

TRF-2.4g, MiRF V1 & V2 Development Boards

5/11/2006

One of the earliest and best selling Spark Fun product lines is the MiRF/SMiRF RF modules, along with the Laipac TRF-2.4G RF module, both based around the Nordic Semiconductor nRF2401a (though the SMiRF has been retired). Now with the release of the Nordic Semiconductor nRF24L01 transceiver chip, Spark Fun Electronics has now created two new development boards to help ease the learning curve with the Nordic transceiver chips.

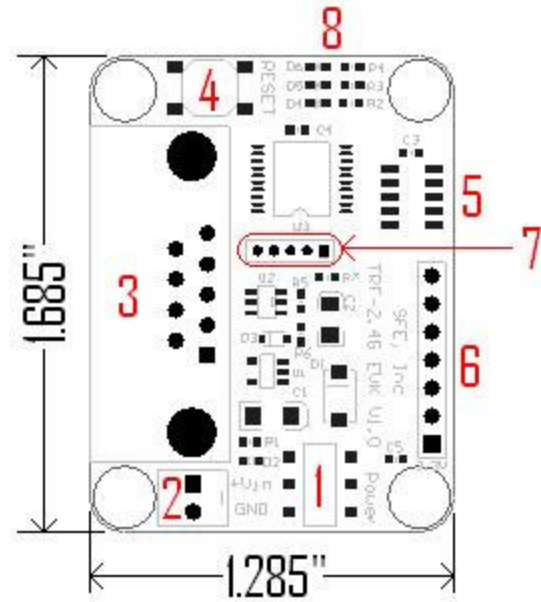
Both development boards come with open source code that demonstrates the function of the RF modules. Additionally, the source code is compact enough to compile on the free version of the CC5X compiler from BKD (<http://www.bknd.com/>

). With two development boards and 2 RF modules, along with an \$11 PG1 programmer from Olimex and a \$4 programming adapter (SKU# PCB-SFEAdapter), you have a complete 2-way RF development platform.

1 TRF-2.4G EVK V1.0

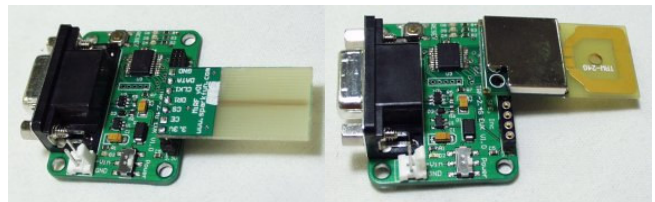
There are just a few things to familiarize yourself with before powering up:

- 1) Power Switch
- 2) Power Connector, 4V to 7V Input



- 3) DB9 Serial Connector
- 4) Reset Switch
- 5) TRF-2.4G Module Receptacle
- 6) MiRF V1 Receptacle
- 7) ICSP Programming Port
- 8) Status LEDs

As you can see from the pictures below, the demo board will work with either a TRF-2.4G module or with a MiRF V01 module since both use the nRF2401a chip.



There are two versions of demo code for this board, 24g_232demo_V01 and 24g_232demo_V02. Both are available for download right off of the product description page in both hex and C file formats.

MiRF V1 & V2 Development Boards

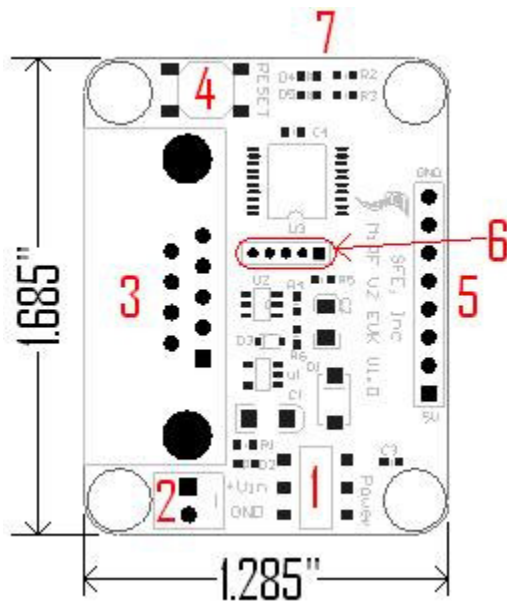
Version 1 of the code (V01) is an automated demonstration of the RF modules function. No input is taken from the serial line, instead characters are exchanged between modules automatically. The status LED's will augment their state with the reception of a packet from the other module.

Version 2 of the code (V02) takes user input from the serial line (at 9600/8/1/none) and transmits it to any other listening module. Upon reception of a packet from another module, the status LED's will augment their state and the received text will be output on the serial line (again 9600/8/1/none).

All of the TRF-2.4g EVK V01 boards are shipped with V02 of the code.

2 MiRF V2 EVK V01

This board is designed to work with the MiRF V2, which uses the nRF24L01. There a similar list of items to be familiar with on the MiRF V2 EVK V01 board as with the 2.4g EVK:



- 1) Power Switch
- 2) Power Connector, 4V to 7V Input

- 3) DB9 Serial Connector
- 4) Reset Switch
- 5) MiRF V2 Receptacle
- 6) ICSP Programming Port
- 7) Status LEDs

Currently there is only one version of demo code for this board (though more may follow), 24L01demo_V01, also available for download from our product description page. This code demonstrates the backward compatibility of the nRF24L01 with the nRF2401a. The code doesn't take any input from the user, but it does exchange text between two MiRF V2 modules and you can plug into the serial line (9600/8/1/none) to see the text as each module receives it. The status LED's also augment their state upon reception of a packet from another module.

3 Programming

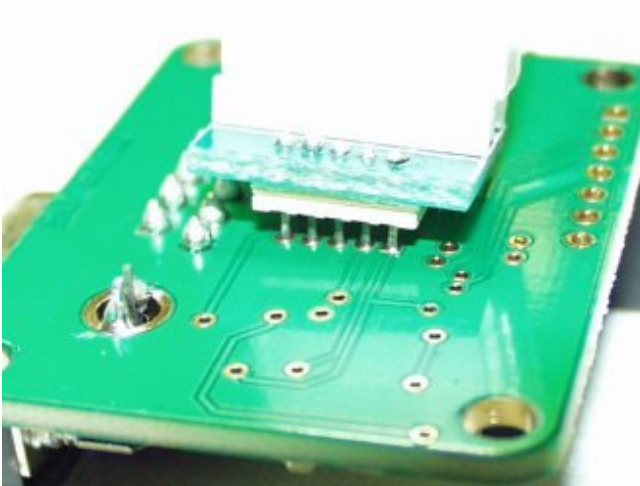
This section assumes that you already have a PG1 programmer and an SFE programming adapter (seen below), along with some working knowledge of WinPic.



To program either of the development

MiRF V1 & V2 Development Boards

boards, insert the SFE programming adapter into the PG1 programmer, then insert the free end of the programming adapter into the back of the development board as shown in the picture below.



Pay careful attention to the orientation of the adapter and the programming port. The square pad indicating pin one of the port is at the rightmost position in the picture. Also, the programming port is a friction fit, so you may need to lean the adapter over a bit to insure good contact.

The general sequence for programming is:

- 1) Connect your PG1 to a serial port and start WinPic. Make sure that your programmer initializes. If not, select the correct com port on the "Interface" tab and hit "Initialize".
- 2) Select PIC16F88 from the list of components on the "Device/Config" tab. You don't have to worry about the configuration bits, they will be imported when you import the code.
- 3) Plug the programmer into the development board.
- 4) Go to the "File" tab in WinPic and select "Load and Program Device (all)" and select the code you wish to load. WinPic will do the rest.

4 Hints and Tips

One of the underlying philosophies at Spark Fun Electronics is freedom of information exchange and general education. This is what drives us to make the development tools that we do, and it's also why all of our schematics and source code are freely available. But in spite of our best efforts, you may experience difficulties when you start altering the code for these boards. For those times late at night when it seems that nothing will work and the coffee's gone cold, check out the Spark Fun Support Forum (<http://www.sparkfun.com/cgi-bin/phpbb/login.php>). There's a large community of electronic guru-types there that may be able to help out.

Good Luck!