MODMYPI

# YouTube

## Workshop Kit v2.0

## Tutorial 2/9: LED On & Off

# Contents

## Things you will need

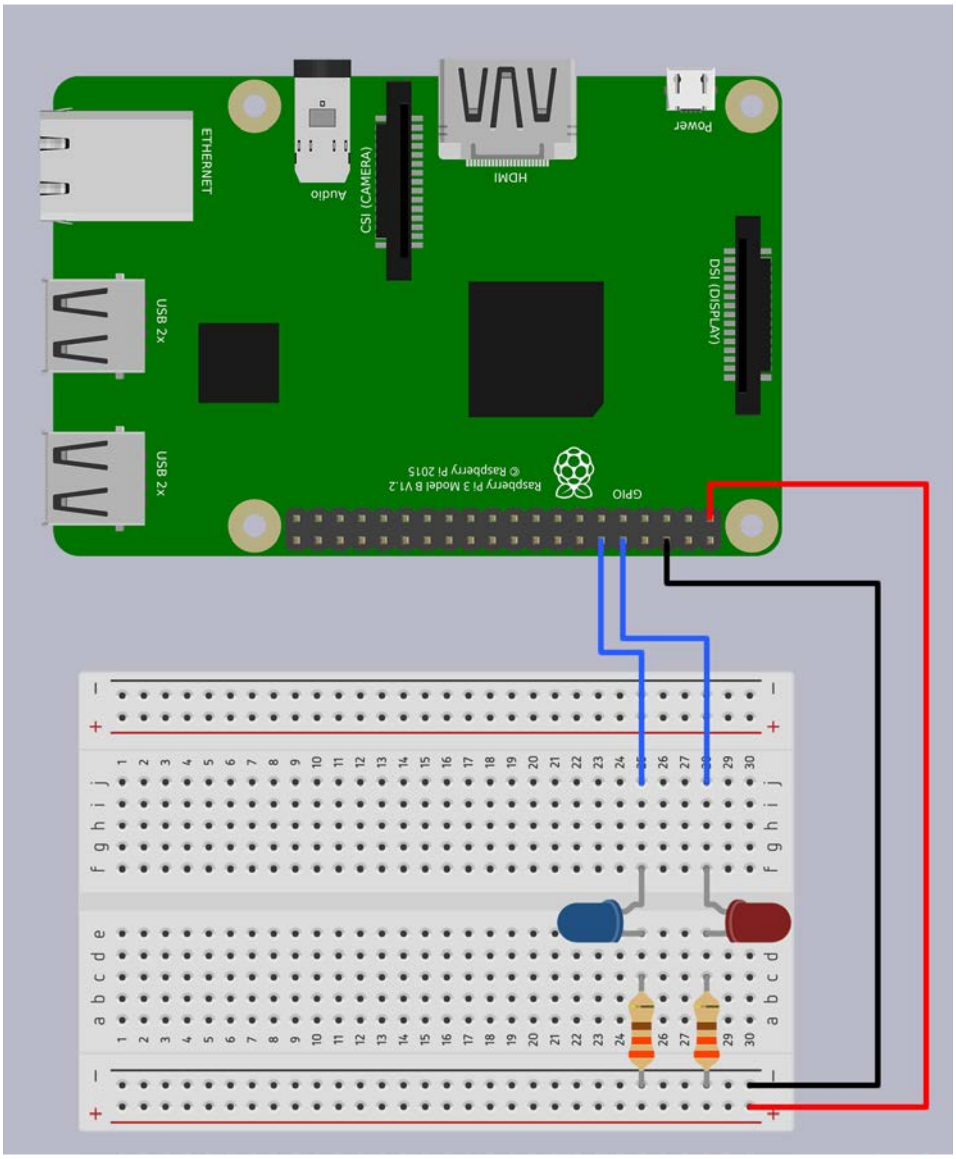
Raspberry Pi 3 Model B
Class 10 Micro SD Card
Keyboard + Mouse
Monitor + HDMI Cable
Power Supply (Recommended: 5V 2.5A)
Breadboard
1x Red LED
1x Blue LED
2x 330Ω Resistor
3x M/F Jumper Wire

*If you are connecting to your Raspberry Pi remotely using VNC or other means then Keyboard, mouse Monitor and HDMI cable are optional.*

## Prerequisites

You will need to install the latest version of Raspbian on to your Micro SD Card. Initial setup will require a keyboard, mouse, HDMI cable and Monitor/TV.

## Wiring Diagram

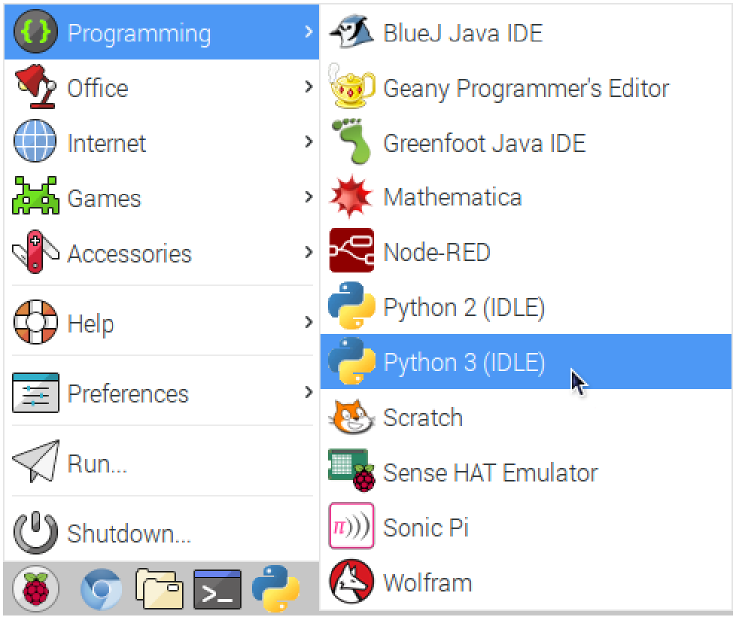


## Introduction

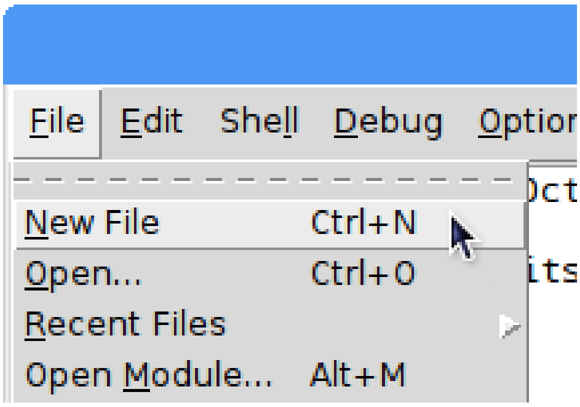
In this tutorial we will be making a Light Emitting Diode (LED) turn on and off

## Getting Started

To get started, first you need to open Python 3 (IDLE). To do this click on the Raspberry Pi icon on the task bar, highlight "Programming" then click on "Python 3 (IDLE)"



When IDLE has loaded, you will want to start working on a new file. You can do this by clicking on File and select "New File" or by pressing Ctrl+N

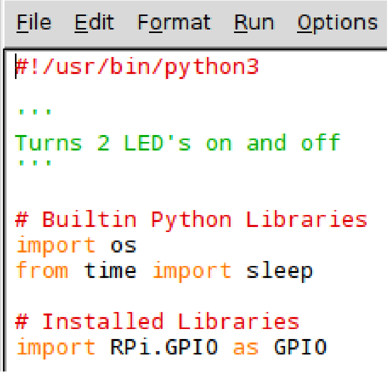


Now it is time to start coding

# Writing the Code

The first thing you should type is a shebang line, docstring and import your modules

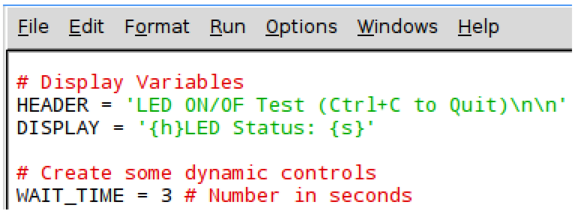
```
#!/usr/bin/python3

'''
Turns 2 LED's on and off
'''

# Builtin Python Libraries
import os
from time import sleep

# Installed Libraries
import RPi.GPIO as GPIO
```



```
#!/usr/bin/python3

'''
Turns 2 LED's on and off
'''

# Builtin Python Libraries
import os
from time import sleep

# Installed Libraries
import RPi.GPIO as GPIO
```

## Display Variables

Now you need to create some variables, 2 string variables and 1 integer variable. The string variables will be used to display text in the terminal to let us know what the program is doing. The integer variable will be used for a sleep timer.



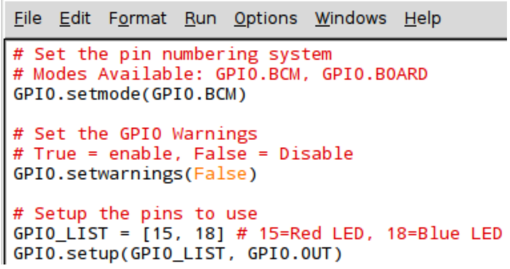
```
# Display Variables
HEADER = 'LED ON/OF Test (Ctrl+C to Quit)\n\n'
DISPLAY = '{h}LED Status: {s}'

# Create some dynamic controls
WAIT_TIME = 3 # Number in seconds
```

```
# Display Variables
HEADER = 'LED ON/OF Test (Ctrl+C to
Quit)\n\n'
DISPLAY = '{h}LED Status: {s}'

# Create some dynamic controls
WAIT_TIME = 3 # Number in seconds
```

## Setting up the GPIO

Time to setup the GPIO. We will set mode and warnings and also which pins that you will be using.

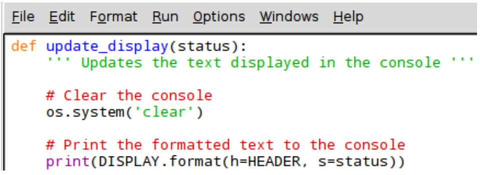
```
# Set the pin numbering system
# Modes Available: GPIO.BCM, GPIO.BOARD
GPIO.setmode(GPIO.BCM)

# Set the GPIO Warnings
# True = enable, False = Disable
GPIO.setwarnings(False)

# Setup the pins to use
GPIO_LIST = [15, 18] # 15=Red LED, 18=Blue LED
GPIO.setup(GPIO_LIST, GPIO.OUT)
```



```
# Set the pin numbering system
# Modes Available: GPIO.BCM, GPIO.BOARD
GPIO.setmode(GPIO.BCM)

# Set the GPIO Warnings
# True = enable, False = Disable
GPIO.setwarnings(False)

# Setup the pins to use
GPIO_LIST = [15, 18] # 15=Red LED, 18=Blue LED
GPIO.setup(GPIO_LIST, GPIO.OUT)
```

## Update Display Method

Next, it is time to create a Method that will update the text that will be displayed in the console. This will be called when you want to update the information in the console.



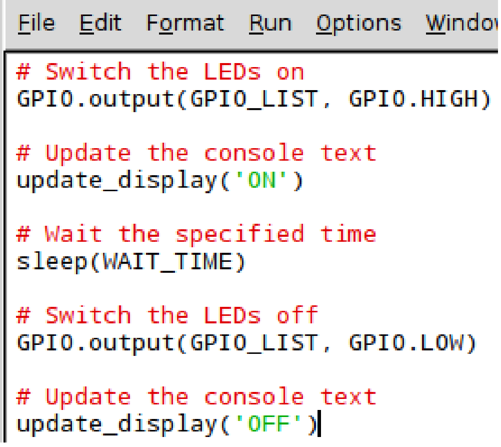
```
def update_display(status):
    ''' Updates the text displayed in the console '''

    # Clear the console
    os.system('clear')

    # Print the formatted text to the console
    print(DISPLAY.format(h=HEADER, s=status))
```

```
def update_display(status):
    ''' Updates the text displayed in the console '''

    # Clear the console
    os.system('clear')
    # Print the formatted text to the console
    print(DISPLAY.format(h=HEADER, s=status))
```

## The Action Code

The last thing you will need to add is to tell the program to turn on an LED, update the console text, wait for a couple seconds, turn the LED off then update the console text again.

```
# Switch the LEDs on
GPIO.output(GPIO_LIST[0], GPIO.HIGH)

# Update the console text
update_display('ON')

# Wait the specified time
sleep(WAIT_TIME)

# Switch the LEDs off
GPIO.output(GPIO_LIST[0], GPIO.LOW)

# Update the console text
update_display('OFF')
```



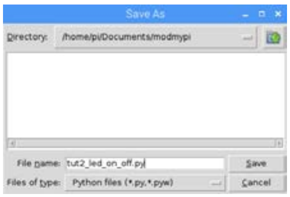
```
# Switch the LEDs on
GPIO.output(GPIO_LIST, GPIO.HIGH)

# Update the console text
update_display('ON')

# Wait the specified time
sleep(WAIT_TIME)

# Switch the LEDs off
GPIO.output(GPIO_LIST, GPIO.LOW)

# Update the console text
update_display('OFF')
```

If you would like to turn on the Blue LED instead of the red replace GPIO_LIST[0] with GPIO_LIST[1]. If you would like to turn on both LED's at the same time, replace GPIO_LIST[0] with GPIO_LIST

## Save the Program

That is all there is for the code. Now you need to save it. First open up the File Manager by clicking on this icon on the taskbar and open the Documents folder.

In here, create a new folder, right click select "Create New" and click Folder, or press Ctrl+Shift+N. Call the folder modmypi and click ok.



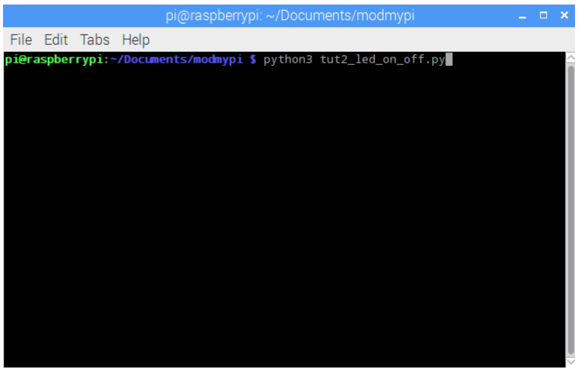
Directory: /home/pi/Documents/modmypi
File name: tut2_led_on_off.py
Files of type: Python files (*.py,*.pyw)

Go back to the IDLE window where you typed in your code and click on File and select "Save As". Navigate to /home/pi/Documents/modmypi and enter tut2_led_on_off.py for the filename then click Save.

## Running the Program

Now that you have saved your code, it is time to run it so that you can make sure that it works as it should. Go back to the File Manager and open the modmypi folder you created. Next click on tools and select "Open Current Folder in Terminal" or press F4.

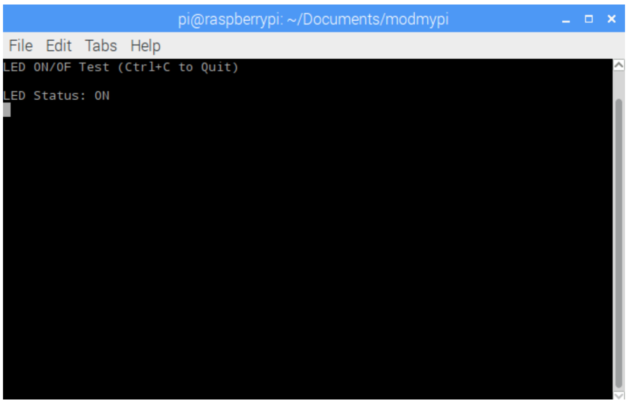
In the terminal, type python3 tut2_led_on_off.py and press enter



```
pi@raspberrypi:~/Documents/modmypi $ python3 tut2_led_on_off.py
```

## Results

If everything is working correctly you should see the LED(s) turn on and then after a few seconds they LED(s) should turn off. In the console you should see something like



```
LED ON/OF Test (Ctrl+C to Quit)

LED Status: ON
```

## Code on GitHub

If you would like to download a copy of the code, you can download it from along with all the other tutorials, code and wiring diagrams from GitHub here

## Thanks

Thank you for taking the time to follow this tutorial and hope that you have found this useful. Please feel free to follow the other tutorials that have been created for the ModMyPi YouTube Workshop Kit.