RF RANDOM ACCESS INTEGRATED NODE USER's GUIDE

REVISION 1.0 DRAFT
AUGUST 2nd, 2003

MAT LAIBOWITZ
mat@media.mit.edu
RESPONSIVE ENVIRONMENTS GROUP
MIT MEDIA LAB

Introduction

The RF Random Access Integrated Node (RFRAIN) is a complete system of hardware and software for the development of multi-node wireless applications.

The RFRAIN hardware is based around the Chipcon CC1010 Integrated Circuit which contains an RF Transceiver and a Microcontroller in one package. This allows development of standalone wireless applications on the RFRAIN. The hardware also provides all the analog tuning and support hardware for the RF transceiver and two choices of antennas. A full spectrum rgb led, an additional white led, a miniature 4-way navigation switch, and an analog temperature sensor are also included on the RFRAIN to further demonstrate and test distributed wireless applications with no additional hardware.

The range of the RFRAIN has been tested to work at 50 meters indoors.

The RFRAIN software implements a random access peer to peer network for up to 64000 nodes. It implements a collision avoidance/detection algorithm, acknowledgement with timeouts and retries, broadcast addressing, crc checking on both the header and the data payload, 512 byte buffers in both directions, and a detailed serial interface that allows the RFRAIN to be easily embedded into any application that needs wireless communications.

This guide describes the use of the RFRAIN. It assumes some familiarity with the RFRAIN's hardware and software. Please see http://www.media.mit.edu/resenv/rfrain for the latest information. This guide is organized in three main sections. The first section contains general information about the RFRAIN. The second section contains information on how to embed the daughtercard with its pre-loaded software into an application device. The third section contains the steps for getting started writing custom code and standalone applications that run directly on the RF Random Access Integrated Node.

Section 1 -- General Information

Antenna

The RFRAIN comes with a monopole antenna in the form of a wire. This wire is 16.4cm long. It is a tuned length for applications that work at the 433MHz base frequency. If the operating frequency is changed, the length of this wired should be changed. See the antenna guide at http://www.media.mit.edu/resenv/rfrain/docs.html for alternate lengths.

Alternatively, the wire antenna can be removed and replaced with an SMA connector. Straight and right-angle SMA connectors are available that fit the pcb. An off-the-shelf whip antenna tuned to the correct frequency can then be attached to SMA connector.

To switched antennas, remove the 0R resistor at R63 and stuff a 0R resistor at R62.

Passive Components

The inductors and capacitors on the RFRAIN are chosen for use at the base frequency of 433MHz. To use a different base frequency, the component values need to be changed. Use the provided windows application, SmartRF Studio, to calculate component values. SmartRF Studio takes in frequency and power settings and outputs required component values. The designators on the pcb match those in SmartRF Studio. For move information on SmartRF studio, see the SmartRF Studio user's guide at http://www.media.mit.edu/resenv/rfrain/docs.html. The software is available at http://www.media.mit.edu/resenv/rfrain/software.html.

Changing Base Frequencies

Besides changing the antenna and the passive components, to change the base operation frequency you need to change the register settings by changing the software. If you are using the provided RFRAIN software, you need to flash the CC1010 on the RFRAIN with the hexfile provided for that frequency. Hexfiles for 433, 868, and 915MHz are provided. If you are using a different frequency or developing your own software, you need to setup the MODEM registers in the CC1010 with the appropriate value. SmartRF Studio generates register values for all the RF registers in the CC1010.

Physical Dimensions

The RFRAIN is 1.25" wide and 2.75" long. It has two mounting holes in the top two corners and two 2x9 header pin-outs on the lower half. The mating connector for these pin-outs is: Molex Part# 87267-1850 available from Digikey. These are surface mount on the application device side and the mating header for the RFRAIN on the other side.

P1 Pin-out

Pin 1	3.3V Power Pin
Pin 2	General Purpose I/O P1_7
Pin 3	General Purpose I/O P1_6
Pin 4	General Purpose I/O P2_6
Pin 5	General Purpose I/O P1_5
Pin 6	General Purpose I/O P2_7
Pin 7	General Purpose I/O P0_3
Pin 8	Flash Programming Pin
Pin 9	SPI MISO
Pin 10	Reset Pin
Pin 11	Serial RX
Pin 12	ADC Channel 0
Pin 13	Serial TX
Pin 14	ADC Channel 1
Pin 15	INT1
Pin 16	RSSI Analog Output
Pin 17	General Purpose I/O P2_5
Pin 18	Ground

P2 Pin-out

Pin 1	3.3V Power Pin
Pin 2	N/C
Pin 3	SPI MOSI
Pin 4	General Purpose I/O P1_1
Pin 5	SPI CLK
Pin 6	General Purpose I/O P1_2
Pin 7	INT2
Pin 8	General Purpose I/O P1_3
Pin 9	PWM Output 2
Pin 10	General Purpose I/O P1_4
Pin 11	PWM Output 3
Pin 12	General Purpose I/O P2_2
Pin 13	Serial2 TX
Pin 14	General Purpose I/O P2_3
Pin 15	Serial2 RX
Pin 16	General Purpose I/O P2_4
Pin 17	General Purpose I/O P1_0
Pin 18	Ground

Section 2 – Using the RFRAIN with provided software

The provided software handles all RF wireless networking functions. It buffers incoming and outgoing messages, handles addressing of messages, crc and error checking, retries and timeouts, and collision avoidance/detection.

It provides a buffered serial interface to an application circuit. It requires only 4 pins connected to an application circuit, 3.3V, Ground, Serial RX, and Serial TX. The serial interface runs at 115200bps with 8 data bits, no parity bits, and 1 stop bit. The RFRAIN's serial interface is full-duplex, buffered, and interrupt driven, so that an application can send it commands no matter what it is doing and the RFRAIN will buffer them until it can handle them. All commands from the application circuit have a packet id and generate and return packet from the RFRAIN with a matching packet id. This allows the application circuit to work how it wants, either synchronously or asynchronously.

Packet Structure
The RFRAIN uses the following structure for its wireless communications:

Preamble and Sync Byte, handled by the RF Transceiver
Destination MSB
Destination LSB
Source MSB
Source LSB
Packet ID
Data Packet Length
Flags (d0: is high to request an ACK, future bits for encryption)
CRC8 check byte on the header
Data Payload
CRC16 MSB check on the data payload
CRC16 LSB

The RFRAIN currently has a maximum data payload packet size of 64 bytes. The CRC checking is handled by the RFRAIN and not exposed to the application device.

A destination address of 0xFFFF is a broadcast address and the packet will be sent to everyone in range. Future addressing schemes for group broadcast messages are in development.

RFRAIN Settings

The RFRAIN has a few settings that can be set over the serial interface. The most important is the address of the RFRAIN. This is added by the RFRAIN to outgoing packets and used by the RFRAIN to discard packets not intended for this node.

This can be set over the serial interface and is stored in nonvolatile flash memory. Although it is okay if the application circuit wants to reset this on power up. If this is not set, a default address is used.

The other settings are the timeout to wait for an acknowledgement after transmission and the number of times to retry.

Serial Protocol

All commands issued to the RFRAIN begin with the header of:

FFh	header byte
PID	Packet ID, any byte
OPC	Opcode, see next page for description of opcodes
Data	Zero or more data bytes, depending on opcode

And all commands receive a return packet of:

FFh	header byte
PID	matching packet id
Data	Return values or return data

It is possible to send more commands to the RFRAIN before receiving the return packet from the last one. The packet ID should be used to identify which return belongs to which packet. Of course, the application can also do everything as a blocking call and wait for the return.

Serial Protocol Opcodes

```
01h       Turn on RFRAIN
          This powers up the transceiver and begins
          listening for packets.
          The RFRAIN startups up in on mode.
          Return:   FFh PID RET
               Where RET  =   00h if the power up was okay
                              FEh if the power up failed

02h       Turn off RFRAIN
          This powers down the transceiver and goes to a
          low power mode.
          Return:   FFh PID RET
               Where RET  =   00h if the power off was okay
                              FEh if the power off failed

03h       Send settings
          Packet:   FFh PID 03h NAM NAL TTO TRC
               Where NAM =    Node Address MSB
                     NAL =    Node Address LSB
                     TTO =    Tx Time Out
                     TRC =    Tx Retry Count
          Return:   FFh PID RET
               Where RET  =   00h if the settings were okay
                              FEh if there was an error

04h       Check Status
          Returns Status Byte
          Return:   FFh PID STA
               Where STA  = d0: rx buffer contains packets
                            d1: rx buffer is full
                            d2: tx buffer is full

05h       Clear Tx Buffer
          Empties tx buffer discarding all unsent packets
          Return:   FFh PID RET
               Where RET  =   00h if the buffer was cleared
                              FEh if there was an error

06        Clear R Buffer
          Empties rx buffer discarding all received packets
          Return:   FFh PID RET
               Where RET  =   00h if the buffer was cleared
                              FEh if there was an error


07h       Read Packet from Rx Buffer
          Returns the next packet in the Rx Buffer
```

```
            Return: FFh PID SAM SAL LNG FLG D0 … Dn
                 Where SAM =    Source Address MSB
                       SAL =    Source Address LSB
                       LNG =    Data Payload Length
                       FLG =    Packet Flags
                       D0…Dn =  Data Payload

08h         Place packet in Tx buffer for transmission
            Packet: FFh PID 08h DAM DAL LNG FLG D0 … Dn
                 Where DAM =    Destination Address MSB
                       DAL =    Destination Address LSB
                       LNG =    Data Payload Length
                       FLG =    Packet Flags
                       D0…Dn =  Data Payload
            Return: FFh PID RET
                 Where RET =    00h if successful (acked)
                                00h if sent with no ack req
                                FDh if tx buffer is full
                                FEh if transmission failed
```

Section 3 – Developing Standalone RFRAIN applications

Requirements

CC1010 Development Kit

Available for online purchase from Chipcon at http://www.chipcon.com. This allows you program the flash in-circuit on the RFRAIN as well as use the Keil in-circuit debugger. The CC1010DK is available in several different frequencies. For development for the RFRAIN is does not matter which one you get, this frequency is the frequency of the included Evaluation Module. If you wish to have your RFRAINs talk to the Chipcon Evaluation Module, then get the matching CC1010DK.

Keil C51 Compiler and uVision IDE

The RFRAIN code, and most likely, any code that uses the CC1010's RF transceiver will be too large to use the free evaluation version of the C51 compiler. The full version is necessary. The uVision IDE comes both with the CC1010DK and the Keil C51 compiler package. The Keil C51 package is available from http://www.keil.com.

Setup

1) Install the Keil software and the CC1010 software included in the development kit.
2) Connect the CC1010DK board to your windows pc using both the parallel cable and the included null modem cable. The null modem cable connects between the Serial 1 port on the CC1010DK and any free serial port on your PC. The Parallel is used to download the code into the flash of the CC1010 and the serial cable is used for the in-circuit debugger in the Keil uVision IDE.
3) Remove the CC1010 Evaluation Module if it is mounted on the CC1010DK board.
4) Attach a 2x5 standard ribbon cable from the green port marked SPI on the CC1010DK board. Attach the other end to the matching programming port on the RFRAIN.
5) Power up either the CC1010DK board or the RFRAIN board.

Programming the Flash

After setting it up as above, use the Chipcon Flash programming utilty, either in its standalone form or by selecting it from the Keil uVision tools menu.
Load a hex file and hit program. The default settings are all correct for the RFRAIN.

Developing Code

Refer to the CC1010DK User's guide for information on how to setup the compiler and Keil uVision IDE for use with the CC1010, or use one of sample uVision projects at
http://www.media.mit.edu/resenv/rfrain/software.html.

Using the In-circuit debugger

1. Write and compile your application under Keil uVision.
2. Make sure the serial cable and parallel cables are connected to the CC1010DK board and the ribbon cable is connected to the RFRAIN.
3. Download the Debug Bootloader to the CC1010 flash by choosing Download Debug Bootloader from the Keil uVision Tools menu.
4. Setup the comm. Port under Options for Target in the Project menu of the Keil uVision IDE. Under the Debugger tab, make sure the Chipcon debugger is selected. Hit settings and set the correct serial port.
5. Start Debug Session under the Debug menu in the Keil uVision IDE. This will download your code and begin debugging.