



Application Note: **Communicating with the RM3000 Eval Board using the Bus Pirate**

Introduction

The Bus Pirate is an open-source, universal bus interface that enables communication between a computer's USB port and the SPI interface of PNI's RM3000 Evaluation Board. Bus Pirate design files can be found at <http://code.google.com/p/the-bus-pirate/>, and fully assembled and tested boards are available from several companies, including Sparkfun, Adafruit, and Dangerous Prototypes. PNI's testing was with a Bus Pirate board and cable procured from Sparkfun. RealTerm software was used as the terminal interface, and this can be downloaded at <http://sourceforge.net/projects/realterm/>.

Getting Started

1. Make connections from the RM3000 Evaluation Board to the ribbon cable, per the following table.

Function	RM3000 Evaluation Board Pin #	Ribbon Cable Color
SCLK	1	Yellow
MSIO	2	Brown
MOSI	3	Orange
SSNOT	4	Red

Function	RM3000 Evaluation Board Pin #	Ribbon Cable Color
CLEAR	6	Green
GND	7	Black
3.3 VDC	12	White

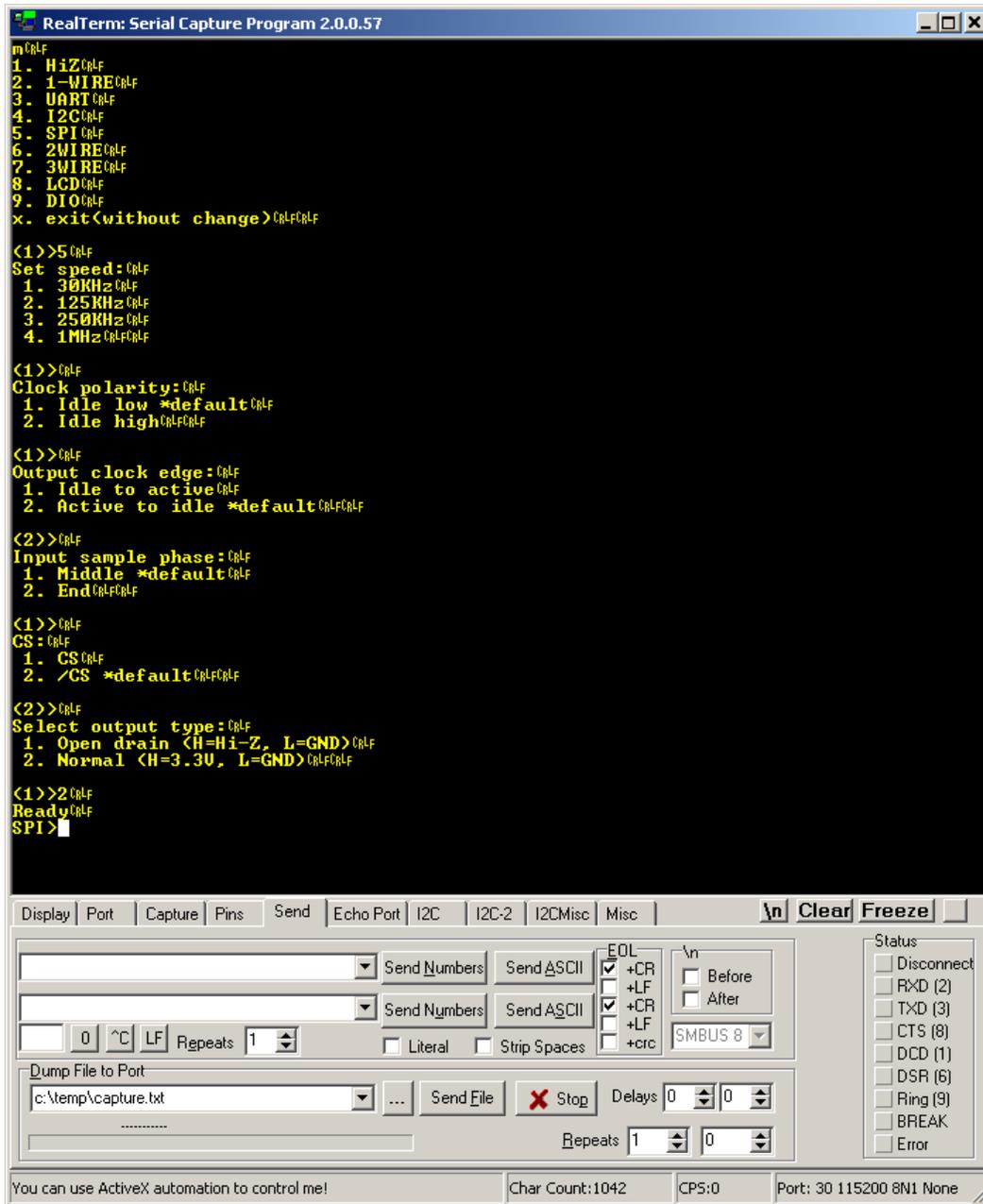
NOTE: The DRDY line is not used since running RealTerm essentially is a manual operation which is too slow to take advantage of the DRDY line, and because the Bus Pirate's input line can cause noise spikes on the SPI bus. Rather, fixed time delays are implemented. For integrated systems, the preferred implementation is to use the DRDY line as it allows for hardware handshaking.

2. Ensure the Bus Pirate is plugged into the USB port before launching RealTerm.
3. Launch RealTerm.

4. Configure RealTerm as follows:
 - a. Display Tab – If desired, increase the number of rows for a bigger window. Everything else should remain as the default setting.
 - b. Port Tab – Set the Baud rate to 115.2.k. Select the appropriate Port. Click on the Change button.
 - c. Echo Port Tab – Set the Baud rate to 115.2.k. Select the appropriate Port. Click on the Change button.
 - d. Send Tab – In the EOL box, check both “+CR” boxes. Communication with the Bus Pirate will be through the Send tab.

5. Configure the Bus Pirate as follows:
 - a. On the Send tab Click in RealTerm, click in the window area.
 - b. Type: m <CR> to bring up the menu. The menu consists of 6 selections to be made; each presented one at a time.
 - First selection: 5 <CR> - for SPI
 - Second selection: customer choice - choose appropriate SPI clock speed
 - Third selection: 1 <CR> or simply <CR> - for Idle low
 - Fourth selection: 2 <CR> or simply <CR> - for Active to Idle
 - Fifth selection: 1 <CR> or simply <CR> - for Middle
 - Sixth selection: 2 <CR> or simply <CR>
 - Seventh selection: 2 <CR> - for Normal

The screen shot below shows what the screen should look like after completing these selections. Note that for several of the selections no number was typed in, as the appropriate value was also the default value.



- Power on the RM3000 Evaluation Board by typing “W” <CR>. You can use the SPI> prompt or the boxes on the Send tab and Send ASCII button.

Communication

When communicating with the RM3000 Evaluation Board via the Bus Pirate, please keep the following in mind:

- Communication either can be a single character at a time or a string of characters in RealTerm’s interactive window.

- The Bus Pirate has two modes for controlling the slave select. The “[” character brings the slave select line LOW (slave select enabled) and ignores the return byte. The “{” character brings the slave select line LOW (slave select enabled) and reads the return byte. By reading back the return byte the user can verify that the sent command was received properly, and for this reason the examples will use the “{” character.
- The Bus Pirate runs slow enough that no commanded delays are required for SPI purposes. However, delays are required to take sensor measurements.

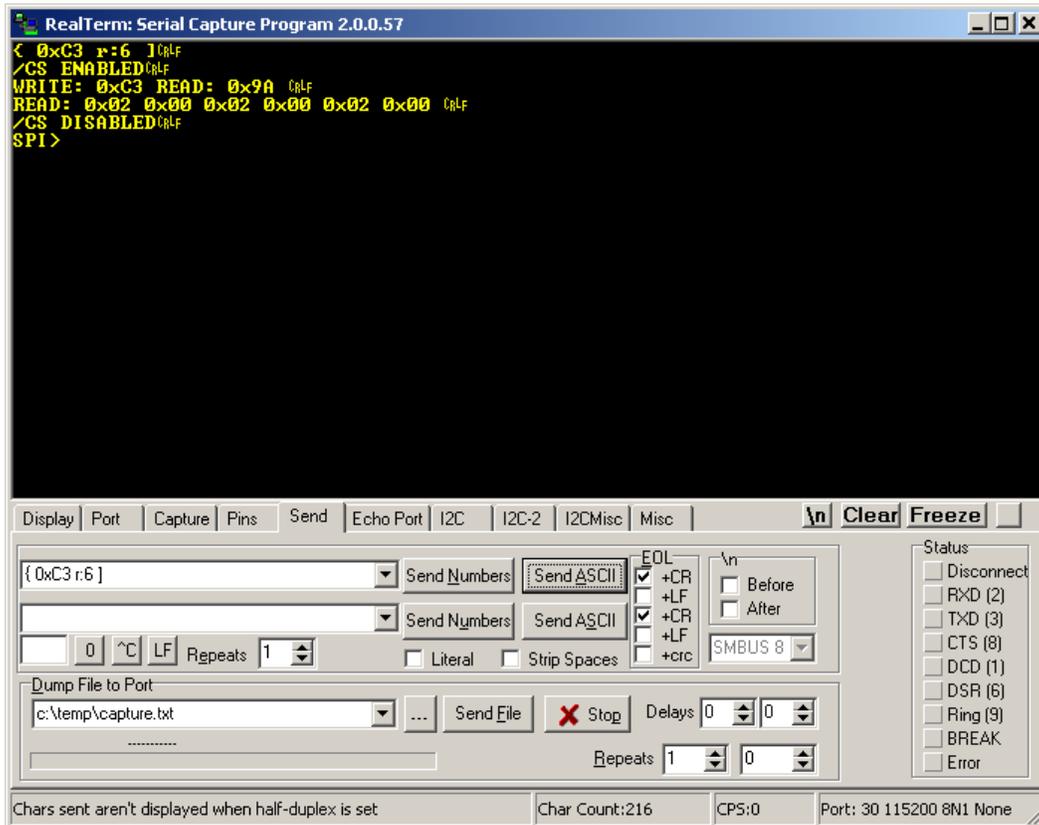
PNI recommends referring to the Bus Pirate documentation for a full list of commands.

Setting the Cycle Count Register

To read the Cycle Count Register the following string is sent:

```
{ 0xC3 r:6 ]
```

and the following will be displayed in the RealTerm Window:

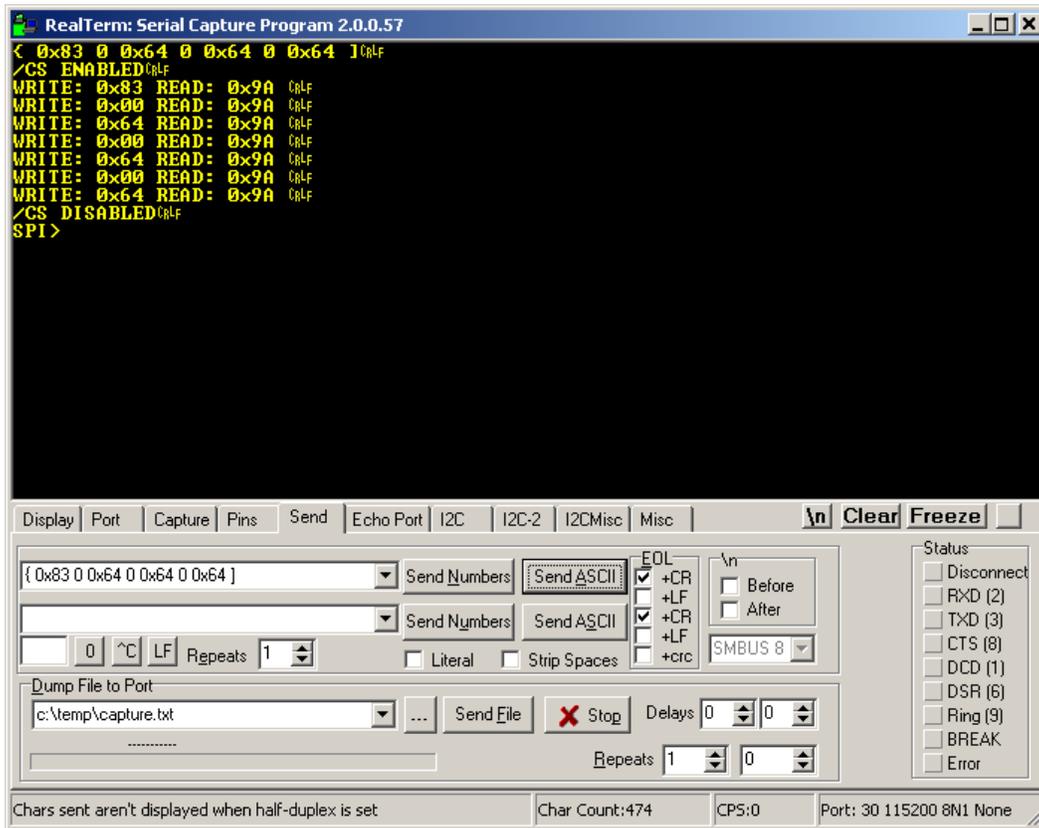


This shows the cycle count register is still set at the default value of “0x02 0x00”, or 512 decimal.

To change the cycle count to 100_d for example the string would be:

```
{ 0x83 0 0x64 0 0x64 0 0x64 }
```

This would result in the following display:

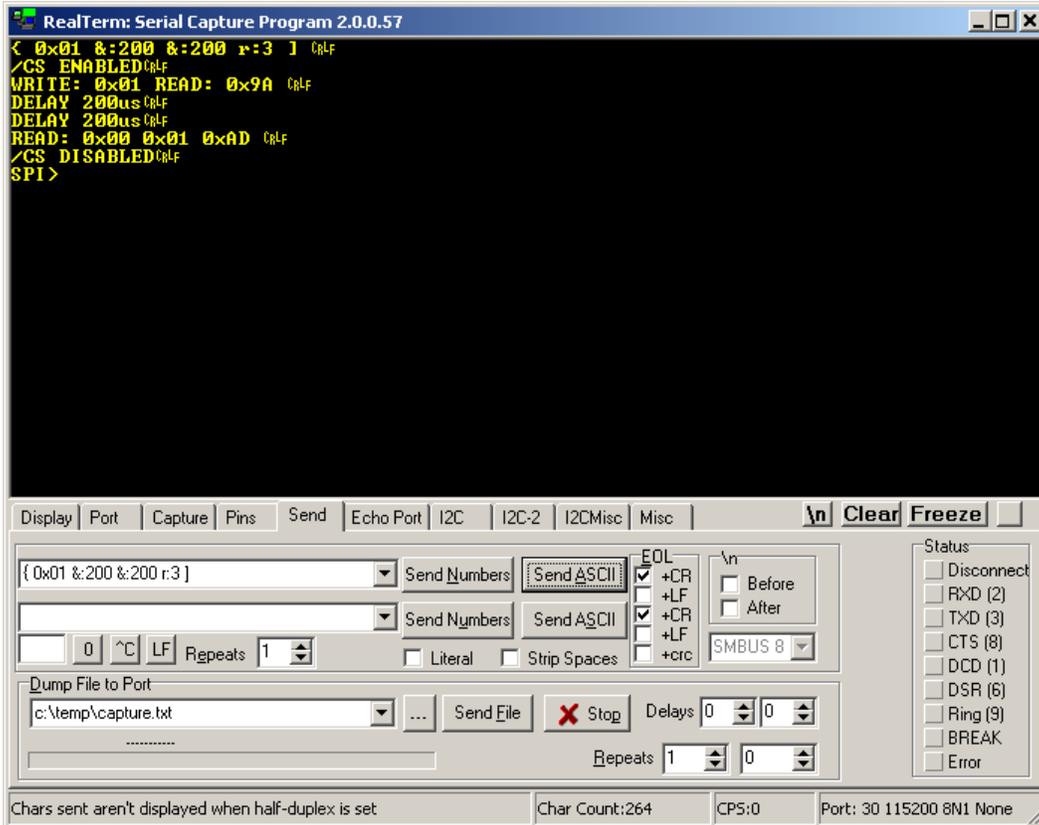


Using the Delay Command

To make sensor measurements, it is necessary to implement a delay between initiating a measurement and reading back the results, as it takes some finite time to actually take the measurements. The “&” character is the Bus Pirate delay command and is for 1 µsec. Placing a colon and a number after the “&” character results in multiplying the delay by the value of the number. The number can be up to 255, such that a maximum delay of 255 µsec is obtained with the “&:255” delay command. To get delays longer than 255 µsec, multiple delay commands must be sent. However, when sending multiple delay commands the total delay is greater than the sum of the individual delays. For instance, sending:

```
{ 0x01 &:200 &:200 r:3 }
```

seems to imply a delay of 400 μ sec between the send command (0x01) and the read command (r:3), but the actual delay is more like 1 msec.



Single Axis Measurement (SAM) operation:

To initiate a single-axis measurement, send the following commands

```
{ 0x01 &:200 &:200 r:3 } Take X-axis reading
{ 0x02 &:200 &:200 r:3 } Take Y-axis reading
{ 0x03 &:200 &:200 r:3 } Take Z-axis reading
```

The actual required delay will depend on the Cycle Count Register setting.

Multi-Axis Measurement (MAM) operation:

To initiate a multi-axis measurement (for all 3 sensors), send the following command

```
{ 0x82 0x01 } &:200 &:200 &:200 &:200 { 0xC9 r:9 }
```

The actual required delay will depend on the Cycle Count Register setting. A screen shot for sending this command the subsequent activity is shown below.

