MIT AI2 204

IoT with MIT App Inventor

Fundamental

X. Tang

IoT with MIT App Inventor

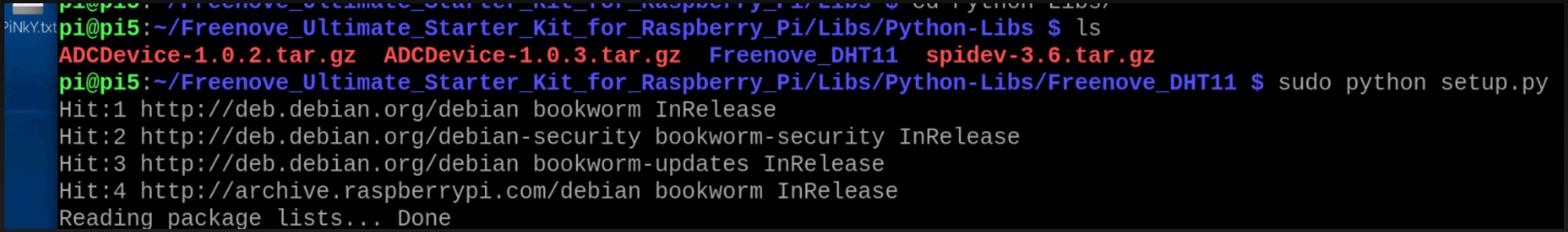
Start 7:05PM

Please update your name with studentID

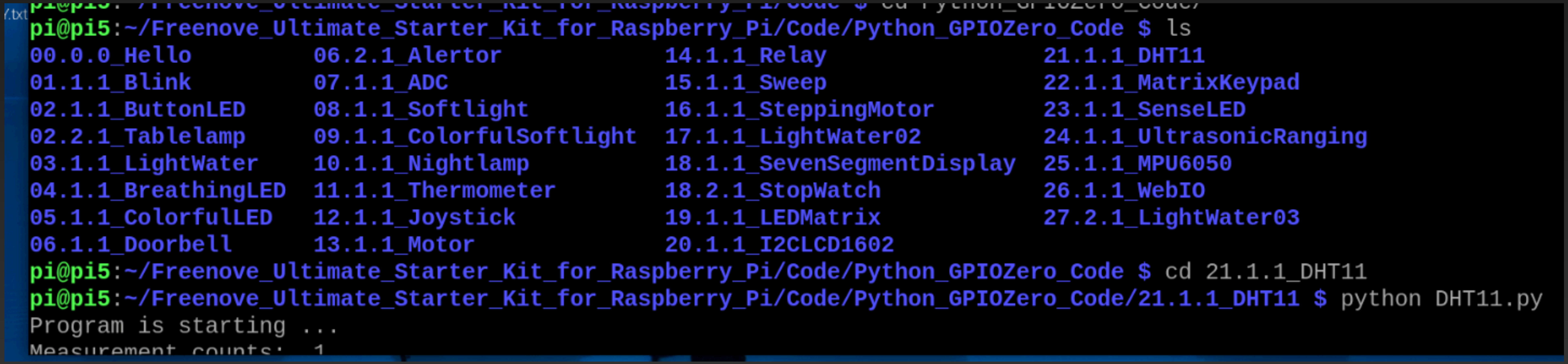
In Zoom meeting

Prepare the Env and Test the Sensor

```
git clone --depth 1 https://github.com/freenove/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi

mv Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi  ~/Freenove_Kit/

cd Freenove_Kit/Code/Python_GPIOZero_Code/21.1.1_DHT11/

python DHT11.py
```

- 

```
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Libs $ cd Python-Libs/
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Libs/Python-Libs $ ls
ADCDevice-1.0.2.tar.gz  ADCDevice-1.0.3.tar.gz  Freenove_DHT11  spidev-3.6.tar.gz
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Libs/Python-Libs/Freenove_DHT11 $ sudo python setup.py
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease
Reading package lists... Done
```

- 

```
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code $ cd Python_GPIOZero_Code/
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code $ ls
00.0.0_Hello         06.2.1_Alertor          14.1.1_Relay                21.1.1_DHT11
01.1.1_Blink         07.1.1_ADC              15.1.1_Sweep                22.1.1_MatrixKeypad
02.1.1_ButtonLED     08.1.1_Softlight        16.1.1_SteppingMotor        23.1.1_SenseLED
02.2.1_Tablelamp     09.1.1_ColorfulSoftlight 17.1.1_LightWater02        24.1.1_UltrasonicRanging
03.1.1_LightWater    10.1.1_Nightlamp        18.1.1_SevenSegmentDisplay  25.1.1_MPU6050
04.1.1_BreathingLED  11.1.1_Thermometer      18.2.1_StopWatch            26.1.1_WebIO
05.1.1_ColorfulLED   12.1.1_Joystick         19.1.1_LEDMatrix            27.2.1_LightWater03
06.1.1_Doorbell      13.1.1_Motor            20.1.1_I2CLCD1602
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code $ cd 21.1.1_DHT11
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code/21.1.1_DHT11 $ python DHT11.py
Program is starting ...
Measurement counts:  1
```

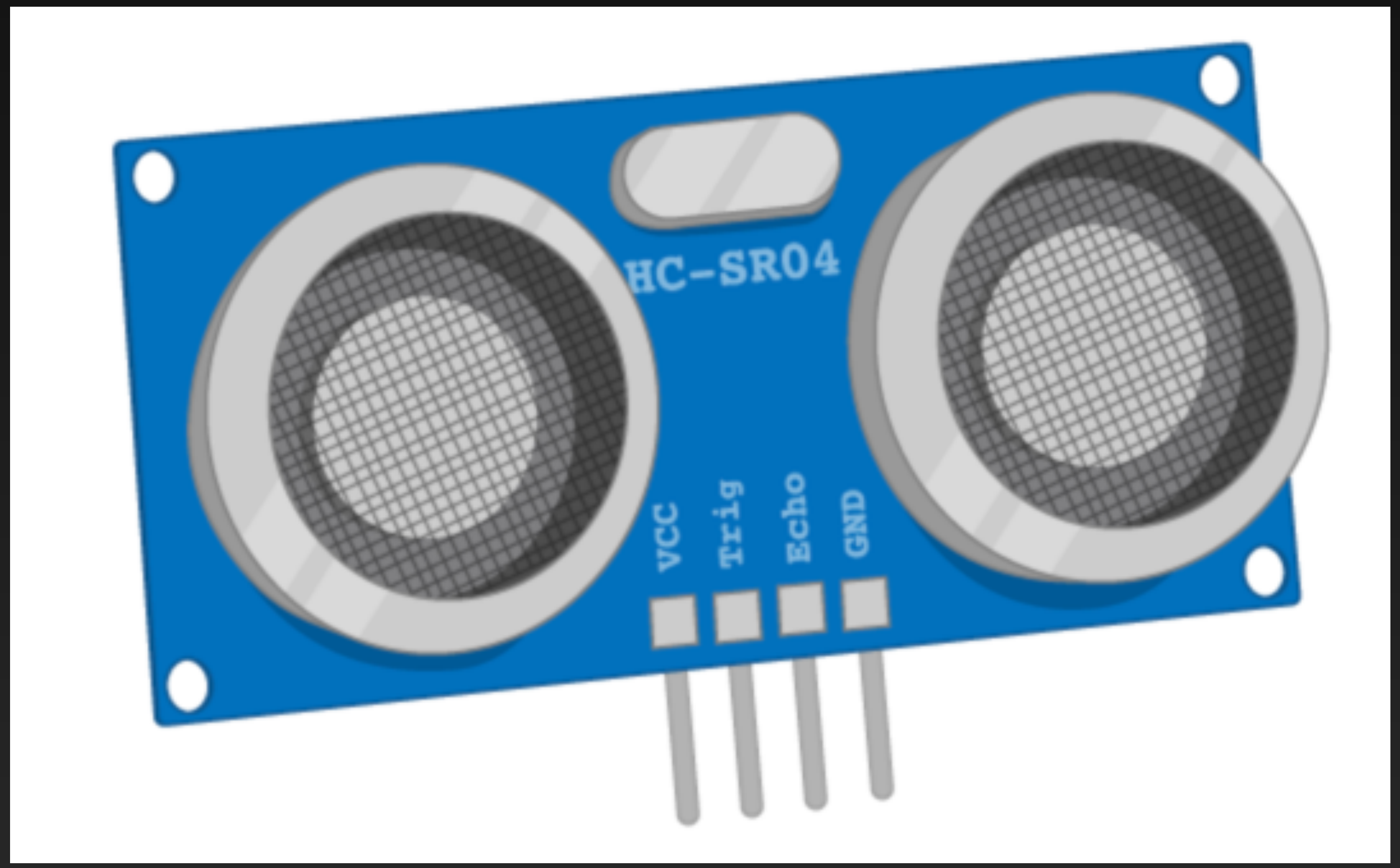
Setup Distance Sensor

An ultrasonic distance sensor is a device that sends out pulses of ultrasonic sound, and measures the time they take to bounce off nearby objects and be reflected back. They can measure distances fairly accurately (up to about a meter).

Setup Distance Sensor

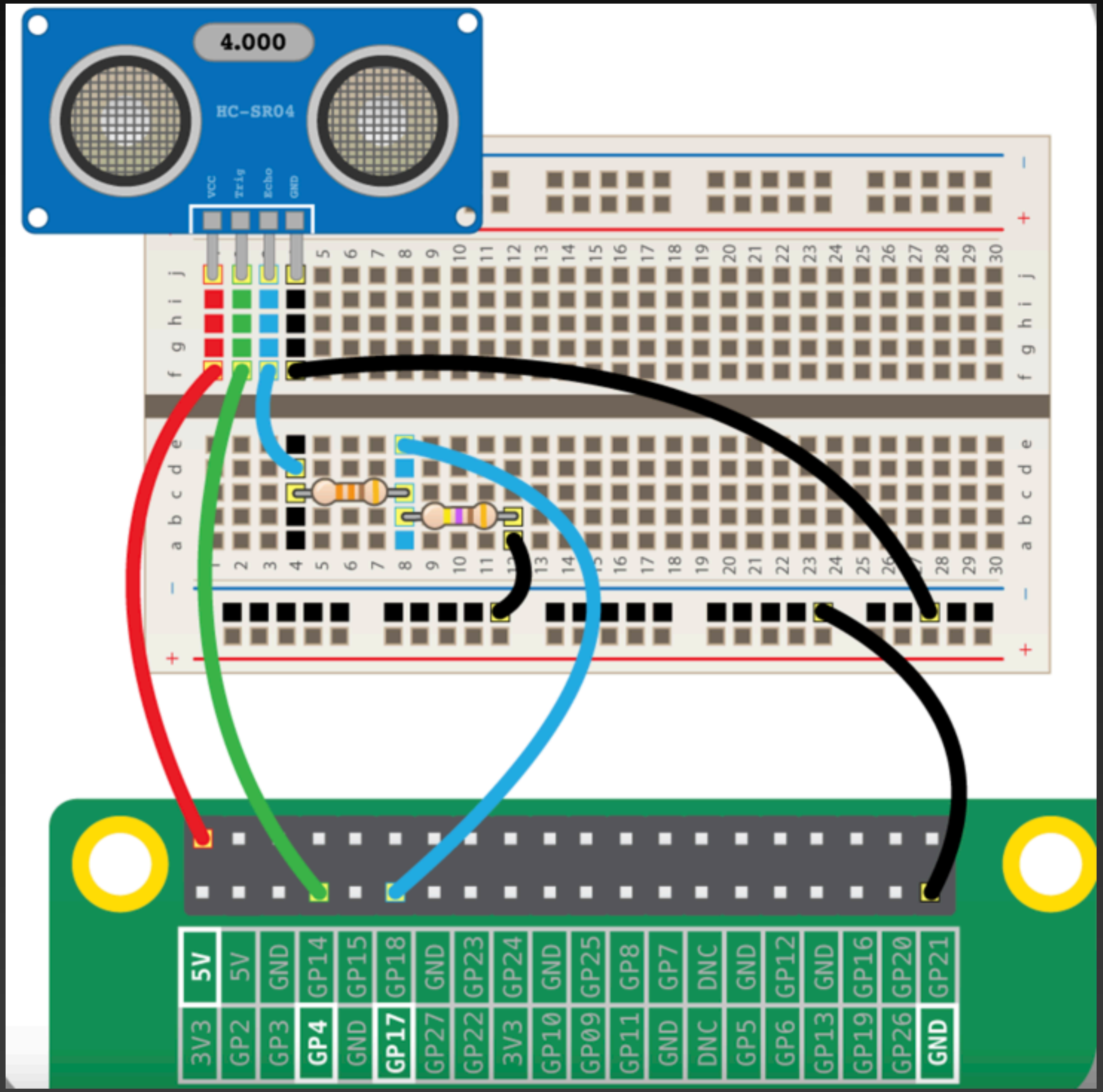
An ultrasonic distance sensor has four pins. They are called Ground (Gnd), Trigger (Trig), Echo (Echo), and Power (Vcc).

To use an ultrasonic distance sensor, you need to connect the Gnd pin to the ground pin on the Raspberry Pi, the Trig and Echo pins to GPIO pins on the Raspberry Pi, and the Vcc pin to the 3V3 pin on the Raspberry Pi.



Setup Distance Sensor

The circuit connects to two GPIO pins (one for echo, one for trigger), the ground pin, and a 5V pin. You’ll need to use a pair of resistors (220Ω) as a potential divider:

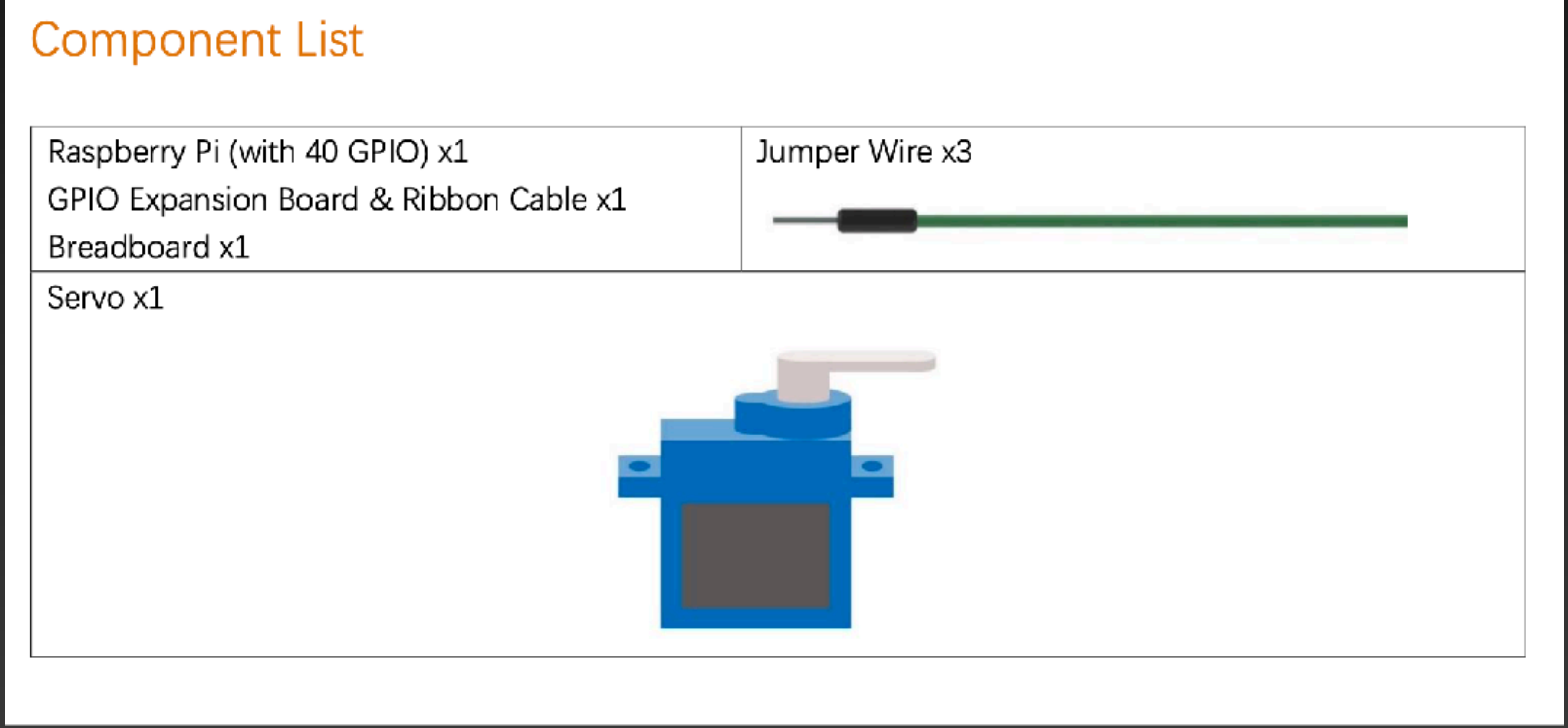


Setup Distance Sensor

```python
from gpiozero import DistanceSensor
ultrasonic = DistanceSensor(echo=17, trigger=4)
while True:
    print(ultrasonic.distance)
```

Servo Motor Control

We will learn how to make motor rotate.

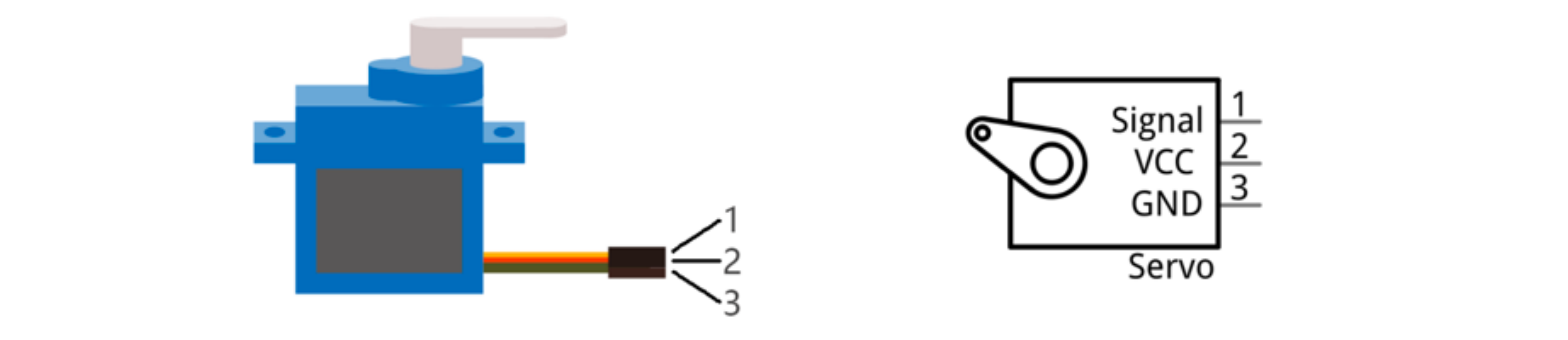


Component List

Raspberry Pi (with 40 GPIO) x1 GPIO Expansion Board & Ribbon Cable x1 Breadboard x1	Jumper Wire x3
Servo x1	

Servo Motor Control

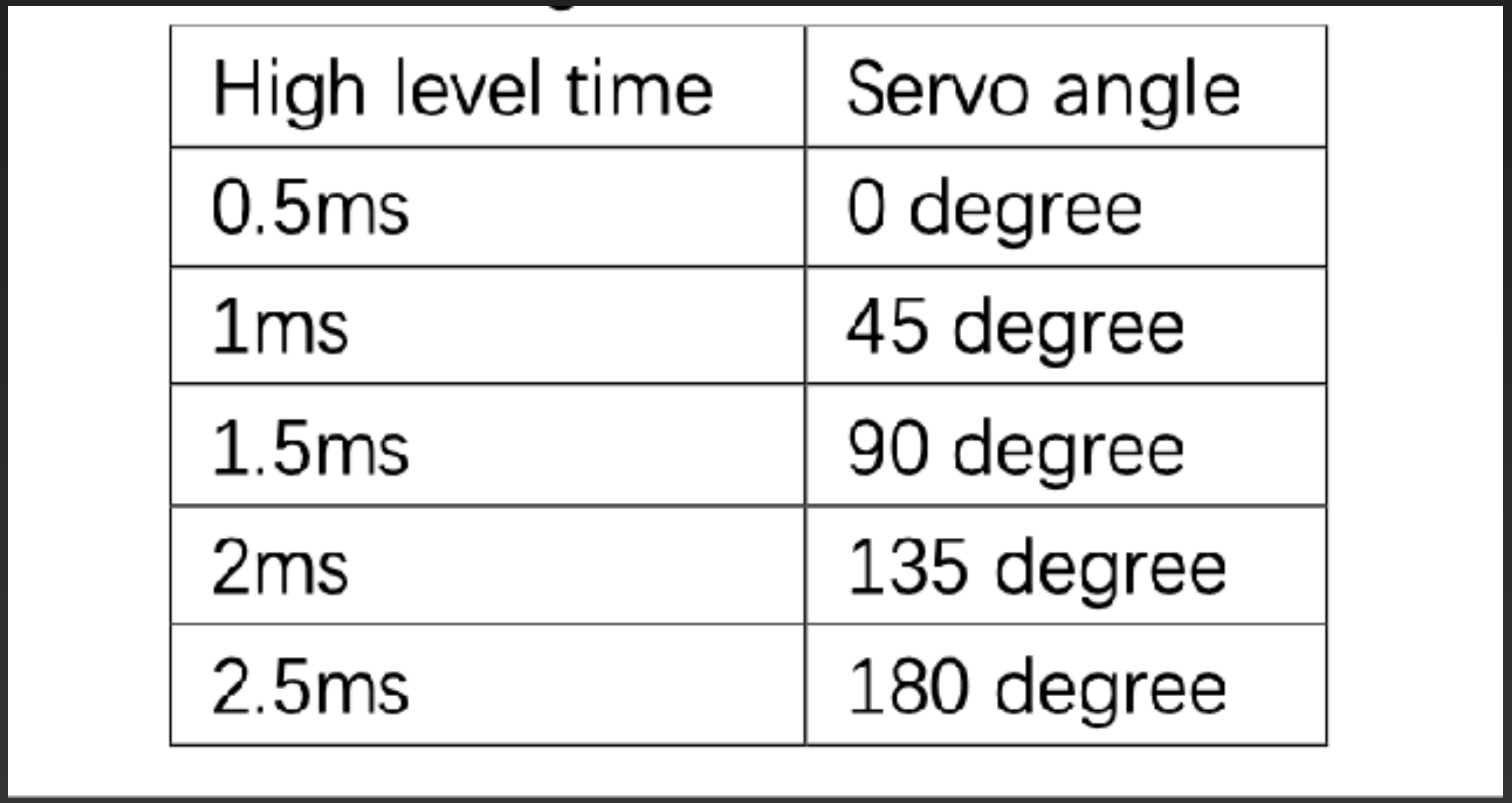
Servo is a compact package which consists of a DC Motor, a set of reduction gears to provide torque, a sensor and control circuit board. Most Servos only have a 180-degree range of motion via their “horn”. Servos can output higher torque than a simple DC Motor alone and they are widely used to control motion in model cars, model airplanes, robots, etc. Servos have three wire leads which usually terminate to a male or female 3-pin plug. Two leads are for electric power: Positive (2-VCC, Red wire), Negative (3-GND, Brown wire), and the signal line (1-Signal, Orange wire) as represented in the Servo provided in your Kit.



Servo Motor Control

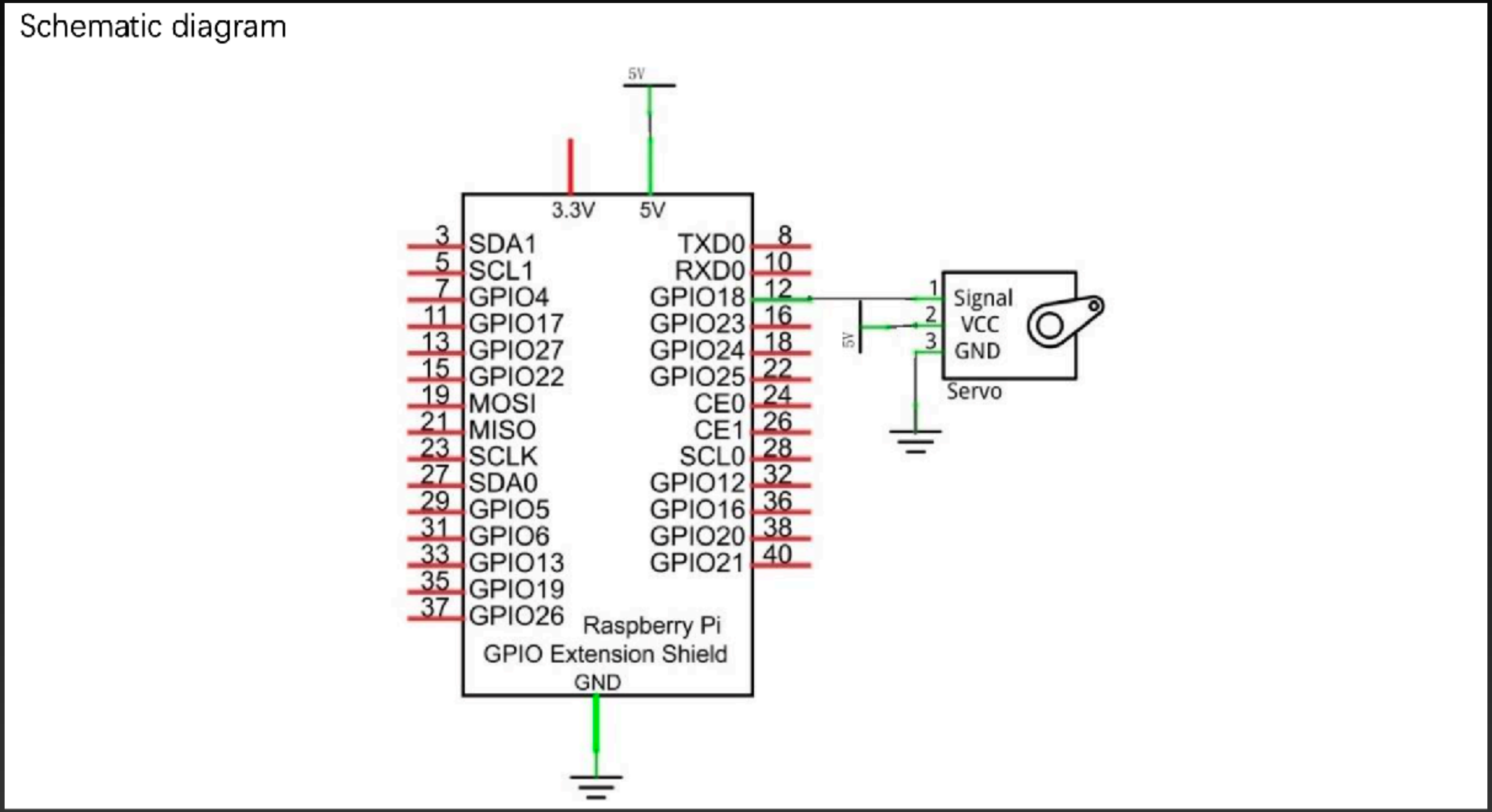
We will use a 50Hz PWM signal with a duty cycle in a certain range to drive the Servo. The lasting time 0.5ms- 2.5ms of PWM single cycle high level corresponds to the Servo angle 0 degrees - 180 degree linearly. Part of the corresponding values are as follows:

Note: the lasting time of high level corresponding to the servo angle is absolute instead of accumulating. For example, the high level time lasting for 0.5ms correspond to the 0 degree of the servo. If the high level time lasts for another 1ms, the servo rotates to 45 degrees.

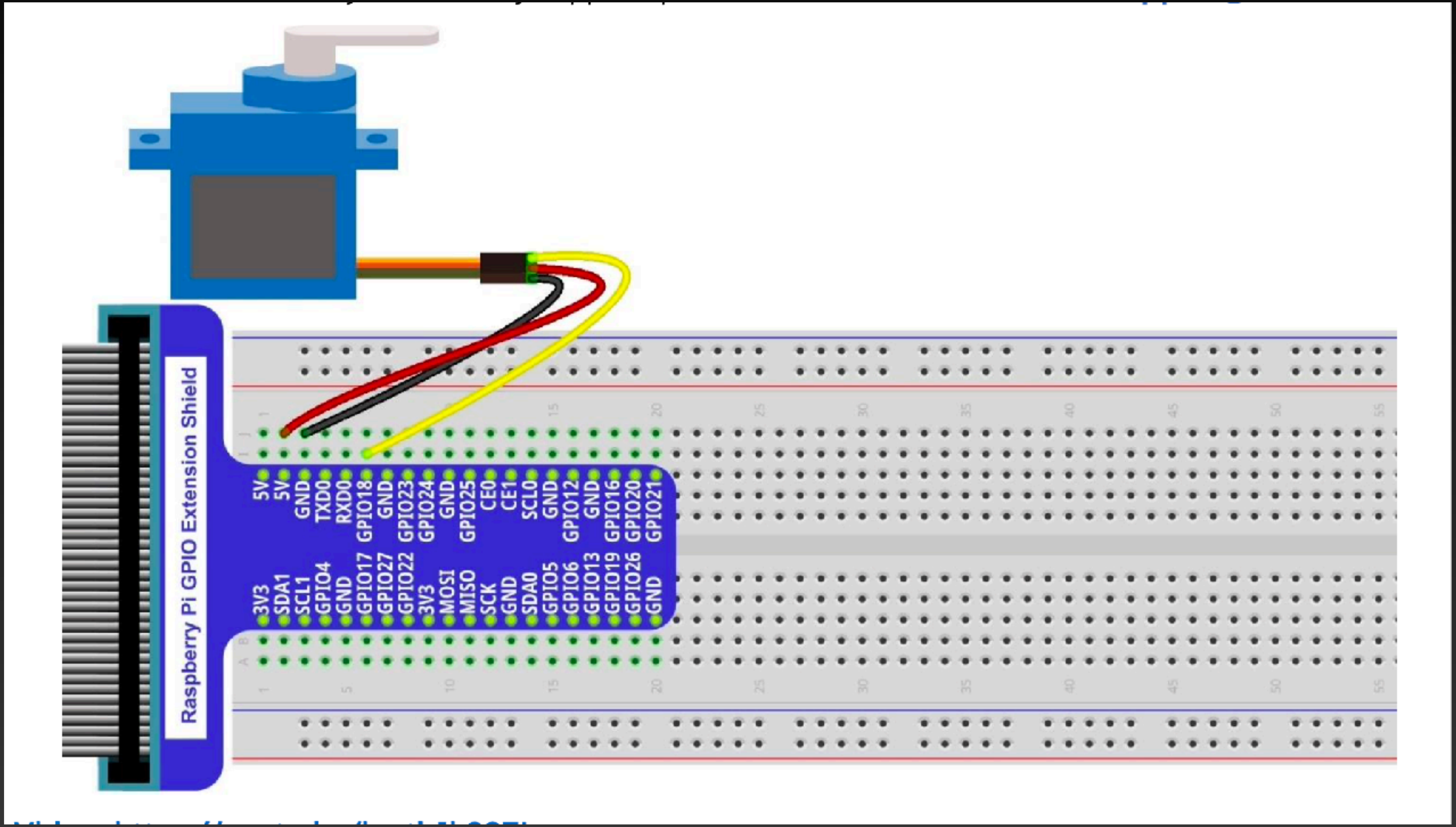


High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

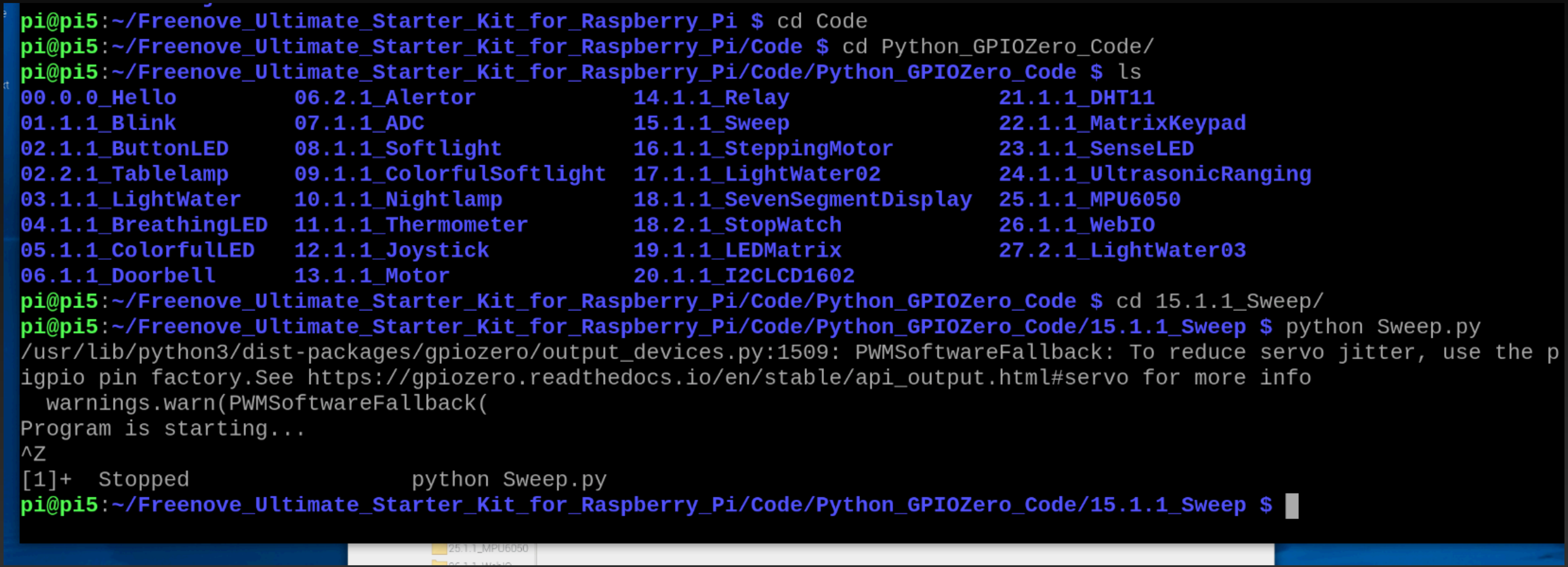
Servo Motor Control



Servo Motor Control



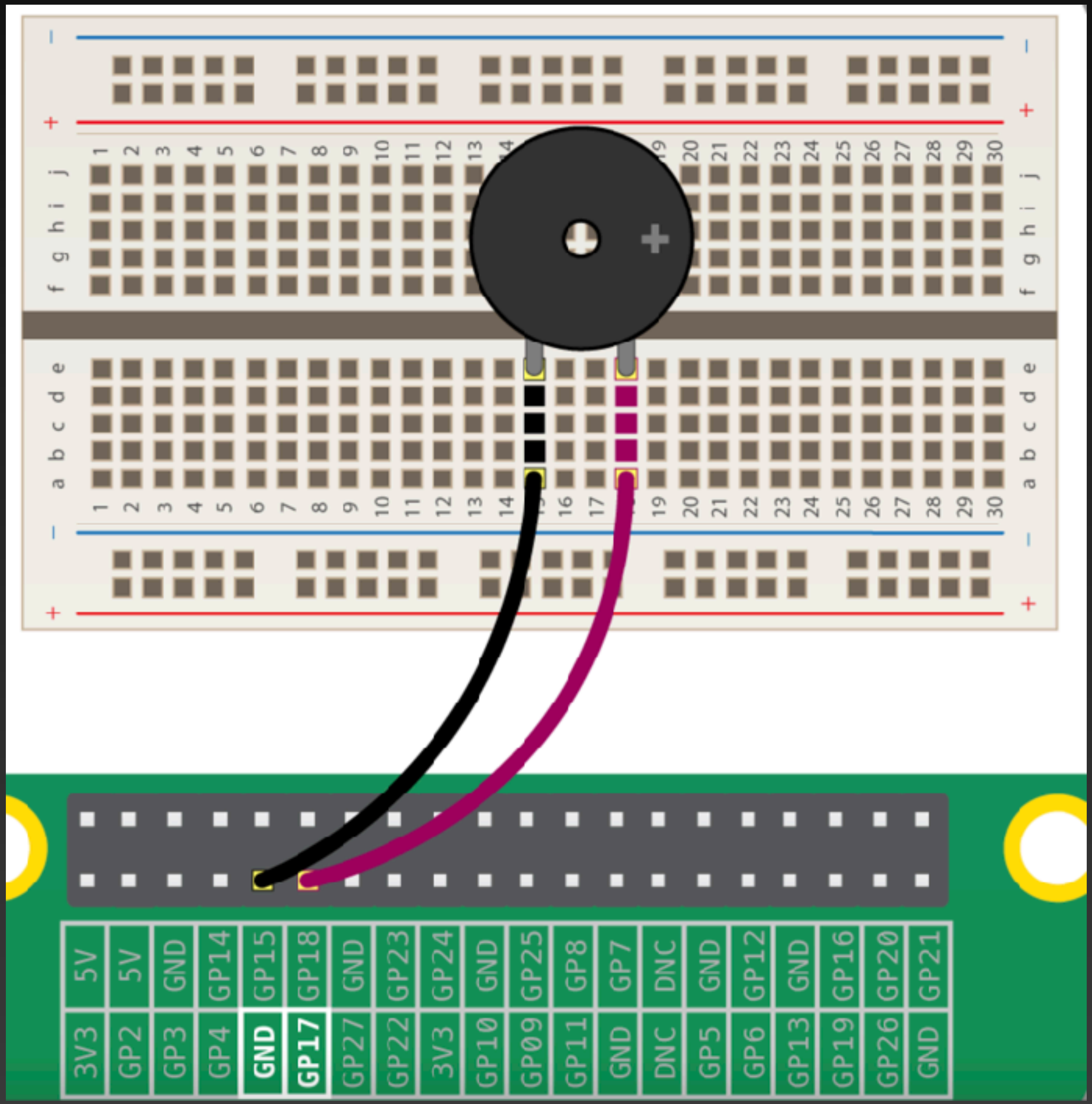
Servo Motor Control



```
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi $ cd Code
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code $ cd Python_GPIOZero_Code/
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code $ ls
00.0.0_Hello         06.2.1_Alertor            14.1.1_Relay                 21.1.1_DHT11
01.1.1_Blink         07.1.1_ADC                15.1.1_Sweep                 22.1.1_MatrixKeypad
02.1.1_ButtonLED     08.1.1_Softlight          16.1.1_SteppingMotor         23.1.1_SenseLED
02.2.1_Tablelamp     09.1.1_ColorfulSoftlight  17.1.1_LightWater02          24.1.1_UltrasonicRanging
03.1.1_LightWater    10.1.1_Nightlamp          18.1.1_SevenSegmentDisplay   25.1.1_MPU6050
04.1.1_BreathingLED  11.1.1_Thermometer        18.2.1_StopWatch             26.1.1_WebIO
05.1.1_ColorfulLED   12.1.1_Joystick           19.1.1_LEDMatrix             27.2.1_LightWater03
06.1.1_Doorbell      13.1.1_Motor              20.1.1_I2CLCD1602
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code $ cd 15.1.1_Sweep/
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code/15.1.1_Sweep $ python Sweep.py
/usr/lib/python3/dist-packages/gpiozero/output_devices.py:1509: PWMSoftwareFallback: To reduce servo jitter, use the p
igpio pin factory.See https://gpiozero.readthedocs.io/en/stable/api_output.html#servo for more info
  warnings.warn(PWMSoftwareFallback(
Program is starting...
^Z
[1]+  Stopped                 python Sweep.py
pi@pi5:~/Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi/Code/Python_GPIOZero_Code/15.1.1_Sweep $ 
```

Setup Buzzer

An active buzzer can be connected just like an LED, but as they are a little more robust, you won’t be needing a resistor to protect them.

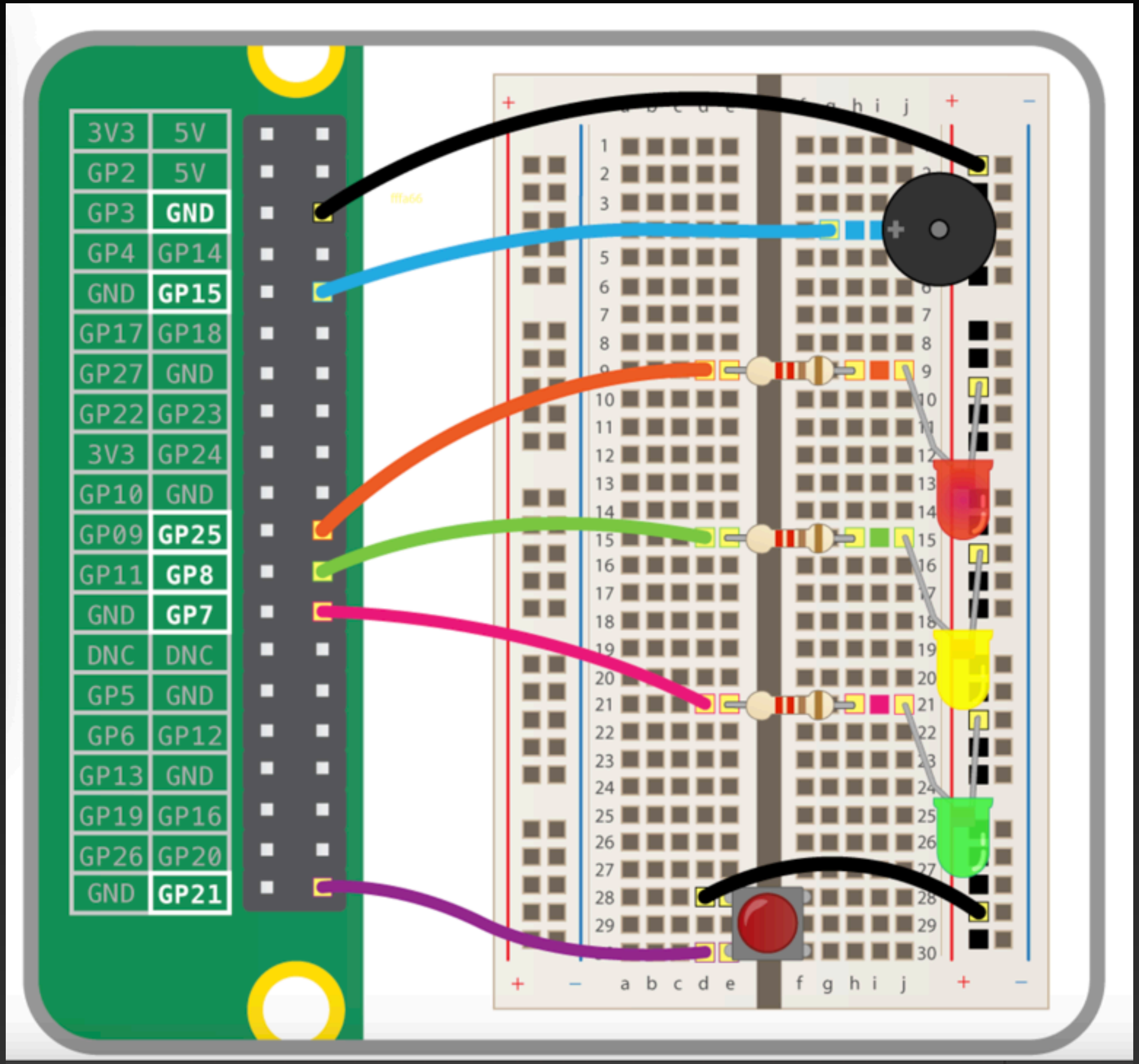


Setup Buzzer

An active buzzer can be connected just like an LED, but as they are a little more robust, you won’t be needing a resistor to protect them.

```
from gpiozero import Buzzer
from time  import sleep

buzzer = Buzzer(17)

while True:
    buzzer.on()
    sleep(1)
    buzzer.off()
    sleep(1)
```

Setup Traffic light program


3V3
5V
GP2
5V
GP3
GND
GP4
GP14
GND
GP15
GP17
GP18
GP27
GND
GP22
GP23
3V3
GP24
GP10
GND
GP09
GP25
GP11
GP8
GND
GP7
DNC
DNC
GP5
GND
GP6
GP12
GP13
GND
GP19
GP16
GP26
GP20
GND
GP21
a b c d e
f g h i j
+
−