

Getting Switch/Button Input from Picon Zero

Requirements:

- Picon Zero
- Push button Switch
- Raspberry Pi Connected to Internet, keyboard and screen
- Raspberry Pi already setup following Worksheet 01

Check Your Version of Picon Zero:

The Picon Zero continues to evolve and new features are occasionally added to the firmware. One of the features not in the original version 06 release is the ability to add pullup resistors to digital inputs in the software. This was added in Firmware version 08 for both Digital Inputs and DS18B20 temperature sensors and makes the wiring much simpler.

To check which version of Picon Zero you have, change to the piconzero folder on your Raspberry Pi and enter the following command:

Python version.py

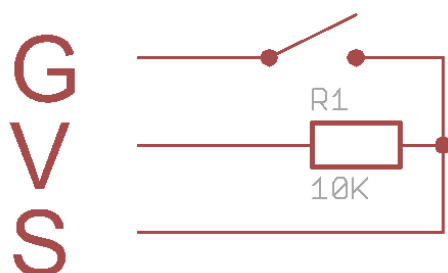
The board type (Picon Zero) and the Firmware version will be printed on the screen

```
pi@raspberrypi: ~/piconzero
File Edit Tabs Help
pi@raspberrypi:~ $ cd piconzero
pi@raspberrypi:~/piconzero $ python version.py
Board Type: Picon Zero
Firmware version: 8
pi@raspberrypi:~/piconzero $
```

If your version is 7 or earlier, you will need to add the pullup resistor in the following activity

Wire up the Switch

To make use of the pullup resistor (internal or external) we wire the switch between the signal input and ground. This gives an inverted output – ie. it LOW (zero) when pressed, and HIGH (1) when not pressed. Here is the schematic for the circuit. With version 08 firmware and above, resistor R1 is not required.



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



Without Pullup resistor

With 10K Pullup resistor

What are Pullup Resistors For?

Inputs on microcontrollers are usually “floating”. This means that they don’t have a particular value of either On or Off – or High or Low.

If you touch a floating input, it is likely to change from Off to On, or vice versa. This is often used as a cheap sort of touch input method.

For real programming, we need a defined state for the pin, so we “pull it up” to 5V using a resistor, so that we know it is high. The resistor can generally be any value from 1K to 100K, but 10K is a normal value.

Programming the Test

If you have firmware before vsn 08, then set the input configuration to Digital Input:

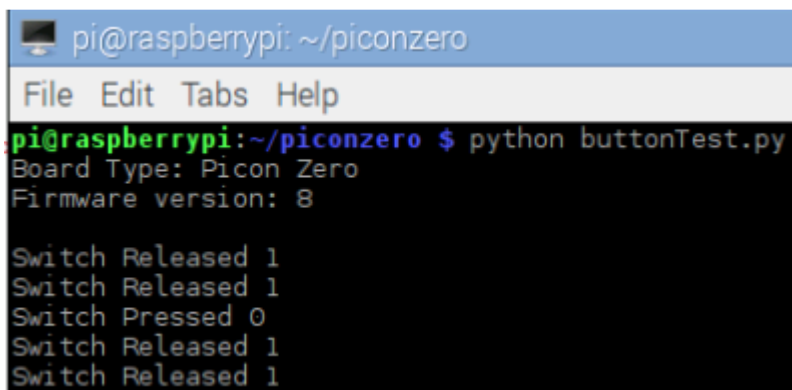
```
pz.setInputConfig (0, 0)
```

But if you have vsn 08 or later, then you can select a pullup resistor like this:

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

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

```
import piconzero as pz, time
pz.init()
pz.setInputConfig(0, 0, True) # request pullup on input
while True:
    switch = pz.readInput(0) # 0 = pressed, 1 = not pressed
    if (switch == 0):
        print "Switch Pressed", switch
    else:
        print "Switch Released", switch
    time.sleep(1)
```



```
pi@raspberrypi: ~/piconzero
File Edit Tabs Help
pi@raspberrypi:~/piconzero $ python buttonTest.py
Board Type: Picon Zero
Firmware version: 8

Switch Released 1
Switch Released 1
Switch Pressed 0
Switch Released 1
Switch Released 1
```