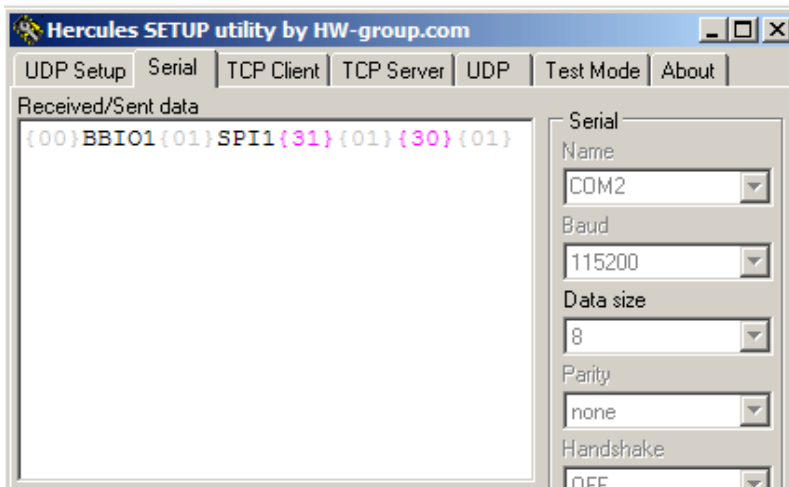


Bitbang



There's two new binary I/O libraries in the v2.3 Bus Pirate firmware. Raw bitbang mode provides direct control over the Bus Pirate pins and hardware using a simple single-byte protocol.

Hopefully the raw access modes inspire some interesting new Bus Pirate tools using PC software. There's already [a patch for AVRdude](#) AVR programmer in the works.

[Example scripts in Python, Perl, etc.](#)

Contents [\[hide\]](#)

1 Overview

2 Commands

- 2.1 00000000 - Reset, responds "BBIO1"
- 2.2 00000001 - Enter binary SPI mode, responds "SPI1"
- 2.3 00000010 - Enter binary I2C mode, responds "I2C1"
- 2.4 00000011 - Enter binary UART mode, responds "ART1"
- 2.5 00000100 - Enter binary 1-Wire mode, responds "1W01"
- 2.6 00000101 - Enter binary raw-wire mode, responds "RAW1"
- 2.7 00000110 - Enter OpenOCD JTAG mode
- 2.8 0000xxxx - Reserved for future raw protocol modes
- 2.9 00001111 - Reset Bus Pirate
- 2.10 0001000x - Bus Pirate self-tests
- 2.11 00010010 - Setup pulse-width modulation (requires 5 byte setup)
- 2.12 00010011 - Clear/disable PWM
- 2.13 00010100 - Take voltage probe measurement (returns 2 bytes)
- 2.14 00010101 - Continuous voltage probe measurement
- 2.15 00010110 - Frequency measurement on AUX pin
- 2.16 010xxxxx - Configure pins as input(1) or output(0): AUX|MOSI|CLK|MISO|CS
- 2.17 1xxxxxxx - Set on (1) or off (0): POWER|PULLUP|AUX|MOSI|CLK|MISO|CS

3 PWM Computation

4 Updates

Overview

The bitbang protocol uses a single byte for all commands. The default start-up state is pin input (HiZ).

Send 0x00 to the user terminal (max.) **20 times** to enter the raw binary bitbang mode. Check in between if **BBIO1** is returned. [Example binary mode entry functions.](#)

Send 0x0F to exit raw bitbang mode and reset the Bus Pirate.

Other raw protocol modes are accessible from within bitbang mode, 0x00 always returns to raw bitbang mode and prints the version string.

There is a slight settling delay between pin updates, currently about 5us.

Commands

00000000 - Reset, responds "BBIO1"

This command resets the Bus Pirate into raw bitbang mode from the user terminal. It also resets to raw bitbang mode from raw SPI mode, or any other protocol mode. This command always returns a five byte bitbang version string "BBIOx", where x is the current protocol version (currently 1).

Some terminals send a NULL character (0x00) on start-up, causing the Bus Pirate to enter binary mode when it wasn't wanted. To get around this, you must now enter 0x00 **at least 20 times** to enter raw bitbang mode.

Note: The Bus Pirate user terminal could be stuck in a configuration menu when your program attempts to enter binary mode. One way to ensure that you're at the command line is to send <enter> at least 10 times, and then send '#' to reset. Next, send 0x00 to the command line 20+ times until you get the BBIOx version string.

After entering bitbang mode, you can enter other binary protocol modes.

00000001 - Enter binary SPI mode, responds "SPI1"

[Binary SPI mode is documented here](#)

00000010 - Enter binary I2C mode, responds "I2C1"

[Binary I2C mode is documented here.](#)

00000011 - Enter binary UART mode, responds "ART1"

[Binary UART mode is documented here.](#)

00000100 - Enter binary 1-Wire mode, responds "1W01"

[Binary 1-Wire mode is documented here.](#)

00000101 - Enter binary raw-wire mode, responds "RAW1"

[Binary raw-wire mode is documented here.](#)

00000110 - Enter OpenOCD JTAG mode

OpenOCD mode is documented in the source only.

0000xxxx - Reserved for future raw protocol modes

00001111 - Reset Bus Pirate

The Bus Pirate responds 0x01 and then performs a complete hardware reset. The hardware and firmware version is printed (same as the 'i' command in the terminal), and the Bus Pirate returns to the user terminal interface. Send 0x00 20 times to enter binary mode again.

Note: there may be garbage data between the 0x01 reply and the version information as the PIC UART initializes.

0001000x - Bus Pirate self-tests

[Binary self tests are documented here.](#) Available only in v2go and v3 hardware.

00010010 - Setup pulse-width modulation (requires 5 byte setup)

Configure and enable pulse-width modulation output in the AUX pin. Requires a 5 byte configuration sequence. Responds 0x01 after a *complete sequence* is received. The PWM remains active after leaving binary bitbang mode!

Equations to calculate the PWM frequency and period are in the [PIC24F output compare manual](#). Bit 0 and 1 of the first configuration byte set the prescaler value. The Next two bytes set the duty cycle register, high 8bits first. The final two bytes set the period register, high 8bits first.

00010011 - Clear/disable PWM

Clears the PWM, disables PWM output. Responds 0x01.

00010100 - Take voltage probe measurement (returns 2 bytes)

Take a measurement from the Bus Pirate voltage probe. Returns a 2 byte ADC reading, high 8bits come first. To determine the actual voltage measurement: $(ADC/1024)*3.3volts*2$; or simply $(ADC/1024)*6.6$.

00010101 - Continuous voltage probe measurement

Sends ADC data (2bytes, high 8 first) as fast as UART will allow. A new reading is not taken until the previous finishes transmitting to the PC, this prevents time distortion from the buffer. Added for the [oscilloscope script](#).

00010110 - Frequency measurement on AUX pin

Takes frequency measurement on AUX pin. Returns 4byte frequency count, most significant byte first.

010xxxxx - Configure pins as input(1) or output(0): AUX|MOSI|CLK|MISO|CS

Configure pins as an input (1) or output(0). The pins are mapped to the lower five bits in this order:

AUX|MOSI|CLK|MISO|CS.

The Bus pirate responds to each direction update with a byte showing the current state of the pins, regardless of direction. This is useful for open collector I/O modes.

1xxxxxxx - Set on (1) or off (0): POWER|PULLUP|AUX|MOSI|CLK|MISO|CS

The lower 7bits of the command byte control the Bus Pirate pins and peripherals. Bitbang works like a [player piano](#) or [bitmap](#). The Bus Pirate pins map to the bits in the command byte as follows:

1|POWER|PULLUP|AUX|MOSI|CLK|MISO|CS

The Bus pirate responds to each update with a byte in the same format that shows the current state of the pins.

PWM Computation

Here is a [Python Script](#) for computing the PWM for Bus Pirate.

For example, setup PWM with Period of 1msec, 50% duty cycle. Using 1:1 Prescaler.

Modify only the 3 lines:

```
Prescaler=1 # 1:1
PwmPeriod=1e-3 # 0.1msec
DutyCycleInPercent=.5 # 50%
```

It will output:

```
=====
PwmPeriod: 0.0010000000000000000 sec.
Tcy: 0.00000006250000000000 sec.
Prescaler: 1
PR2: 0x3e7f = 15999
OCR: 0x1f3f = 7999
5 Byte Setup:
1st Byte: 0x00
2nd Byte: 0x1f
3rd Byte: 0x3f
4th Byte: 0x3e
5th Byte: 0x7f
```

Use the 5byte setup to set the PWM.