Raspberry Pi - Hardware Interface

ECE 4564 - Network Application Design

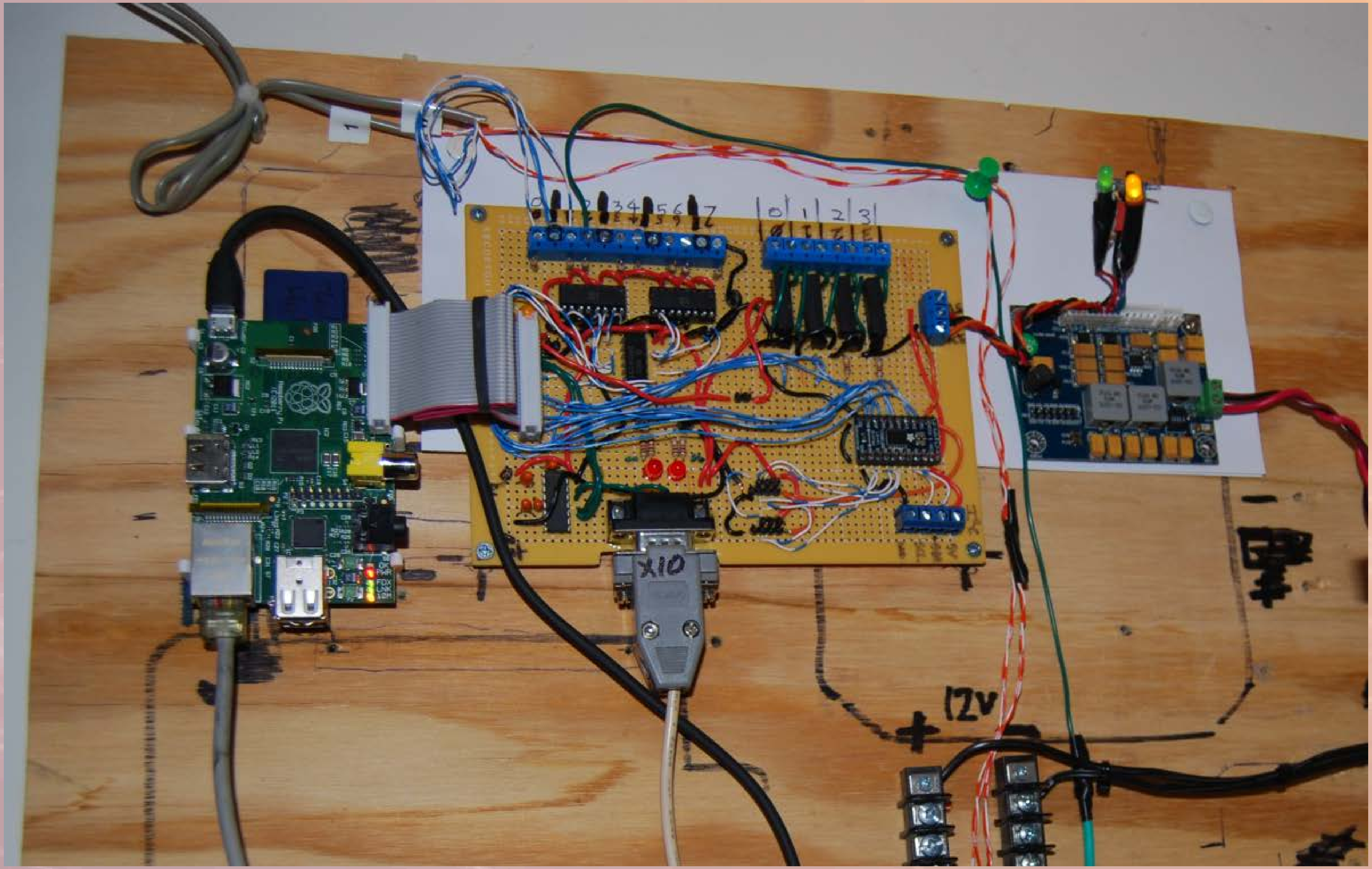
Dr. William O. Plymale

Topics

Interactive Hardware

- General-Purpose Input/Output
- Raspberry Pi – GPIO
- Sysfs
- Python Rpi.GPIO Module

Interactive Hardware



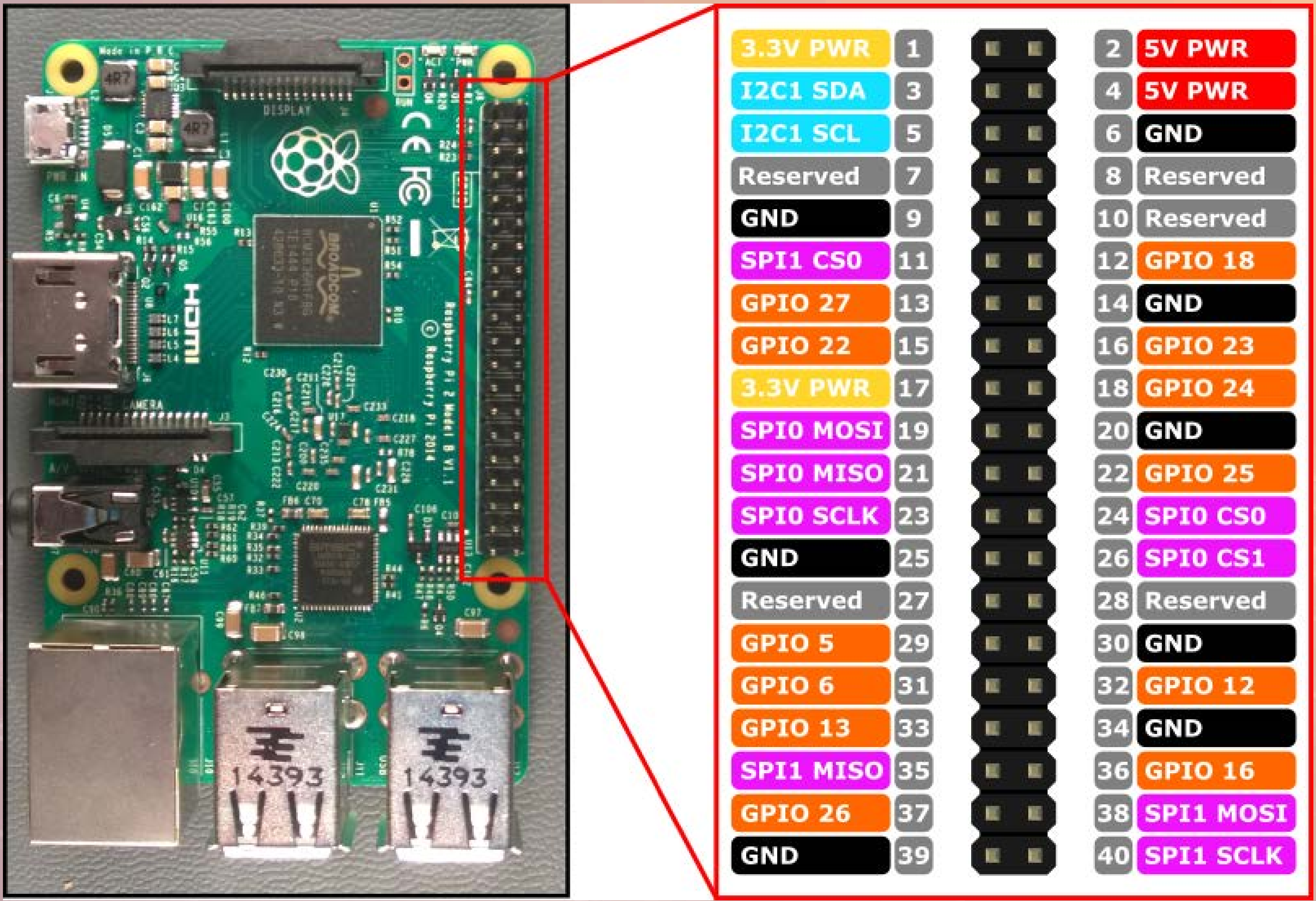
General-Purpose Input/Output

A generic pin on a microcontroller whose behavior, including whether it is an input or output pin, can be controlled by the user at run time.

GPIO capabilities may include:

- GPIO pins can be configured to be input or output
- GPIO pins can be enabled/disabled
- Input values are readable (typically high=1, low=0)
- Output values are writable/readable
- Input values can often be used as IRQs (typically for wakeup events)

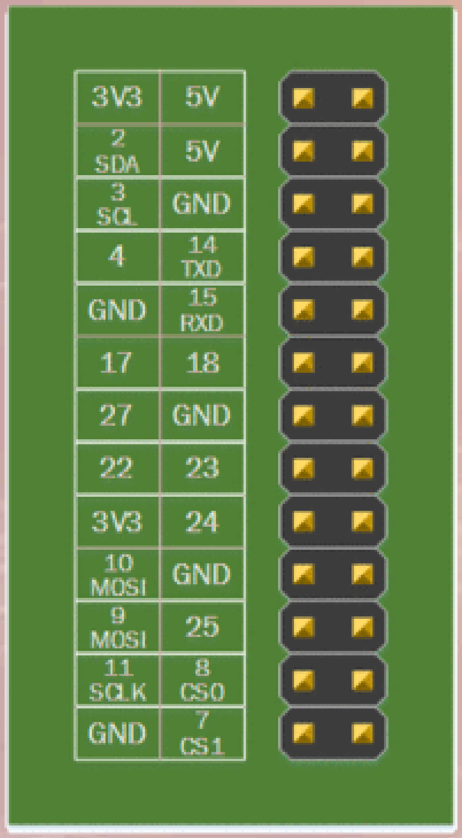
Raspberry Pi GPIO



3.3V PWR	1	2	5V PWR
I2C1 SDA	3	4	5V PWR
I2C1 SCL	5	6	GND
Reserved	7	8	Reserved
GND	9	10	Reserved
SPI1 CS0	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3.3V PWR	17	18	GPIO 24
SPI0 MOSI	19	20	GND
SPI0 MISO	21	22	GPIO 25
SPI0 SCLK	23	24	SPI0 CS0
GND	25	26	SPI0 CS1
Reserved	27	28	Reserved
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
SPI1 MISO	35	36	GPIO 16
GPIO 26	37	38	SPI1 MOSI
GND	39	40	SPI1 SCLK

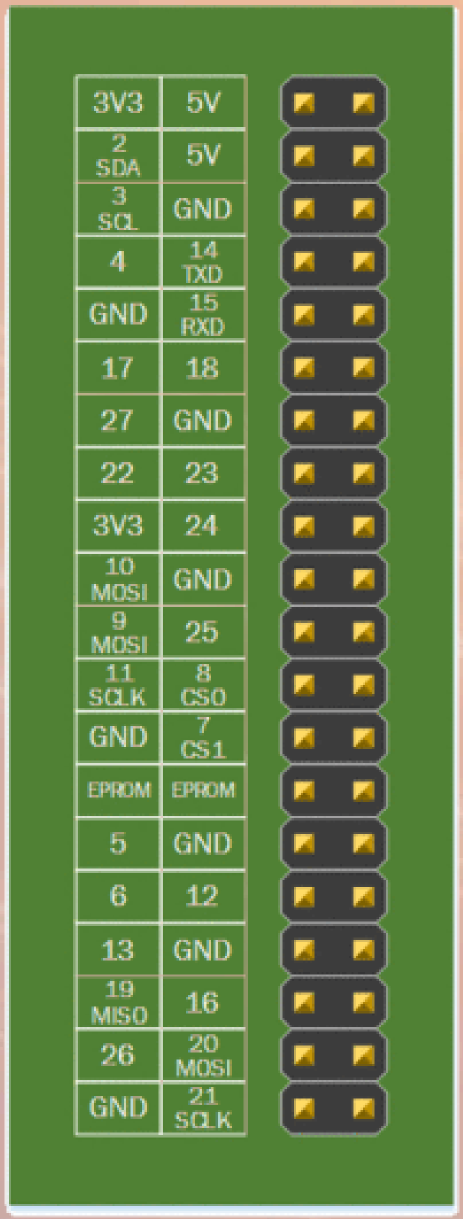
Raspberry Pi

Models A & B



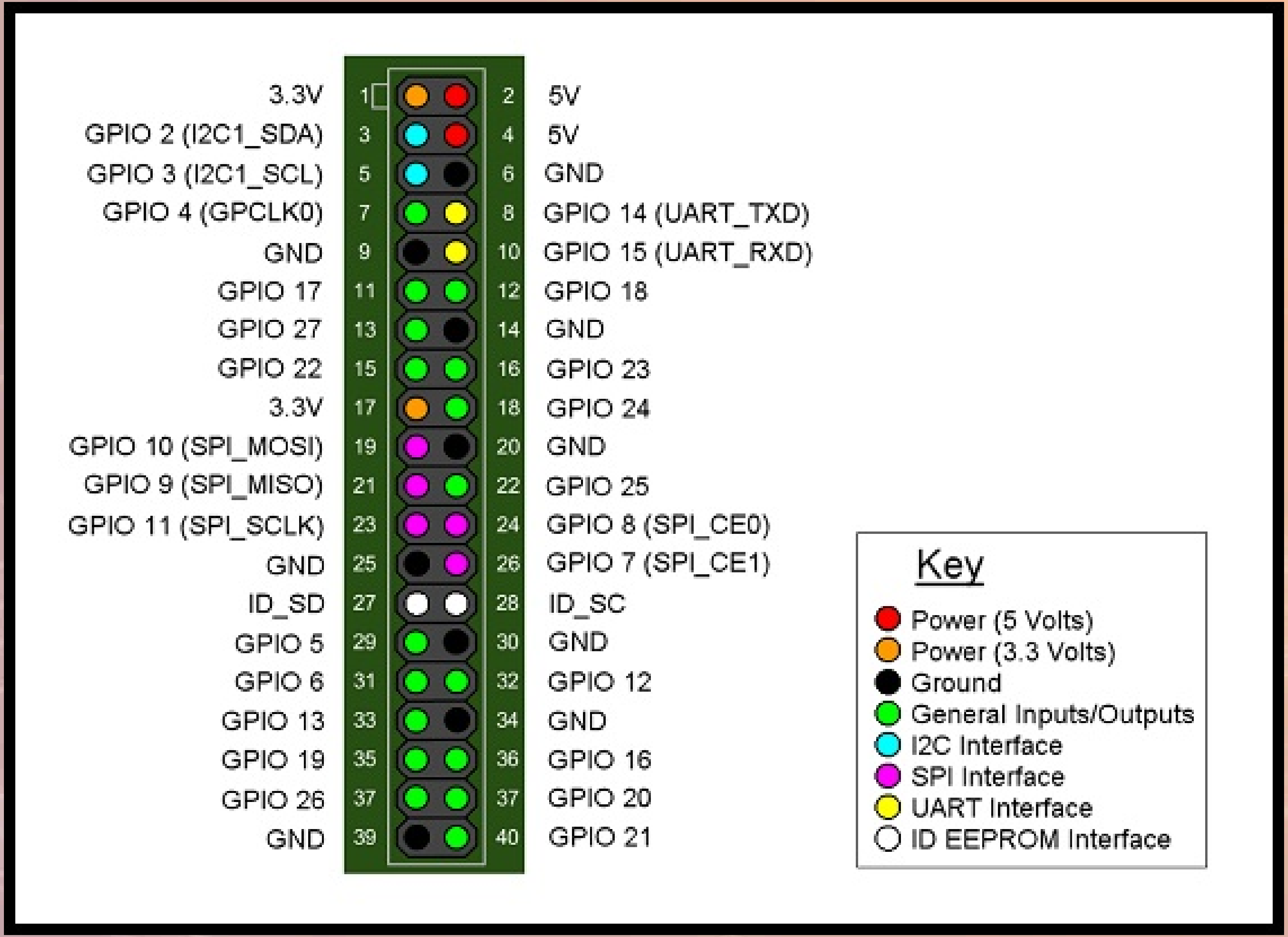
3V3	5V
2 SDA	5V
3 SCL	GND
4	14 TXD
GND	15 RXD
17	18
27	GND
22	23
3V3	24
10 MOSI	GND
9 MOSI	25
11 SCLK	8 CS0
GND	7 CS1

Models A+, B+ & Pi2



3V3	5V
2 SDA	5V
3 SCL	GND
4	14 TXD
GND	15 RXD
17	18
27	GND
22	23
3V3	24
10 MOSI	GND
9 MOSI	25
11 SCLK	8 CS0
GND	7 CS1
EPROM	EPROM
5	GND
6	12
13	GND
19 MISO	16
26	20 MOSI
GND	21 SCLK

Raspberry Pi



3.3V	1	🟠	🔴	2	5V
GPIO 2 (I2C1_SDA)	3	🔵	🔴	4	5V
GPIO 3 (I2C1_SCL)	5	🔵	⚫	6	GND
GPIO 4 (GPCLK0)	7	🟢	🟡	8	GPIO 14 (UART_TXD)
GND	9	⚫	🟡	10	GPIO 15 (UART_RXD)
GPIO 17	11	🟢	🟢	12	GPIO 18
GPIO 27	13	🟢	⚫	14	GND
GPIO 22	15	🟢	🟢	16	GPIO 23
3.3V	17	🟠	🟢	18	GPIO 24
GPIO 10 (SPI_MOSI)	19	🟣	⚫	20	GND
GPIO 9 (SPI_MISO)	21	🟣	🟢	22	GPIO 25
GPIO 11 (SPI_SCLK)	23	🟣	🟣	24	GPIO 8 (SPI_CE0)
GND	25	⚫	🟣	26	GPIO 7 (SPI_CE1)
ID_SD	27	⚪	⚪	28	ID_SC
GPIO 5	29	🟢	⚫	30	GND
GPIO 6	31	🟢	🟢	32	GPIO 12
GPIO 13	33	🟢	⚫	34	GND
GPIO 19	35	🟢	🟢	36	GPIO 16
GPIO 26	37	🟢	🟢	37	GPIO 20
GND	39	⚫	🟢	40	GPIO 21

Key

- 🔴 Power (5 Volts)
- 🟠 Power (3.3 Volts)
- ⚫ Ground
- 🟢 General Inputs/Outputs
- 🔵 I2C Interface
- 🟣 SPI Interface
- 🟡 UART Interface
- ⚪ ID EEPROM Interface

GPIO Reference

More on Pin Numbering

The GPIO pins are sometimes renamed with another set of numbers.

In order to avoid damaging your Pi you need to be sure what pins you are connecting to other hardware and that your program is referring to the correct pins.

http://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering

http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/

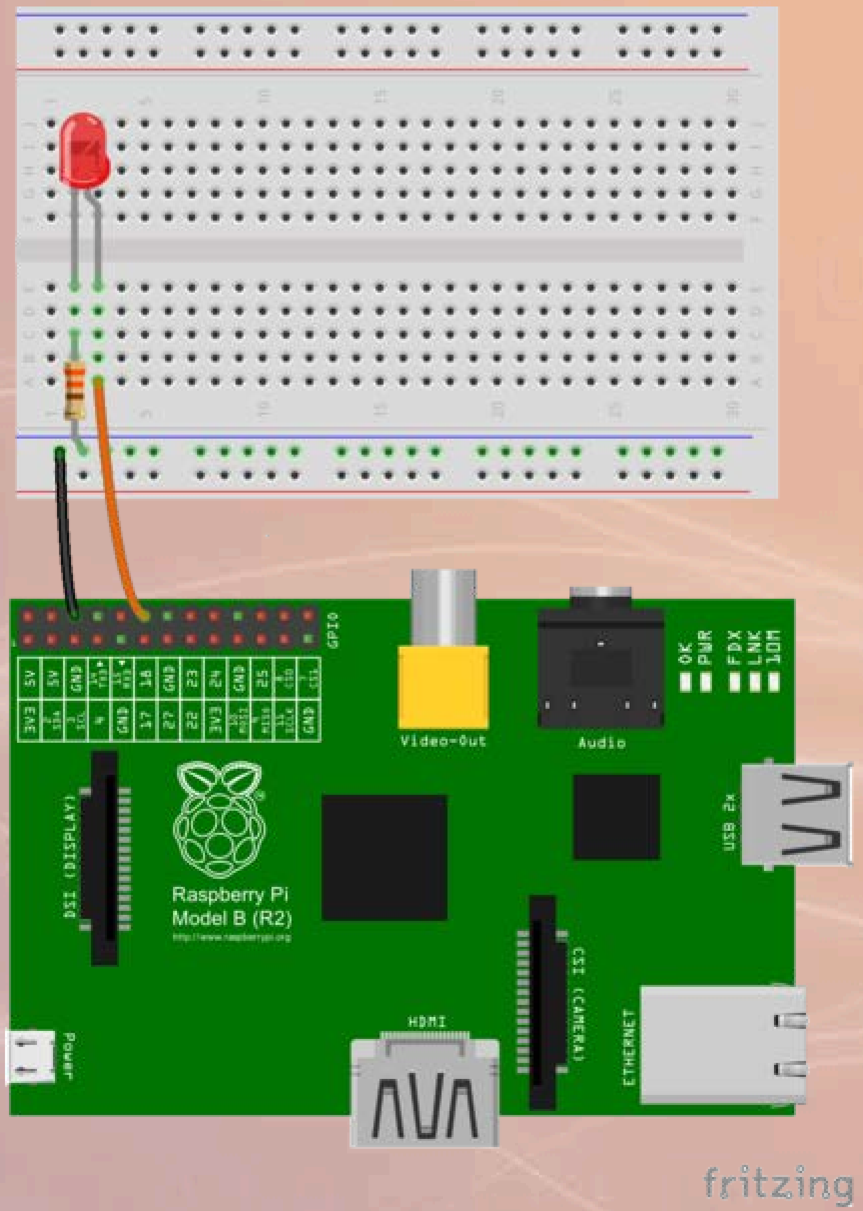
GPIO Pins – Raspberry Pi

- GPIO voltage levels are 3.3 V and are not 5 V tolerant.
- There is no over-voltage protection on the board
  - the intention is that people interested in serious interfacing will use an external board with buffers, level conversion and analog I/O rather than soldering directly onto the main board.

(Sending 5V to a pin may kill the Pi)

Rpi Low-level Peripherals

Raspberry Pi



Turning on an LED

GPIO with sysfs on Raspberry Pi

- In Linux everything is a file: /dev/ttyUSB0, /sys/class/net/eth0/address, /dev/mmcblk0p2,…
- sysfs is a kernel module providing a virtual file system for device access at /sys/class
  - provides a way for users (or code in the user-space) to interact with devices at the system (kernel) level
- Advantages / Disadvantage
  - Allows conventional access to pins from userspace
  - Much slower the digitalWrite()/digitalRead() of Arduino

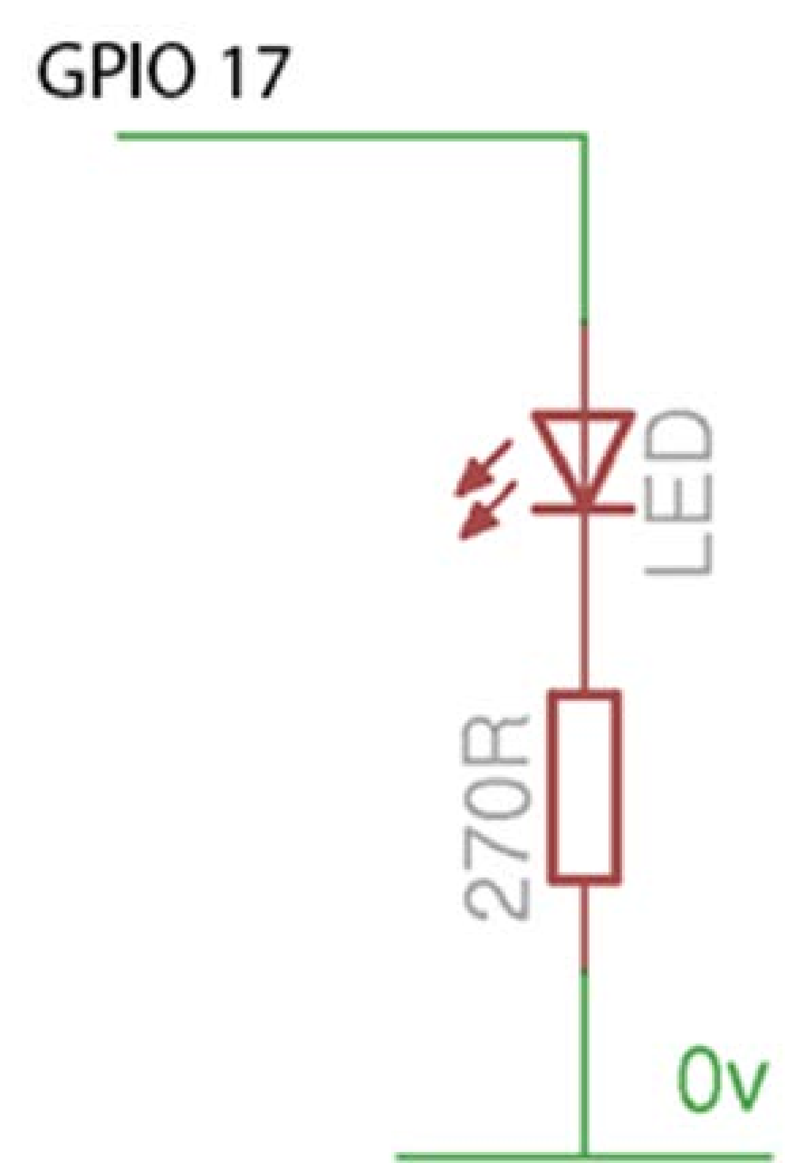
/sys/class/gpio

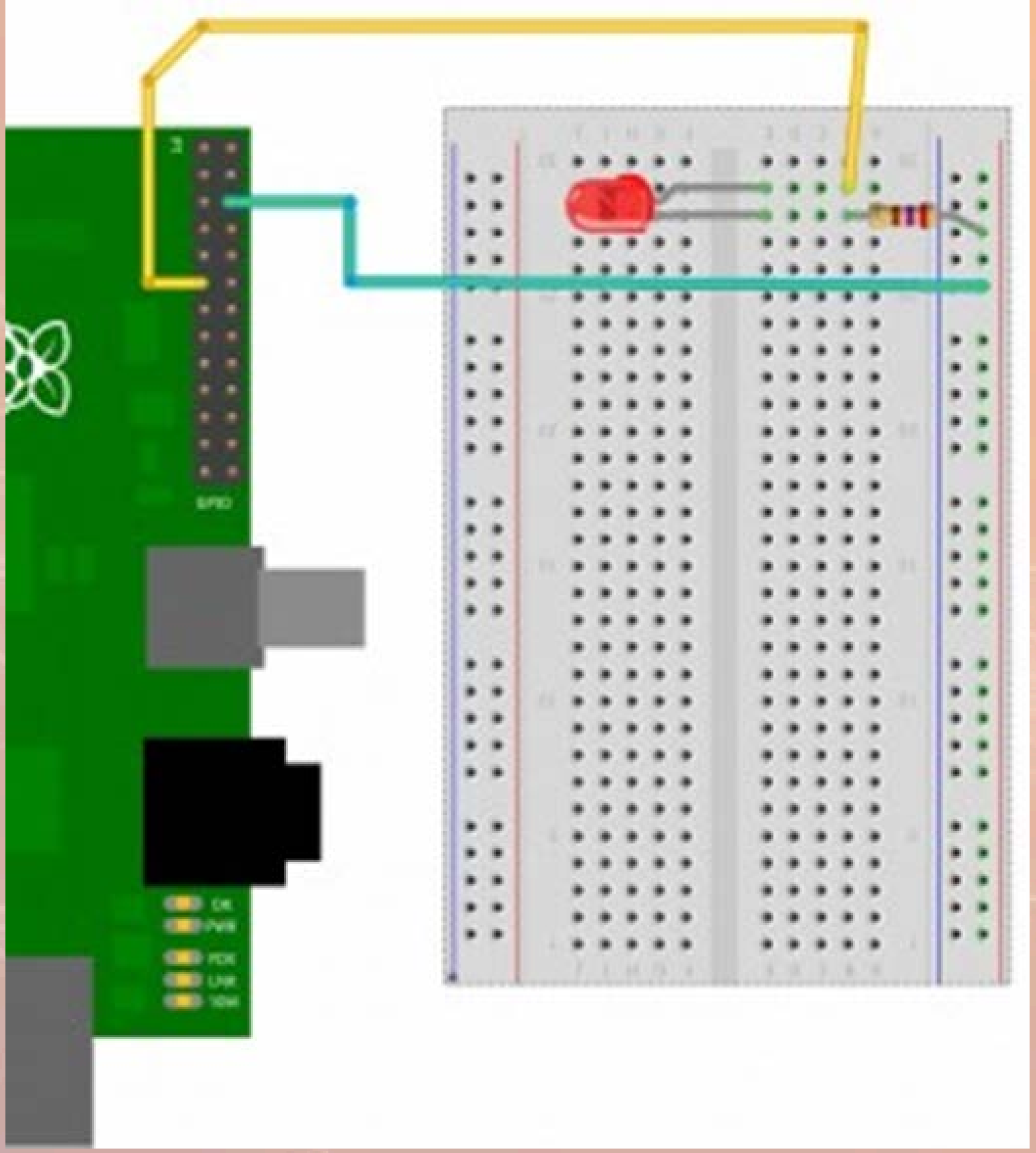
- Explore this directory
- As *root,* cd /sys/class/gpio
- List files
  - export
  - gpiochip0 – sym link
  - unexport
- Create the sysfs alias for a pin by *exporting* the pin
  - echo 4 > export
- sysfs monitors these files, and updates the links between user-space and kernel-space when they're updated
- When finished, echo 4 > unexport

/ sys / class / gpio

- Export the pin we want to use
  - Write the pin number to /sys/class/gpio/export
    - echo 17 > /sys/class/gpio/export
- Set the direction
  - Write "in" or "out" to /sys/class/gpio/gpio??/direction
    - echo out > /sys/class/gpio/gpio17/direction
- Set the value
  - Write "1" or "0" to /sys/class/gpio/gpio??/value
    - echo 1 > /sys/class/gpio/gpio17/value

Connect an LED between GPIO 17 (P1-11) and GND





blink.sh

```
#!/bin/sh
echo 17 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
while true
do
        echo 1 > /sys/class/gpio/gpio17/value
        sleep 1
        echo 0 > /sys/class/gpio/gpio17/value
        sleep 1
done
```

Rpi.GPIO

A module to control Raspberry Pi GPIO channels

RPi.GPIO 0.7.0

Installation

Rpi.GPIO Demo

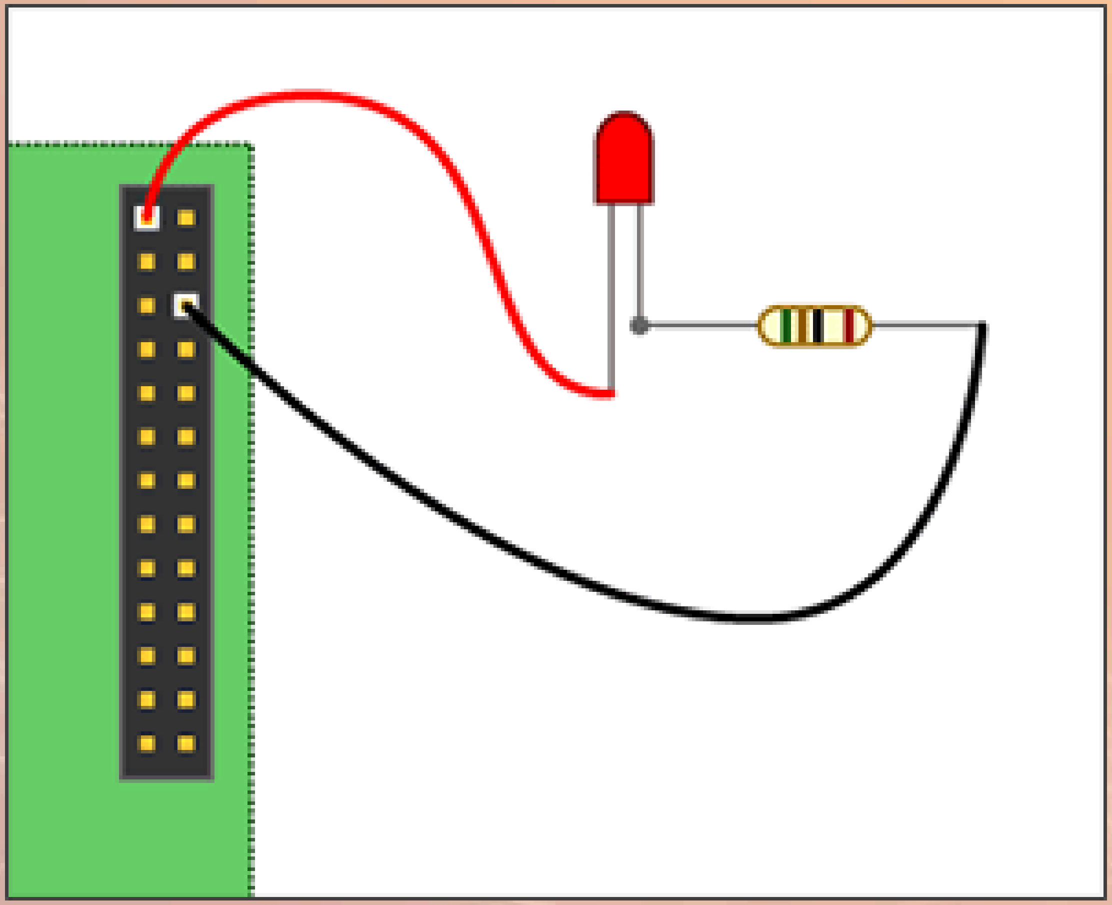
```
#!/usr/local/bin/python

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(17, GPIO.OUT)
GPIO.output(17, False)

while True:
        GPIO.output(17, True)
        time.sleep(2)
        GPIO.output(17, False)
        time.sleep(2)
```



Emulator

PWM Control

Pulse width modulation (PWM) is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts.

PWM in Python

Flex Sensor

Flex Sensor with Raspberry Pi

Absolute Orientation Sensor

Absolute Orientation Sensor

Stepper Motor

Stepper Motor

Closing