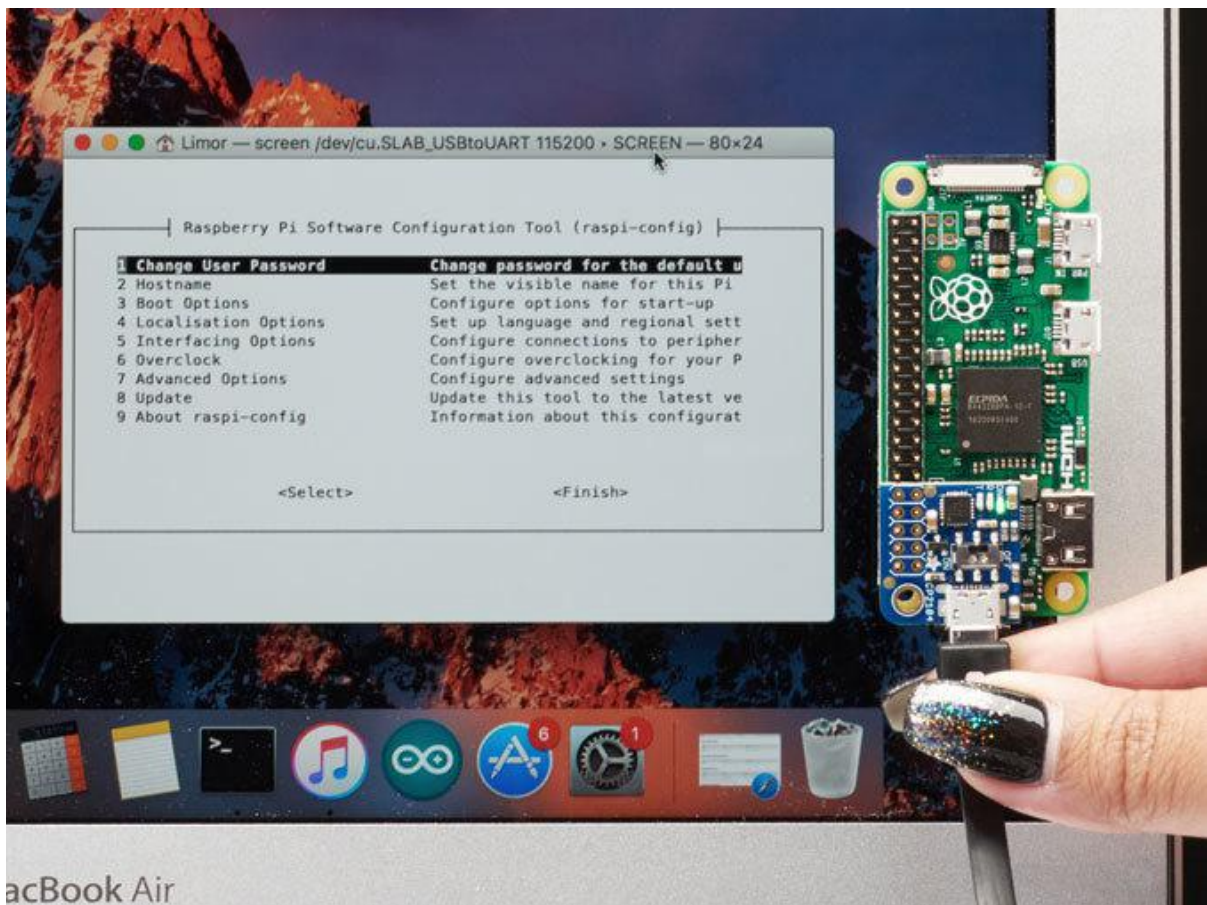




Adafruit PiUART - USB Console and Power Add-on for Raspberry Pi

Created by lady ada



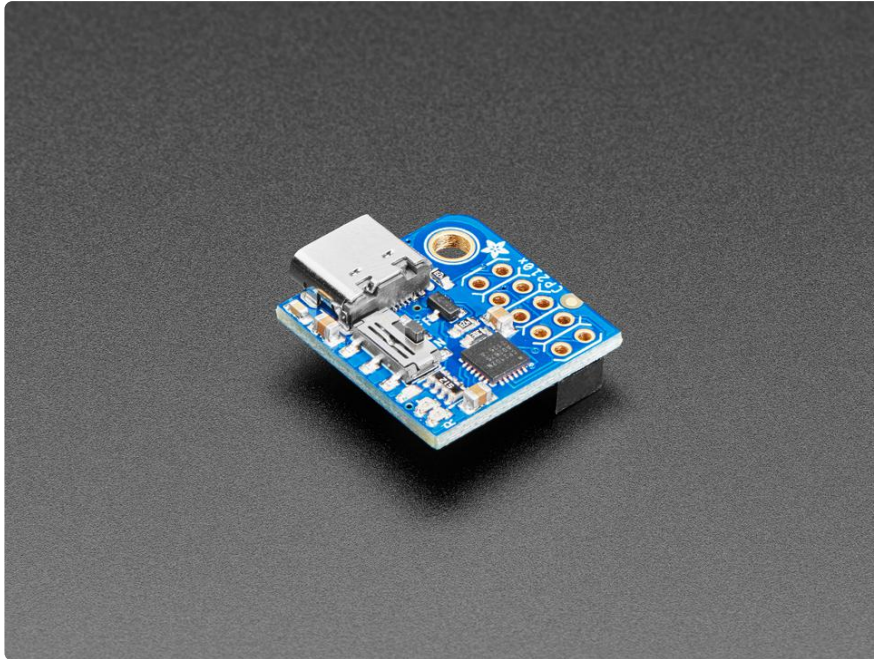
<https://learn.adafruit.com/adafruit-piuart-usb-console-and-power-add-on-for-raspberry-pi>

Last updated on 2022-12-01 01:54:28 PM EST

Table of Contents

Overview	3
Pinouts	5
Enabling Serial Console	6
<ul style="list-style-type: none">• Option 1. Enabling in /boot/config.txt• Option 2. Enabling via Raspi-Config	
Setup Software	9
<ul style="list-style-type: none">• Windows• Mac OS X• Linux	
Test & Configure	10
<ul style="list-style-type: none">• Mac OS X• Linux• Windows	
Downloads	16
<ul style="list-style-type: none">• Files• Schematic and Fab Print for USB C Version• Schematic and Fab Print for USB Micro B Version	

Overview



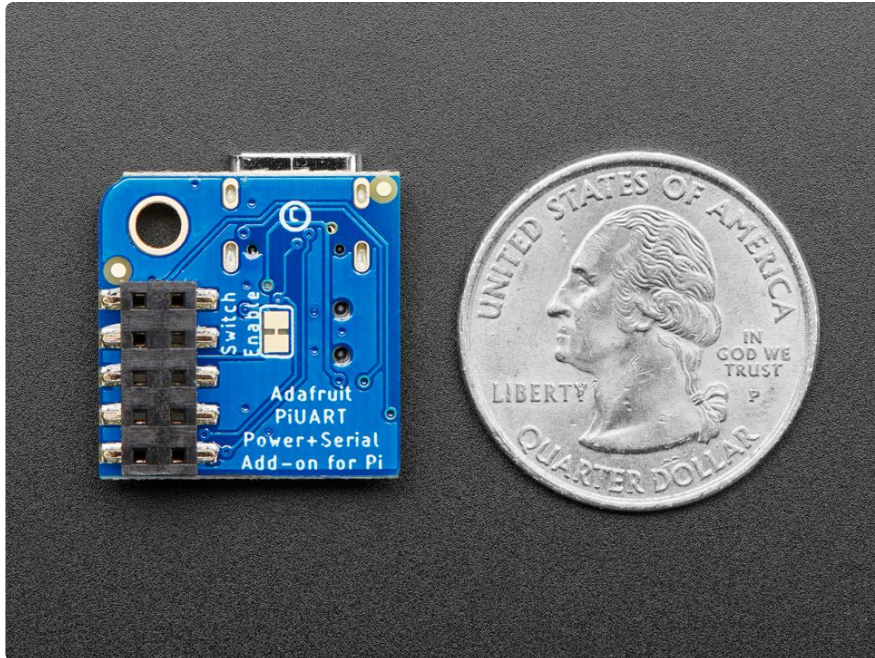
Here's another super handy add-on for your Raspberry Pi computer, perfect for 'head-less' setups! The [PiUART](#) () adds a USB Type-C to serial connection so you can use any serial port software to connect to the Pi's console. It plugs in and is fast and easy to add whenever you need to connect to your Pi. Two LEDs connect to RX and TX on the serial converter chip so you get blinking whenever data is sent or received.



We had some space left over, so the PiUART also comes with an on-off switch with a 4 Amp transistor. You can power your Pi through the USB-C port and then use the

switch whenever you want to cut power, without having to unplug the cable. Low-power usage Pi's like the Pi Zero and A+ can thus be powered and controlled from a single cable connected to your computer.

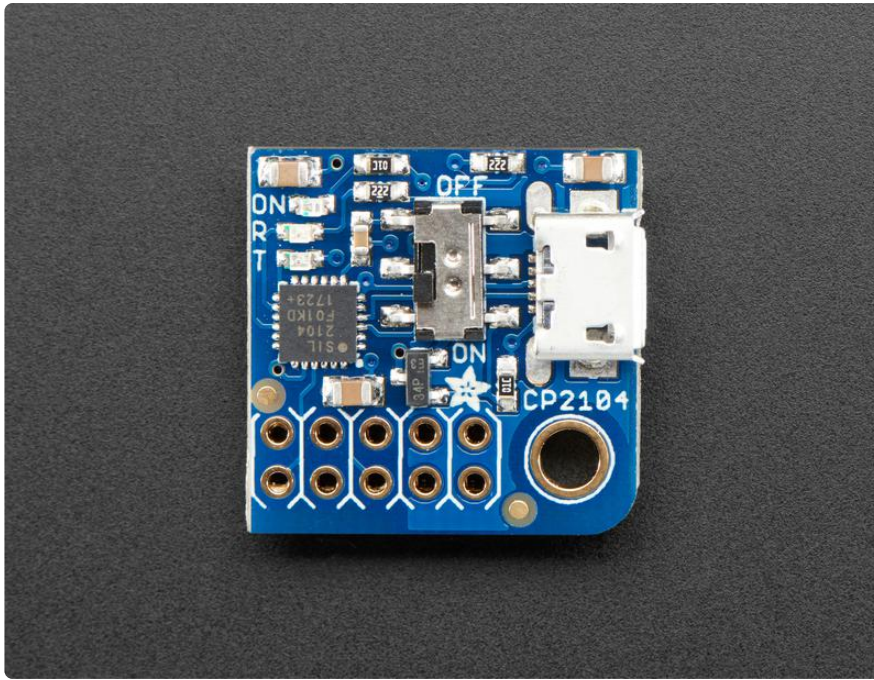
Heavy-hitter Pi's like the Pi 2 and Pi 3 may draw too much power from a computer USB port, so check if your motherboard has a high-current USB port before trying.



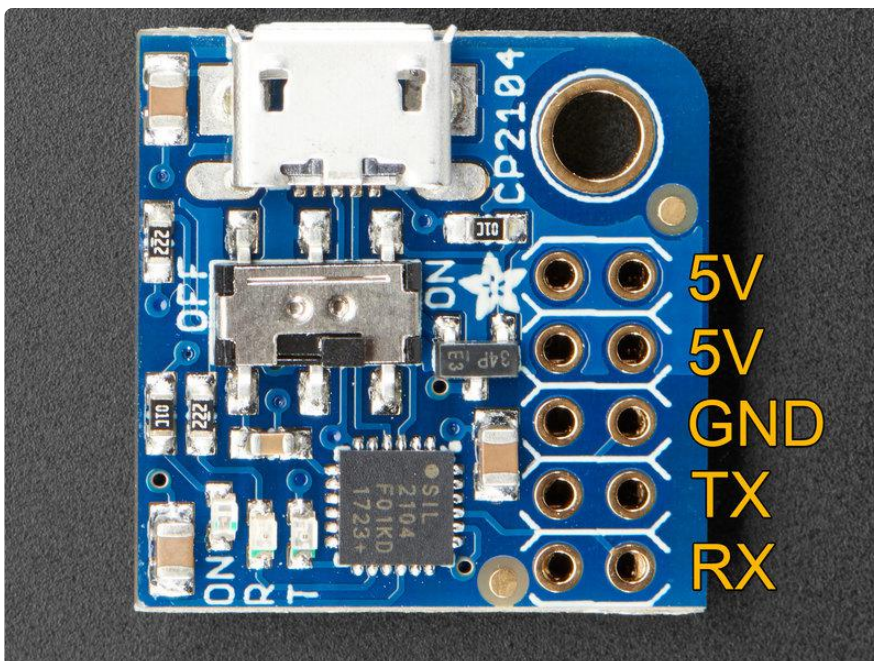
Comes fully assembled and ready to go, plug into your Pi, and on Mac OS X install the driver - within 2 minutes and you'll be ready to go.

Works with any Raspberry Pi computer (Pi 1, 2, 3, Zero, etc)

The original version of this board used USB Micro B, as seen below. This guide will work with both.



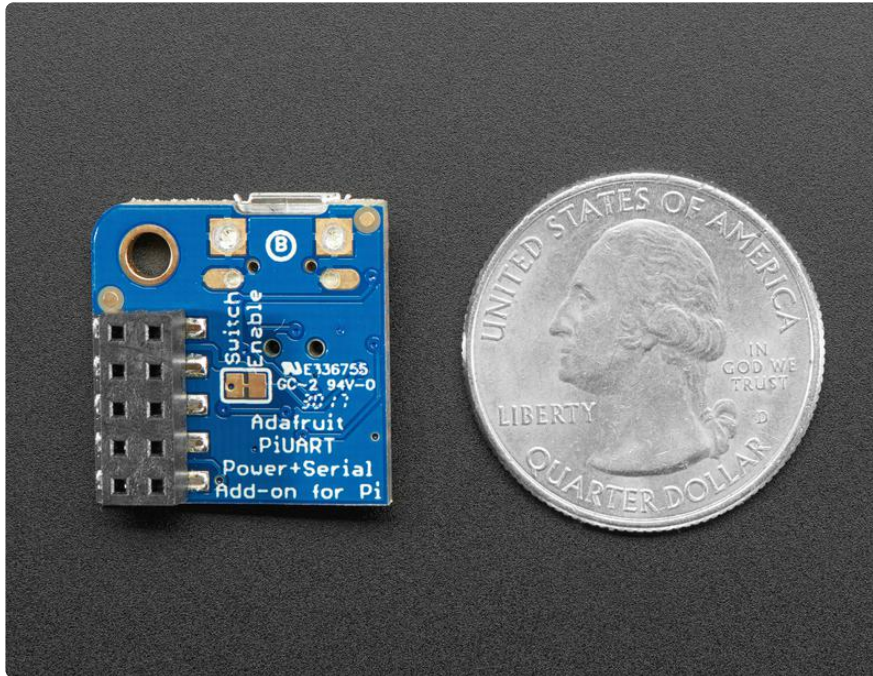
Pinouts



We only use the 'right' strip of the dual header but to make it more mechanically stable we have a dual header to plug into.

The 5V power pins can be fed from the MicroUSB connector when the switch is flipped to ON. Otherwise you can leave the switch OFF and power the Pi from the standard power port.

Note this will back-power the computer USB port. We've never had an issue with this as long as the Pi's power supply is 5V, but if you want to avoid it, cut the bottom Switch Enable trace. (Also good to do if you want to make sure the switch isn't activated by accident)



Enabling Serial Console

As of Jessie Raspbian, you may need to enable the serial console.

You can enable/disable the serial console with either editing `/boot/config.txt` or `raspiconfig` (which will edit `/boot/config.txt` for you)

Option 1. Enabling in `/boot/config.txt`

You can pop your SD card into a computer and edit `config.txt` with a text editor like SimpleText, WordPad or whatnot. You can also edit on a pi with `sudo nano /boot/config.txt`

At the bottom, last line, add `enable_uart=1` on it's own line

```
COM53 - PuTTY
GNU nano 2.2.6 File: /boot/config.txt

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

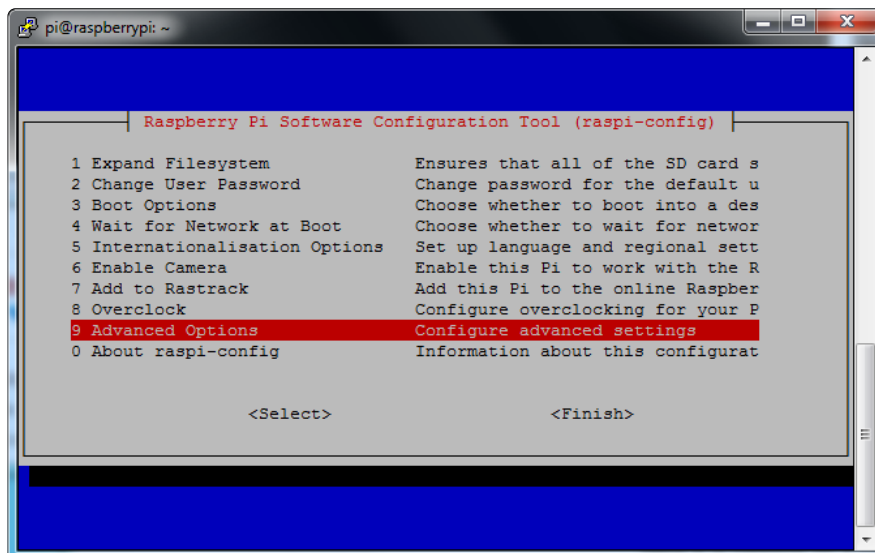
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
enable_uart=1
```

Option 2. Enabling via Raspi-Config

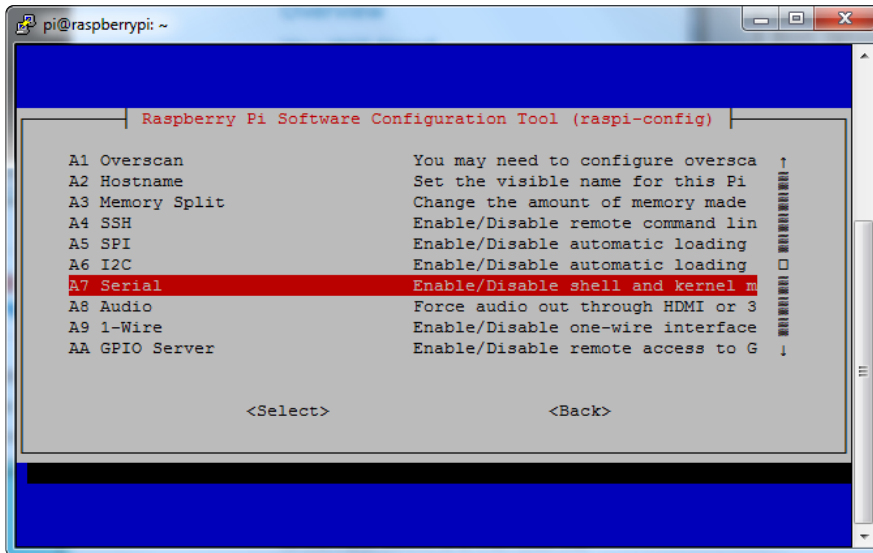
Using a monitor and keyboard, log into the shell and run

```
sudo raspi-config
```

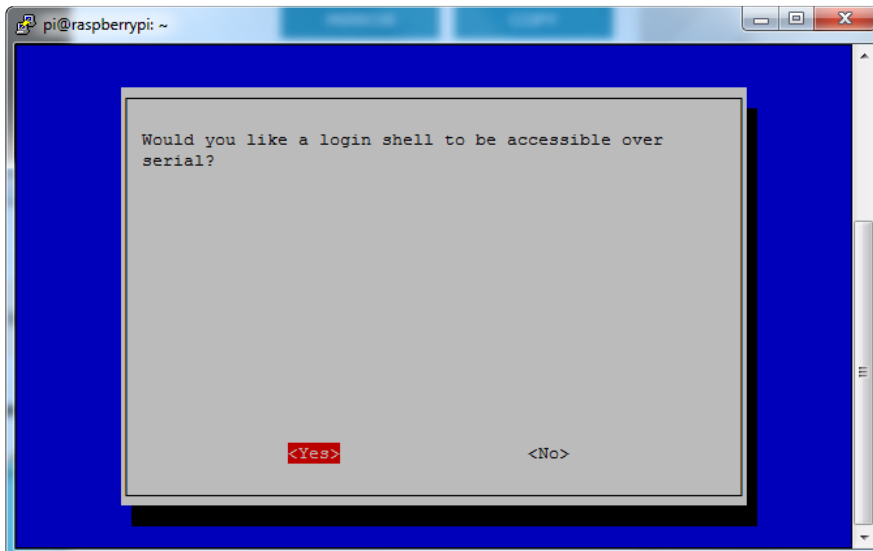
go down to Advanced Options



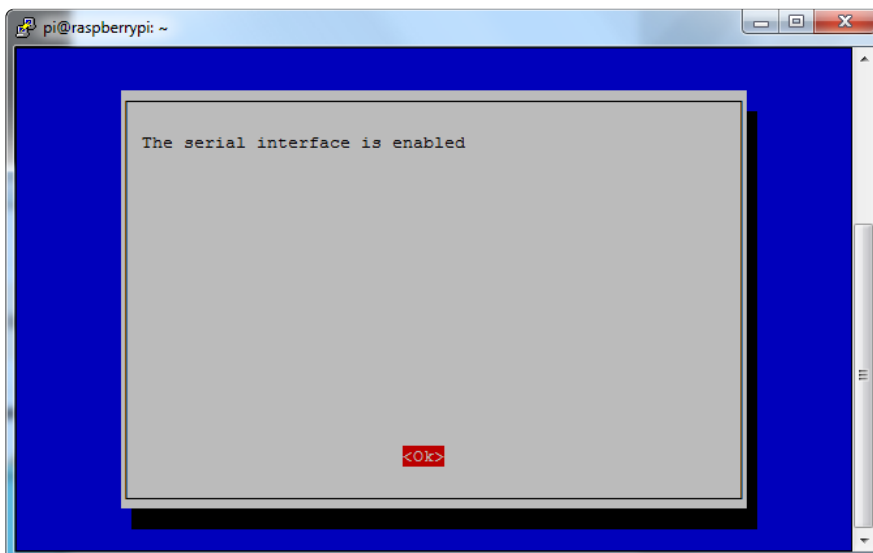
Hit enter and then go down to Serial



Select Yes

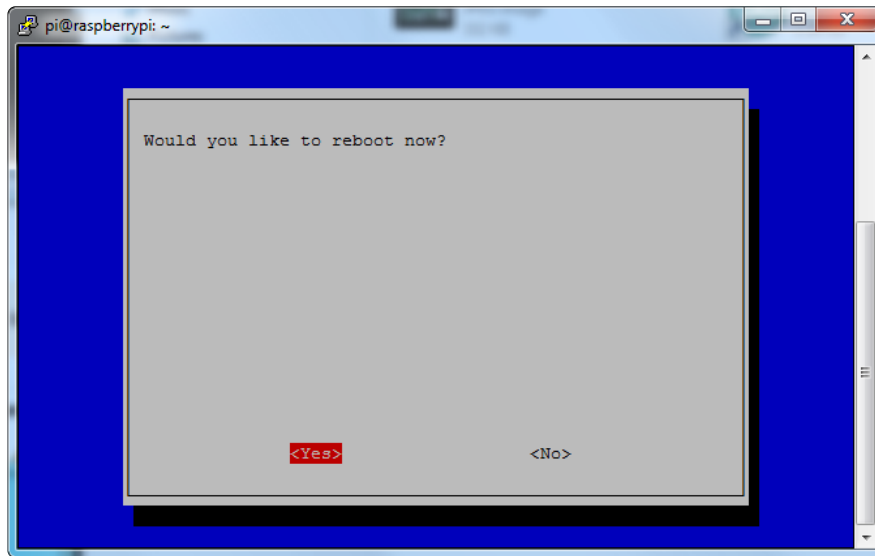


It should now be enabled



Hit return then select Finish

When it asks you to reboot, go to Yes and hit return



OK the serial console is now enabled!

Setup Software

Windows

Windows will automatically install the CP2104 driver for use so no driver is required. The Windows 10 driver can also be found [here](#) () if needed.

For terminal software, you can use Putty to start. Download and install Putty from here: <http://www.putty.org/> ()

From the list of downloads select the binary called just putty.exe from the section For Windows on Intel x86. This will prompt you to save the file. Save it onto the Desktop for now.

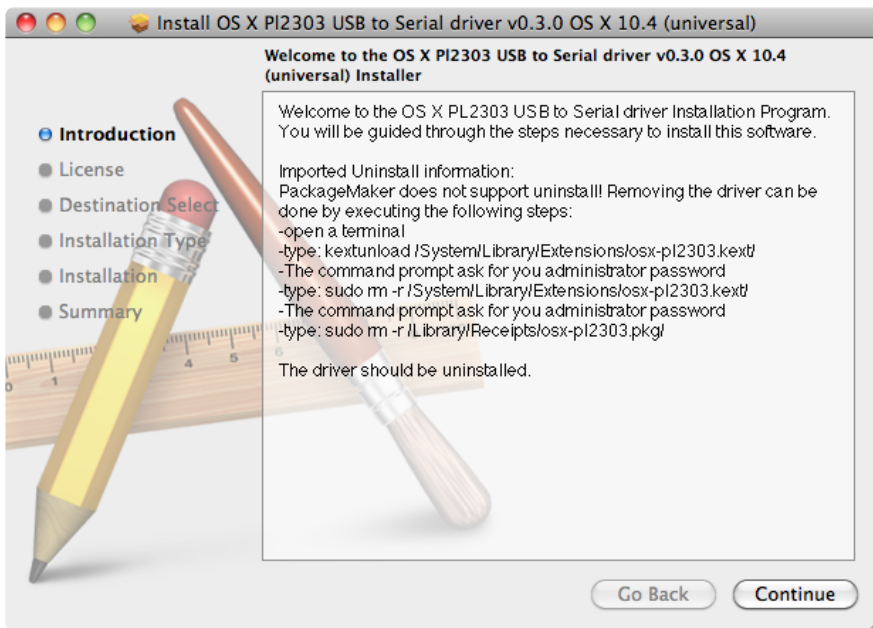
Note that this actually saves the Program itself not an installer. Simply double click putty.exe to run putty!

Mac OS X

If you are using Mac OS X you will need to install drivers. For the SiLabs chipset, you can grab the drivers from here:

[Install Mac OS X driver](#)

The download is a standard Mac installer. Accept all defaults when prompted.



Linux

Linux has the CP2104 driver built in!

Some distributions such as Ubuntu 12.10 do not include the "screen" command. Try running the command "screen" and if you get an error message, you can install it by typing the following command: `sudo apt-get install screen`

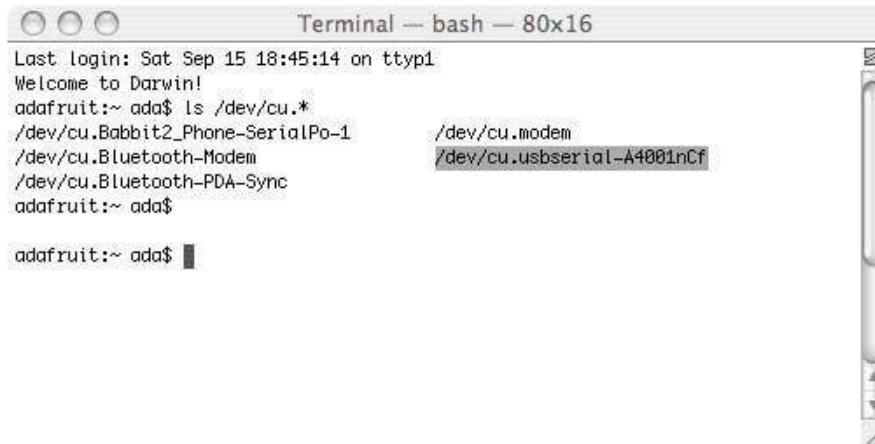
Test & Configure

Mac OS X

If you are using a Mac, then all you need to do is open a Terminal window and issue the command

```
ls /dev/cu.*
```

To list all of the available serial ports. You should see something like this:

A screenshot of a macOS Terminal window titled "Terminal — bash — 80x16". The window shows the following text: "Last login: Sat Sep 15 18:45:14 on ttty1", "Welcome to Darwin!", "adafruit:~ ada\$ ls /dev/cu.*", and the output: "/dev/cu.Babbit2_Phone-SerialPo-1", "/dev/cu.modem", "/dev/cu.Bluetooth-Modem", "/dev/cu.usbserial-A4001nCd", "/dev/cu.Bluetooth-PDA-Sync", "adafruit:~ ada\$". The output lines are wrapped. A vertical scrollbar is visible on the right side of the terminal window.

```
Terminal — bash — 80x16
Last login: Sat Sep 15 18:45:14 on ttty1
Welcome to Darwin!
adafruit:~ ada$ ls /dev/cu.*
/dev/cu.Babbit2_Phone-SerialPo-1      /dev/cu.modem
/dev/cu.Bluetooth-Modem              /dev/cu.usbserial-A4001nCd
/dev/cu.Bluetooth-PDA-Sync
adafruit:~ ada$
adafruit:~ ada$
```

You'll see a few items listed including a /dev/cu.Bluetooth device. You're looking for something like /dev/cu.usbserial-NNNN or /dev/cu.SLAB_USBtoUART or /dev/cu.usbmodem

Once you've identified the name, you can then run:

```
screen /dev/cu.PL2303-00001004 115200
screen /dev/cu.SLAB_USBtoUART 115200
screen /dev/cu.usbserial-A4001nCd 115200
```

The device will have a slightly different name than the above. One way to quickly figure out the name is start typing screen /dev/cu. and then press the TAB key to auto-complete to whatever your device is called, before adding 115200 (which is the baud rate) to the end.

You can also try `screen /dev/cu.SLAB* 115200` or `screen /dev/cu.PL2303* 115200` or `screen /dev/cu.usbserial* 115200` To have the shell complete the file name for you

[You may need to disable system integrity protection \(according to feedback from a tutorial-reader\) \(\)](#)

Linux

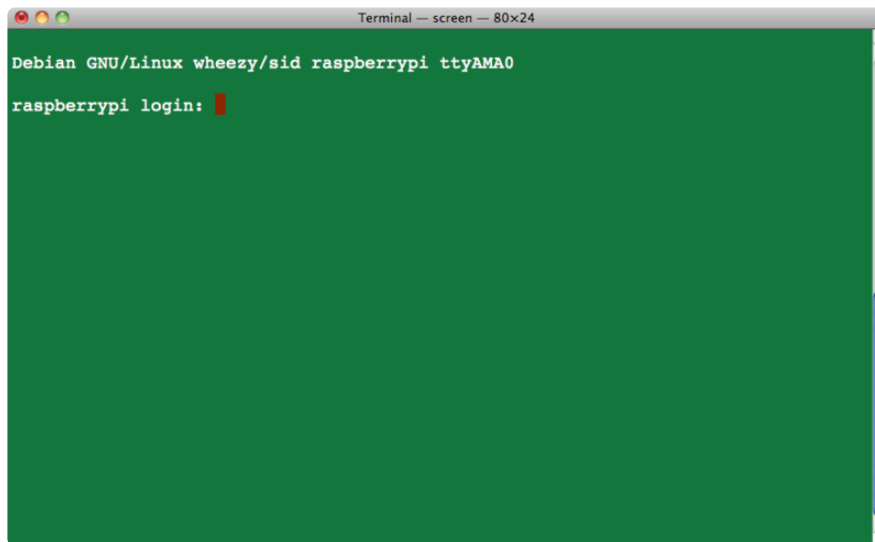
If you are using Linux, its much like the above but often times the device is called /dev/ttyUSB0 - you may want to run `sudo dmesg` after plugging in and looking for hints on what the device is called.

Then use the command:

```
sudo screen /dev/ttyUSB0 115200
```

To start communication with the Pi, press ENTER and you should see the login prompt from the Pi.

Here it is running on a Mac.



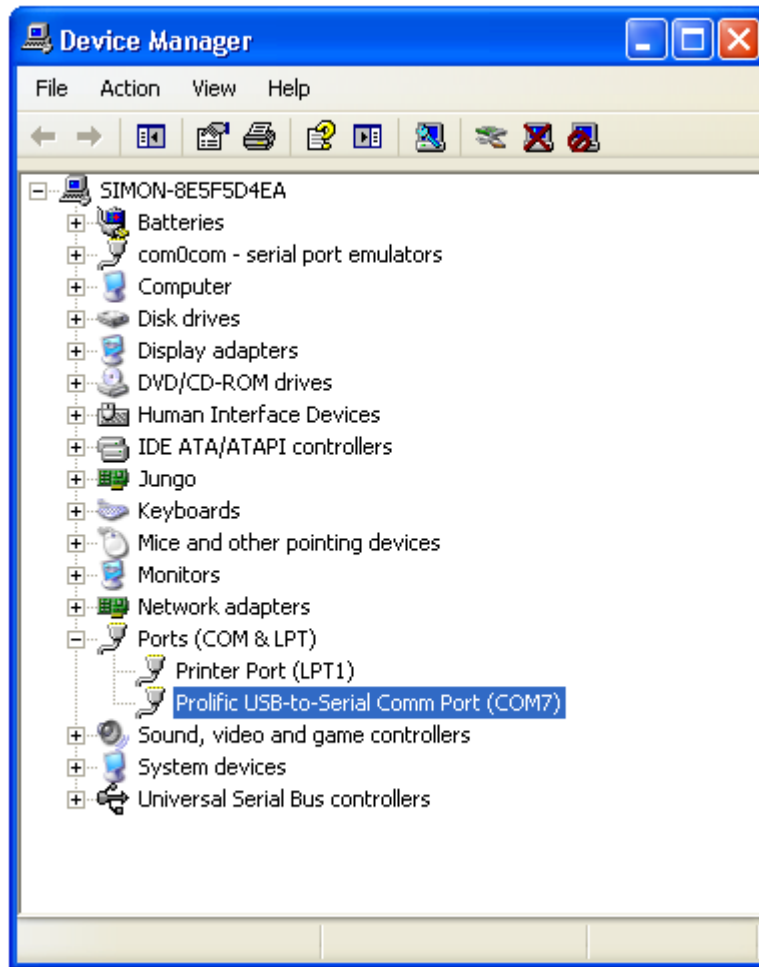
and here is what it looks like in Ubuntu.

```
simon@simon-UB: /dev
[ 2.889118] usb 1-1.1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[ 2.902423] smsc95xx v1.0.4
[ 2.974585] smsc95xx 1-1.1:1.0: eth0: register 'smc95xx' at usb-bcm2708_usb-1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:d8:9e:69
[ 5.317736] bcm2708_i2c bcm2708_i2c.0: BSC0 Controller at 0x20205000 (irq 79)
[ 5.392119] Adafruit Industries' Raspberry Pi PWM driver v1.0
[ 5.480127] bcm2708_spi bcm2708_spi.0: SPI Controller at 0x20204000 (irq 80)
[ 5.531192] bcm2708_i2c bcm2708_i2c.1: BSC1 Controller at 0x20804000 (irq 79)
[ 9.763013] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[ 10.116994] ### snd_bcm2835_alsa_probe c067cbf8 ##### PROBING FOR bcm2835 ALSA device (0):(1) #####
[ 10.134267] Creating card...
[ 10.138094] Creating device/chip ..
[ 10.143196] Adding controls ..
[ 10.147169] Registering card ....
[ 10.158814] bcm2835 ALSA CARD CREATED!
[ 10.165128] ### BCM2835 ALSA driver init OK ###
[ 10.260888] i2c /dev entries driver
◆◆
Debian GNU/Linux wheezy/sid raspberrypi ttyAMA0
raspberrypi login: █
```

Windows

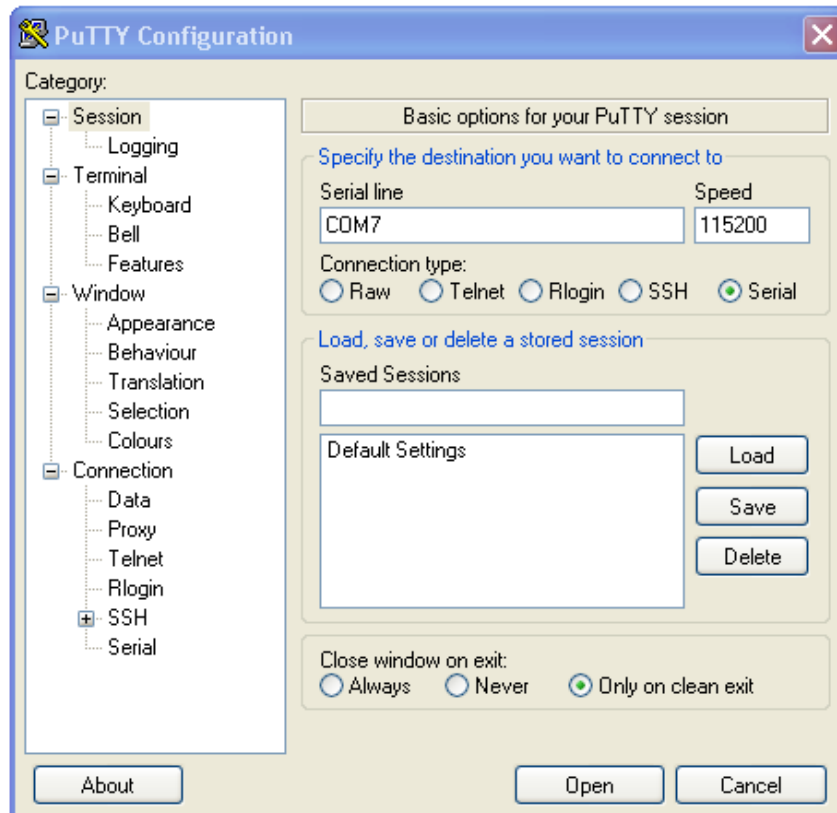
If you are using a PC, then before you start Putty, you need to know which com port is being used for the cable. You can find this by looking in the Ports section of the Windows Device Manager.

The Device Manager is accessible from the Control Panel under System.



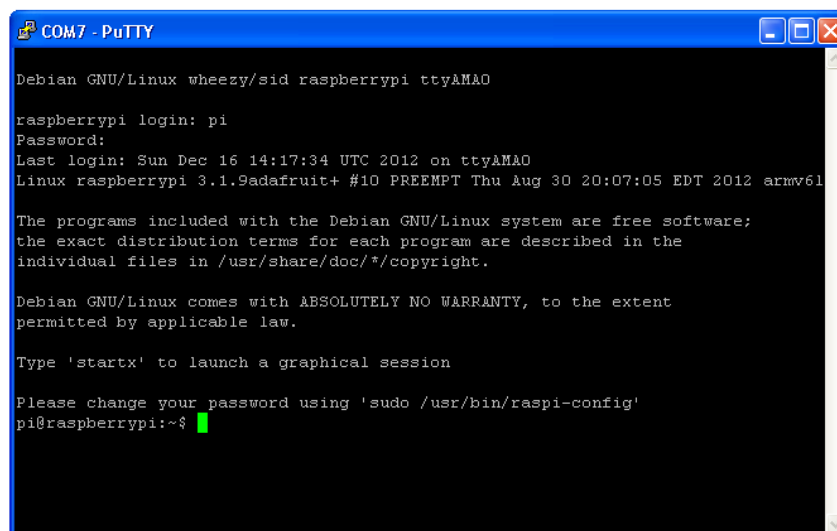
It isn't going to be COM1 so never pick that. Chances are its the one right below COM1. It will often say "SiLabs" or "Prolific" or similar next to it. In this case it is COM7 that is in use.

Now start Putty and you will see a connection window.



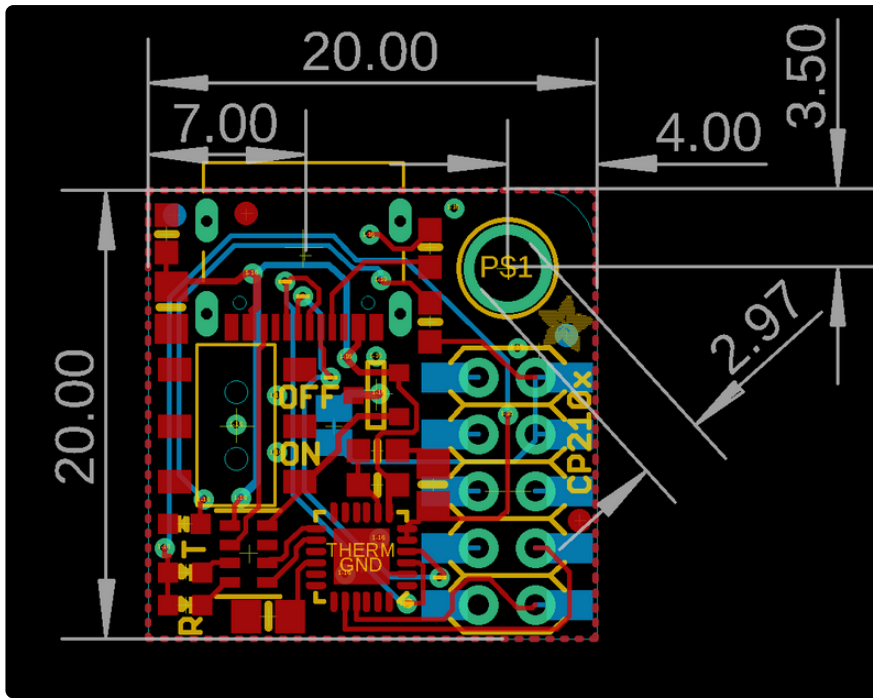
Select a connection type of "Serial" from the radio buttons, then set the speed to 1152 00 and the serial line to COM7

Finally click 'Open' to connect. Remember to press ENTER to start communications.



For a new installation of Raspbian, the default username is pi and the default password is raspberry

Thats it! You are connected and can use the command line to navigate around your Pi.



Schematic and Fab Print for USB Micro B Version

