

[Skip to Navigation](#)



DANGEROUS PROTOTYPES
A new open source hardware project every month



- [Blog](#)
- [Projects](#)
- [Forum](#)
- [About](#)
- [Contact](#)
- [Shop](#)
- [Free PCBs](#)

[Facebook](#) [Twitter](#) [RSS](#) [Comments](#)

Bus Pirate 102 tutorial

From DP

Language:	English • Español • Français • Русский
-----------	--



By popular demand, here's a second part to the original Bus Pirate 101 tutorial. This time we walk through the Bus Pirate user interface and demonstrate some basic stuff like bus modes, power supplies, pull-up resistors, and number entry. Even if you've worked with the Bus Pirate for a while, you might learn some new tricks in this detailed use tutorial written by the primary developer.

Unboxing photo by Uwe Bannow (<http://img338.imageshack.us/img338/2870/pb130503.jpg>) , thanks!

Contents

- 1 Enter a bus mode
 - 1.1 Power supplies
 - 1.2 Pull-up resistors
 - 1.3 Macros
 - 1.4 Number entry and formats
 - 1.5 Repeat command
 - 1.6 When you're done
- 2 Taking it further

Enter a bus mode

```
* Syntax error, type ? for help
HiZ>W
No mode set, M for mode <<<not allowed in HiZ
HiZ>
```

The Bus Pirate always starts in high impedance (http://en.wikipedia.org/wiki/High_impedance) mode (HiZ), a safe mode with all outputs disabled. It's intended to protect any connected devices from conditions beyond their specifications.

If your terminal is blank, press enter to see the command prompt. Type '?' to see a complete list of menu options and syntax available for the current hardware and software.

```
HiZ>m <<<configure bus mode
1. HiZ
...
4. I2C
...
10. LCD
(1) >4 <<< select I2C (option 4)
READY
I2C>
```

We'll be using a 24AA I2C EEPROM (<http://en.wikipedia.org/wiki/EEPROM>) (data storage chip) on the 3EEPROM explorer board to demonstrate some features. *You don't need the chip to follow along, you won't damage anything by following the tutorial without a chip.*

1. Press 'M' to configure a bus mode. Choose the I2C bus by typing 4 followed by enter
2. If requested, choose the default speed by simply pressing enter
3. If requested, choose the default hardware mode by pressing enter

Steps 2 and 3 may not be available, depending on your hardware and firmware version.

The Bus Pirate is now in I2C mode, you should see a *I2C>* command prompt.

Power supplies

3.3volt and 5volt on-board power supplies can provide up to 150mA for your project. Activate them with the w/W command from any mode except HiZ mode. HiZ mode is a safe mode and all outputs are disabled.

```
I2C>w<<<power supplies off
POWER SUPPLIES OFF
I2C>v<<<voltage report
Voltage monitors: 5V: 0.0 | 3.3V: 0.0 | VPULLUP: 0.0 |
I2C>W<<<power supplies on
POWER SUPPLIES ON
I2C>v<<<voltage monitor report
Voltage monitors: 5V: 4.9 | 3.3V: 3.2 | VPULLUP: 0.0 |
I2C>
```

Capital 'W' activates the on-board supplies, small 'w' turns them off. Turn the power supplies on, then press v to show a power supply voltage report.

Note that W is syntax and not a menu option, it can be used with other syntax to toggle the power in the middle of complex bus operations.

The power supplies will try to protect your project with over current protection. Additionally, the Bus Pirate will measure the voltage shortly after the supplies are enabled. If a short is detected it will disable them and show a warning. This may help minimize damage to a project when there is a short or problem on a new board.

Pull-up resistors

Some bus types, such as I2C, require a resistor to create a '1' on the bus. This is called an open collector or open drain bus (<http://hackaday.com/2009/07/01/mixed-voltage-interfacing-with-the-bus-pirate/>) . Without a pull-up resistor (http://en.wikipedia.org/wiki/Pull-up_resistor) , the bus will only read '0'. Some bus modes are open collector only (1-Wire, I2C), most have an open collector option.

```
I2C>v <<<voltage monitor report
Voltage monitors: 5V: 4.9 | 3.3V: 3.2 | VPULLUP: 4.9 |
I2C>
```

Connect the Vpu pin of the IO header to a voltage. The on-board pull-up resistors apply the voltage connected to the Vpu pin to the bus pins through 10Kohm resistors. Press v and verify that the Vpullup voltage shows the desired value.

```
I2C>p <<<configure pull-up resistors
1. Pull-ups off
2. Pull-ups on
(1) >2 <<<enable
Pull-up resistors ON
I2C>
```

Press p to configure the on-board pull-up resistors, enable them. For more, see the practical guide to Bus Pirate pull-up resistors.

Macros

Macros are preprogrammed shortcuts. Some of the macros available include: smart card interrogation, I2C address scanners, JTAG chain probes, transparent UART bridges, LCD initialization, and bus sniffers.

```
I2C>(0)<<< show macro menu
0.Macro menu
1.7bit address search
2.I2C sniffer
I2C>(1)<<< macro 1, I2C address search
Searching 7bit I2C address space.
Found devices at:
0xA0(0x50 W) 0xA1(0x50 R)
I2C>
```

Type (0) to get a list of macros in any mode. The macro number can be entered in any format, (0x00) and (0b0) are also valid macro entries.

I2C devices respond to a 7bit address. We could look up the address in the datasheet, but instead we'll use the address scanner macro (1). The Bus Pirate scanned all possible addresses and found the chip at 0x50 (0xa0/0xa1).

If you don't have an I2C chip attached for this tutorial, the Bus Pirate won't find any I2C devices. If it finds devices at every address, make sure there's powered pull-up resistors on the I2C bus pins.

Read more about the I2C address scan (<http://dangerousprototypes.com/2009/09/07/scan-for-i2c-read-and-write-addresses/>) .

Number entry and formats

Numbers can be entered as binary (http://en.wikipedia.org/wiki/Binary_numeral_system) , decimal (<http://en.wikipedia.org/wiki/Decimal>) , or hexadecimal (<http://en.wikipedia.org/wiki/Hexadecimal>) values, mix and match, use whatever best suits the datasheet or task.

```
I2C>= <<<convert number formats
CONVERT (1) >10 <<<enter value in any format
0x0A
10
0b00001010
I2C>
```

Use the '=' menu to convert between formats. Decimal 10 is equivalent to 0x0a (HEX) and 0b1010 (BIN). More about the value converter (<http://dangerousprototypes.com/2009/08/27/bus-pirate-hexdecbin-converter>) .

```
CONVERT (1) >0x0f
0x0F
15
0b00001111
```

0x0f is a HEX formatted value. The leading 0 is optional for values less than 0x10, 0xf is also valid. HEX can also be entered as 0h0f, if that's more familiar.

```
CONVERT (1) >0b11
0x03
3
0b00000011
```

0b11 is a binary 3. It can also be entered with or without leading 0s: 0b00000011, 0b011, etc. More about number entry and output display.

Repeat command

Most syntax can be repeated by appending ':' and a repeat value, the repeat value can be in any number format.

```
I2C>0b10:0x03
WRITE: 0x02 , 0x03 TIMES ACK <<<'0b10' three times
I2C>
```

0b10:0x03 uses the repeat command(:) to write 2 (0b10) to the I2C bus three times (:0x03).

```
I2C>[0xa1 r r r:3]
I2C START BIT
WRITE: 0xA1 ACK
READ: 0x0F ACK<<< read a single byte
```

```
READ: 0x0E ACK<<< read a single byte
READ 0x03 BYTES:<<< read multiple bytes
0x03 ACK 0x0E ACK 0x02 NACK
I2C STOP BIT
I2C>
```

r is the generic command to read a single byte in all bus modes, it can also be repeated with the ':' command (r:3). The Bus Pirate output is HEX formatted by default, but you can change it to binary or decimal from the output mode menu 'o'.

When you're done

```
I2C>m<<<mode menu
1. HiZ
...
10. LCD
(1) ><<<HiZ is the default
Mode selected
HiZ>
```

When you're done, press 'm'. The Bus Pirate immediately turns off all outputs, power supplies, and pull-up resistors, and prompts for a new mode. Choose 1 (or just press enter for the default option) to return to HiZ mode.

Taking it further

In Bus Pirate 101 we looked at the Bus Pirate hardware, installed the driver, and configured the serial terminal. In Bus Pirate 102 we entered a bus mode and demonstrated some basic commands.

Now you're ready to interface some devices (<http://dangerousprototypes.com/bus-pirate-manual/#demos>) . The 3EEPROM explorer board might be a good place to start, it has three common EEPROM chips with three popular buses (1-Wire, I2C and SPI).

The Bus Pirate manual (<http://dangerousprototypes.com/bus-pirate-manual/>) is continuously updated. Check out the expanded IO header pin reference (<http://dangerousprototypes.com/bus-pirate-manual/#ioheader>) , complete menu options guide (<http://dangerousprototypes.com/bus-pirate-manual/#menus>) , and individual bus mode documentation (<http://dangerousprototypes.com/bus-pirate-manual/#busmodes>) .

If you program on a PC, you could automate your work with a script for the Bus Pirate binary interface mode. There's a bunch of examples (<http://code.google.com/p/the-bus-pirate/source/browse/#svn/trunk/scripts>) to get you started.

There's also an active Bus Pirate forum (<http://dangerousprototypes.com/forum/index.php?board=4.0>) .

Retrieved from "http://dangerousprototypes.com/docs/Bus_Pirate_102_tutorial"
Categories: Projects | Bus Pirate

jump to

- Projects homepage
- Bus Pirate
- USB Infrared toy
- Web platform
- Bus Blaster
- Logic Sniffer
- CPLD dev boards
- Logic Shrimp
- Breakout boards
- Proto boards
- #twatch network LCD
- Flash Destroyer
- In development
- Partlist and reference
- Recent changes

projects



buy stuff

- Bus Pirate v3
- USB Infrared Toy
- web platform
- Bus Blaster v2
- Logic Sniffer

Search

Toolbox

- What links here
- Related changes

- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)

Views

- [Page](#)
- [Discussion](#)
- [View source](#)
- [History](#)

Personal tools

- [Log in](#)
-

Site

- This page was last modified on 27 August 2012, at 13:09.

Copyright 2011 Where Labs, LLC.
CC BY-SA unless otherwise noted.

Connect

- [Follow us on Twitter](#)
- [Subscribe to our feed](#)
- [Subscribe to our comments](#)
- [Visit our Facebook page](#)
- [Send us an email](#)

Projects

- [Get a Bus Blaster v2](#)
- [Get a Bus Pirate v3](#)
- [Get a Flash Destroyer kit](#)
- [Get a Logic Sniffer](#)
- [Get a USB Infrared Toy](#)

About Us

Our goal is to release a new open source hardware project every month. Copies of our projects are available through Seeed Studio and our distributors.