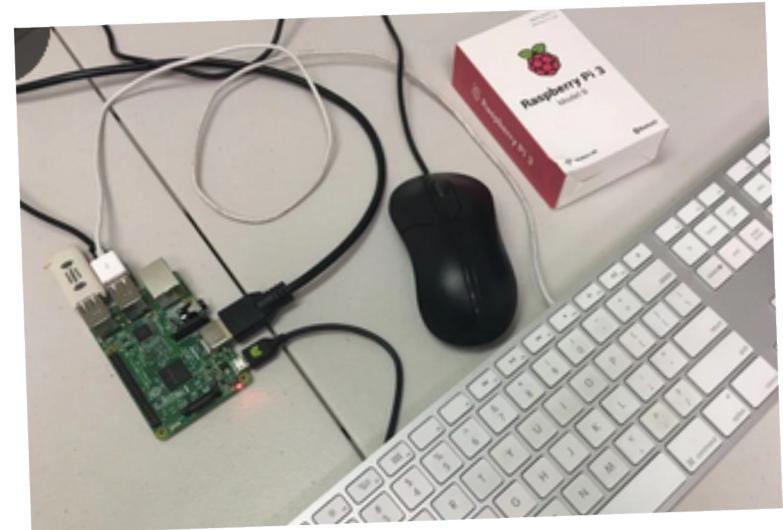


Pie-3 System Environment

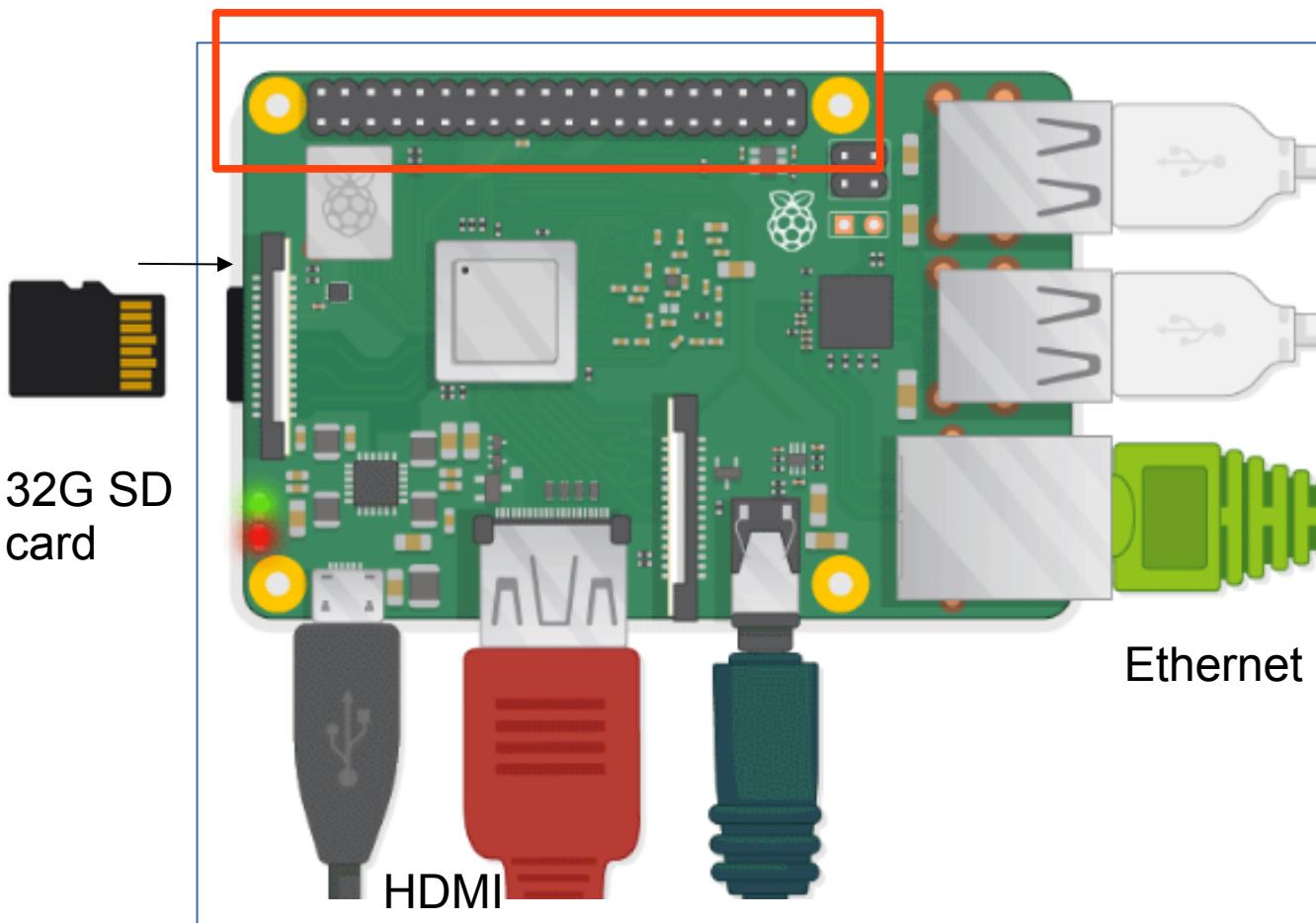


USB keyboard, USB mouse;
USB cable for power
USB wifi
HDMI cable for monitor
SD Card for Raspbian OS

Pie-3 Board

[https://www.raspberrypi.org/
help/](https://www.raspberrypi.org/help/)

Expansion Connectors



The Raspbian Operating System via NOOBS

Using the NOOBS software to install Raspbian OS on your SD card. Download NOOBS at (<https://www.raspberrypi.org/downloads>).

OS: Raspbian, comes pre-installed with many software. It supports Python, Scratch, Sonic Pi, Java and more.

C++/C programming for pie

[https://raspberry-
projects.com/pi/category/
programming-in-c](https://raspberry-projects.com/pi/category/programming-in-c)

Eclipse Linux

Using A Linux PC With A Cross Compiler: this page does not exist

C programming for pie

[https://raspberry-projects.com/pi/
programming-in-c/getting-your-
raspberry-pi-ready-for-c-programming](https://raspberry-projects.com/pi/programming-in-c/getting-your-raspberry-pi-ready-for-c-programming)

Raspbian OS for Pie-3

<https://www.raspberrypi.org/downloads/raspbian/>

Raspbian Stretch with desktop and recommended software

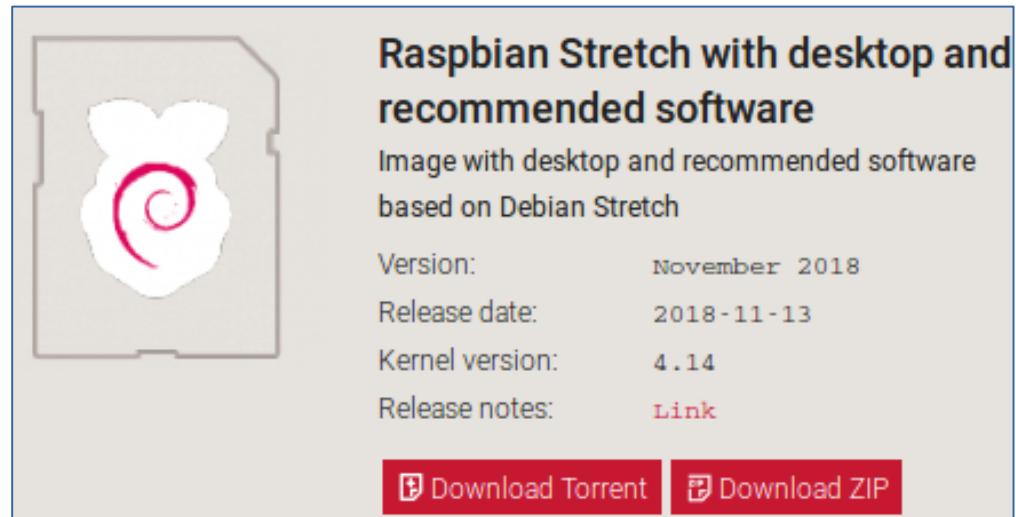
Image with desktop and recommended software based on Debian Stretch

Version: November 2018

Release date: 2018-11-13

Kernel version: 4.14

Release notes: [Link](#)
download: zip

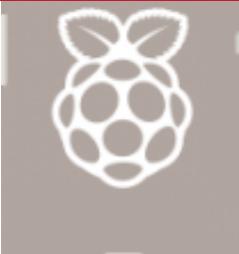


Raspbian Stretch with desktop and recommended software
Image with desktop and recommended software
based on Debian Stretch

Version: November 2018
Release date: 2018-11-13
Kernel version: 4.14
Release notes: [Link](#)

[Download Torrent](#) [Download ZIP](#)

NOOBS



NOOBS

Offline and network install

Version: 3.0.0

Release date: 2018-11-16

[Download Torrent](#)

[Download ZIP](#)

Go to the downloads page, grab a copy of the NOOBS zip file, and unpack it onto a freshly formatted 32GB (or larger) SD card.

<https://www.raspberrypi.org/blog/introducing-noobs/>

When the pie boot up for the first time, you'll see a menu prompting you to install one of several operating systems into the free space on the card. Select the boot of the Pi with a regular OS Raspbian, or with a media-centre OS like RaspBMC.

Prototype Board Layout Design

Prototype Board

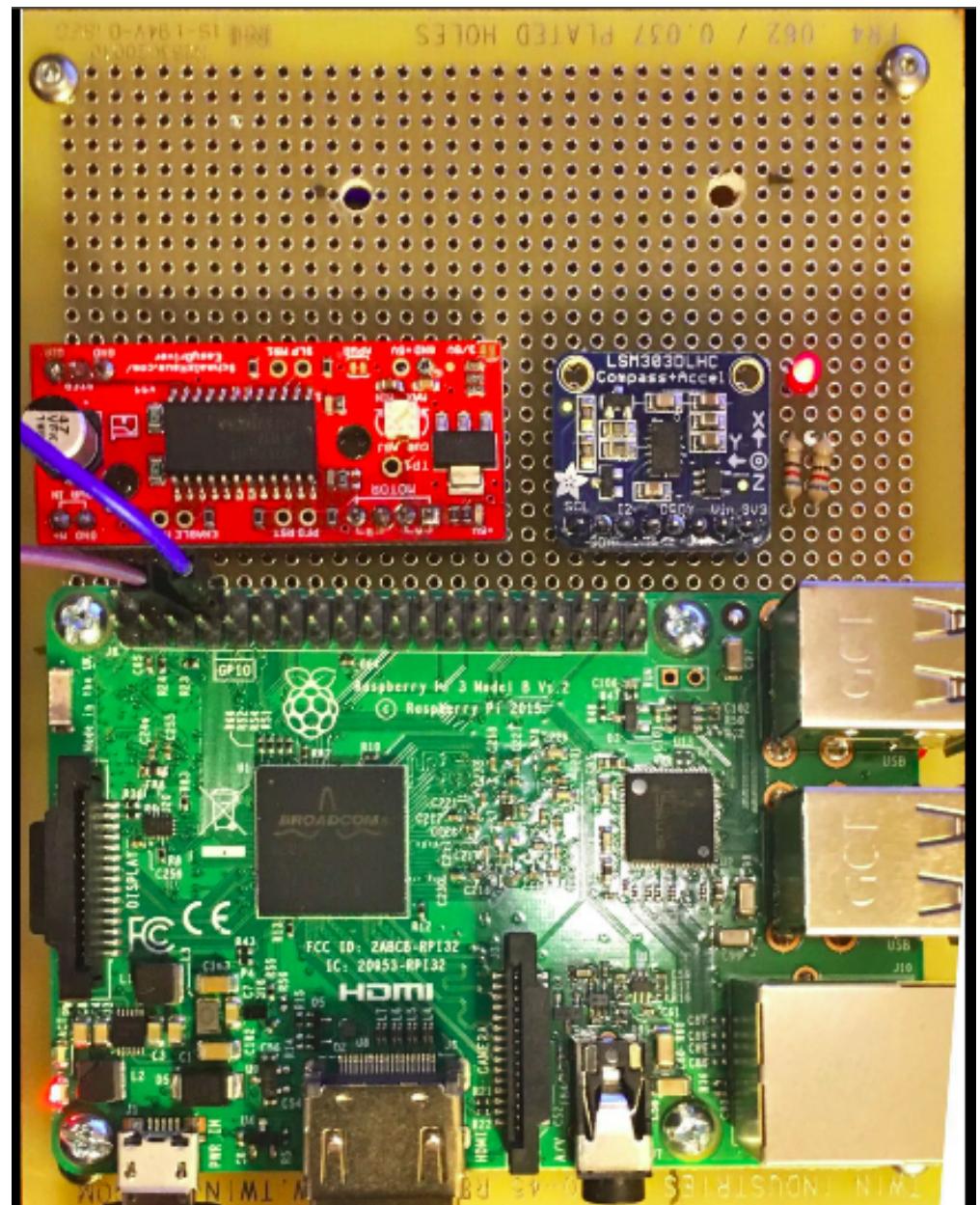
GPIO I/
O
Testing

Stepper
Motor Drive

LSM303

Pwr Unit
(Tier 1)

Pie-3 B+
(Tier 2)



Power Unit Design

INSTRUCTIONS

3900, 4300, 4400
CMD2040WC

5100 SERIES
CMD333UWC

WARNING:

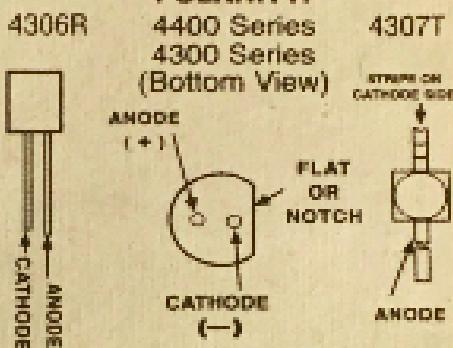
Rectifier Diode must be used for AC applications

Voltage AC DC

6 V.	100 ohms	220 ohms
12 V.	330 ohms	680 ohms
28 V.	750 ohms	1500 ohms

RESISTOR: 1/2 WATT

IDENTIFICATION OF POLARITY:



Model: 3990, 5100, 5110, 5210

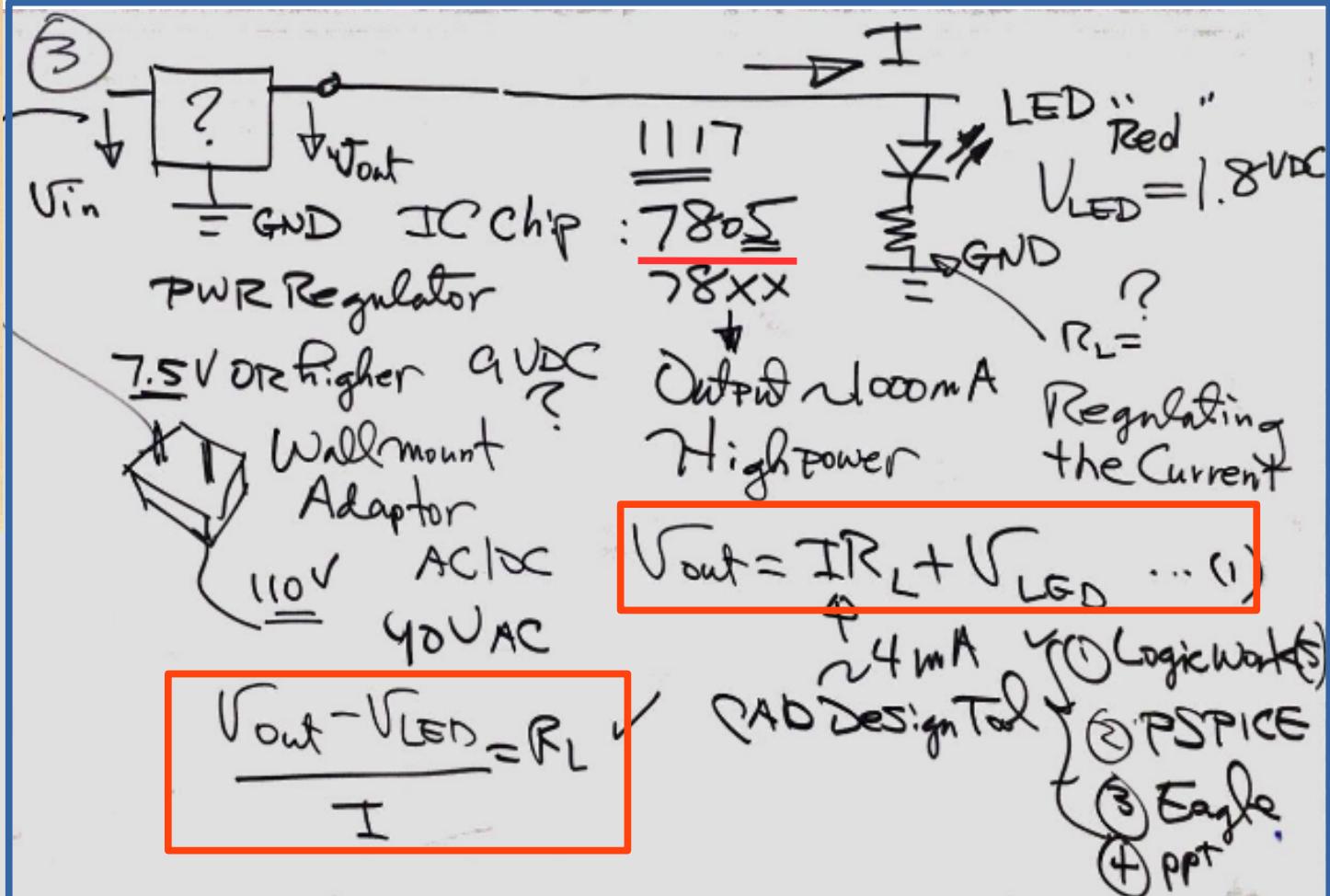
Redlead: Anode (+)

Mfg. hole: 3990-7/32", 5100-1/4"
5210-3/16"

To mount Panel Clip
for 4304 series use 17/64 drill
or punch .265 hole in panel.

For 4305 CH series,
drill 1/4" hole.

MODELS	VOLTS	2 MA	2.5 MA
4300 LC SERIES	1.5 VDC	100 ohms	100 ohms
	3 VDC	620 ohms	470 ohms
RESISTOR: 1/4 WATT	6 VDC	2.2K ohms	1.8K ohms
	12 VDC	5.1K ohms	3.9K ohms
	24 VDC	10K ohms	9.1K ohms



LED Spec

Estimated Power Budget total: 1750 mA

ARM CPU Board

MLS303

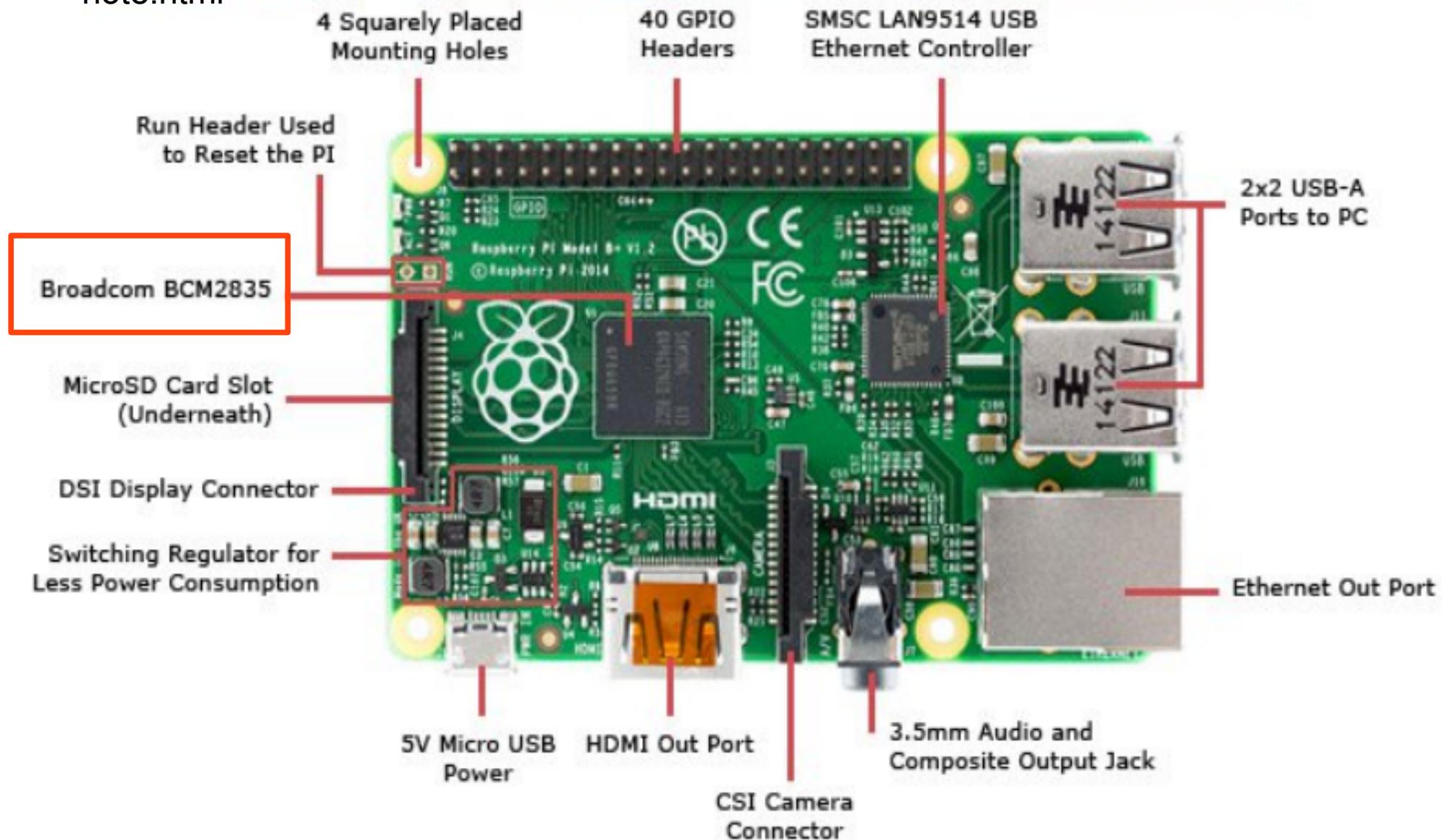
Motor Drive bd

Rest Glue

GPIO Testing

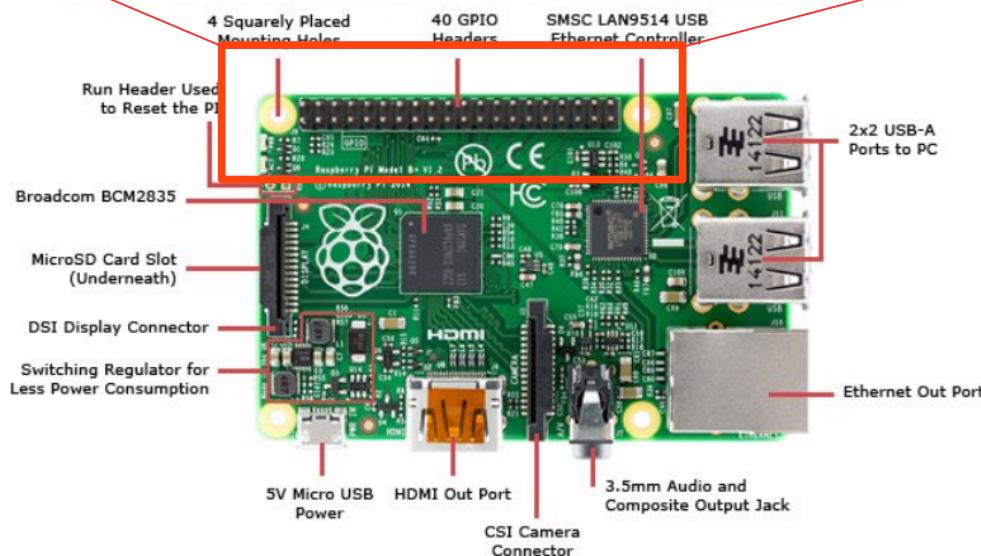
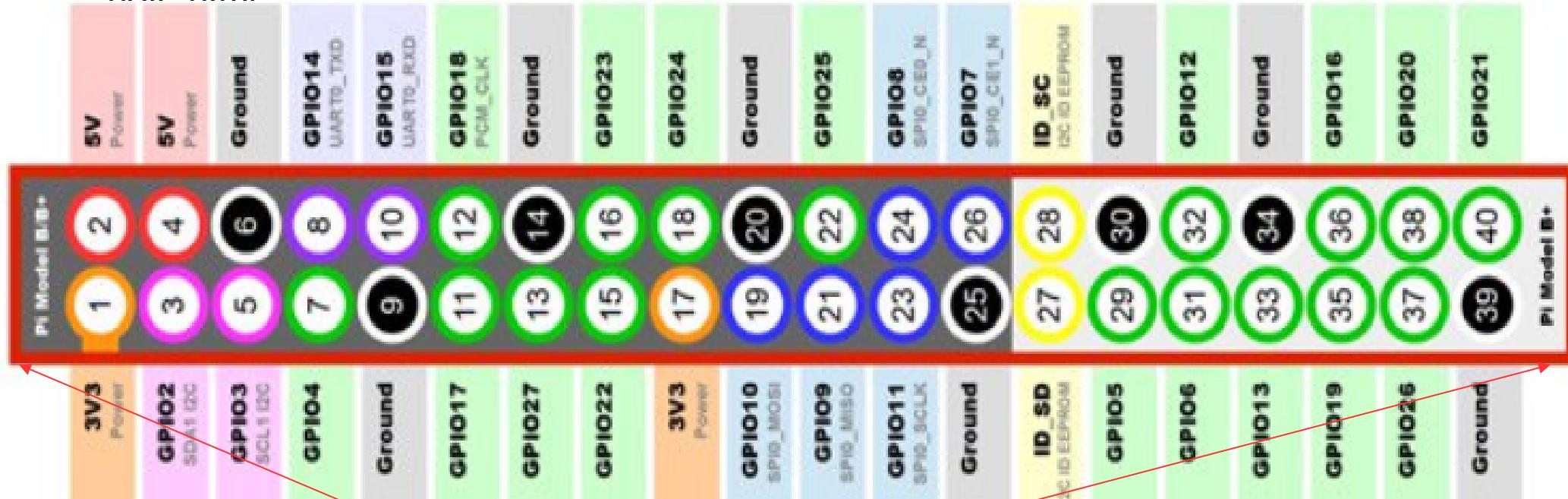
Pie-3 Version B GPIO Pins

<https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html>



Pie-3 Version B GPIO Pins

<https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html>



Python GPIO Interface Testing

```
""  
Date: Feb 2019; Coded by: HL  
sample raspberry pie GPIO code  
""  
  
#import the GPIO and time package  
import RPi.GPIO as GPIO  
import time  
print('----GPIO testing program----')  
print(GPIO.RPI_INFO)  
GPIO.setmode(GPIO.BCM)  
GPIO.cleanup()  
Boardpin = 8  
GPIO.setup(Boardpin, GPIO.OUT)  
# loop 5 times, on/off for 1 second  
for i in range(5):  
    print('GPIO testing program',i)  
    GPIO.output(Boardpin,GPIO.HIGH)  
    time.sleep(1)  
    GPIO.output(Boardpin,GPIO.LOW)  
    time.sleep(1)  
GPIO.cleanup()  
print('End')
```



RPi.GPIO Python package

“Thonny” Python IDE On Pie

The screenshot shows the Thonny Python IDE interface. At the top, the title bar reads "Thonny - /home/pi/harry/gpio/gpio.py @ 15:37". Below it is a menu bar with File, Edit, View, Run, Device, Tools, and Help. A toolbar with various icons follows. The main area contains a code editor with the file "gpio.py" open. The code is as follows:

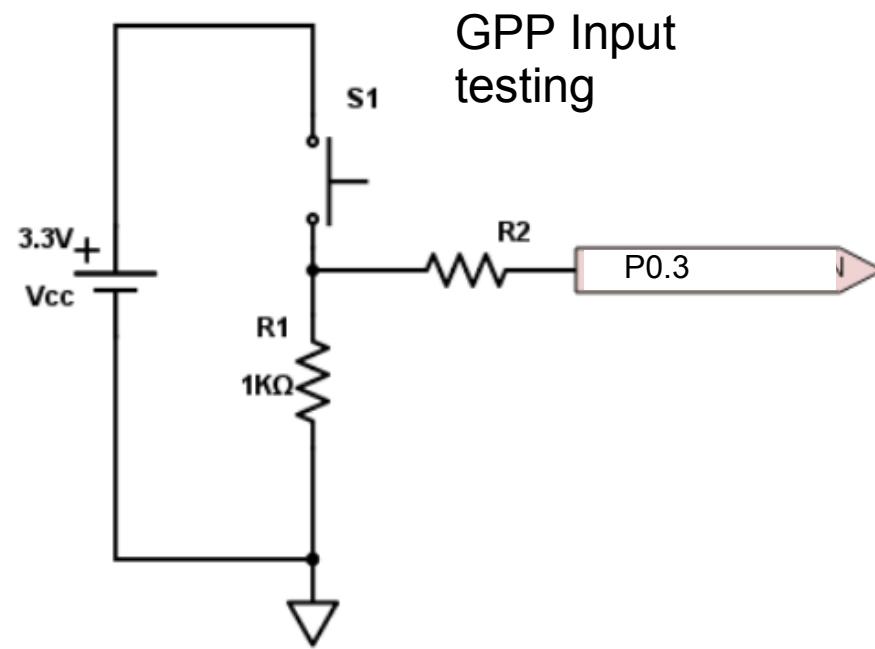
```
4
5 #import the GPIO and time package
6 import RPi.GPIO as GPIO
7 import time
8 print('-----GPIO testing program-----')
9 print(GPIO.RPI_INFO)
10
11 GPIO.setmode(GPIO.BOARD)
12 GPIO.cleanup()
13 Boardpin = 8
14 GPIO.setup(Boardpin, GPIO.OUT)
15 # loop 10 times, on/off for 1 second
16 for i in range(5):
17     print('GPIO testing program',i)
18     GPIO.output(Boardpin,GPIO.HIGH)
19     time.sleep(1)
20     GPIO.output(Boardpin,GPIO.LOW)
21     time.sleep(1)
22 GPIO.cleanup()
```

Below the code editor is a "Shell" window with the following output:

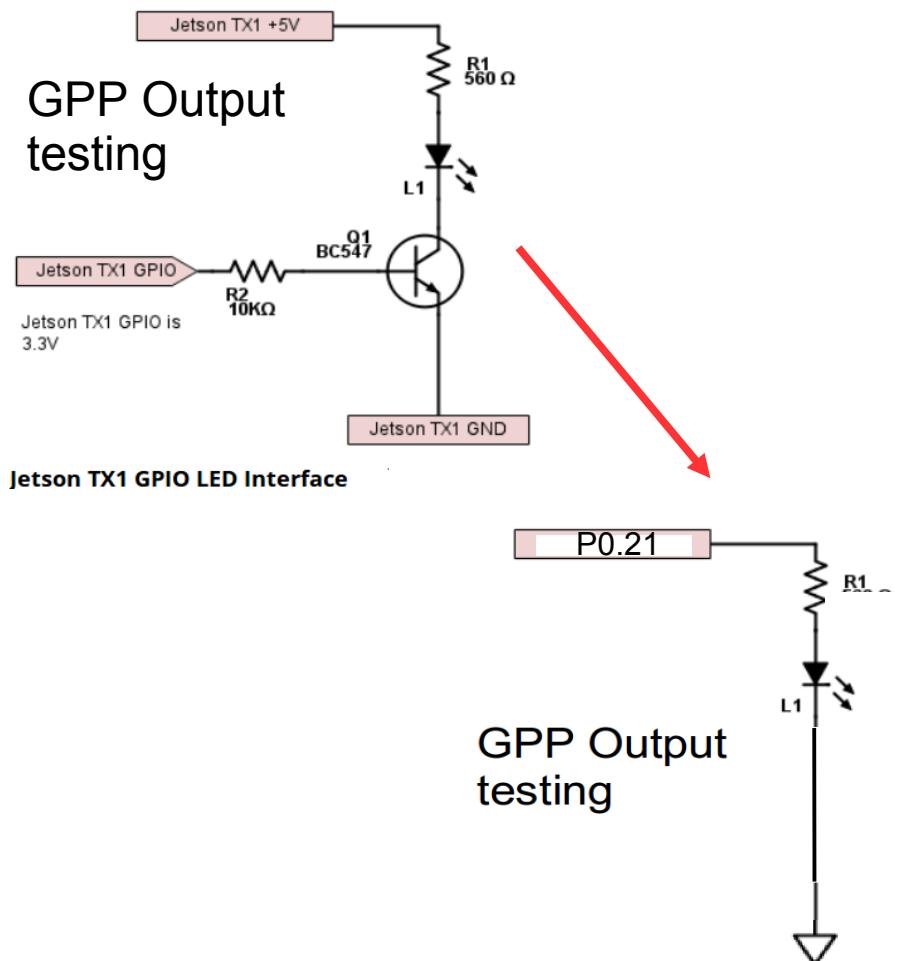
```
gpio.py:12: RuntimeWarning: No channels have been set up yet - nothing to clean up! Try cleaning up at
the end of your program instead!
    GPIO.cleanup()
GPIO testing program 0
GPIO testing program 1
GPIO testing program 2
GPIO testing program 3
GPIO testing program 4
End

>>>
```

GPIO Interface Testing

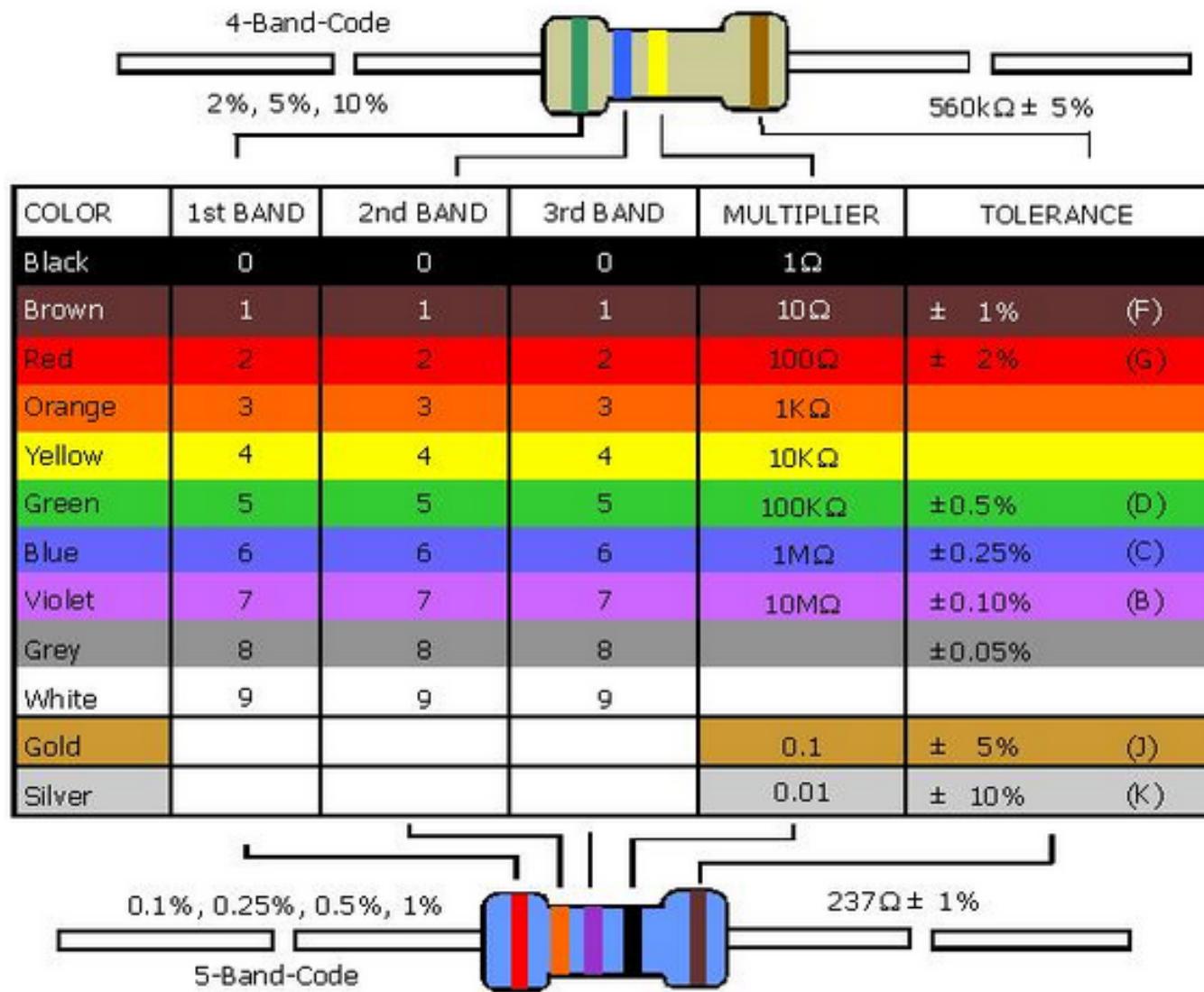


Schematic GPIO



<http://www.jetsonhacks.com/2015/12/29/gpio-interfacing-nvidia-jetson-tx1/>

Color Code Resistor



Color	Digit	Multiplier	Tolerance (%)
Black	0	10^0 (1)	
Brown	1	10^1	1
Red	2	10^2	2
Orange	3	10^3	
Yellow	4	10^4	
Green	5	10^5	0.5
Blue	6	10^6	0.25
Violet	7	10^7	0.1
Grey	8	10^8	
White	9	10^9	
Gold		10^{-1}	5
Silver		10^{-2}	10
(none)			20

Screen Capture with Scrot

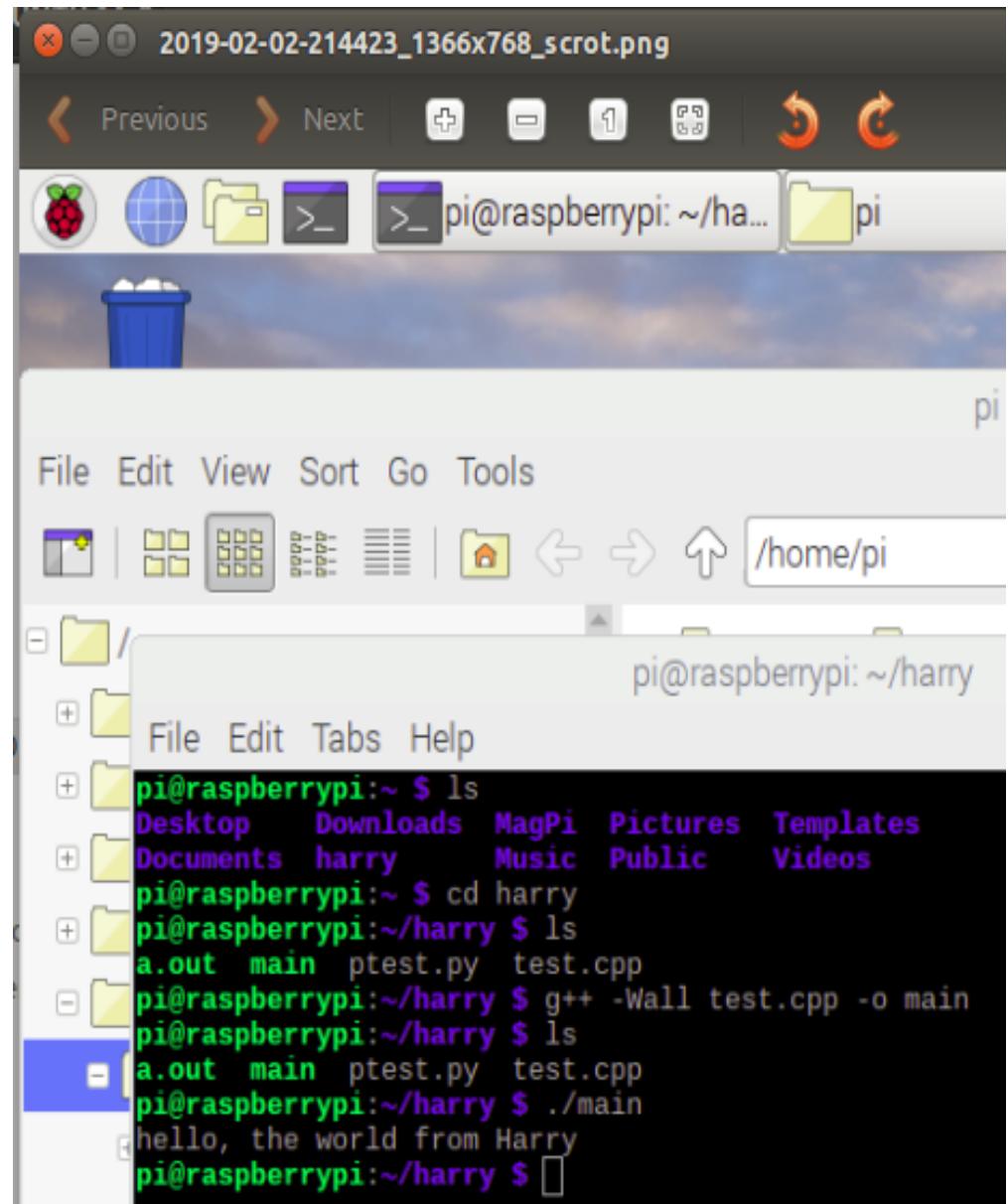
1) Download the scrot a screen capture tool

```
$ sudo apt-get install scrot
```

2) once done, restart your system, then open a terminal do

```
$scrot
```

This will capture the whole screen for you.



Pie-3 C Programming

1) Download the Raspbian Wheezy SD card image from the Raspberry Pi website downloads page

2) Copy it to a SD card and unzip it.

3) Boot your RPi, log in and start the GUI, then select Raspian to boot.

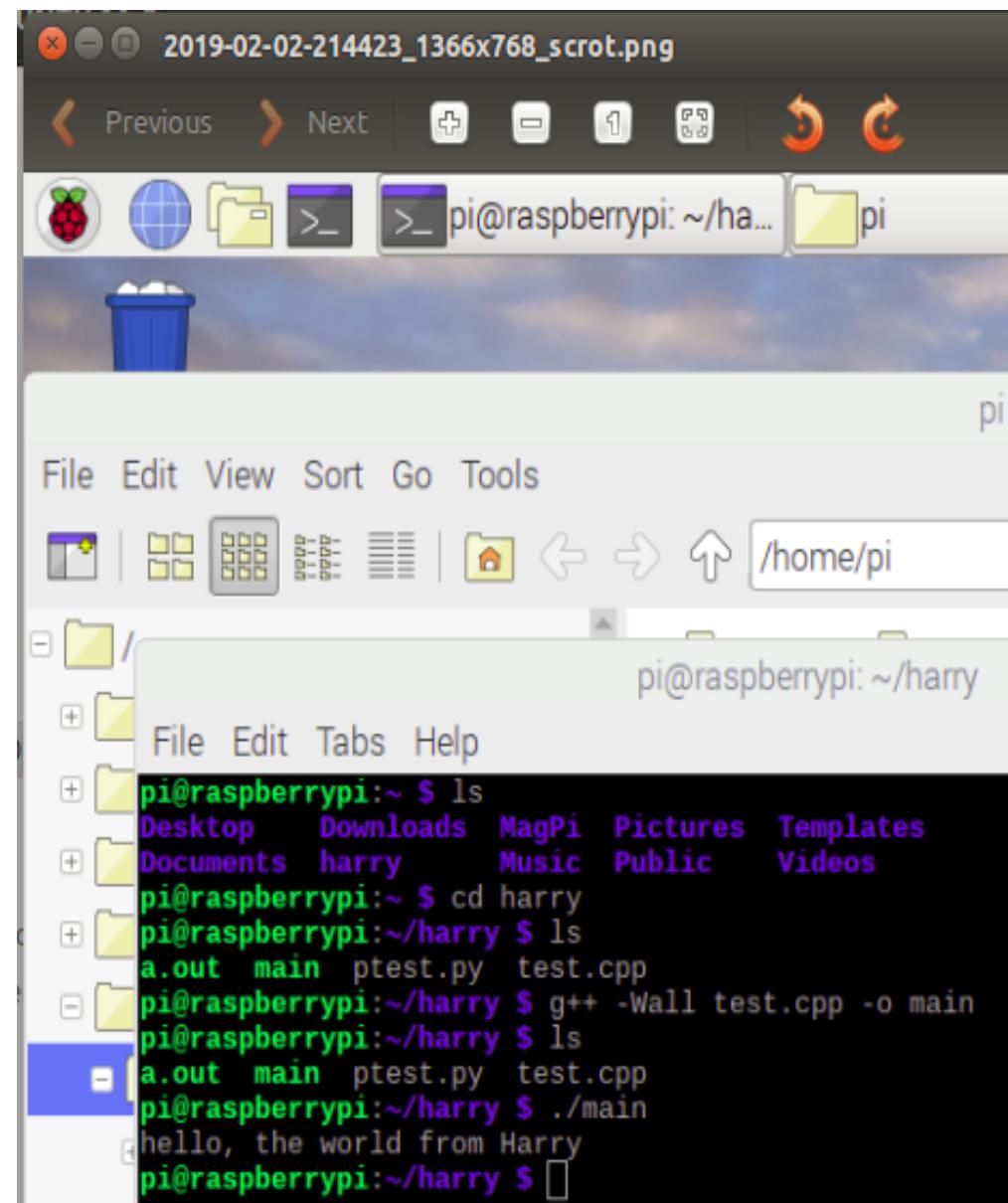
4) Once booted, at the top left select the terminal icon, click on it to open.

5) Then create your working directory, for example, under /home/pi directory, create your working directory.

6) Then use your preferred word editor to create your first test.c program, for example, use "vi" to create test.c.

7) Compile and build: \$gcc -Wall test.cpp -o main

8) To execute the program, \$./main



Configure Device Drivers

<https://pimylifeup.com/raspberry-pi-gpio/>

GPIO is your standard pins that simply be used to turn devices on and off. For example, a LED.

I2C (Inter-Integrated Circuit) pins allow you to connect and talk to hardware modules that support this protocol (I2C Protocol). This protocol will typically take up two pins.

SPI (Serial Peripheral Interface Bus) pins can be used to connect and talk to SPI devices. Pretty much the same as I2C but makes use of a different protocol.

UART (Universal asynchronous receiver/transmitter) is the serial pins used to communicate with other devices.

GPIO Input Testing and exINT

<https://opensourceforu.com/2017/07/introduction-raspberry-pi-gpio-programming-using-python/>

Setting up an input pin

Just like a ground pin was used to complete the circuit of the output pin, the circuit of an input pin should be completed using a ground pin or 3v3 pin. A lone input pin in a circuit is said to be ‘floating’. Since its voltage can be of any value between 0 and 3.3V, it cannot be used. That should be avoided by using a 3v3 pin or a ground pin and an in-built pull up or pull down resistor.

A pin can be set as input as follows:

```
GPIO.setup(channel, GPIO.IN, PUD)
```

Here, the channel is the pin number. PUD can be either GPIO.PUD_DOWN or GPIO.PUD_UP. It tells you whether to use the inbuilt pull up or the pull down resistor. If the 3v3 pin is used, we have to use GPIO.PUD_DOWN (pull down resistor), and for the ground pin, we need to use GPIO.PUD_UP (pull up resistor). The choice of 3v3 or ground is up to you. The only difference is the value read from an open circuit and closed circuit. For the 3v3 pin (GPIO.PUD_DOWN), the input value of the open circuit will be 0 and of a closed circuit will be 1. For a ground pin (GPIO.PUD_UP), open circuit is 1 and closed circuit is 0.