

[Skip to Navigation](#)



DANGEROUSPROTOTYPES
A new open source hardware project every month



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

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

Bus Pirate 101 tutorial

From DP

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

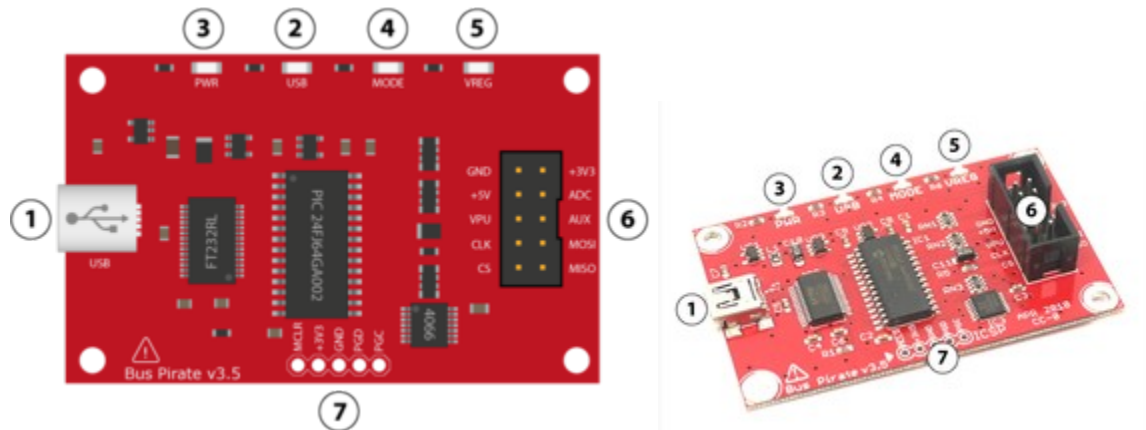
In this introductory tutorial we cover the basics of the Bus Pirate universal serial interface (<http://www.buspirate.com>) . First, we identify the different parts of the Bus Pirate and explain their function. Next, we cover driver installation and terminal configuration. Finally, we explain the user interface and demonstrate some basic terminal menu commands. There's links to everything you need to start hacking with the Bus Pirate.

Contents

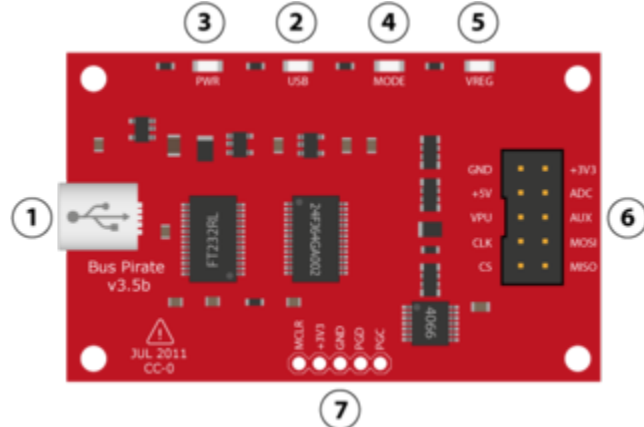
- 1 Bus Pirate parts
- 2 FTDI driver install and configuration
- 3 Terminal setup
- 4 Get to know the terminal interface
- 5 Bus modes, protocol libraries
- 6 Taking it further
 - 6.1 Part 2

Bus Pirate parts

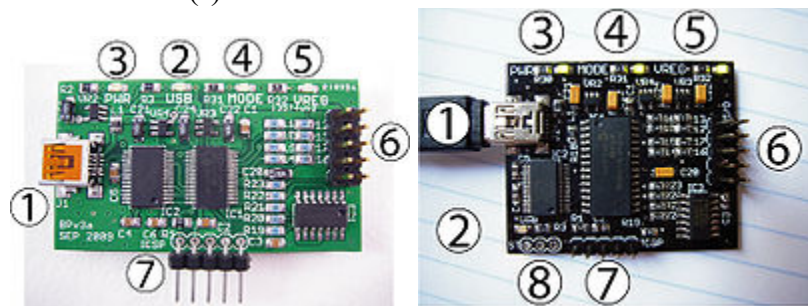
SOIC Version



Last Version



Older Version(s)



1. *Mini-B USB port.* Connects the Bus Pirate to a PC. The Bus Pirate draws power from the USB port, and uses the data connection to communicate with the PC.

2. *USB transmit indicator.* This LED flashes when there's traffic from the PIC to the PC.

3. *Power indicator.* This LED lights when the Bus Pirate is powered by the USB supply.

4. *Mode indicator*. This LED lights when the Bus Pirate is configured for a protocol mode from the user terminal (menu 'm'). The I/O pins might be active when the mode indicator is on. The pins should be in a safe, non-powered, high-impedance state when the mode LED is off.

5. *Voltage regulator indicator*. This LED lights when the on-board power supplies have been activated from the user terminal (command capital 'W').

6. *I/O pins*. This 2x5 block of 0.1" pin header connects the Bus Pirate to external circuits. See the pinout table below, or the Bus Pirate manual (<http://dangerousprototypes.com/bus-pirate-manual/>) .

Bus Pirate - IO Pins

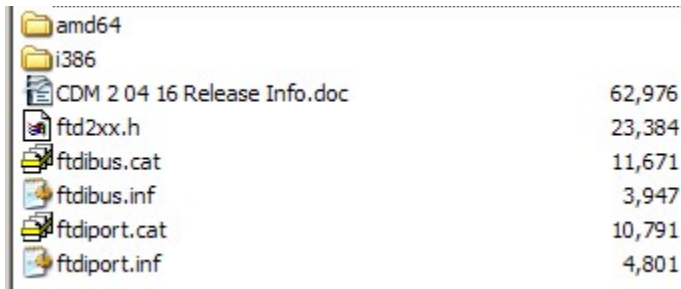
Pin Name	Description (Bus Pirate is the master)
MOSI	Master data out, slave in (SPI, JTAG), Serial data (1-Wire, I2C, KB), TX* (UART)
CLK	Clock signal (I2C, SPI, JTAG, KB)
MISO	Master data in, slave out (SPI, JTAG) RX (UART)
CS*	Chip select (SPI), TMS (JTAG)
AUX	Auxiliary IO, frequency probe, pulse-width modulator
ADC	Voltage measurement probe (max 6volts)
Vpu	Voltage input for on-board pull-up resistors (0-5volts).
+3.3v	+3.3volt switchable power supply
+5.0v	+5volt switchable power supply
GND	Ground, connect to ground of test circuit

Notes: * TX moved from CS to MOSI in firmware v0g

7. *In circuit serial programming (ICSP) header*. This 1x5 block of 0.1" pin header is the programming connection for the PIC 24FJ64GA002 microcontroller. These pins can be used to write new firmware to the microcontroller with a programmer like the PICKIT2 or ICD2 . The Bus Pirate firmware can also be updated over the USB connection using a bootloader, so the ICSP header is normally only used to program it the first time at the factory. Put a jumper between the PGC and PGD pins to trigger the on-board bootloader for firmware updates.

8. *Serial terminal (ST) header*. Version v2go only. This unpopulated header is a tap into the UART connection between the PIC microcontroller and the FTDI 232BL chip that provides the USB connection. The Bus Pirate firmware defaults to a 115200bps/8/N/1 UART.

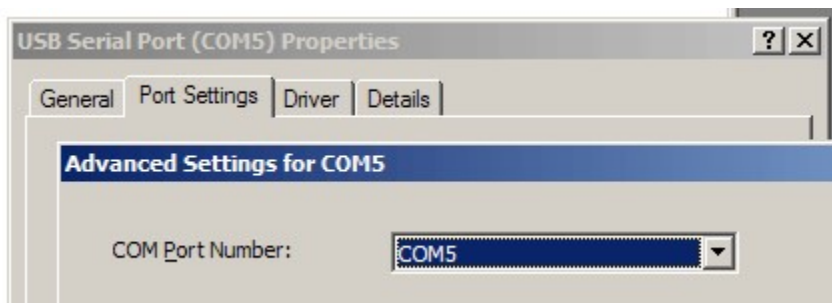
FTDI driver install and configuration



Windows will request a driver the first time the Bus Pirate connects to a PC. Extract the 2.08.28 virtual com port drivers from FTDI (<http://www.ftdichip.com/Drivers/VCP.htm>) into a folder and browse to them using the 'Found New Hardware' wizard. Note: the 2.08.30 drivers gave some people connection errors between the Windows terminal and the Bus Pirate. For now we advise to install the 2.08.28 drivers. Install guides and drivers for other systems are also available on the FTDI driver download page.



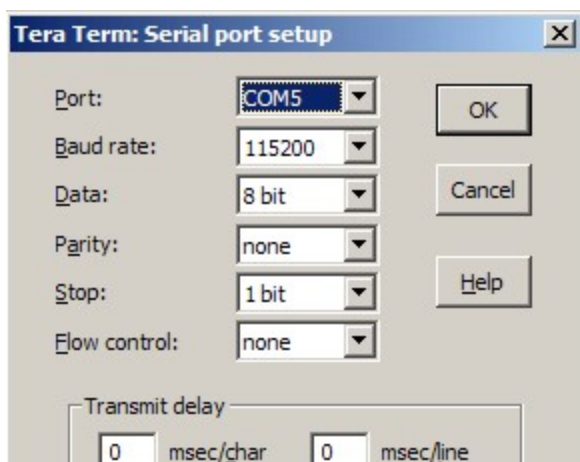
To find the COM port number assigned to the Bus Pirate go to the Windows device manager (*Start->Settings->Control panel->System->Hardware->Device manager*). Look in 'Ports (COM & LPT)' for 'USB Serial Port', ours is COM5.



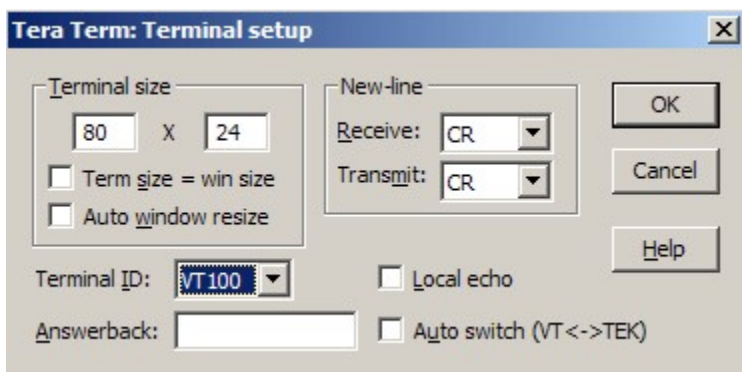
You can change the serial port assigned to the FTDI chip. Go to *USB Serial Port properties->Port settings tab->Advanced*, change the COM port in the drop-down box.

Terminal setup

Windows terminal is cranky, but it appears to work (<http://whereisian.com/forum/index.php?topic=4.0>) with the Bus Pirate when VT100 emulation is enabled. We highly recommend a better terminal, we like Tera Term Pro (<http://www.ayera.com/teraterm/>) for Windows.

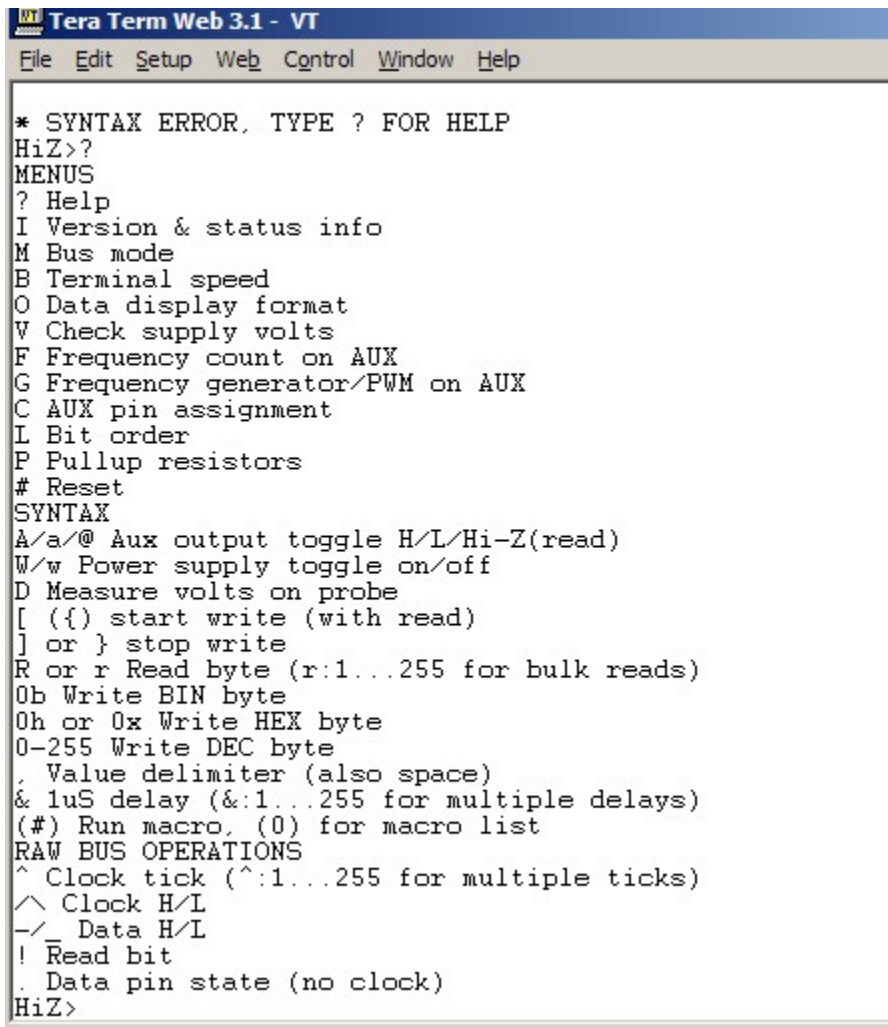


First, configure the correct COM port and settings. The Bus Pirate operates at 115200bps/8/N/1 on the COM port assigned by Windows. Flow control is no longer required. Disable it!



Next, check the terminal setup. Turn off local echo and use a VT100 terminal type. The Bus Pirate should work with any type of new-line character, but we use the CR setting.

Get to know the terminal interface



```

Tera Term Web 3.1 - VT
File Edit Setup Web Control Window Help

* SYNTAX ERROR, TYPE ? FOR HELP
HiZ>?
MENUS
? Help
I Version & status info
M Bus mode
B Terminal speed
O Data display format
V Check supply volts
F Frequency count on AUX
G Frequency generator/PWM on AUX
C AUX pin assignment
L Bit order
P Pullup resistors
# Reset
SYNTAX
A/a/@ Aux output toggle H/L/Hi-Z(read)
W/w Power supply toggle on/off
D Measure volts on probe
[ ({} start write (with read)
] or } stop write
R or r Read byte (r:1...255 for bulk reads)
0b Write BIN byte
0h or 0x Write HEX byte
0-255 Write DEC byte
. Value delimiter (also space)
& 1uS delay (&:1...255 for multiple delays)
(#) Run macro, (0) for macro list
RAW BUS OPERATIONS
^ Clock tick (^:1...255 for multiple ticks)
^/ Clock H/L
-/_ Data H/L
! Read bit
. Data pin state (no clock)
HiZ>
```

The Bus Pirate is controlled by text commands entered through the serial terminal. If the terminal is blank, press *enter* to show the command prompt. Press '?', followed by *enter*, to show the help menu.

- **Menus** configure various Bus Pirate options like pull-up resistors, terminal speed, data display format (DEC, HEX, BIN), etc. Type the menu command, followed by *enter*, to display the options.
- **Syntax** is used to interact with a device connected over a bus. Commands are mostly single characters, such as 'r' to read a byte. Enter up to 4000 characters of syntax, press *enter* to execute the sequence.

Each menu and syntax option is documented fully in the Bus Pirate manual (<http://dangerousprototypes.com/bus-pirate-manual/>) .

Most menus have a default option shown in () before the prompt:

Output type:

1. High-Z outputs (H=input, L=GND)

2. Normal outputs (H=Vcc, L=GND)
- (1) > <<< **option 1 is the default**

Press *enter* to select the default option.

Bus modes, protocol libraries

HiZ>M <<< **bus mode menu**

1. HiZ
2. 1-WIRE
3. UART
4. I2C
5. SPI
6. JTAG
7. RAW2WIRE
8. RAW3WIRE
9. PC KEYBOARD
10. MIDI
11. LCD

(1) >5 <<< **enter SPI bus mode**

The 'bus mode' menu (M) configures the Bus Pirate for a specific protocol, like 1-Wire, I2C, SPI, etc. The default start-up mode is HiZ, all pins are inputs and all power supplies are off.

Most modes have additional options, like open drain outputs (<http://hackaday.com/2009/07/01/mixed-voltage-interfacing-with-the-bus-pirate/>) or speed settings. Each bus mode is documented in the Bus Pirate manual (<http://dangerousprototypes.com/bus-pirate-manual/>) [1] (<http://www.buspirate.com/>) .

SPI READY

SPI>p <<< **pull-up configuration menu**

1. Pullup off
2. Pullup on

(1) >2 <<< **pull-up on**

PULLUP RESISTORS ON

SPI>W <<< **power supplies ON syntax**

POWER SUPPLIES ON

SPI>v <<< **check power supply output**

VOLTAGE MONITOR: 5V: 4.9 | 3.3V: 3.3 | VPULLUP: 0.0 |

SPI>

Once a mode is configured, you can enable other options. Turn on the on-board voltage regulators (capital 'W') and check the output levels (v). Configure the pull-up resistors (p).

Taking it further

Now might be a good time to run a hardware self-test and update to the latest firmware (Windows (<http://dangerousprototypes.com/2009/07/24/bus-pirate-firmware-upgrades/>) , OSX

(<http://dangerousprototypes.com/2009/10/24/bus-pirate-firmware-updates-on-osx/>) , Python (<http://dangerousprototypes.com/2009/08/06/bus-pirate-firmware-upgrades-on-linux-osx/>)).

For lots of device demonstrations, and to learn more about the Bus Pirate syntax, menus, pinout, modes, and more, check the Bus Pirate manual (<http://dangerousprototypes.com/bus-pirate-manual/>) and the Bus Pirate forum (<http://whereisian.com/forum/index.php?board=4.0>) .

Part 2

Continue reading Bus Pirate 102.

Retrieved from "http://dangerousprototypes.com/docs/Bus_Pirate_101_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 19 December 2013, at 11:27.

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.