EmbedRF DesktopPro Software Manual

1. Introduction

This document provides a complete overview of the EmbedRF DesktopPro software. Use this document as a reference as you learn to fully configure and control your EmbedRF Boards using EmbedRF DesktopPro.

2. Hardware Set-up and Software Installation

Please see the "Getting Started with EmbedRF Development Kit" document for a full set of instructions on setting up the hardware properly and installing the EmbedRF Desktop Pro Software.

3. Software Overview

The EmbedRF DesktopPro software is used to configure and control EmbedRF Boards through a USB interface. You may use this software to do the following things which will be further described in this document:

- Configure all of the settings of an EmbedRF Board
- Set which data an EmbedRF Board will transmit
- View data an EmbedRF Board has received

These three primary functionalities may be accessed through three tabs (Configure, Transmit, and Receive) that may be clicked on at the top of the EmbedRF DesktopPro software window.

Each of these main three categories will be further discussed in this document.

4. Configuring an EmbedRF Board

A number of settings on the EmbedRF Board may be set using the Configure tab including:

- Configure as either a receiver, transmitter, or a bidirectional transceiver
- Set a transmit ID, a receive ID, and a network ID
- Set the interval at which data is transmitted and received
- Configure to store and transmit analog sensor data
- Configured to set digital output values
- Configured to notify a host microcontroller or the EmbedRF DesktopPro software when it has received a valid data packet.
- Find a transmitter
- Configure network mode as point-to-point or multi-point to point
- Determine the firmware version of the EmbedRF Board

This section of the document describes how to configure all of these settings on an EmbedRF Board.

Before beginning to configure your EmbedRF Board, you should ensure that you are in the Configure tab of the EmbedRF DesktopPro software window and the window should look like Figure 1.

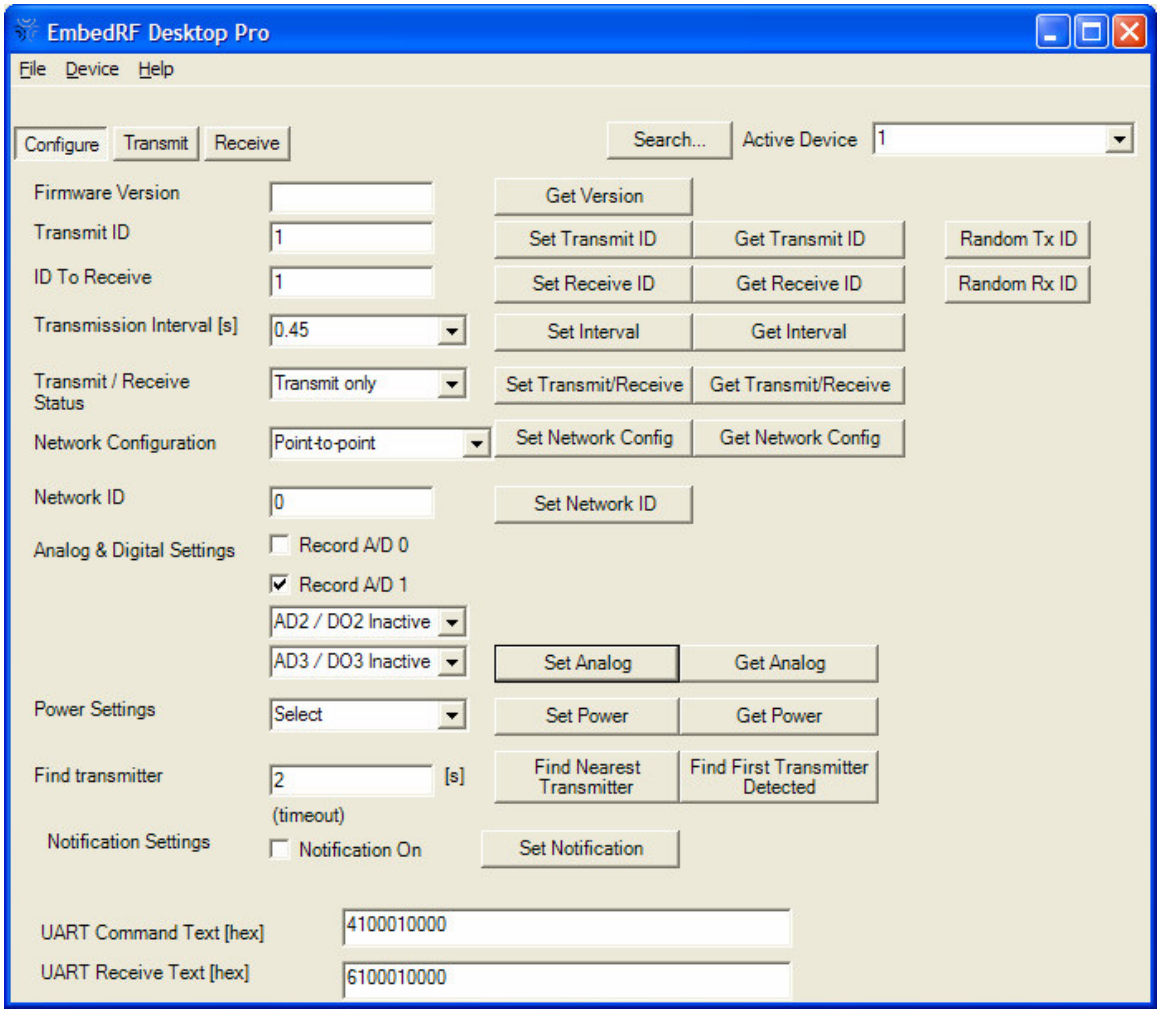

Figure 1: The Configure tab window for the EmbedRF DesktopPro software. This window may be used to configure an EmbedRF board's transmission ID, receive ID, network ID, transmission interval, analog data settings, and notification settings. Notice that the two boxes at the very bottom of the window provide the commands in hexadecimal notation. These can be used as a reference when writing software for a host microcontroller to verify what commands should be in raw hex format.

4.1. Raw Hexadecimal UART Commands and Replies

At the very bottom of the main EmbedRF DesktopPro window are two boxes that show what UART commands are being sent to an EmbedRF module and what UART replies are being sent back from the EmbedRF module. These boxes are useful in that they show the raw hexadecimal commands that are being sent serially between the PC and the EmbedRF modules. These raw commands are all listed and fully specified within the EmbedRF Datasheet. However, some people find it more convenient to use the PC

software to reference a raw hex command when writing software for a host microcontroller that will be interfacing with an EmbedRF module.

4.2. Configure as Receiver, Transmitter, or bidirectional Transceiver

You may set an EmbedRF Board as either a transmitter, a receiver, or a bidirectional transceiver.

4.2.1. Configure to Receive Data

Select the option to "Receive Only" from the Receive / Transmit Status pull-down combo-box. Then press the "Set Receive / Transmit" button.

4.2.2. Configure to Transmit Data

Select the option to "Transmit Only" from the Receive / Transmit Status pull-down combo-box. Then press the "Set Receive / Transmit" button.

4.2.3. Configure to a Bidirectional Transceiver

When a board is set to be a bidirectional transceiver, you may set it to either transmit and then receiver, or alternatively receive and then transmit. It is important that when configuring two modules such that both will be bidirectional transceivers, then one should be configured to "transmit then receive" while the other should be configured to "receive then transmit". Otherwise, the two devices will not be able to communicate properly. Please see the application "Bidirectional wireless communication using EmbedRF" for more information about executing bidirectional communication.

To configure a device to be a transceiver, select either "Receive then transmit" or "Transmit then Receive" option from the Receive/Transmit Status pull-down combobox. Then press the "Set Receive / Transmit" button.

4.3. Setting Receive, Transmit, and Network IDs

Each EmbedRF Board has a receive ID, a transmit ID, and a network ID. These IDs are used to provide control over which devices send/transmit to other devices. An EmbedRF Board that is configured to transmit (a transmitter), should be given a transmit ID and a network ID. An EmbedRF Board that is configured to receive data (a receiver), should be given a receive ID and a network ID. The receiver will only receive data from a transmitter if the transmitter's transmit ID matches the receiver's receive ID and the transmitter's network ID matches the receiver's network ID.

4.3.1. Setting the Transmit ID

Enter a number between 0 and 16777215in the Transmit ID text box, then press the "Set Transmit ID" button. To verify that you have properly set the ID, you may press the "Get Transmit ID" button and the transmit ID that has been previously set on the EmbedRF Board should appear in the "Transmit ID" text box.

4.3.2. Setting the Receive ID

Enter a number between 0 and 16777215in the "ID to Receive" text box, then press the "Set Receive ID" button. To verify that you have properly set the ID, you may press the "Get Receive ID" button and the receive ID that has been previously set on the EmbedRF Board should appear in the "ID to Receive" text box.

Setting Transmit and Receive ID to Random Number
You may wish to set the transmit and receive IDs to random numbers. You may do this by pressing the "Random Tx ID" and "Random Rx ID" buttons to respectively set the transmit ID or receive ID to a random number between 0 and 65535.

4.3.3. Setting the Network ID

Enter a number between 0 and 16777215, then press the "Set Nework ID". For security purposes, you are not able to prompt an EmbedRF Board for its network ID. You are only capable of setting it.

4.4. Set Interval at which Data is Transmitted and Received

The EmbedRF Boards may be configured to transmit data at an interval from 0.25 seconds up to 12.75 seconds in 0.05 second intervals. For two boards to effectively transmit/receive data, they must have the same transmission interval.

To set the transmission interval, select a number from the Transmission Interval pull-down menu. Then press the "Set Interval" button. To check the transmission interval for an EmbedRF board, you may press the "Get Transmission Interval" button and the Transmission Interval list box will automatically select which interval that EmbedRF Board has been given.

4.5. Configure Analog Sensor Data and Digital Outputs

The EmbedRF Boards may be configured to store and transmit up to 4 analog sensor readings. Each of the 4 channels (channels A/D0, A/D1, A/D2, and A/D3) consist of 10-bit analog sensor readings. Each channel has a 3V range that is represented by the numbers 0 – 1023 where 0 is equivalent to 0 V and 1023 is equivalent to 3V.

The analog sensor channel data is sent in bytes 0 – 7 of the data packet:

A/D 0 = bytes 0 and byte 1 (byte 0 is the high byte, byte 1 is the low byte)
A/D 1 = bytes 2 and byte 3 (byte 2 is the high byte, byte 3 is the low byte)
A/D 2 = bytes 4 and byte 5 (byte 4 is the high byte, byte 5 is the low byte)
A/D 3 = bytes 6 and byte 7 (byte 6 is the high byte, byte 7 is the low byte)

A/D Channels 2 and 3 may be alternatively set as digital outputs. These two channels may not be set as both a digital output and as an analog input. You may only configure them as one or the other.

All A/D Channels are set at the same time when you press the "Set Analog" button.

To configure A/D channel 0, check or uncheck the "Record A/D 0" checkbox. Checking the checkbox implies that you are configuring the EmbedRF Board to record analog data on A/D Channel 0.

A/D channel 1 is set the same way as A/D channel 0 using the "Record A/D1" checkbox instead.

To configure A/D channel 2 to be an analog input, select AD2 Active from the pull-down list box. To configure A/D channel 2 to be a digital output, logic 1, select DO2 Active High. To configure A/D channel 2 to be a digital output logic 0, select DO2 Active Low. If you want to set A/D channel 2 to be neither an analog input nor a digital output, select the AD2 / DO2 Inactive option from the pull-down listbox.

A/D channel 3 is set the same way as A/D channel 2 using the AD3 pull-down list box.

4.6. Configure to Notify when Receiving Data

An EmbedRF Board that has been configured to receive data can also be configured to put its received data out the UART (USB) line every time a packet is received. This can be useful when using the EmbedRF DesktopPro software as you can view the data as it arrives.

To configure an EmbedRF Board to send the data packet to the USB port every time it has been received, select the "Notification On" checkbox and then press the "Set Notification" button.

4.7. Find a Transmitter

An EmbedRF Board that has been configured to receive data (a receiver) may search for an EmbedRF Board that has been configured to transmit data (a transmitter). Once the receiver detects a valid transmitter, they lock-in on each other and become synchronized. A valid transmitter is one which has the same network ID. If a transmitter and receiver do not have the same network ID, they will never synchronize.

There are two ways for a receiver to search for a transmitter

- Find the transmitter that is closest in proximity to the receiver
- Find the first transmitter detected after a find is initiated

For both of the find transmitter methods, a timeout may be specified. This timeout does not affect how the EmbedRF performs the search. It is only used by the EmbedRF

Desktop Pro PC software as a timeout from waiting the reply from the EmbedRF device. The internal EmbedRF device timeout period is 10 times the transmit/receive interval. So you should enter a timeout that is greater than 10 times the transmit/receive interval. For example, if your transmit interval is 0.25 seconds, a good timeout would be 3 seconds. If you enter a timeout less than 10 times the transmit interval value, the Desktop Pro screen will not update when an ID has been found and sent over the serial UART.

4.7.1. Proximity Find

The proximity find uses the receive signal strength indicator (RSSI) as a measure of proximity. The transmitter that has the highest RSSI will be selected as the transmitter to which the receiver will synchronize.

To begin a Proximity Find, first enter the timeout in seconds after which the EmbedRF Desktop Pro software will stop waiting for a reply from the EmbedRF device. Then press the "Find Nearest Transmitter" button. The EmbedRF device will search for 10 times the transmit / receive interval and synchronize with whichever transmitter is closest in proximity. Once the ID has been found, it sends the ID over the serial UART to notify the host microcontroller or the EmbedRF Desktop Pro software. You may verify which transmitter has been found by pressing the "Get Transmit ID" after the find is complete and see which transmit ID has been synchronized. If the ID is 0, then the receiver did not find a transmitter.

4.7.2. Find First

The Speed Find synchronizes the receiver with the first transmitter which shares the receiver's network ID.

To begin a Find First, first enter the timeout in seconds after which the EmbedRF Desktop Pro software will stop waiting for a reply from the EmbedRF device. Then press the "Find First Transmitter Detected" button. The receiver will search for a transmitter either until it finds one or until 10 times the transmit / receive interval has expired. Once the ID is found, the EmbedRF device it sends the transmit ID over the serial UART to notify the host microcontroller or the EmbedRF Desktop Pro software. You may verify which transmitter has been found by pressing the "Get Transmit ID". If the ID is 0, then the receiver did not find a transmitter.

4.8. Set Network Mode

The EmbedRF device may configured as a point-to-point transceiver, or as a multi-point to point receiver. When the device is in point-to-point mode, data is transferred only between devices that have the appropriate transmit / receive IDs. When a device is configured as a multi-point to point receiver, data is received from any EmbedRF transmitter that shares its network ID number. To set the network ID, use the Network Configuration pull-down menu and select either "Point-to-point" or "Multi-point-to-point". Then press the "Set Network Config" button. For more information about setting up a multi-point-to-point network, please see our application note entitled "Setting up an analog sensor network using EmbedRF".

4.9. Get Firmware Version

Each EmbedRF Board has firmware loaded on that is essentially like an operating system. This operating system controls the low-power wireless transmission protocols. If you are having difficulty with an EmbedRF Board, technical support may request that you get the firmware of the EmbedRF Board that you are using. You can prompt the EmbedRF Board for its firmware by pressing the "Get Version" button. The Firmware appears in the "Firmware Version" text box.

5. Setting Data to Transmit

The EmbedRF Board may be configured to wirelessly transmit a 12-byte data packet on a set interval (see section 4.2 for details). Ten bytes of the 12-byte data packet can be set to any value between 0 and 255 by using the Transmit tab window of the EmbedRF DesktopPro software when the EmbedRF Board is connected to the PC via the USB Interface Board.

Before configuring the data that the EmbedRF Board will transmit, ensure that you have selected the Transmit tab from the upper left-hand corner of the window. The Transmit tab window is shown in the figure below.

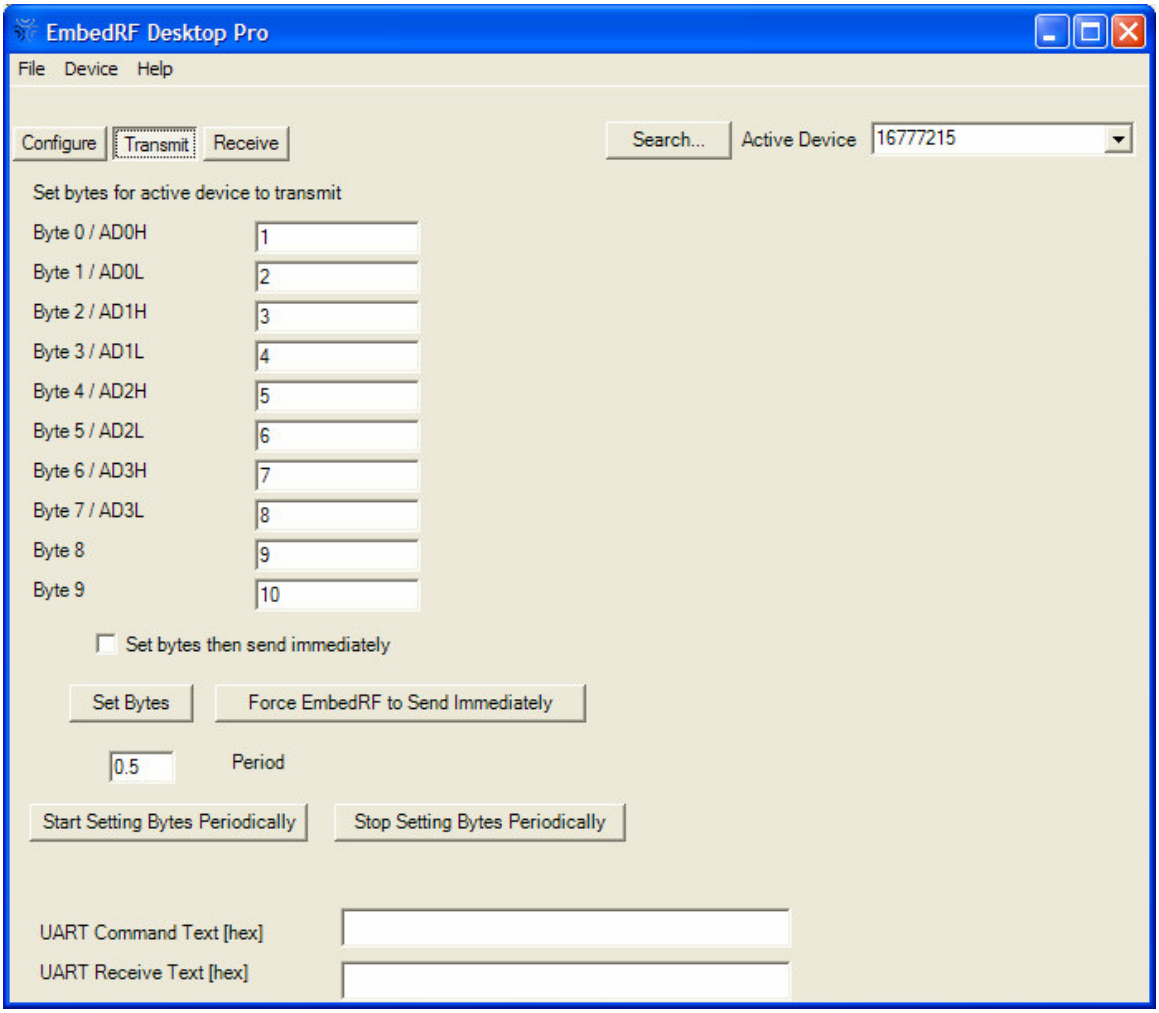

Figure 2: Transmit tab window of the EmbedRF Desktop Pro software. This window may be used to set the data an EmbedRF Board will transmit wirelessly to another EmbedRF Board.

5.1. Restrictions when Setting Bytes Containing Analog Sensor Data

Notice in figure 2 above, that bytes 0-7 are labeled with the analog sensor channel numbers (AD0L, AD0H etc) as well as being labeled Byte 0-7. If an analog sensor channel has been configured to record data, then you will not be able to set that byte to a value since it will be sending the analog data in that byte. For example, if analog sensor channel 1 has been set to record analog data (see section 4.4), then bytes 2 and 3 will contain analog sensor data and may not be configured to a custom value.

5.2. Setting Bytes to Transmit

Using the text boxes in figure 2, enter any number between 0 and 255 for each of the 10 bytes of data. Then press the "Set Bytes" button. If you do not receive an error message

back from the EmbedRF Board, you have successfully set each of the 10 bytes that will be transmitted by the EmbedRF Board every transmission interval

5.3. Setting Bytes to be Sent Immediately

You will notice on the Transmit tab window, that there is a checkbox labeled "Set Bytes then Send Immediately". This tells the EmbedRF Board to transmit the data being sent to it immediately after it has received the command to update the data bytes. It will continue to send this data on the regular transmission intervals. There is also a button labeled "Force EmbedRF to Send Immediately". This is a command that forces the EmbedRF board to immediately send wirelessly any data that it has in its buffer. Pressing this button does not update the data on the EmbedRF transmitter prior to the wireless send.

5.4. Simulation for Periodically Updating the Bytes

Using the DesktopPro software, you can periodically update the data bytes of data on an EmbedRF board configured as a transmitter. To do this, enter a value in the textbox labeled as Period. The value in the Period textbox is the amount of time in seconds that the EmbedRF board will update the data (and send if the "Set Bytes and Send Immediately" checkbox has been checked). This feature is meant to mimic what a host microcontroller might do to periodically update the EmbedRF transmit data prior to the data being wirelessly sent to an EmbedRF receiver. To illustrate that the data is changing, Byte0 is incremented by the EmbedRF DesktopPro software immediately prior to sending the update command. Therefore, you can verify that the update is occurring by viewing the data on an EmbedRF receiver and checking that the data in Byte0 increments at the interval specified.

6. Viewing Data that Has Been Received

An EmbedRF Board can be configured to receive data at regular transmission intervals (see section 4.1 for details). You may use the Receive tab window to view the 16-bytes of data when they arrive at the EmbedRF Board over the serial UART.

Before beginning this section, make sure you have selected the Receive tab window from the upper left hand corner of the EmbedRF DesktopPro software window. The Receive tab window is shown in Figure 3. The Receive tab window is shown in the figure below.

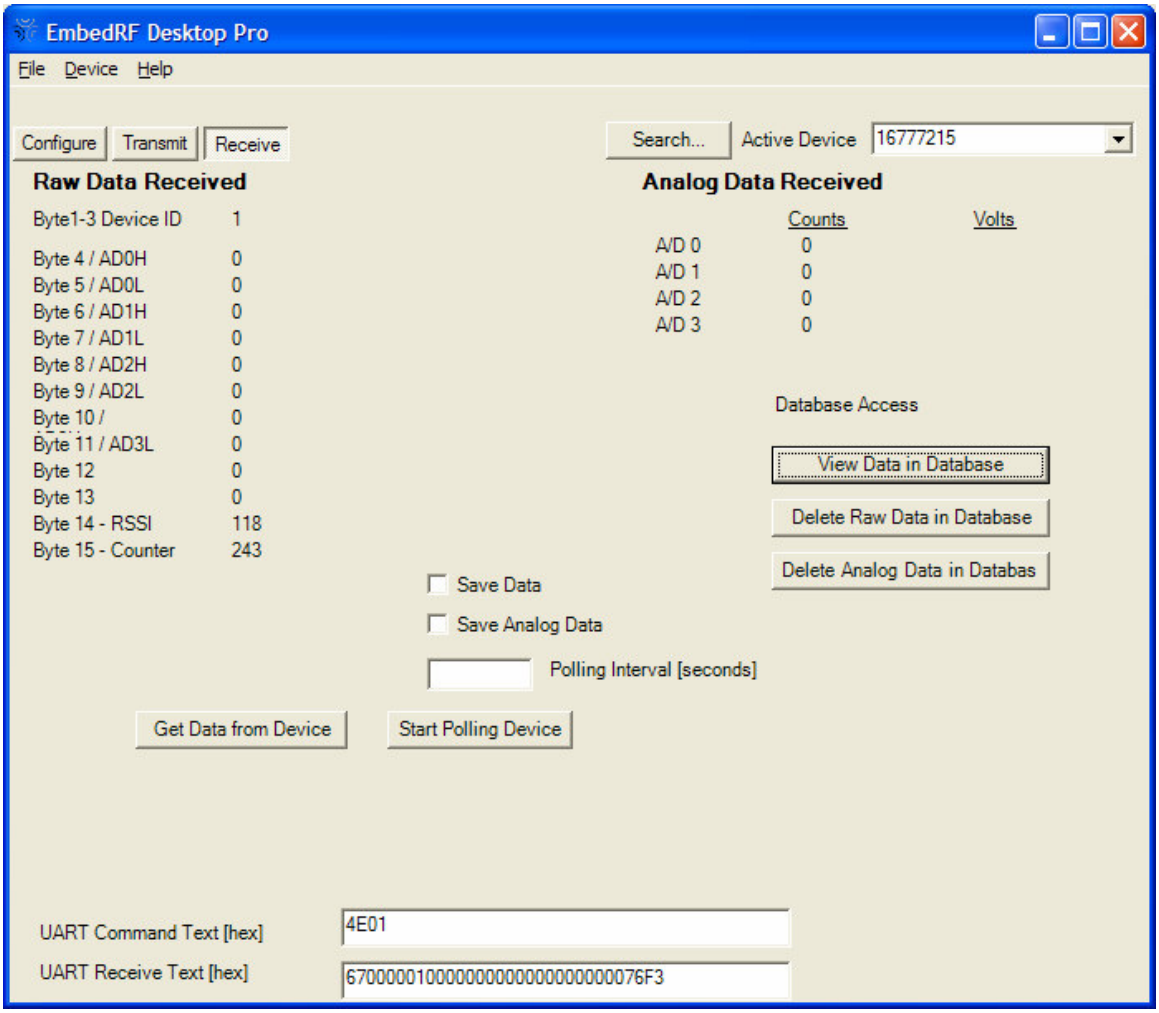

Figure 3: Receive tab window of the EmbedRF DesktopPro software. This window may be used to observe which data has been wirelessly received by an EmbedRF Board.

Sixteen bytes of data are sent over the serial UART when a wireless data packet has been received and the device is in "Notify" mode or when a device is prompted for its received data.

The transmit ID of the transmitter is included within each notify UART packet and is located at bytes 1, 2, and 3 with the most-significant byte sent first.

Following the transmit ID, the next ten bytes of data (bytes 4-13) are either raw payload data or analog sensor data, depending on whether the transmitter was configured to transmit analog data. In this example, we configured the transmitter to send analog data on A/D channel 1. A/D channel 1 analog data comes through on bytes 6 and 7.

Byte 14 contains the received signal strength indicator or (RSSI). This byte is a 2's complement number representing the strength of the received signal. A larger RSSI means that the transmitter is closer to the receiver while a smaller RSSI indicates that the transmitter is further away from the receiver.

Byte 15 in the UART notify data packet is a packet counter. Every time the transmitter sends a packet, it increments its counter by 1. This byte can be used to indicate whether a receiver has missed a packet sent by a transmitter. It should increment at each interval. In this example, the counter should increment by 1 every 0.25 seconds.
Ten bytes of data are included within each data packet. Bytes 0-9 are either raw data or they include analog sensor data, depending on whether the transmitter was configured to transmit analog data.

Byte 10 contains the received signal strength indicator or (RSSI). This byte is a 2's complement number representing the strength of the received signal. A larger RSSI means that the transmitter is closer to the receiver while a smaller RSSI indicates that the transmitter is further away from the receiver.

Byte 11 is a packet counter. Every time the transmitter sends a packet, it increments its counter by 1. This byte can be used to determine whether a receiver has missed a packet sent by a transmitter. It should increment at each interval.

6.1. Polling the EmbedRF Board for Data

In Figure 3, you will notice a button called "Start Polling Device". When this button is pressed, the EmbedRF DesktopPro software will poll an EmbedRF Board for data every 0.5 seconds. Polling may be disabled by pressing the button again.

6.2. Viewing Data when it Arrives during Notification

Section 4.5 described how to set up an EmbedRF Board that is receiving data to notify the EmbedRF DesktopPro software when a data packet has arrived. If notification has been enabled for an EmbedRF Board, the Receive tab window in Figure 3 will continually update every interval, whenever a data packet has been received.

7. Storing Data to the Database

EmbedRF DesktopPro comes with a Microsoft Access database (embedrf.mdb) that is located in the installation directory (for a typical installation, the location of this database is c:\Program Files\EmbedRF\EmbedRF.mdb). An EmbedRF Board can be configured to receive data and the EmbedRF DesktopPro software can be used to log and store that data to the database when the receiver EmbedRF board is connected to the PC via the USB Interface Board.

7.1. Storing Raw Data

To begin storing received data to the database, select the "Receive" tab in the upper left hand of the EmbedRF DesktopPro software window. The window in Figure 10 should appear.

Click on the "Save Data" checkbox. Now, when new data packets arrive at a receiver that is connected to the computer via the USB Interface Board, the raw data will be stored to the database for later retrieval.

7.2. Storing Analog Data

The analog data is included within the 10 bytes of raw data. The analog data is sent within the data bytes according to Table 1 below:

Analog Channel	Bytes
0	0 and 1
1	2 and 3
2	4 and 5
3	6 and 7

Table 1: Analog data is stored in bytes 0-7 of the raw data packet.

Analog channel 0 is included in bytes 0 and 1, analog channel 1 is included in bytes 2 and 3, analog channel 2 is included in bytes 4 and 5, and analog channel 3 is included in bytes 6 and 7. It may be convenient for you to store the analog data rather than storing the raw data. The analog data is formatted such that the two bytes of data which make up a given analog data reading are concatenated together to make a single value. This value is stored into the database. The reason you need two bytes to store an analog reading is because the analog-to-digital converter is 10-bits long. So you need two bytes to store the reading.

To convert raw bytes into an analog reading, use the following formula – where Analog_Data_X represents one of 4 analog channels, where X is the number 0-3. ADXH and ADXL represent raw data bytes. For example, AD0H represents byte 0 and AD0L represents byte 1

Eq 1. $$Ana\log\_Data\_X[V] = \frac{((ADXH \times 256) + ADXL) \times 3.0}{1023}$$

To store the analog data, click on the "Receive" tab in the upper left-hand corner of the EmbedRF DesktopPro software window. Then click on the "Save Raw Data" checkbox. When new data packets arrive at a receiver that is connected to the computer via the USB Interface Board, the raw analog data will be stored to the database for later retrieval.

7.3. Viewing Data

Data that has been saved in the database can be viewed even if you do not own a copy of Microsoft Access. To view the data stored to the database, click the "View Data in Database" button on the right side of the window in the "Receive" tab window. Both the raw data and the analog data will appear as shown in the window below.

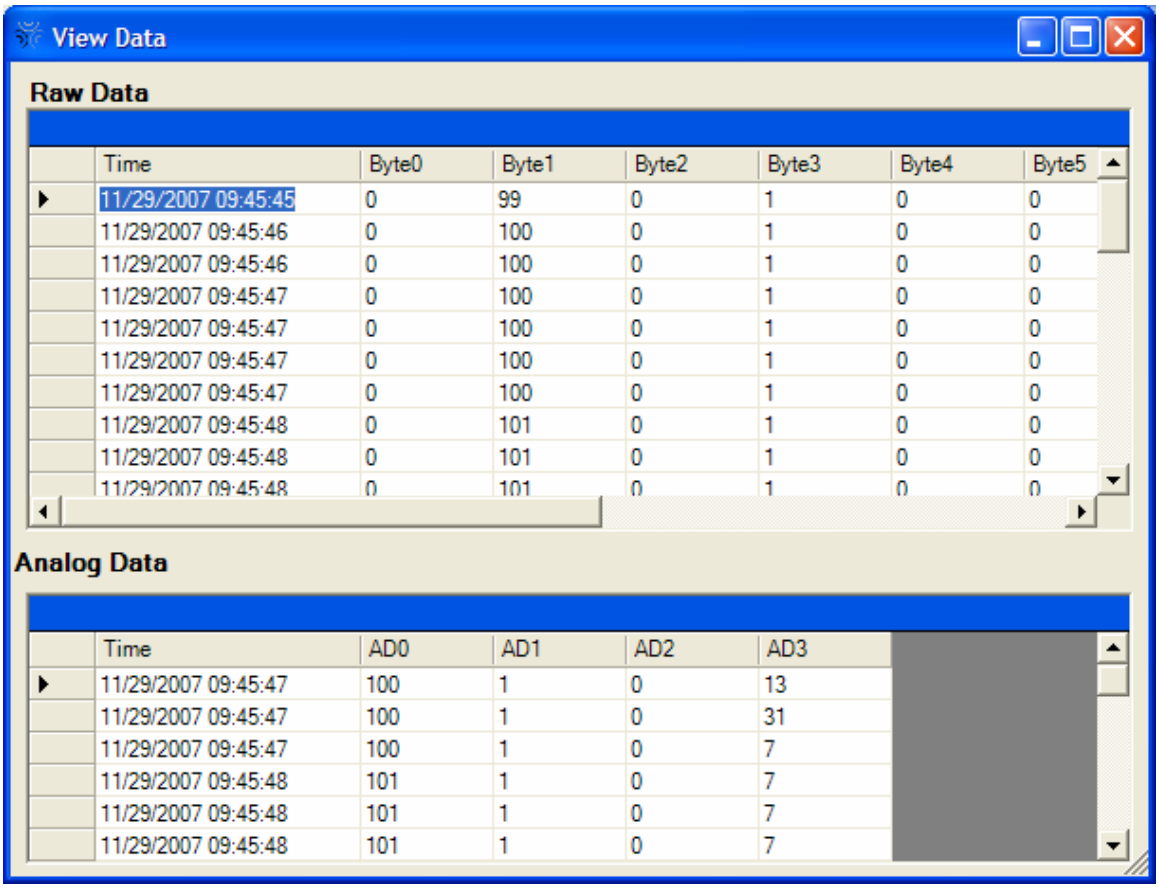

Figure 4: Data viewer window. This window may be used to view and to export both raw and analog data stored to the database.

7.4. Delete Data

You may delete raw data that has been previously stored to the database by clicking on the "Delete Raw Data" button under the Receive tab.

You may delete analog data that has been previously stored to the database by clicking on the "Delete Analog Data" button.

You may store as many as 50,000 data packets in both the Raw Data table and the Analog Data table. After 50,000 data points have been received, you will receive an error message indicating that the database is full. You will then need to delete data to make room for new data.

