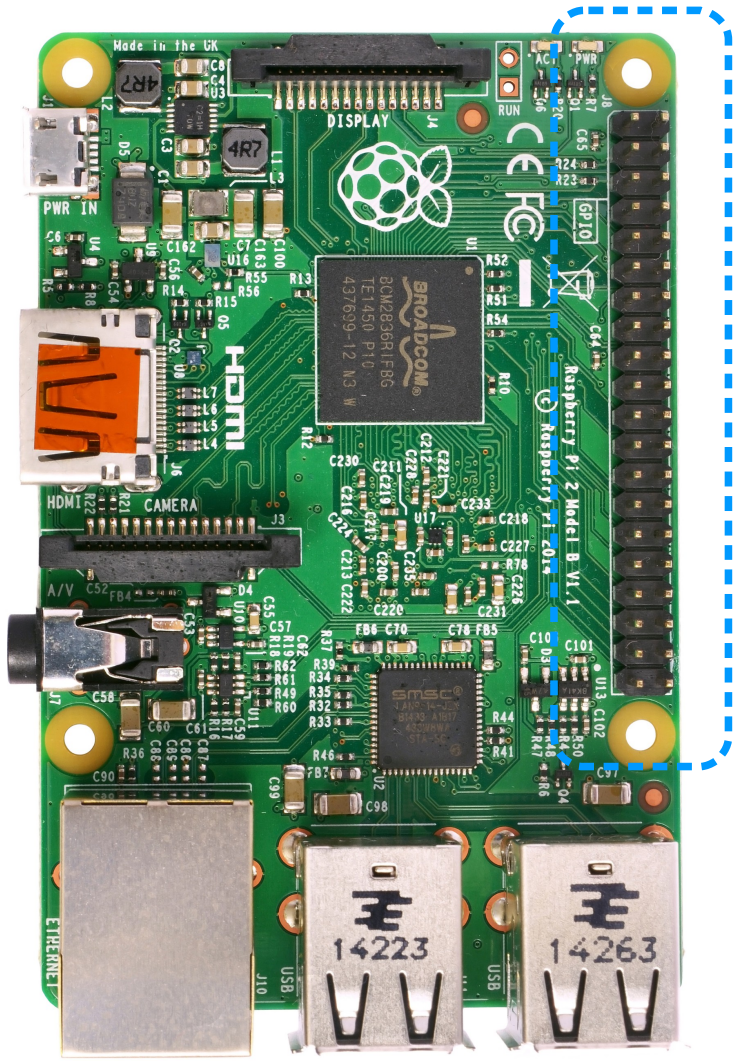
Connecting to Electronics

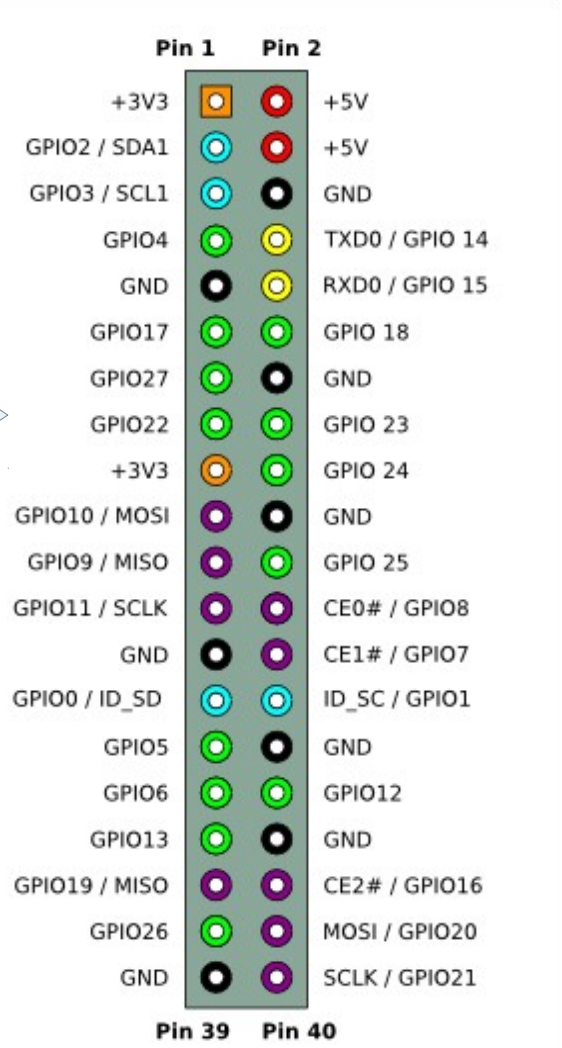
with the Raspberry Pi

http://www.fnc.co.uk

Raspberry Pi Day
University of Strathclyde
12/12/2015

GPIO Header





http://elinux.org/RPi_Low-level_peripherals

Connections

- **GPIO** – 3.3V logic
- **SPI** (Serial Peripheral Interface)
- **PWM** (Pulse width modulation)
- **I²C** (Inter Integrated Circuit Communications)
- **UART** (Universal Asynchronous Receiver Transmitter)

GPIO properties

- The GPIO pins will accept 0 or 3.3V
  - There is no protection on the board.
  - Putting a 5V signal into a GPIO pin will destroy a Raspberry Pi.
  - They will source or sink up to 16mA.
- Can be configured as inputs or outputs
  - Configurable internal pull-up and pull-down resistors.

Possible GPIO uses

- Can be connected to 3.3V logic devices.
  - A passive infrared sensor is one example.
  - Be careful though, since some sensors when powered with 5V will produce 5V logic.
  - Possible to use a logic level converter board to switch from 5V to 3.3V.
- Can be used to drive low power devices.
  - A single LED connected in series with a 300Ohm resistor to a ground terminal.
- Can connect a switch to a GPIO pin and then to ground.
  - Using the internal pull up resistor.

Driving LEDs

```
#!/usr/bin/env python
import time
import RPi.GPIO as GPIO

red = 4
green = 25
blue = 24
yellow = 23
leds = [ red, green, blue, yellow ]

waitTime = 0.1

GPIO.setmode(GPIO.BCM)

for led in leds:
  GPIO.setup(led,GPIO.OUT)
  GPIO.output(led,0)
```

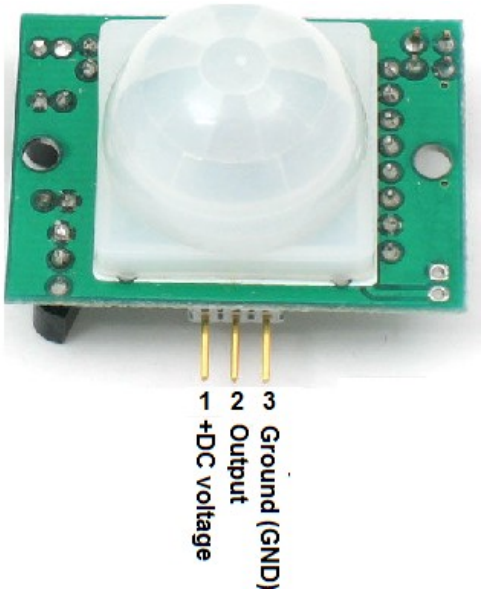
```
try:
  print("Type CTRL-C to quit the loop")
  while True:
    for led in leds:
      GPIO.output(led,1)
      time.sleep(waitTime)
    for led in leds:
      GPIO.output(led,0)
      time.sleep(waitTime)


except KeyboardInterrupt:
  GPIO.cleanup()
```

git clone https://github.com/williamhbell/GpioExamples.git
./GpioExamples/SimpleLED/python/LED.py

Need to use sudo on Wheezy, but not on Jessie

Connecting to a PIR

```
#!/usr/bin/env python
import time
import RPi.GPIO as GPIO

pir = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(pir, GPIO.IN)

try:
  print("Now checking the PIR.")
  print("Type CTRL-C to quit the loop.")

  while True:
    if GPIO.input(pir):
      print("The PIR has triggered")

    time.sleep(0.5)

except KeyboardInterrupt:
  GPIO.cleanup()
```



git clone https://github.com/williamhbell/GpioExamples.git
./GpioExamples/PIR/python/simplePIR.py

Need to use sudo on Wheezy, but not on Jessie

Connected to a switch

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

def callback_down(channel):
  print('channel %s'%channel)

inputPin = 10
GPIO.setmode(GPIO.BCM)
GPIO.setup(inputPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

try:
  GPIO.add_event_detect(inputPin, GPIO.FALLING, callback=callback_down)
  while True:
    time.sleep(100)
except KeyboardInterrupt:
  GPIO.cleanup()
```

git clone https://github.com/williamhbell/GpioExamples.git
./GpioExamples/Switch/python/switch.py

Need to use sudo on Wheezy, but not on Jessie

I²C

- Use `raspi-config` to enable I²C in the kernel
- `sudo apt-get install i2c-tools`
- Shutdown the Raspberry Pi
- Plug in the I²C device(s)
  - More than one device can be connected in parallel
  - Read the data-sheet and markings on the board.
  - If you have a hat, then it should be obvious how to plug it in.
- Power up and check for connected devices
  - `i2cdetect -y 1`
  - Each device has its own address.
- Board may come with drivers or someone else might have written them.

Summary

- There are lots of different devices that can be safely connected to the Raspberry Pi
  - Protect the Raspberry Pi by using a cheaper component to interface with analogue I/O.
- Break projects up into steps and then look for suitable hardware modules.
  - GPS, ADC, DAC, H-bridge, DIO and more.