multipoint mode. If an EmbedRF device is configured to be a receiver and it is in point-to-multipoint mode, then it will receive data from any EmbedRF device that has an identical network ID.

This command sets the network mode for the EmbedRF device to either point-to-point or point-to-multipoint.

#### Command

0x4B XX

XX: data byte indicating whether the network mode is point-to-point (0x00) or point-to-multipoint (0x01).

#### Reply

0x6B XX

XX: data byte indicating whether the network mode is point-to-point (0x00) or point-to-multipoint (0x01).

#### Example

This example sets the device to point-to-multipoint mode.

Command 0x4B 0x01

Reply: 0x6B 0x01

### **3.4.21 Get Network Mode**

This command gets the current network mode setting of the EmbedRF device.

#### Command

0x4A

Reply

0x6A XX

XX: data byte indicating whether the network mode is point-to-point (0x00) or point-to-multipoint (0x01).

Example

This example gets the device to point-to-multipoint mode which is currently set to point-to-point.

Command 0x4A

Reply: 0x6A 0x00

## 4 Factory Configurations

The table below lists the configurations which are pre-programmed into the EmbedRF's non-volatile memory. These values will be programmed along with the code; however several of these variables may be changed using the EmbedRF Desktop Pro software tool included with the EmbedRF Development Kit. The variables which may be written are indicated in the table below. Additionally these variables could be read or written using a host microcontroller.

Variable Name	Description	Factory Setting	Read / Write
Frequency of RF Transmission	Frequency at which the data is transmitted	915 MHz	No / No
Data rate	Wireless data transmission rate	100 KBaud	No / No
Transmit power	Power setting, between -10 and +5 dbm	0 dbm	Yes / Yes
Data field 10-bytes	10-bytes	All zeros	Yes / Yes
Network ID	3-byte network identifier.	16777215	No / Yes
Transmit ID	3-byte ID that will be transmitted by EmbedRF	16777215	Yes / Yes
Receive ID	3-byte ID that specifies which transmit ID an EmbedRF board will receive	16777215	Yes / Yes
Analog & Digital default settings	Indicates whether analog data will be acquired by	Analog and digital disabled.	Yes / Yes

	EmbedRF and whether digital data will be used.		
Receive / Transmit Mode	Setting indicates if receive / transmit modes are enabled or disabled	Both transmit and receive disabled	Yes / Yes
Transmit Interval	Timing interval over which wireless data is transmitted. May range from 0.25 to 12.75 seconds in 0.05 second intervals.	12.75 seconds	Yes / Yes
Network Mode	Network mode can be either point-to-point or point-to-multi-point	Point-to-point	Yes / Yes

**Table 4: Factory settings of EmbedRF parameters.**

\*\* This can not be over written from the host controller

Development kit will have the “public” network available; ID = 0xfffff

This Manufacturer identifier prevents misuse of a receiver from a third-party.