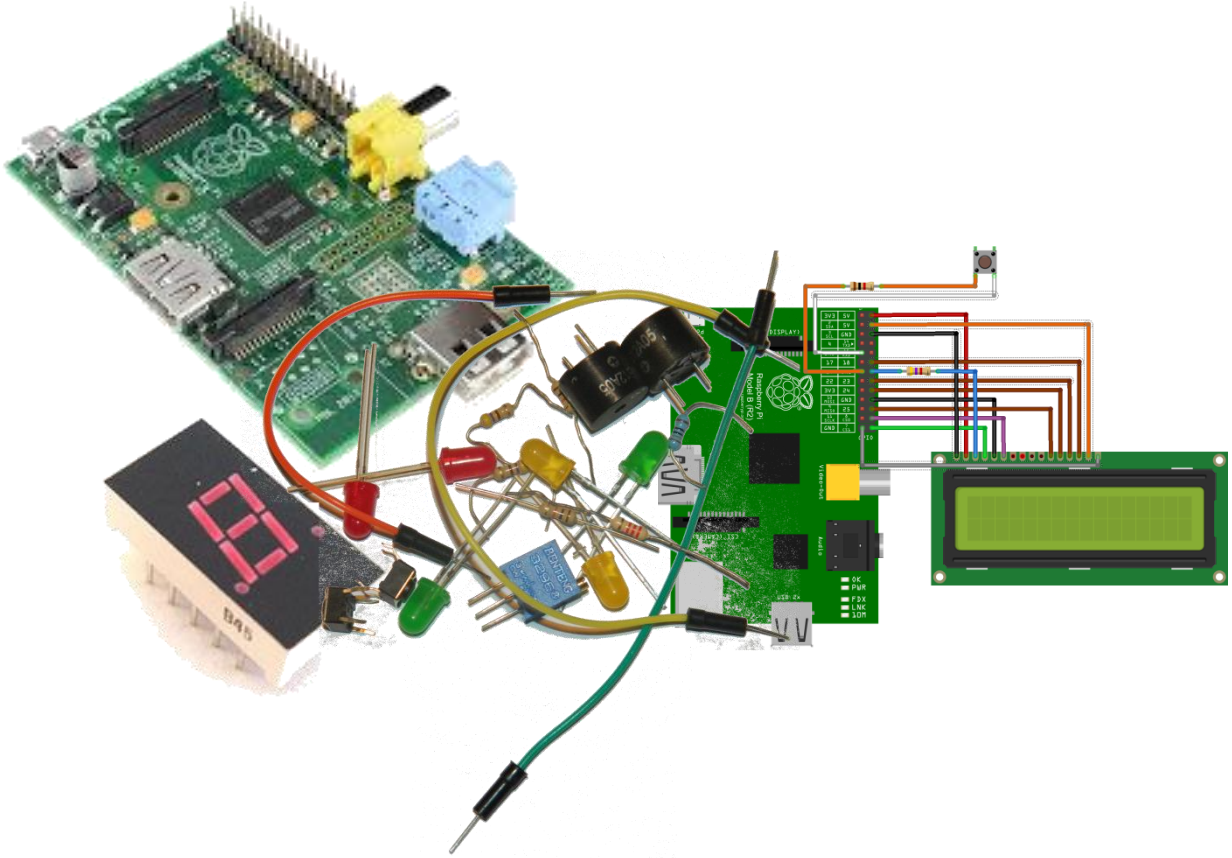
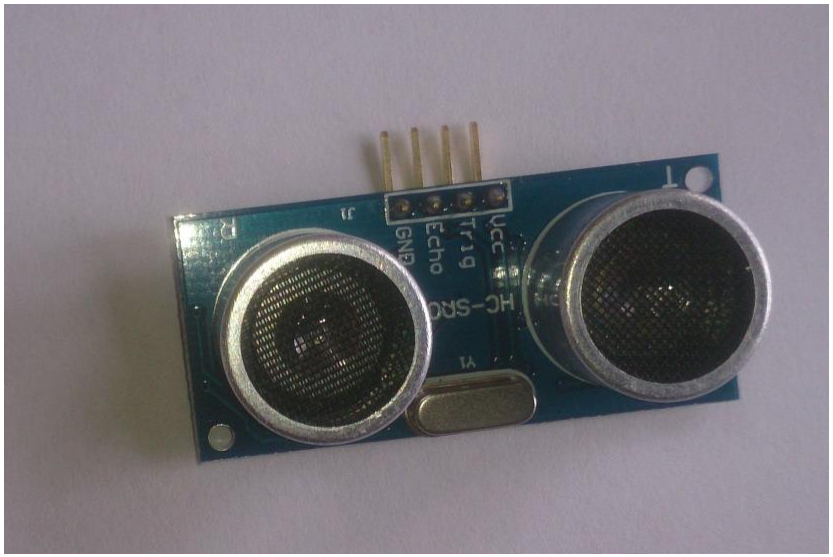
Raspberry Pi and Electronics



Distance Sensor

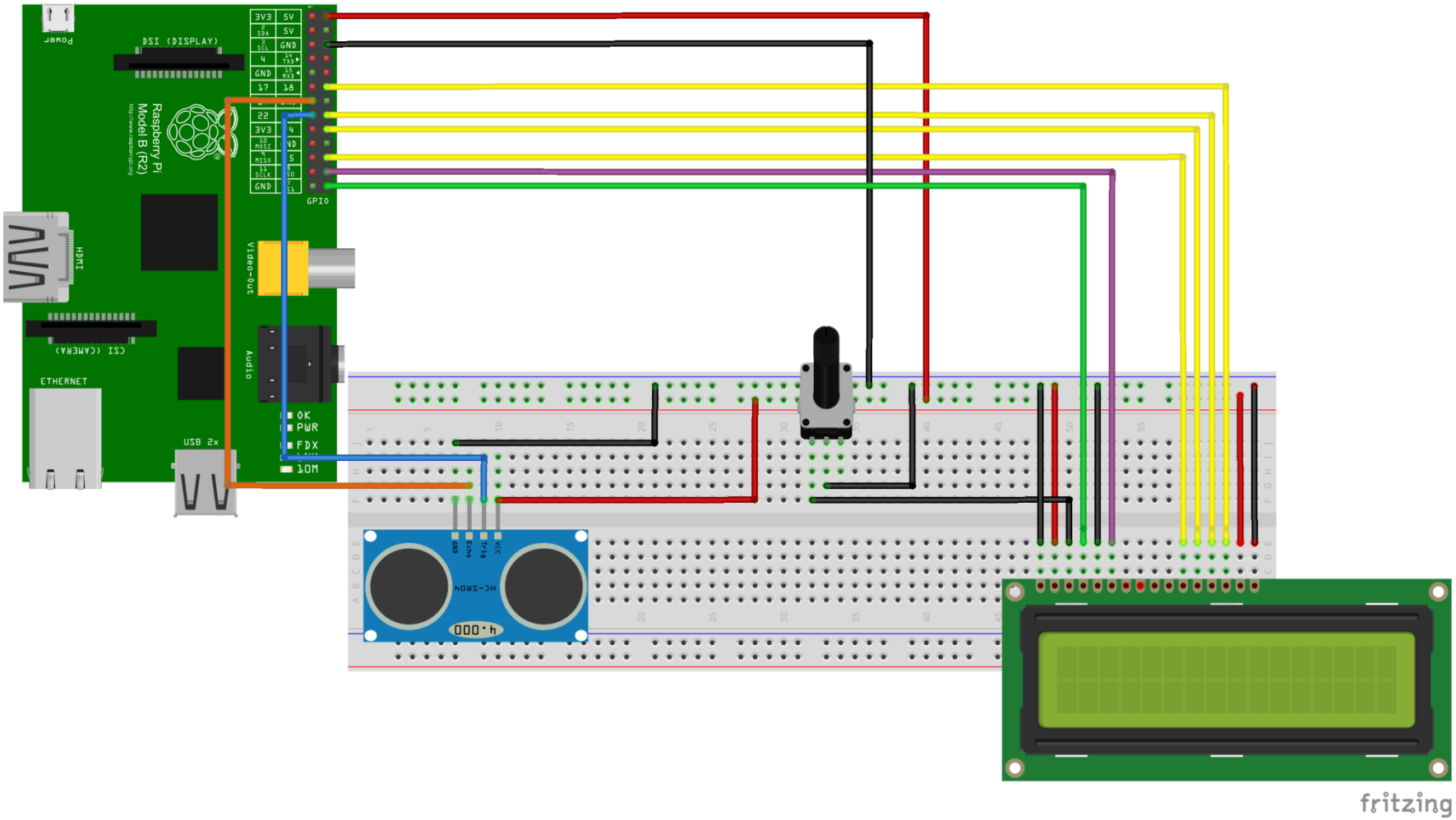


Today we're going to take measurements with the HC-SR04 distance sensor and output these to our LCD Displays

The HC-SR04

The HC-SR04 module costs around 5 Euros and is the size of a box of matches. It has a 5V pin, an input pin, an output pin and a ground (0v) pin. The module works by sending an ultrasonic pulse into the air and measuring the time it takes to bounce back. This value can then be used to calculate the distance the pulse travelled.

Wiring Diagram



Wiring the HC-SR04

Powering the module is easy. We'll just connect the +5V and Ground pins to the 5v and gnd rails on our breadboard. The input pin on the module is called the "Trig" as it is used to trigger the sending of the pulse. Ideally it takes a 5V signal but it works fine with a 3.3V signal from the GPIO. We'll connect it to GPIO 27 on our Raspberry Pi.

Wiring the HC-SR04

The module's output is called the "echo". The output pin is low (0V) until the module has taken its distance measurement. It then sets this pin high (+5V) for the same amount of time that it took the pulse to return. So our script needs to measure the time this pin stays high to calculate the distance.

Wiring the HC-SR04

There is a slight problem with the output the module uses a +5V level for a “high” but this is too high for the inputs on the GPIO pins which only like 3.3V. In order to ensure the Pi doesn’t get damaged we use a 10 K ohm resistor to protect it. We’ll connect the echo pin to GPIO pin 22 on our Pi.

Our Code

```
#!/usr/bin/python3

import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO_TRIGGER = 22
GPIO_ECHO = 27

print("Ultrasonic Measurement")

GPIO.setup(GPIO_TRIGGER,GPIO.OUT)
GPIO.setup(GPIO_ECHO,GPIO.IN)

GPIO.output(GPIO_TRIGGER,False)
time.sleep(0.5)
GPIO.output(GPIO_TRIGGER,True)
time.sleep(0.001)
GPIO.output(GPIO_TRIGGER,False)
start = time.time()
while GPIO.input(GPIO_ECHO)==0:
    start = time.time()

while GPIO.input(GPIO_ECHO)==1:
    stop = time.time()

elapsed = stop - start
distance = elapsed * 34000
distance = distance / 2

print('distance = ', str(round(distance,-1)))

GPIO.cleanup()
```

This weeks Challenge

Next week we'll use the code we've just looked at and the code we used for our jokes last week to write our measurements to our LCD Displays