TURNING AN LED ON AND OFF

Project

How to control LED's using python

Description

In this tutorial you will be getting taught how to connect and control LED's (Light Emitting Diode) using a raspberry pi.

Equipment You Will Need

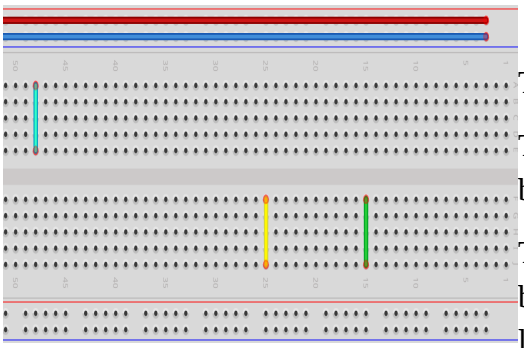
- Raspberry pi and an SD card
- Mouse and Keyboard
- Monitor and HDMI cable
- 1 x Red LED
- 2 x male to female jumper wires
- 1 x 400 point breadboard
- Power supply
- 1 x 330Ω resistor

The Parts To Be Used

Within this section I will go through how the parts that we are going to use work and how they should be connected.

The Breadboard

The breadboard is used to connect electronics together instead of soldering. The breadboard is often used in prototyping a product.



The red line connects all the dots within this row.

The blue line connects all the dots within this row.

The turquoise line connects all the dots within this column, but is not connected to either the yellow or green lines

The yellow line connects all the dots within this column, but is not connected to either the green or the turquoise lines.

The green Line connects all the dots within this column, but is not connected to either the yellow or turquoise lines.

The LED's

LED stands for Light Emitting Diode. They glow when electricity is passed through them. When looking at LED's you will notice that there is one leg longer than the other. The longer leg known as the anode is connected to the positive supply of the circuit. The shorter leg known as the cathode is connected the the negative side of the circuit also known as ground.

If you connect the LED's the wrong way round this is not the end of the world because you will not damage them, they just will not light up.

The Resistors

Resistors limit the amount of current that can flow through an electrical circuit. Resistors are measured in ohms Ω, The larger the resistance the more the current is limited.

Resistors can be identified by the coloured bands along the body. For example the resistor that we are using is 330 Ω, this is identified by orange, orange, brown then gold.

We have to use resistors to connect the LED's to the raspberry pi, or the LED's will pull too much current from the raspberry pi and this will cause the raspberry pi to stop working.

The resistors can be connected any way round because the current of the electricity will flow both ways.

The Jumper Wires

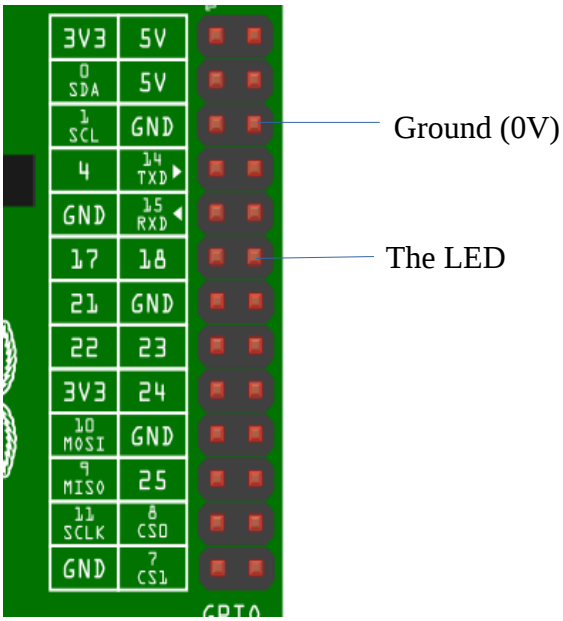
Jumper wires are used on breadboards to jump from one connection to another. The jumper wires we are using are called male to female jumper wires. The end with the pin on it will go into a hole on the breadboard, and the end with the hole will connect to a GPIO pin on our raspberry pi.

Making The Circuit

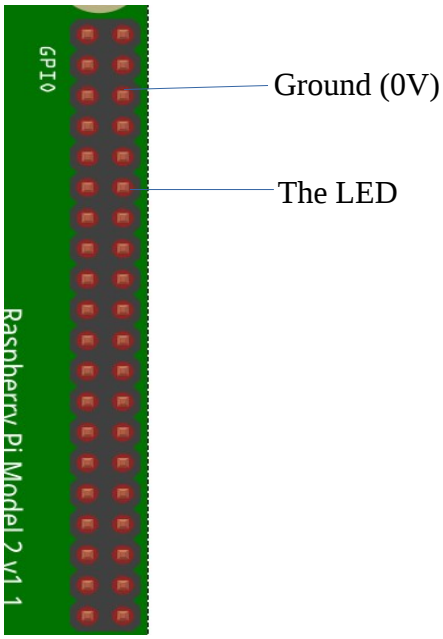
To make the circuit it is always best to have the raspberry pi turned off at this stage.

The first thing we are going to do is take a look at the GPIO (General Purpose Input Output) pins.

Models A & B



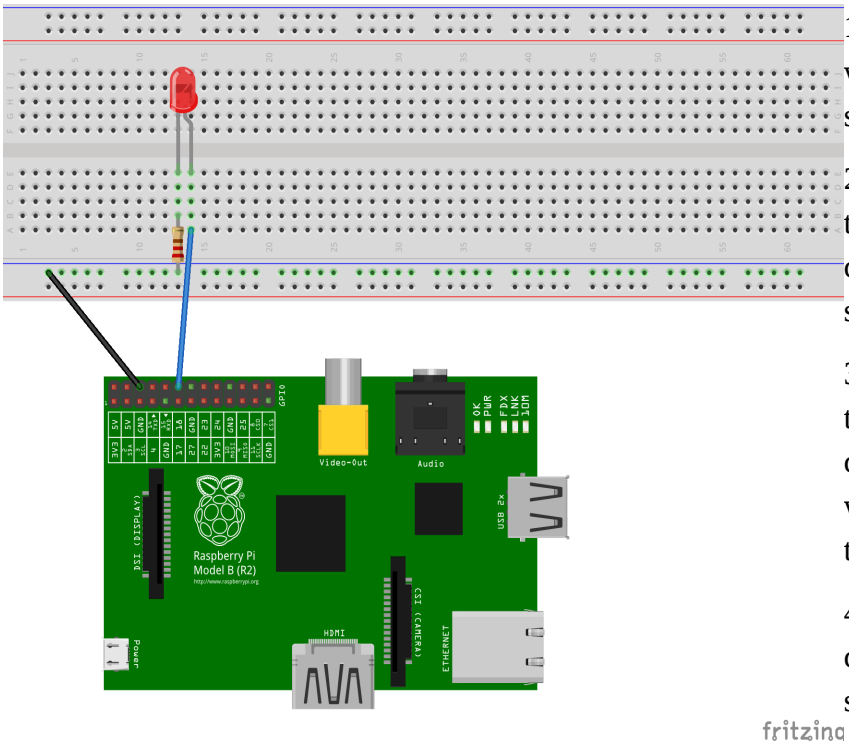
Model B+ & Raspberry Pi 2



It is through the GPIO pins that the raspberry pi can control and even monitor the outside world by being connected into an electronic circuit. The raspberry pi can control LED's Motors, button presses and even buzzers.

The diagrams above show the pin layouts of the model A and B on the left and Models A+, B+ and the raspberry pi 2 on the right. I have marked out the pins we will be using and what will be connected to them.

We are now ready to set up our circuit.



1. Put the LED into the breadboard with the longer leg on the right hand side.

2. Connect the resistor in between the short leg of the LED and the common ground rail, which is the second row from the bottom.

3. Take a jumper wire and connect the end with the pin into the common ground rail and the side with the hole over the ground pin on the raspberry pi.

4. Take the second jumper wire and connect the end with the pin into the same column as the longer leg of

the LED and the side with the hole onto pin 18 on the raspberry pi.

Now we have our circuit ready we can power up the pi and do some coding.

Code

The first thing we need to do is open up a terminal window.

If you did my last tutorial you will remember we created a Python directory, so all we need to do now is change into that directory by typing cd Python

Next we need to create a new text file, to do that type nano OnOff.py

We are now ready to type some code. The code is in red and the text in black explains what the code does.

import Rpi.GPIO as GPIO	Importing the raspberry pi GPIO library as GPIO
import time	Importing the time library for timing functions
GPIO.setmode (GPIO.BCM)	Setting the naming convention for the GPIO pins
GPIO.setwarnings (False)	Telling python not to print GPIO warning messages to the screen
GPIO.setup (18, GPIO.OUT)	Set up pin 18 as an output
print “Light On” GPIO.output (18, GPIO.HIGH)	Turning the LED on
time.sleep (5)	Pause for 5 seconds
print “Light Off” GPIO.output (18, GPIO.LOW)	Turning the LED off
time.sleep (5)	Pause for 5 seconds
GPIO.cleanup()	Setting the GPIO pins back to default

Once you have checked the code for any mistakes, save the code and exit the text editor by pressing ctrl+X then y then enter.

Running The Code

To run the program type sudo python OnOff.py

That is all for this tutorial please join me again for the next one

Happy coding from the RaspiKidd

You can also find all my tutorials on my website: www.raspikidd.me.uk, where you can find my links to facebook and twitter.

If you have any questions please leave a comment tweet post on facebook or just email me. The email address is at the bottom of the page.