Controlling Brightness of LEDs with Picon Zero

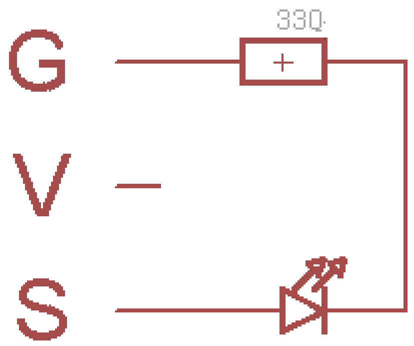
Requirements:

- Picon Zero
- Any LED
- Resistor (100Ω to 1kΩ) - 330Ω recommended
- Raspberry Pi Connected to Internet, keyboard and screen
- Raspberry Pi already setup following Worksheet 01

Wiring the LED

You cannot connect LEDs directly to the output of the Picon Zero. That would be bad. You need to connect them via a resistor, to limit the current. The smaller the value of resistor you use, the higher the current and the brighter the LED will be.

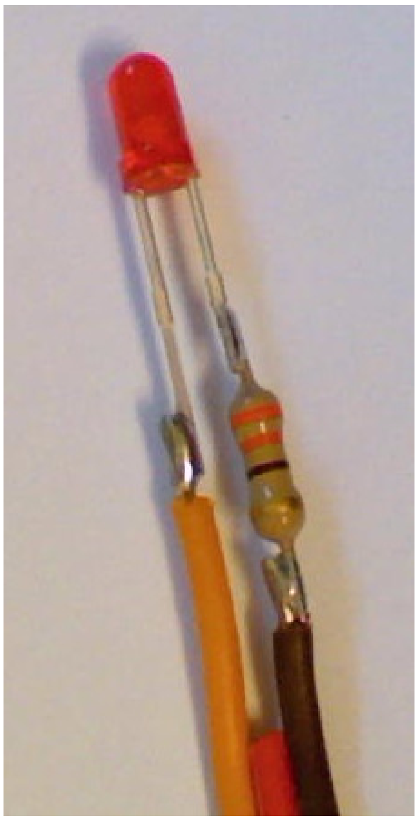
For this circuit, we do not need the Voltage/Power connection at all, only the outer two pins on the 3-way (GVS) output connector are used.



The easiest way to connect input and output 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 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

The long lead of the LED is the Anode and should be connected directly to the Signal wire (Orange). The other end of the LED (the Cathode) should connect to one end of the resistor. Finally, the other end of the resistor should connect to the Ground wire (Brown)



What is PWM?

Pulse Width Modulation (PWM) is a method of using a digital (On or Off) signal to create the semblance of an analog (continuously varying) signal.

PWM works by setting a time period (eg 20ms) during which the output is ON for a varying amount of time.

If it is ON for 10% of the time, ie 2ms out of every 20ms, then the LED will appear quite dim. If it is on for 90% of the time (18ms out of every 20ms) then the LED will be bright.

The same mechanism is used to control the speed of motors on controllers like Picon Zero

In the photo above, I have left the leads long and uncovered, so the connections are clear. In practice, it is best to use insulating tape or heat-shrink tubing so prevent short circuits.

Programming the Test

Set the Output configuration of channel 0 to PWM Output:

pz.setOutputConfig (0, 1)

Now when you send data to channel 0, it will be used to set the PWM percentage for that output. This can vary from 0% (fully Off), to 100% (fully On)

Here is a complete (but trivial) program to cycle through various brightnesses

```
import piconzero as pz, time
pz.init()
pz.setOutputConfig(0, 1)
while True:
    pz.setOutput(0, 5) # 5% - very dim
    time.sleep(1)
    pz.setOutput(0, 30) # 30% - medium
    time.sleep(1)
    pz.setOutput(0, 70) # 70% - bright
    time.sleep(1)
    pz.setOutput(0, 100) # 100% - maximum
    time.sleep(1)
```