

## Controlling Servos with Picon Zero

### Requirements:

- Picon Zero
- Standard or Micro servo
- Raspberry Pi Connected to Internet, keyboard and screen
- Raspberry Pi already setup following Worksheet 01

### Connect the Servo

The servo will come with a cable already connected that can be plugged directly in to any of the 6 outputs on Picon Zero.



The wires on servos can have different colours, but it is important to ensure you connect it the right way round or you can damage it. Use the following connections:

Brown (or Black): Ground

Red: Power / Volts

Orange (or White): Signal

For this test, plug the servo in to Output 0

### Servo Power

Servos can use more power than can be provided from the 5V of the Raspberry Pi. If you use one or two micro servos like the one shown above, then all will probably be OK. (you may still get problems if you stress the servo by holding it in one position while trying to make it move).

You can use more power for the servos by moving the 5V jumper from RPI to USB, then connecting a 5V USB power supply (or USB battery bank) into the micro-USB socket on the Picon Zero.

### Programming the Test

Set the Output configuration of channel 0 to Servo Output:

```
pz.setOutputConfig(0, 2)
```

Now when you send data to channel 0, it will be used to set the position of the servo.

Setting the data to 0, will set the servo 90 degrees clockwise

Setting to data to 90, will set the servo to centre position

Setting the data to 180, will set the servo to 90 degrees anti-clockwise

Here is a complete (but trivial) program to move the servo left and right

```
import piconzero as pz, time
pz.init()
pz.setOutputConfig(0, 2)
while True:
    pz.setOutput(0, 0) # 90 degrees clockwise
    time.sleep(1)
    pz.setOutput(0, 90) # centre position
    time.sleep(1)
    pz.setOutput(0, 180) # 90 degrees anticlockwise
    time.sleep(1)
    pz.setOutput(0, 90) # centre position
    time.sleep(1)
```

### How does a Servo work?

An (analog) servo sets the position of its output shaft depending on the length of the pulse sent on the Signal wire.

Although it varies between servos, a 1.5ms pulse sets the centre position, 0.5ms pulse for 90 degrees clockwise, and 2.5ms pulse for 90 degrees anti-clockwise.

Inside the servo is a motor and gears, and a potentiometer which is turned by the output shaft. When the voltage on the potentiometer matches the voltage generated by the pulse, then the servo "knows" it has reached the correct position