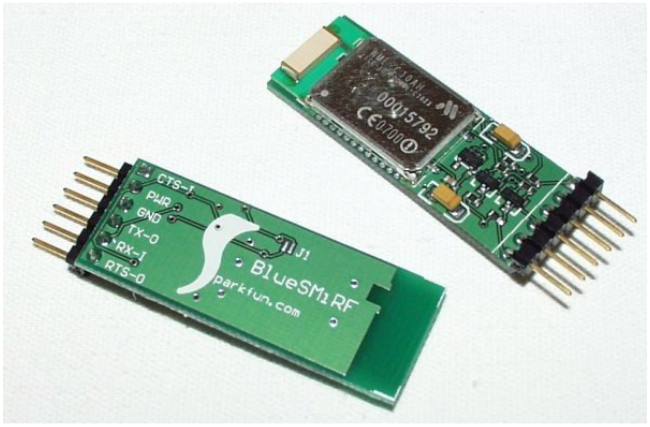
BlueSMiRF v1

Embedded Bluetooth Serial RF Link
5/3/2007



1 Overview

The BlueSMiRF provides embedded developers with an easy to use, easy to integrate serial RF channel over Bluetooth. It has only six electrical connections to deal with: 5V and ground, TX and RX, and CTS and RTS for optional hardware flow control. All communication lines are 5V tolerant (**the BlueSMiRF by itself is not an RS232 device**).

2 Specifications

Electrical specifications are as follows:

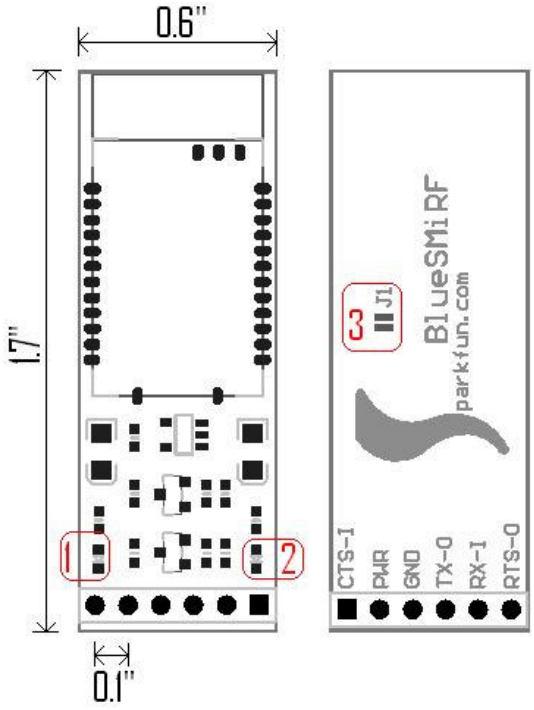
- Input Voltage 4.5V to 5.5V
- Current Consumption

  120mA Worst Case TX

  40mA RX

  2mA Idle
- +20dBm Maximum Output Power

There are only a few items on the unit that the user should be familiar with, as illustrated in the following diagram.



1) Status LED
2) Bluetooth Connection LED
3) CTS/RTS Solder Jumper

3 Operation

What follows is a short description of the operational parameters and AT command set to get the user up and running in the shortest time. This is by no means a full listing of the AT command set. For the full AT command set, please contact support@sparkfun.com.

Generically speaking, any Bluetooth device can communicate with any other Bluetooth device.

However, there are different Bluetooth profiles or "services" that may prevent certain types of connections. The BlueSMiRF is by default a serial port device, and it should be displayed as such when performing a discovery with other Bluetooth devices to find the BlueSMiRF. That being said, there are a wide variety of Bluetooth devices and manufacturers, and we cannot provide connection details for all of them. In the end, it will be up to the user to determine the compatibility of the BlueSMiRF with the user's other Bluetooth equipment.

3.1 Computer to BlueSMiRF Connection

In this section, we assume that the BlueSMiRF is integrated into an embedded device to which the user is trying to connect externally (i.e., you're trying to talk to a BlueSMiRF with a dongle from your computer).

In general, the procedure is to open up whatever Bluetooth software package you're using, do a discovery to find local devices, and connect to the BlueSMiRF as a serial device once it is found. Once the connection is established, the Bluetooth software of your choice will most likely indicate in some fashion as to which virtual serial port to which the BlueSMiRF is connected. Then you simply open up a terminal window to that port and you're "on the air".

Note: The BlueSMiRF has no active security in place by default (although it can be set up that way). But should the user's local Bluetooth device and/or software ask for it, the default PIN is "default".

Unfortunately, we are unable to go into more detail of how to navigate the myriad of third-party software packages used for Bluetooth communications. We don't mean to be cryptic here, but there are many different platforms from which the user may be trying to connect to the BlueSMiRF. Detailing them all goes well beyond the scope of this document. However, please visit our tutorials section at http://www.sparkfun.com/commerce/ for descriptions of connections from a few different Bluetooth devices.

3.2 Bluetooth Connection with the BlueSMiRF as Master

In this section, we will assume that the user has integrated a BlueSMiRF into some piece of hardware and is trying to establish a Bluetooth connection from the users hardware to some other device. However, this procedure is still valid with the BlueSMiRF connected to RS232 conversion circuitry and a serial cable, talking to it with a terminal program. Again, **the BlueSMiRF by itself is not an RS232 device.**

The default communication rate of the BlueSMiRF is 9600 baud, 8 data bits, one stop bit, no parity and hardware flow control. If you do not intend on using the flow control, you may just connect CTS and RTS together, or optionally close the solder jumper J1 on the back of the BlueSMiRF. Upon power up, the status LED will blink at a rate of 1Hz. The user may at this time verify communication with the BlueSMiRF by issuing "AT<cr>" (cr = carriage return). If all of the connections are correct and the command was received and understood, the BlueSMiRF will answer back with "OK<cr,lf> (lf = linefeed).

The first thing to do is to put the BlueSMiRF into idle mode with the command "ATUCL<cr>". The unit will answer back with "OK", and the status LED will stop blinking and go dark.

The user may then perform a discovery for other Bluetooth devices with the command "ATDI,x,yyyyyyyy<cr>", where "x" is the number of devices to locate within a 60 second time-out window and "yyyyyyyy" is an 8-digit, 16-bit class of device code (COD, these codes are published and maintained by the Bluetooth SIG). To search for all types of Bluetooth devices, insert "00000000" in the COD field. For example, if we enter "ATDI,4,00000000<cr>", the BlueSMiRF will first reply with "OK", then it will give a listing of up to 4 devices with any profile, giving both MAC address and COD of whatever is found. The BlueSMiRF will stop searching either when it finds 4 other Bluetooth devices, or at the end of the time-out window. The end of the search will be

indicated by another "OK".

If the target device's MAC address is known, the preceding command may be omitted.

To connect as master, the user will enter the command "ATDM,<target MAC address>, 1101<cr>", where "target MAC address" is the MAC address of the device to which you wish to connect, and "1101" identifies the connection type as serial. The BlueSMiRF will immediately reply with "OK", then will attempt the connection. If successful, the BlueSMiRF will reply with "CONNECT,<target MAC address>". If unsuccessful, it will reply with "NO ANSWER".

Assuming that the connection was successful, the BlueSMiRF will now be in data mode and will not answer to commands. However, if the user issues the escape sequence "+++<cr>" the BlueSMiRF will revert to command mode while maintaining the Bluetooth connection. This is handy for putting the device into fast data mode (turns off the command parser in the BlueSMiRF for the duration of the Bluetooth connection, necessary for high-speed communications.

To terminate the Bluetooth connection from the BlueSMiRF side of the link, the user must first enter the escape sequence "+++<cr>", wait for the ensuing "OK", and then enter "ATDH<cr>". When the connection terminates, the BlueSMiRF will reply with "OK", then "DISCONNECT".

If the user then wishes to reset the device to power up settings, they may send the command "ATURST<cr>". The BlueSMiRF will not reply to this command, but the status LED will again start blinking at 1Hz.

3.3 Changing the Baud Rate

Changing the baud rate on the BlueSMiRF can be a little confusing, so it bears some closer examination.

The first thing the user needs to know is that the baud rate setting has absolutely no effect on the communication speed over the RF link. It only affects the communication between the BlueSMiRF and its host hardware (which is to say the UART connection only). With that in mind, it is possible to have the BlueSMiRF set to a baud rate of 115200 along with a Bluetooth device on your computer that you're talking to at 9600 baud and have everything working just fine. The problem with this is that in this configuration the BlueSMiRF can send more data (by virtue of its increased baud rate with its supporting hardware) over the Bluetooth link than your receiving device can send out in the same amount of time. For this reason, it is advisable to have the same baud rates set at both ends of the link.

Changing the baud rate must be done in command mode, but it can be done from either the hardware side over the UART lines, or over the Bluetooth link by first putting the device into command mode with the escape sequence "+++<cr".

To change the baud rate over the UART lines, enter the command:

"ATSW20,
,<par>,<stop>,<save><cr>"

In this example, "par" is parity (0 = none, 1 = odd, 2 = even), "stop" is stop bits (0 = 1 stop bit, 1 = 2 stop bits), and "save" is save to memory (0 = don't save, 1 = save in memory). If you don't save the setting, it will revert to default values the next time the unit is powered up.

The values to enter for the baud rate are listed in the following table:

Baudrate	Ascii Value	Error
No Change	0	-
1200	5	1.73%
2400	10	1.73%
4800	20	1.73%
9600	39	-0.82%
19.2k	79	0.45%
38.4k	157	-0.18%
57.6k	236	0.03%
115.2k	472	0.03%
230.4k	944	0.03%
460.8k	1887	-0.02%
921.6k	3775	0.00%

As another example, if we wish to set the baud rate to 57600, no parity, 1 stop bit and save

to memory, the command would be:

“ATSW20,236,0,0,1<cr>“

The BlueSMiRF does not give any reply to this command. This is because it is assumed that the user must now change the baud rate settings of the host hardware that was used to change the BlueSMiRF’s baud rate in the first place. Once this is done, the user may query the BlueSMiRF with the command “ATSI,8<cr>“. If the change has taken effect, the BlueSMiRF will reply with “OK”, then “<baud>,<parity>,<stop>“ which will be in hex values. To our previous example, it would reply with “00EC,0000,0000”.

As stated earlier, it is possible to change the BlueSMiRF’s baud rate remotely over the Bluetooth link. To do so, the user must first enter the escape sequence “+++<cr>“ to put the BlueSMiRF into command mode. Then the ATSW20 command may be entered. The user may now immediately verify the change with ATSI,8 since it was the remote device that had it’s baud rate changed, not the local device. To get the BlueSMiRF back into data mode, the user may enter either “ATMF<cr>“ for fast data mode or “ATMD<cr>“ for normal data mode. The BlueSMiRF will answer back with “OK” in both cases, then revert to the data mode of choice.