

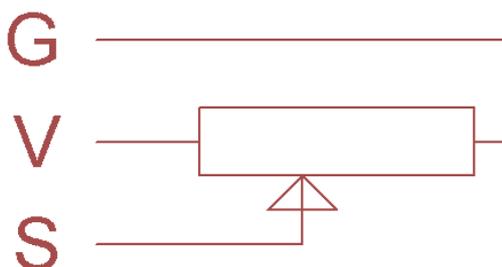
First Steps with Analog Readings using Picon Zero

Requirements:

- Picon Zero
- Potentiometer (rotary or slider) value can be anything from 10K to 100K
- Raspberry Pi Connected to Internet, keyboard and screen
- Raspberry Pi already setup following Worksheet 01

Wire up the Potentiometer

The potentiometer is wired with the outside connections to Power and Ground, and the wiper (centre contact) to the Signal.

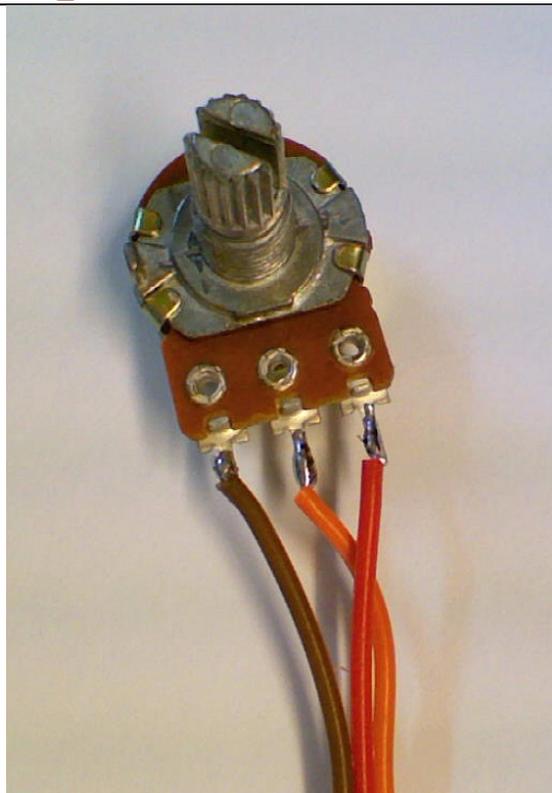


The easiest way to connect sensors, switches and other input devices to Picon Zero is to use a 3-pin “dupont” female header. You can make these up yourself, or easier to get a ready-made cable and attach your input device to the other end. These are the same cables sold as servo cables:



We stock these at 4tronix here:

http://4tronix.co.uk/store/index.php?rt=product/product&product_id=553



Brown: Ground
Red: Volts (Power)
Orange: Signal

Connect this to Input 0

Programming the Test

Set the input configuration to Analog Input:

```
pz.setInputConfig (0, 1)
```

Using the `readInput()` command you will get values between 0 and 1023, where 0 is with the potentiometer turned all the way towards Ground, and 1023 is turned all the way towards Power. In the real world, you may not be able to get to 1023 due to various voltage drops.

What is a Potentiometer?

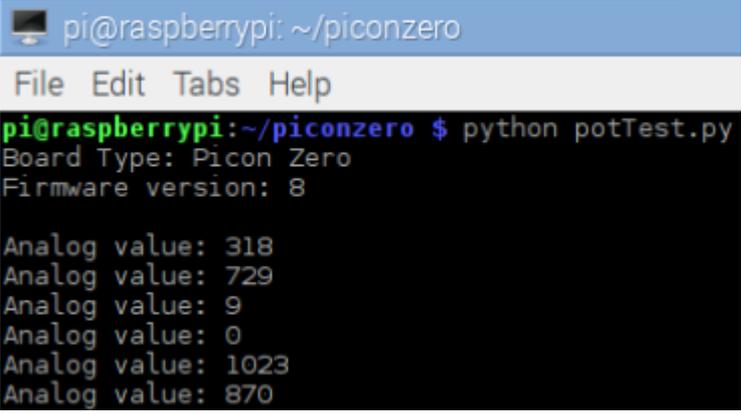
A potentiometer (called a pot) is basically a resistor, with a third terminal that wiper along the length of the resistive material.

As the pot is turned, the wiper moves from one end of the resistor to the other. At one end, the wiper is connected to one end terminal. At the other end, it is connected to the other end terminal.

As the voltage varies from 0V at the Ground end to 5V at the Power end, this allows us to set a voltage which we can read using the analog input function in Picon Zero

Here is a complete (but trivial) program to show the output

```
import piconzero as pz, time
pz.init()
pz.setInputConfig(0, 1)
while True:
    value = pz.readInput(0) # reads the analog value
    print "Analog value:", value
    time.sleep(1)
```



The image shows a terminal window on a Raspberry Pi. The prompt is 'pi@raspberrypi: ~/piconzero'. The terminal has a menu bar with 'File Edit Tabs Help'. The command 'python potTest.py' has been executed, resulting in the following output:

```
pi@raspberrypi:~/piconzero $ python potTest.py
Board Type: Picon Zero
Firmware version: 8

Analog value: 318
Analog value: 729
Analog value: 9
Analog value: 0
Analog value: 1023
Analog value: 870
```