User Input

Project

Interaction between user input and LED's

Description

Within this tutorial you will create a program that will let you control either a red, yellow, or green LED depending on user input.

Equipment You Will Need

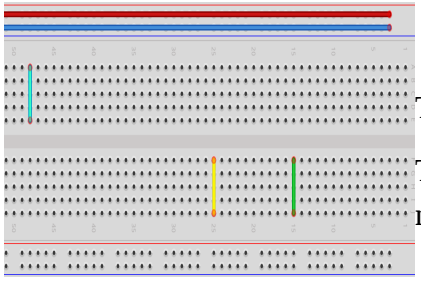
- Raspberry pi and an SD card
- Mouse and Keyboard
- Monitor and HDMI cable
- 1 x Red LED
- 1 x Yellow LED
- 1 X Green LED
- 5 x male to female jumper wires
- 1 x 400 point breadboard
- Power supply
- 3 x 330Ω resistor

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

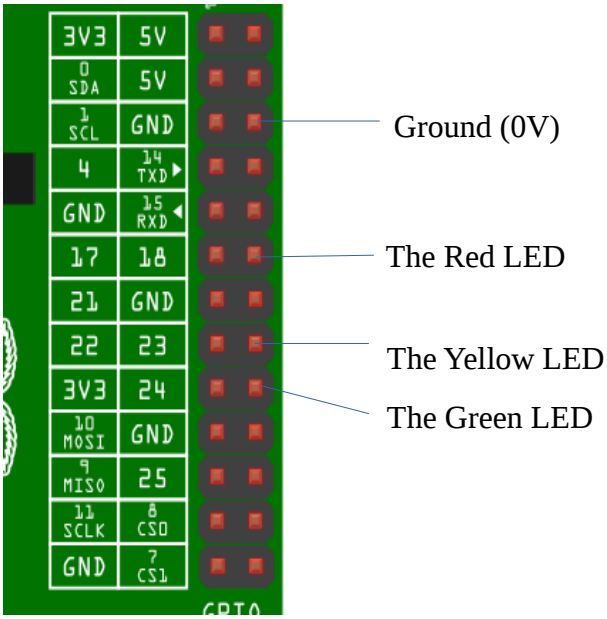
Jumper wires are used on breadboards to jump from one connection to another. The jumper wires we are using are called male to female jumper wires. The end with the pin on it will go into a hole on the breadboard, and the end with the hole will connect to a GPIO pin on our raspberry pi.

Making The Circuit

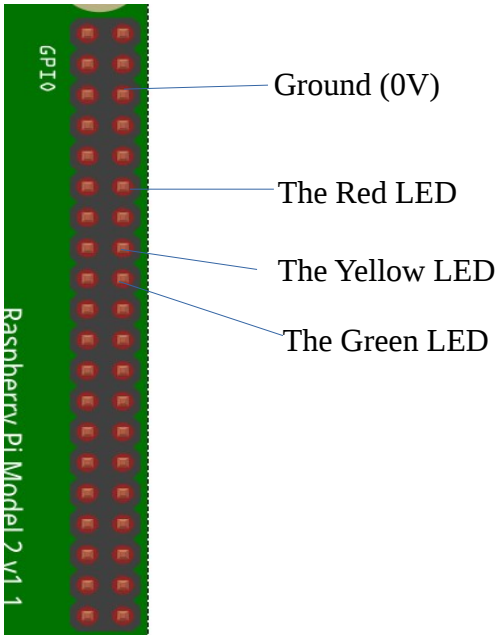
To make the circuit it is always best to have the raspberry pi turned off at this stage.

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

Models A & B



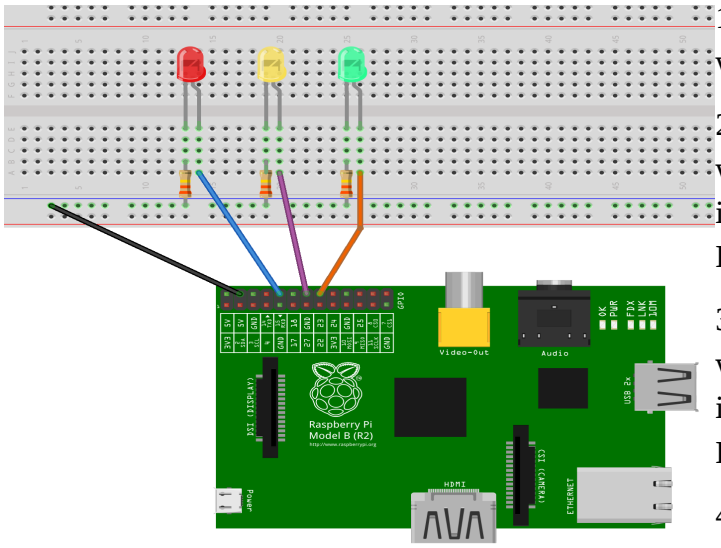
Model A+, B+ & Raspberry Pi 2



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

2. Put the Yellow LED into the breadboard with the longer leg on the right hand side, it is best to leave a gap between the two LED's

3. Put the Green LED into the breadboard with the longer leg on the right hand side, it is best to leave a gap between the two LED's.

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

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

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

7. 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.

8. Take another jumper wire and connect the end with the pin into the same column as the longer leg of the Red LED and the side with the hole onto pin 18 on the raspberry pi.

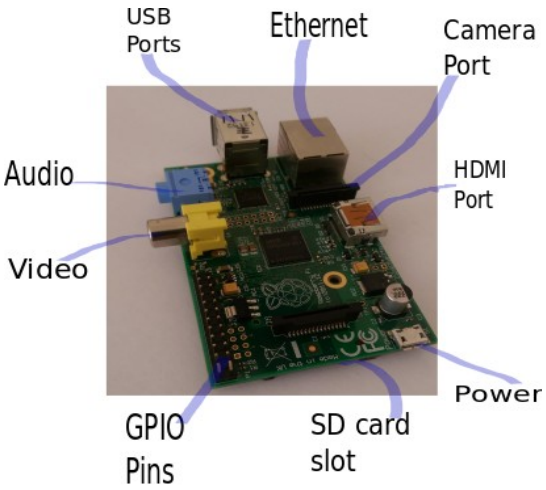
9. Take another jumper wire and connect the end with the pin into the same column as the longer leg of the Yellow LED and the side with the hole onto pin 23 on the raspberry pi.

10. Take the last jumper wire and connect the end with the pin into the same column as the longer leg of the Green LED and the side with the hole onto pin 24 on the raspberry pi.

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

Setting Up The Raspberry Pi

- Plug in your SD card or Micro SD card depending on the raspberry pi you have
- Plug one end of the HDMI cable into the raspberry pi and the other end into the monitor
- Plug in the mouse to one of the USB ports
- Plug the keyboard into another USB port
- plug in the power supply into the micro USB port.

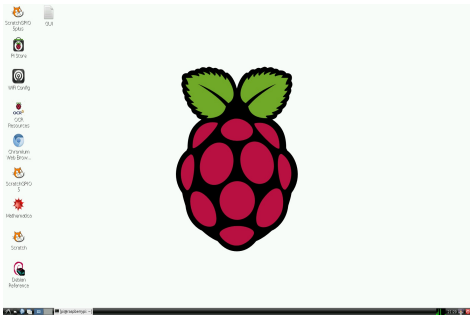


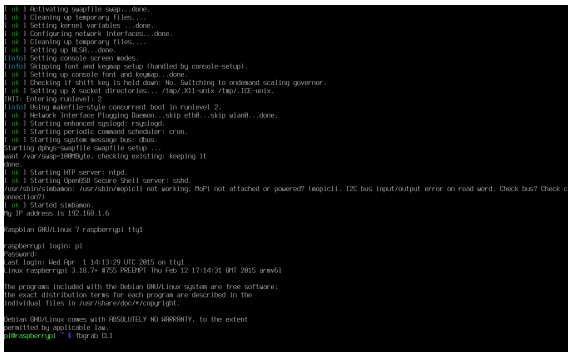
After you have connected all the wires and inserted your SD card, your pi should look a little like this.

After the raspberry pi has finished booting, you will either end up in the

command line interface (CLI) as in the picture below on the right or in the graphical user interface (GUI) as the picture below on the left. If you see the CLI then you will need to:

- Type the user name “pi”
- Type the password. The default password is “raspberry”





Code

You are now going to create your first python program that will take input from you and display it on the screen.

If you are not in the CLI then you will want to open the LxTerminal application. 

The first thing we are going to do is create a new directory called Python. This is where all our python code is stored. Type the following commands which are in blue pressing the “enter” key after every line:

cd ~	Changes to the home directory
mkdir Python	Creates a directory (folder) called Python
cd Python	Changes to the python directory
nano UserInput.py	Opens the “nano” text editor creating the file called “UserInput.py”

You will now be in the “nano” text editor. This is where we will type our code. Type in the code exactly as seen below, which is in red.

#! /usr/bin python	Telling the pi to use the python interpreter
import os	This is for operating system functionality
import time	This is for time related commands
import Rpi.GPIO as GPIO	Importing the GPIO library as GPIO
GPIO.setmode (GPIO.BCM)	Set the GPIO pins naming mode
GPIO.setwarnings (False)	Telling python not to display GPIO warnings to

	screen
LEDRed = 18	Setting pin 18 as a variable for the Red LED
LEDYellow = 23	Setting pin 23 as a variable for the Yellow LED
LEDGreen = 24	Setting pin 24 as a Variable for the Green LED
GPIO.setup (LEDRed, GPIO.OUT)	Setting the Red LED to an output
GPIO.setup (LEDYellow, GPIO.OUT)	Setting the Yellow LED to an output
GPIO.setup (LEDGreen, GPIO.OUT)	Setting the Green LED to an output
led_choice = 0	A variable for the choice of LED from the user. Set to 0 by default.
count = 0	Counter variable set to 0 by default
LEDChosen = 0	A variable for The LED chosen by the user, set to 0 by default
os.system ('clear')	This clears the screen
print “Which LED would you like to blink? “	Prints “Which LED would you like to blink? “ to screen
print “1: Red”	Prints “1: Red” to screen
print “2: Yellow”	Prints “2: Yellow” to screen
Print “3: Green”	Prints “3: Green” to screen
led_choice = input (“Chose your option: “	This asks the user to enter either 1 for Red, 2 for yellow or 3 for Green and sets the variable led_choice to the chosen LED
count = input (“How many times would you like it to blink? “	This asks the user to input the number of times the LED should blink for. Setting the count variable to number of times
if led_count == 1	Determines whether option 1 has been chosen or not
print “You picked the Red LED”	Prints “You picked the Red LED if option 1 was chosen
LEDChosen = LEDRed	Sets the LEDChosen variable to LEDRed if option 1 was selected
If led_count == 2	Determines whether option 2 has been chosen or not
print “You picked the Yellow LED”	Prints “You picked the Yellow LED if option 2 was selected
LEDChosen = LEDYellow	Sets the LEDChosen variable to LEDYellow if option 2 was selected
If led_choice == 3	Determines whether option 3 has been chosen or not

print “You picked the Green LED”	Prints “You picked the Green LED if option 3 was chosen
LEDChosen = LEDGreen	Sets the LEDChosen variable to LEDGreen if option 3 was selected
if LEDChosen <> 0:	Checks if the variable LEDChosen is = to 0 or not
while count > 0:	Determines if count is greater than 0
GPIO.output (LEDChosen, GPIO.HIGH)	While count is greater than 0 the chosen LED is turned on
time.sleep (1)	Program pauses for 1 second
GPIO.output (LEDChosen, GPIO.LOW)	The chosen LED is turned off
time.sleep (1)	Program pauses for 1 second
count = count – 1	This subtracts 1 from the variable count and sets count with the new value
GPIO.cleanup ()	This sets the GPIO pins back to their defaults

Running The Code

sudo python UserInput.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.