



RDM-A4FZ
RF Transceiver Module for 433 MHz
Application Notes
Rev 1.0



Reindeer Systems Pvt Ltd

Excellence Through Innovation

No. 67, 53rd Street,
9th Avenue, Ashok Nagar,
Chennai – 600083
India.



Document Revision History

| Revision No. | Date | Description/Changes |
|--------------|-----------|---------------------|
| V1.0 | 15/9/2009 | Initial Release |



Table of Contents

1.Introduction 4

2.Specifications 4

3.Application Firmware Explanation..... 5

 3.1 Void Main (void)..... 5

 3.2 Interrupt Service Routine (ISR)..... 6

4.Header Files 7

5. Using the sample code with other Microcontrollers 8

6.Contact Us 9

 6.1 Technical Support..... 9

 6.2 Sales Support 9



1. Introduction

This application note describes the usage of RDM-A4FZ module as a transceiver for transmitting and receiving data. The main objective is to explain how wireless transmission of data can be achieved from one device to another using a microcontroller's UART.

2. Specifications

The sample code is developed based on the R8C series of microcontrollers from Renesas. The specifications are mentioned below.

Platform : HEW (High Performance Embedded Workshop)
Language : Embedded C
Controller : Renesas R8C tiny series
Daughter board : RDM-A4FZ
Mother board : RDSRF-232
Frequency : 433 MHz
Interface with RDM-A4FZ: SPI

Application : Data communication between RF transceivers.
Serial configuration : Baud rate: 9600bps, Parity: none, Data Bits: 8, Stop Bits: 1

Below mentioned is the list of files used in this application

- 1) RDM_A4FZ_Main.c – This contains the main program.
- 2) RDM_A4FZ_Function.h – This file contains all the function definitions.
- 3) RDM_A4FZ_IOdefine.h – This file contains all definition for all the I/O ports.
- 4) RDM_A4FZ_Prototype.h – This file contains all the variable declarations.



3.Application Firmware Explanation

3.1 Void Main (void)

The Program enters into the main () function after power-up. All the functions used for configuring the microcontroller and the RF module have been explained below.

Function: Oscillator_configuration ();

This function is used to configure the various register of the microcontroller like register control - 0 used to configure the XIN clock, register control-1 used to configure the low speed on-chip oscillator, select and enable the high speed on-chip oscillator. In the sample application firmware the internal high speed on-chip oscillator is used as the main system clock. The operating frequency of the clock is 20MHz.

Function: Peripheral_Initialization ();

- a. **sfr_init ();** This function is used to configure the input, output ports and also for enabling the corresponding pull-ups for the I/O's.
- b. **serial_init ();** This function is used to configure the UART- 0 of the microcontroller. All the serial settings like serial baud rate, parity etc are configured here.
- c. **timerx_init ();** This function is used for configuring the internal timer of the microcontroller. The timer is configured such that it gives an interrupt every 1 millisecond. This timer is used for creating any delays or any other timing needed.

Function: adf_7020config ();

This function used to initialize the RDM-A4FZ module and its registers. Below mentioned are the list of registers that have to be programmed.

1. Register 0 – This is used to configure the frequency of operation.
2. Register 1 – This is used to configure the oscillator filter settings.
3. Register 2 – This register is used to configure the type of modulation and the output power.
4. Register 3 – This register is used to configure the receiver parameters.
5. Register 4 – This register is used to configure the demodulator settings.
6. Register 5 – This register is used to configure the address of the unit.



Function: receive_mode();

This function is used to enable the receive mode of the module. As soon as this command is issued the module will go into receive mode and can receive data.

Function: int_init ();

This function is used to initialize the external interrupt – 0 of the microcontroller. This interrupt is used to by the module to indicate the microcontroller that a valid data has been received by the RF module.

3.2 Interrupt Service Routine (ISR)

3.2.1 Void uart0 (void)

This uart0 interrupt service routine is used to receive the data from hyper terminal. When data is received from PC it is stored in the 'tx_buffer'. When the complete packet has been received the 'txisr()' function will be enabled so that the packet can be transferred to the RF module.

3.2.2 Void txisr (void)

In this ISR the transfer of data from the microcontroller to the RF module takes place. Before the start of data transfer the 'transmit_mode()' function is used to enable the transmission mode for the RF module.

Function: transmit_mode();

This strobe is used to enable the RF transmission mode of the RF module. After this strobe has been issued any data written into the RF module through the 'DATA' line will be transmitter over the air.

Function: transmit_data();

Before transmitting the data, the address(sync) bytes have to be sent. The function 'send_sync()' is used for transmitting the address of the transmitter over the air. Then the length of the packet should be transmitted. Then the data to be transmitted is Manchester encoded and fed into the RF module through the data line at a low-to-high transition on the 'data_clk'.



3.2.3 Void int0 (void)

This interrupt function is used to call the receive function whenever the RF module indicate that a valid data has been received.

Function: receive_data ();

This function is called when the RF module issue a low-to high pulse on the 'int_lock' pin. This indicates the microcontroller that a valid address has been received.

The microcontroller the receives the data through the 'data' line at a low-to-high transition on the 'data_clk' line. This data is Manchester decoded and stored into a buffer. The first byte received indicates the length of the packet. After the entire packet is received the microcontroller outputs it through the UART.

4.Header Files

The program has 3 header files

1. RDM_A4FZ_IOdefine.h
2. RDM_A4FZ_Function.h
3. RDM_A4FZ_Prototypes.h

RDM_A4FZ_IOdefine.h:

This file contains the macro definition of communication pins. Pin like ce, sle, sclk, sdata, sread, data_io, data_clk, int_lock etc,

RDM_A4FZ_Function.h:

This file contains various function declarations of the program.

RDM_A4FZ_Prototypes.h:

This file consists of the global variable declaration type unsigned char, unsigned int and unsigned long.



5. Using the sample code with other Microcontrollers

The sample code is developed for Renesas R8C/27 microcontroller. For interfacing the RF module with a different microcontroller, use only the below mentioned functions.

- a. `adf_7020config ();`
- b. `receive_mode();`
- c. `transmit_mode();`
- d. `transmit_data();`
- e. `receive_data ();`



6.Contact Us

6.1 Technical Support

Reindeer Systems Pvt Ltd has built a solid technical support infrastructure so that you can get answers to your questions when you need them.

Our technical support engineers are available Mon-Fri between 9:30 am and 6:00 pm Indian standard time. The best way to reach a technical support engineer is to send an email to support@reindeersystems.com. E-mail support requests are given priority because we can handle them more efficiently than phone support requests.

6.2 Sales Support

Our sales department can be reached via e-mail at sales@reindeersystems.com or by phone at 91-44-45022335/337.

Our sales department is available Mon-Fri between 9:30 am and 6:00 pm.



Reindeer Systems Pvt Ltd

Excellence Through Innovation

No. 67, 53rd Street,
9th Avenue, Ashok Nagar,
Chennai – 600083
India.

Phone: 91-44-45022335, 91-44-45022337

Fax: 91-44-45022336

Website: www.reindeersystems.com