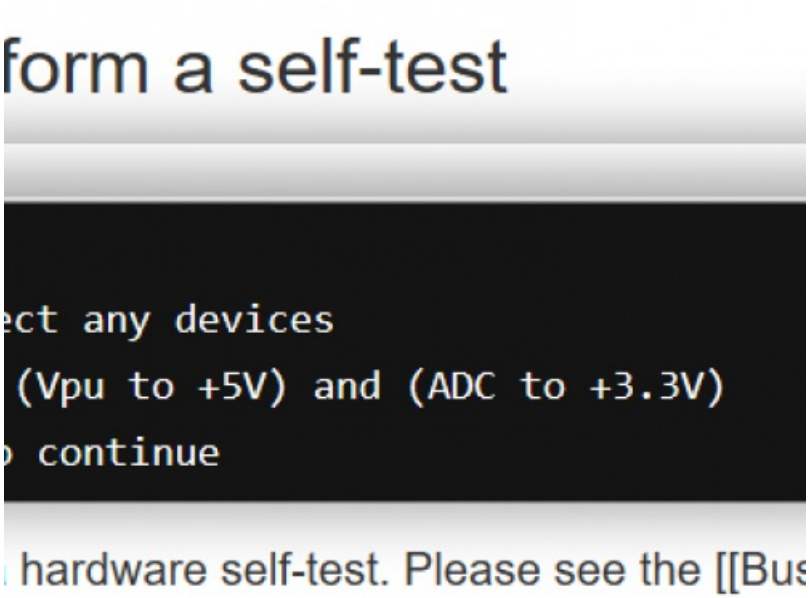

Bus Pirate Command Guide

Bus Pirate v3b Firmware v5.10

http://buspirate.com/tutorial/bus-pirate-command-guide

Generated 2019-01-23

Table of Contents

Introduction

Menu options are single character commands that configure the Bus Pirate. Type a letter and then the **Enter** key to access a menu. Some options are unavailable in some modes and on some hardware.

A simple syntax is used to interact with chips. Syntax characters have the same general function in each bus mode, such as 'R' to read a byte of data.

Up to 256 characters of menu and syntax stuff can be entered into the Bus Pirate terminal at once, press enter to execute the syntax.

Menu options overview

Menu options are single character commands that configure the Bus Pirate. Press enter to show the command prompt if your terminal is blank.

```
HiZ>p
Command not used in this mode
HiZ>
```

Enter a command, followed by the **Enter** key, to access the menu. Most configuration and option prompts have a default value shown in (). Press enter to select the default option. Some options are unavailable in some modes, for example bit order configuration and pull-up resistors.

Basic commands

? Help menu with latest menu and syntax options

```
HiZ>?
General     Protocol interaction
---------------------------------------------------------------------------
? This help   (0) List current macros
=X/|X Converts X/reverse X  (x) Macro x
~ Selftest   [ Start
# Reset    ] Stop
$ Jump to bootloader  { Start with read
&/% Delay 1 us/ms   } Stop
a/A/@ AUXPIN (low/HI/READ)  "abc" Send string
b Set baudrate   123
c/C AUX assignment (aux/CS)  0x123
d/D Measure ADC (once/CONT.) 0b110 Send value
f Measure frequency  r Read
g/S Generate PWM/Servo  / CLK hi
h Commandhistory   \ CLK lo
i Versioninfo/statusinfo  ^ CLK tick
l/L Bitorder (msb/LSB)  - DAT hi
m Change mode   _ DAT lo
o Set output type   . DAT read
p/P Pullup resistors (off/ON) ! Bit read
s Script engine   : Repeat e.g. r:10
v Show volts/states  . Bits to read/write e.g. 0x55.2
w/W PSU (off/ON)  <x>/<x= >/<0> Usermacro x/assign x/list all
HiZ>
```

Print a help screen with all available menu and syntax options in the current firmware and hardware.

i Hardware, firmware, microcontroller version information

```
HiZ>i
Bus Pirate v3b
Firmware v5.10 (r559)  Bootloader v4.4
DEVID:0x0447 REVID:0x3046 (24FJ64GA002 B8)
http://dangerousprototypes.com
HiZ>
```

The information menu displays the hardware, firmware, and microcontroller version. If a bus mode is configured additional information about the configuration options is printed.

~ Perform a self-test

```
HiZ>~
Disconnect any devices
Connect (Vpu to +5V) and (ADC to +3.3V)
Space to continue
```

Perform a hardware self-test. Please see the [[Bus Pirate self-test guide]].

m Set bus mode (1-Wire, SPI, I2C, JTAG, UART, etc)

```
HiZ>m
1. HiZ
2. 1-WIRE
3. UART
4. I2C
5. SPI
6. 2WIRE
7. 3WIRE
8. LCD
9. DIO
x. exit(without change)

(1)>
```

Select a bus mode. The command resets the Bus Pirate and immediately disables all pins, pull-up resistors, and power supplies.

The default mode is HiZ, a safe mode with all pins set to high-impedance and all peripherals disabled.

o Data display format (DEC, HEX, BIN, or raw)

```
HiZ>o
 1. HEX
 2. DEC
 3. BIN
 4. RAW

(1)>1
Display format set
HiZ>
```

The Bus Pirate can display values as hexadecimal (http://en.wikipedia.org/wiki/Hexadecimal), decimal (http://en.wikipedia.org/wiki/Decimal), binary (http://en.wikipedia.org/wiki/Binary_numeral_system), and a raw ASCII (http://en.wikipedia.org/wiki/ASCII) byte. Change the setting in the data display format menu (o). The default display format is HEX.

The RAW display mode sends values to the terminal as raw byte values without any text conversion. This is useful for ASCII serial interfaces. It can also be used to speed up the display of bus sniffers and other high-speed functions where converting raw bytes to text takes too much time. Adjust the display format in your serial terminal to see the raw values as HEX/DEC/BIN.

Utilities

w/W Power supplies (off/ON)

```
1-WIRE>W
Power supplies ON
1-WIRE>w
Power supplies OFF
1-WIRE>
```

Toggle the switchable 3.3volt and 5.0volt power supplies with the w/W command. Capital 'W' enables the supplies, lowercase 'w' disables them. The power supplies on the Bus Pirate v2go and v3 can supply up to 150mA. The current configuration is displayed on the extended information screen.

```
1-WIRE>v
Pinstates:
1.(BR) 2.(RD) 3.(OR) 4.(YW) 5.(GN) 6.(BL) 7.(PU) 8.(GR) 9.(WT) 0.(Blk)
GND 3.3V 5.0V ADC VPU AUX - OWD - -
P P P I I I I I I
GND 3.28V 5.00V 3.29V 5.00V L L L L L
1-WIRE>
```

Use the power supply voltage report 'v' to see the current voltage at each power supply. The 5.0volt supply is powered by the 5.0volt USB supply, so it's normal for it to fall below 5.0volts under load.

v Power supply voltage report


```
1-WIRE>v
Pinstates:
1.(BR) 2.(RD) 3.(OR) 4.(YW) 5.(GN) 6.(BL) 7.(PU) 8.(GR) 9.(WT) 0.(Blk)
GND 3.3V 5.0V ADC VPU AUX - OWD - -
P P P I I I I I I
GND 3.28V 5.00V 3.29V 5.00V L L L L L
1-WIRE>
```


The voltage report shows the current state of all the Bus Pirate pins and peripherals.

The first line is the pin number, according to the silk screen on the v3 PCB, and the "'Seeed Studio'" probe wire color. "The ADC and 3.3V pins are swapped on the v2go and the display is incorrect."

The second line is the pin function in the current bus mode. The power supplies (3.3v, 5.0v), ADC, Vpu, and AUX pins are available in all modes. The other four pins will differ depending on the mode. In 1-Wire mode only one pin is used, one wire data (OWD).

The third line shows the current direction of each pin. I is an input pin, O is an output pin, P is a power supply.

The fourth line shows the current state of each pin. A voltage measurement is displayed for analog pins. The current pin reading, H high and L low, is printed for each digital pins.

p/P Pull-up resistors

```
1-WIRE>P
Pull-up resistors ON
1-WIRE>p
Pull-up resistors OFF
1-WIRE>
```

p and P toggle the pull-up resistors off and on.

The on-board pull-up resistors "'must be powered through the Vpullup pin of the IO header'". A warning is displayed if there's no voltage on the Vpullup pin (v5.2+). Check the voltage report (V) and verify that Vpu is attached to a power supply. See the [[Practical_guide_to_Bus_Pirate_pull-up_resistors|practical guide to Bus Pirate pull-up resistors]] for a simple introduction.

```
(1)>1
Select output type:
 1. Open drain (H=Hi-Z, L=GND)
 2. Normal (H=3.3V, L=GND)

(1)>2
Ready
2WIRE>P
WARNING: pins not open drain (HiZ)
Pull-up resistors ON
Warning: no voltage on Vpullup pin
2WIRE>
```

Pull-up resistors are generally used with open collector/open drain bus types. A warning is displayed when the pull-ups are enabled if the Bus Pirate is configured for normal pin output.

The current configuration is displayed on the extended information screen 'i'.

f Measure frequency on the AUX pin

```
1-WIRE>f
AUX Frequency:  autorange  autorange 0 Hz
1-WIRE>
```

Measures frequency from 0Hz to 40MHz on the AUX pin, the method is an actual 1 second tick count. If the frequency is lower than a few MHz, the Bus Pirate does an 'autorange' and measures the frequency again for an additional second.

g Frequency generator/PWM on the AUX pin

```
1-WIRE>g
1KHz-4,000KHz PWM
Frequency in KHz
(50)>2000
Duty cycle in %
(50)>
PWM active
1-WIRE>g
PWM disabled
1-WIRE>
```

Enable the frequency generator with g, then set frequency and duty cycle. Frequencies from 1kHz to 4MHz are possible. Use g again to disable the PWM.

Note that the resolution at 4MHz is only 1 bit. Anything other than 50% duty cycle will be 100% off or 100% on.

S Servo

```
1-WIRE>S
Position in degrees

(90)>20
Servo active
1-WIRE>S
PWM disabled
1-WIRE>
```

S positions the servo arm to the desired angle, 0-180 degrees. The servo value can be updated as needed, press enter or x to exit. Use 'S' or 'g' again to '"disable"' the servo.

```
1-WIRE>S 90 %:500 S 180
Servo active
DELAY 500ms
PWM disabled
Servo active
1-WIRE>
```

Example of multiple servo positions with delay.

*[[Bus Pirate servo driver documentation]]

NOTE1: Most servos draw more current than the Bus Pirate can supply!! Use an external power supply instead.

=X Convert X to HEX/DEC/BIN number format

```
1-WIRE>=0b110
0x06 = 6 = 0b00000110
1-WIRE>=0xa
0x0A = 10 = 0b00001010
1-WIRE>=12
0x0C = 12 = 0b00001100
1-WIRE>
```

Base conversion command, available in all modes. Press '=' and enter any byte value to see the HEX/DEC/BIN equivalent.

To change the Bus Pirate output display format see the 'o' command.

|X Reverse bits in byte X

```
1-WIRE>|0b10101010
0x55 = 85 = 0b01010101
1-WIRE>|0b10000000
0x01 = 1 = 0b00000001
1-WIRE>|1
0x80 = 128 = 0b10000000
1-WIRE>
```

Reverse bit order in byte X. Displays the HEX/DEC/BIN value of the reversed byte.

To change the Bus Pirate read/write bit order see the l/L command.

s BASIC script engine

```
1-WIRE>s
1-WIRE(BASIC)>
```

Simple BASIC scripts can automate repetitive and tedious tasks. [[Bus Pirate BASIC script reference]]

d/D Measure from voltage probe (once/CONTINUOUS)

```
1-WIRE>d
VOLTAGE PROBE: 0.00V
1-WIRE>
```

'd' takes a single measurement from the voltage measurement probe (ADC pin on the IO header). 'D' takes continuous measurements from the voltage probe, press any key to exit.

The Bus Pirate voltage probe can measure up to 6.0volts (max 6.6volts, but with some margin for error).

a/A/@ Control auxiliary pin (low/HIGH/read)

```
1-WIRE>A
AUX HIGH
1-WIRE>a
AUX LOW
1-WIRE>@
AUX INPUT/HI-Z, READ: 0
1-WIRE>
```

The auxiliary pin is a general purpose digital pin that can be controlled from the Bus Pirate terminal. 'A' makes AUX a 3.3volt output (25mA max). 'a' makes AUX sink to ground (25mA max). '@' makes AUX an input and reads the current state (5volt maximum input).

a/A/@ can also be used to control the CS pin using the c/C commands.

c/C Toggle AUX control between AUX and CS/TMS pins

```
1-WIRE>c
a/A/@ controls AUX pin
1-WIRE>C
a/A/@ controls CS pin
1-WIRE>
```

Sometimes it's useful to control the CS pin from the user terminal. The c/C configures the a/A/@ commands to control the AUX or CS pin.

The current AUX pin configuration is displayed on the extended information screen 'i'.

Bus interaction commands

These commands actually manipulate the bus and interacts with chips. These commands have the same general function in each bus mode, such as 'R' to read a byte of data. See the individual bus mode guides for each protocol.

{ or [Bus start condition.

```
I2C>[
Warning: *Short or no pull-up
I2C START BIT
I2C>
```

This command generally starts bus activity. In various modes it starts (I2C), selects (SPI), resets (1-wire), or opens (UART).

] or } Bus stop condition.

```
SPI>]
/CS DISABLED
SPI>
```

This command generally stops bus activity. In various modes it stops (I2C), deselects (SPI), or closes (UART).

r Read byte

```
I2C>r
READ: 0x00
I2C>r:3
READ:  ACK 0x00  ACK 0x00  ACK 0x00
I2C>
```

r reads a byte from the bus. Use with the repeat command (r:1…255) for bulk reads.

0b01 Write this binary value

```
I2C>0b1001
WRITE: 0x09 ACK  ACK
I2C>0b1001:2
WRITE: 0x09 ACK 0x09 ACK
I2C>
```

Enter a binary value to write it to the bus.

Binary (http://en.wikipedia.org/wiki/Binary_numeral_system) values are commonly used in electronics because the 1's and 0's correspond to register 'switches' that control various aspects of a device. Enter a binary number as 0b and then the bits. Padding 0's are not required, 0b00000001=0b1. Can be used with the repeat command.

0x01 Write this HEX value

```
SPI>0x15
WRITE: 0x15
SPI>0x15:5
WRITE: 0x15 0x15 0x15 0x15 0x15
SPI>
```

Enter a HEX value to write it to the bus.

Hexadecimal (http://en.wikipedia.org/wiki/Hexadecimal) values are base 16 numbers that use a-f for the numbers 10-15, this format is very common in computers and electronics. Enter HEX values as shown above, precede the value with 0x or 0h. Single digit numbers don't need 0 padding, 0x01 and 0x1 are interpreted the same. A-F can be lower-case or capital letters.

0-255 Write this decimal value

```
SPI>18
WRITE: 0x12
SPI>13:5
WRITE: 0x0D 0x0D 0x0D 0x0D 0x0D
SPI>
```

Any number not preceded by 0x, 0h, or 0b is interpreted as a decimal value and sent to the bus.

Decimal (http://en.wikipedia.org/wiki/Decimal) values are common base 10 numbers. Just enter the value, no special designator is required.

"abc" Write this ASCII string

```
SPI>"abc"
WRITE: "abc"
SPI>
```

The ASCII values enclosed in "" are sent to the bus.

space/, Value delimiter

```
SPI>[1 2,3rr]
/CS ENABLED
WRITE: 0x01
WRITE: 0x02
WRITE: 0x03
READ: 0x00
READ: 0x00
/CS DISABLED
SPI>
```

Use a coma or space to separate numbers. Any combination is fine, no delimiter is required between non-number values.

&/% Delay 1uS/MS

```
SPI>&
DELAY 1us
SPI>&:10
DELAY 10us
SPI>%
DELAY 1ms
SPI>%:7
DELAY 7ms
SPI>
```

& delays 1us, % delays 1ms. Use the repeat command for multiple delays.

: Repeat (e.g. r:10)

```
SPI>&:10
DELAY 10us
SPI>r:0b10
READ: 0x00 0x00
SPI>5:0x03
WRITE: 0x05 0x05 0x05
SPI>
```

Many Bus Pirate commands can be repeated by adding ': ' to a command, followed by the number of times to repeat the command. To read five byte, enter r:5, etc. The repeat values can be HEX/DEC/BIN.

; Partial (<16 bit) read/write (e.g. 0x55;3)

```
2WIRE>0xaa;4
WRITE: 0x0A;4
2WIRE>
```

Will write 0x0a (4 bits) to the bus.

```
2WIRE>0xFFFF;12
WRITE: 0x0FFF;12
2WIRE>
```

Will write 0x0FFF (12 bits) to the bus.

```
2WIRE>0x55:4;2
WRITE: 0x01;2 0x01;2 0x01;2 0x01;2
2WIRE>
```

Can be combined with the repeat command.

NOTE: works currently only with the raw2wire and raw3wire bus.

l/L Set MSB/LSB first in applicable modes

```
2WIRE>l
MSB set: MOST sig bit first
2WIRE>L
LSB set: LEAST sig bit first
2WIRE>
```

The l/L command determines the bit order (http://en.wikipedia.org/wiki/Most_significant_bit) for reading and writing bytes in some bus modes. The bitorder command is available in all modes.

The current bit order configuration is displayed on the extended information screen 'i'.

Bitwise bus commands

Bitwise commands are only available in certain bus modes.

^ Send one clock tick

```
2WIRE>^
CLOCK TICKS: 0x01
2WIRE>
```

Send one clock tick. ^:1…255 for multiple clock ticks.

/ or \ Toggle clock level high (/) and low (\)

```
2WIRE>/\
CLOCK, 1
CLOCK, 0
2WIRE>
```

Set the clock signal high or low. Includes clock delay.

- or _ Toggle data state high (-) and low (_)

```
2WIRE>-_
DATA OUTPUT, 1
DATA OUTPUT, 0
2WIRE>
```

Set the data signal high or low. Includes data setup delay.

! Read one bit with clock

```
2WIRE>!
READ BIT: 0  *pin is now HiZ
2WIRE>
```

Send one clock tick and read one bit from the bus.

On a bus with a bi-directional data line (raw2wire, 1-Wire), the data pin is left as a high-impedance input after this command.

. Read data pin state (no clock)

```
2WIRE>.
DATA STATE: 02WIRE>
```

Make the data pin an input and read, but do not send a clock. This can be used as /.\ to achieve the same thing as the ! command.

On a bus with a bi-directional data line (raw2wire, 1-wire), the data pin is left as a high-impedance input after this command.

Macros, user macros

Macros perform complex actions, like scanning for I2C addresses, interrogating a smart card, or probing a JTAG chain. Macros are numbers entered inside (). Macro (0) always displays a list of macros available in the current bus mode.

(0) List mode macros

```
I2C>(0)
 0.Macro menu
 1.7bit address search
 2.I2C sniffer
I2C>
```

Macro (0) always displays a list of macros available in the current bus mode.

(#) Run macro

```
I2C>(1)
Searching I2C address space. Found devices at:
Warning: *Short or no pull-up

I2C>
```

Execute a macro by typing the macro number between ().

Assign user macro

```
I2C><1=[0xa1 r:8]>
I2C>
```

5 user macros can be stored to automate common commands. Each position can store 32 chars (including space).

<0> List user macros

```
I2C><0>
1. <[0xa1 r:8]>
2. <>
3. <>
4. <>
5. <>
I2C>
```

User macro <0> lists the currently stored use macros.

<#> Run user macro

```
I2C><1>

I2C>[0xa1 r:8]
```

Enter the macro number to recall the command. Press enter to execute.

System commands

B Set PC side serial port speed

```
I2C>b
Set serial port speed: (bps)
 1. 300
 2. 1200
 3. 2400
 4. 4800
 5. 9600
 6. 19200
 7. 38400
 8. 57600
 9. 115200
10. BRG raw value

(9)>9
Adjust your terminal
Space to continue
Space to continueI2C>
I2C>
```

Adjust the speed of the serial port facing the computer (and USB->serial converter chip).

After choosing a speed you must adjust your serial terminal speed and press space to continue. The Bus Pirate will pause until the space key is pressed to verify that the terminal speed is correct.

```
(9)>10
Enter raw value for BRG

(34)>34
Adjust your terminal
Space to continue
Space to continueI2C>
I2C>
```

A custom baud rate can be set with a raw BRG value. The value can be calculated according to the datasheet or with a utility (http://www.micromagicsystems.com/#/pic-baud/4523812801) (key constants: PIC24, 32MHz/16MIPS, BRGH=1).

- 230400 baud is '16' (2.2% error)
- 460800 baud is '8' (3.3% error)
- 921600 baud is '3' (8.51% error)

One thing to note is that on some early PIC revisions (A3) the UART is weird and the exact values won't work. On these chips try a value +/-1.

Reset

```
I2C>#
RESET

Bus Pirate v3b
Firmware v5.10 (r559)  Bootloader v4.4
DEVID:0x0447 REVID:0x3046 (24FJ64GA002 B8)
http://dangerousprototypes.com
HiZ>
```

Reset the Bus Pirate.

$ Jump to bootloader

```
HiZ>$
Are you sure? y
BOOTLOADER
BL4+
```

Enter the bootloader for a firmware update without connecting the PGC and PGD pins. Remember to disconnect your terminal program before the upgrade.

```
BL4+BL4+BBL4
```

Bootloader v4.3+ will respond with a version string if a key is pressed while it's active.

Bootloader v4+, firmware v4+. [[Bus_Pirate#Firmware_upgrades |Bus Pirate upgrade instructions]].

Scan or click me to return to the original page
(http://buspirate.com/tutorial/bus-pirate-command-guide)

Visit us:

BusPirate.com Breakout boards, chips, devices, debugging tutorials
DirtyPCBs.com Cheap prototype PCBs, SLA prints, custom cables, and more!
DangerousPrototypes.com Open source hardware, tools, and toys