INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi
1 x breadboard
1 x LED
1 x 330 ohm resistor
Some jumper wires

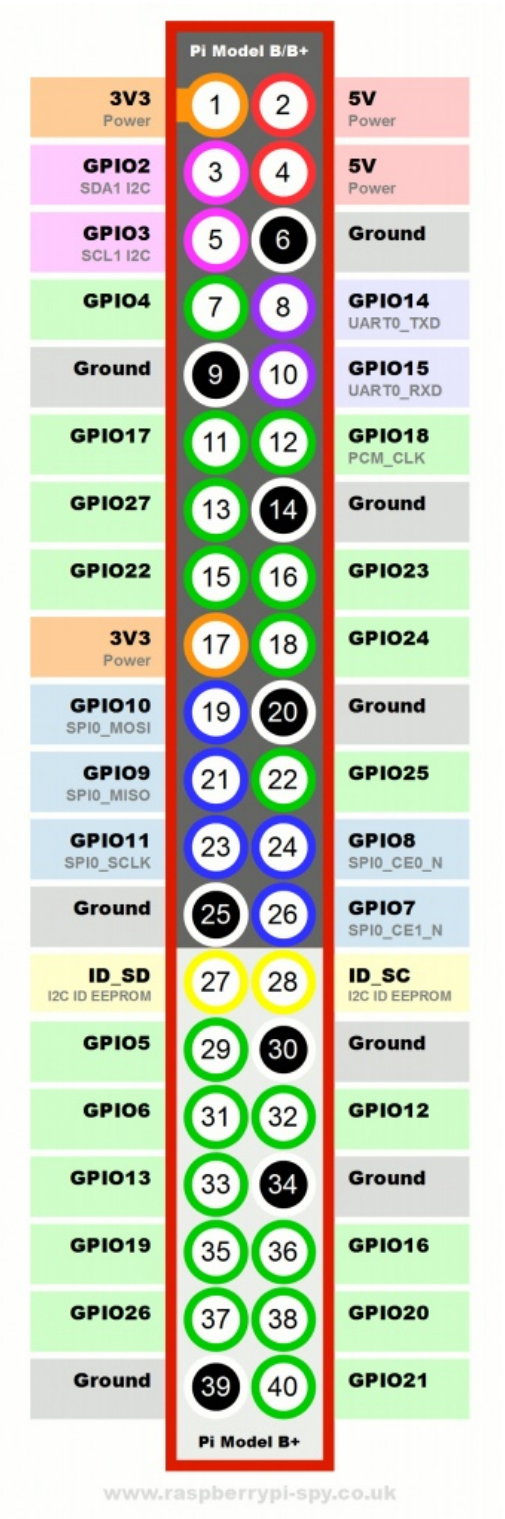
What is GPIO?

GPIO is the General Purpose Input / Output system that enables the Pi to interact with the physical world.

They are the bank of 40 pins that runs along the top of the Pi.

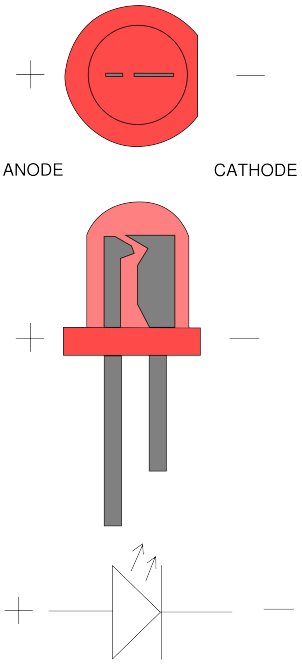
With these you can control robots to take over the world!!

To the right you can see the how the pins are laid out. The numbers inside the circles are the physical pin numbers which will be used shortly with ScratchGPIO. The tags to the left and right are the Broadcom mappings and are as the Raspberry Pi sees the pins.



A quick introduction to LED's

A Light Emmititng Diode is a type of diode that, when connected correctly, produces light.
Being a type of diode, current can only flow one way through the LED. This means that power must flow via the annode to the cathode. One thing to remember is that a LED must have a resistor in series to prevent too much current from destroying the LED.
The cathode will be the shorter of the two legs. It can also be found by feeling the LED body for a slighty flattend edge.

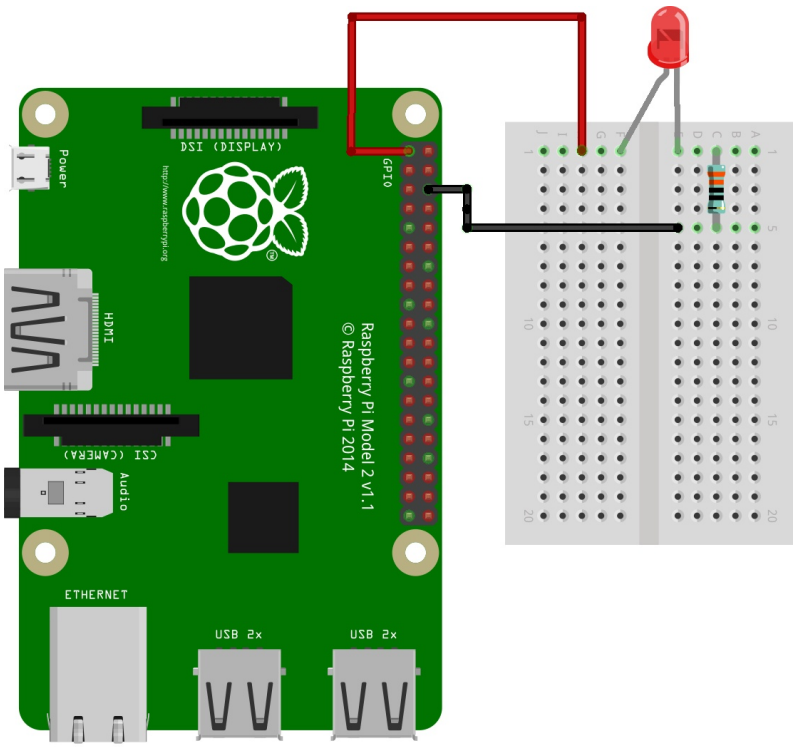


INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi
1 x breadboard
1 x LED
1 x 330 ohm resistor
Some jumper wires

Connecting it all up!



Task 1

Lighting up the LED!

In this task we are simply going to wire up the LED to get it to glow.

Firstly, Power down the Pi
Wire up the circuit as to the left:

Pin 1 (3.3v) to the Annode
Pin 6 (Ground) to the cathode

When power is applied the LED should glow.
Make sure the LED is the right way round!

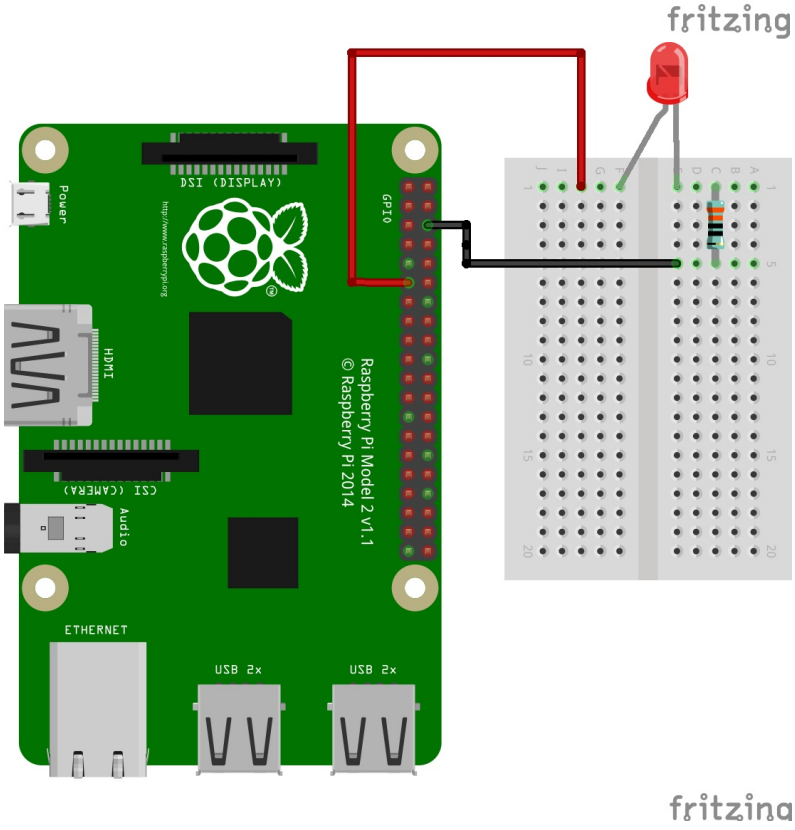
Task 2

Blinking the LED

Now that we have confirmed the LED works it time to make it blink.

Power down the Pi and rewire the circuit to connect the annode to pin 11 and the cathode to pin 6 as per the diagram to the left.
Check the wiring and reconnect the power.

We shall shortly make it blink but first we need to learn about Scratch GPIO...



INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi	1 x 330 ohm resistor
1 x breadboard	Some jumper wires
1 x LED	

Connecting it all up!
(Task 2 cont.)

Scratch GPIO

To make the LED flash we are going to use a variant of Scratch which has been modified to be able to use the GPIO pins called scratchGPIO7.

Scratch GPIO automatically configures the following pins:

For output: 11,12,13,14,15,16,18,21
For input: 7,10,19,22,24,26
(Scratch GPIO uses the physical pin numbers...the one in the circles)

Sending an output:

To send values to the output pins we use the broadcast block to broadcast the following commands:

Command	Example	Result
pin*X*on	pin11on	Turns pin *X* on
pin*X*off	pin11off	Turns pin *X* off
allon		Turn all pins on
alloff		Turn all pins off
pinpatternXXXXXXX	See below	1 for on / 0 for off

Need pin 21,16,12,11 on and all others off:

Pins: 21|18|16|15|14|13|12|11
On/Off: 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1

pinpattern10100011

INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi	1 x 330 ohm resistor
1 x breadboard	Some jumper wires
1 x LED	

Connecting it all up!
(Task 2 cont.)

Scratch GPIO

Sensing

As seen before Scratch GPIO automatically sets certain pins to be inputs. Clicking on the slider block in the Sensing section you will enable you to choose which input pin to read the input from.

As an example, connecting one side of a switch to ground and the other to the selected pin will allow to sense when the switch is closed.

In the following example the input pin will have a value of 0 until the switch is closed. When this happens then the pin value will change to 1 and the message pin11high will be broadcasted.

INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi	1 x 330 ohm resistor
1 x breadboard	Some jumper wires
1 x LED	

Connecting it all up!
(Task 2 cont.)

Time to flash!

Build and run the following code block into Scratch GPIO:

This turns on pin 11, waits for a second and then turns it off. Your LED should be flashing.

What do you think will happen if one or more *wait* blocks are removed? Try it and see.

Can you:

- Make the LED flash faster
- Flash it randomly
- Add another LED
- Make traffic lights
- Have the LED react to an event in scratch, eg. when Scratch Cat touches a red block

In the next section we shall achieve the same using the Python programming language.

INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi	1 x 330 ohm resistor
1 x breadboard	Some jumper wires
1 x LED	

Connecting it all up!
(Task 2 cont.)

Doing the same the Python way!

Python is a relativly simple, but powerful, language to learn. It is used in wide variety of situations from simple scripts to controlling complicated systems.

In this case, we will be using the Python 3 Idle IDE to create our code. As we will be using the GPIO pins we will have to start the IDE with special permissions which allows Python to use the GPIO pins. To do this we start a terminal, type the following and pressing enter:

```
pi@raspberrypi ~ $ sudo idle3 &
```

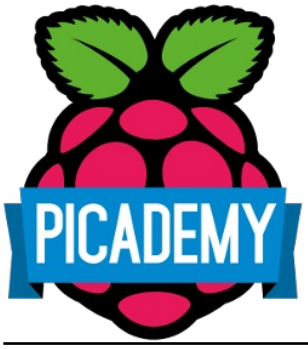
Once loaded start a new file by clicking on New in the File menu and type in the following code:

```
                                    Explanation (No need to type these)

import time                         Import the time library
import RPi.GPIO as GPIO             Import the GPIO library and call it GPIO
GPIO.setmode(GPIO.BOARD)            Set the GPIO library to use the physical pin numbers
GPIO.setup(11, GPIO.OUT)            Set pin 11 to be an output pin

while True:                         Do the following until stopped
    GPIO.output(11, True)           Set pin 11 to output a signal and turn on the LED
    time.sleep(1)                   Wait for a second
    GPIO.output(11, False)          Set pin 11 to not send a signal and turn off the LED
    time.sleep(1)                   Wait for a second and then loop back to the begining
```

Note that in the *while True* loop anything after colon needs to be indented in by one tab. This allows Python to know that this code needs to be ran in the *while True* loop.

Once you have entered the code press F5. You will then be asked to save the code and then the code will be executed causing the LED to flash.

If it's not flashing then check for any error messages displayed and that the circuit is still connected correctly.



INTRODUCTION TO GPIO ON THE RASPBERRY PI

Needed:

1 x ready Raspberry Pi	1 x 330 ohm resistor
1 x breadboard	Some jumper wires
1 x LED	

Wrapping Up....

As can be seen by using a simple bit of code we have made the LED turn on and off.

We can now apply this new knowledge to interact with other types of components.

As an example, a signal could be sent via the GPIO which activates a relay....that causes a motor to start...which turns a wheel...

We now have a way of making things happen in the physical world from inside the Raspberry Pi.

Notes: