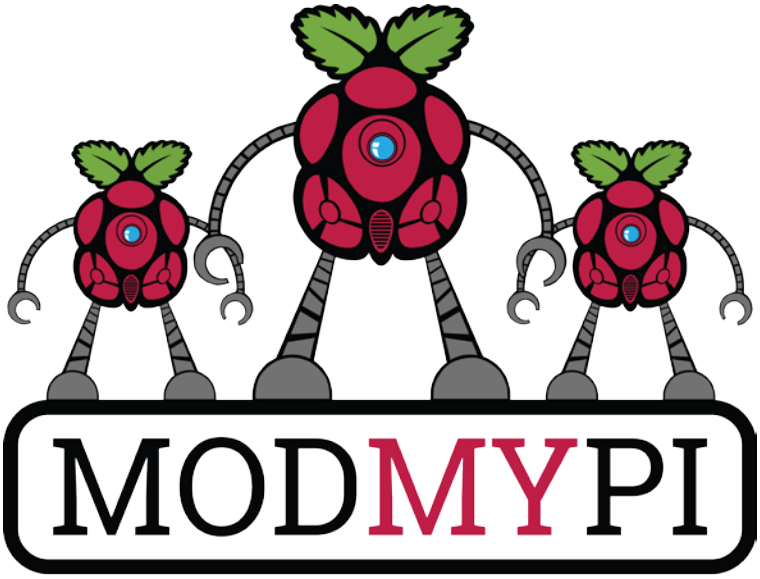


YouTube

Workshop Kit v2.0

Tutorial 5/9: User Input Blink

Contents

Things you will need

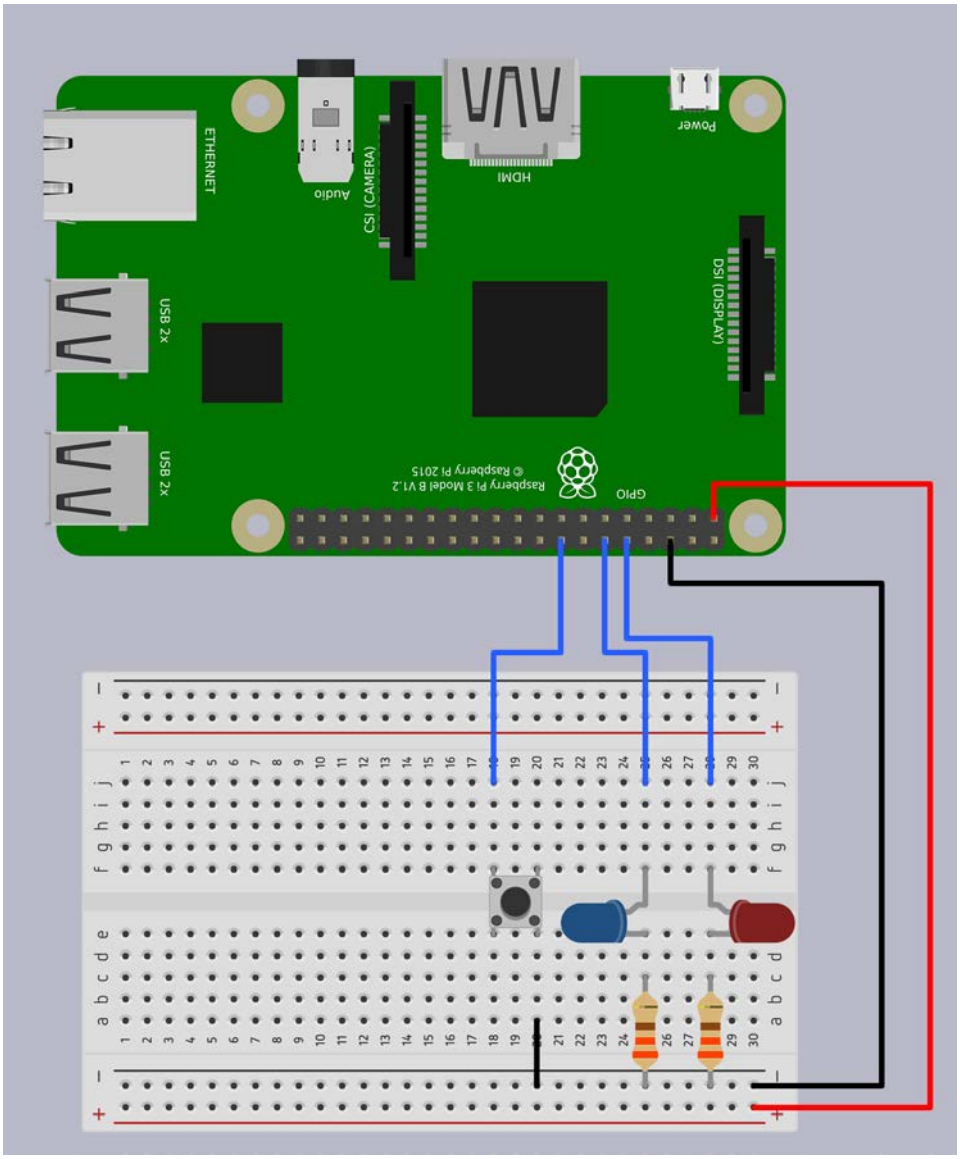
Raspberry Pi 3 Model B
Class 10 Micro SD Card
Keyboard + Mouse
Monitor + HDMI Cable
Power Supply (Recommended: 5V 2.5A)
Breadboard
1x Red LED
1x Blue LED
2x 330Ω Resistor
1x M/M Jumper Wire
4x M/F Jumper Wire
1x Button

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

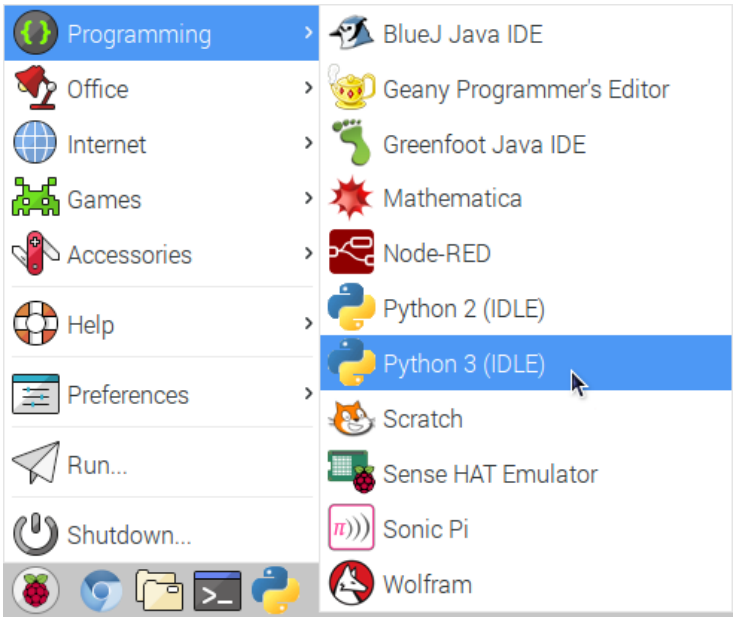
In this tutorial we will be making a Light Emitting Diode (LED) blink on and off as specified by the users input. This tutorial is similar to tutorial 3, however, we are adding in user input to specify which LED blinks and how many times.

These tutorials are written on a Raspberry Pi 3 Model B with a clean install of Rasbian 4.4.

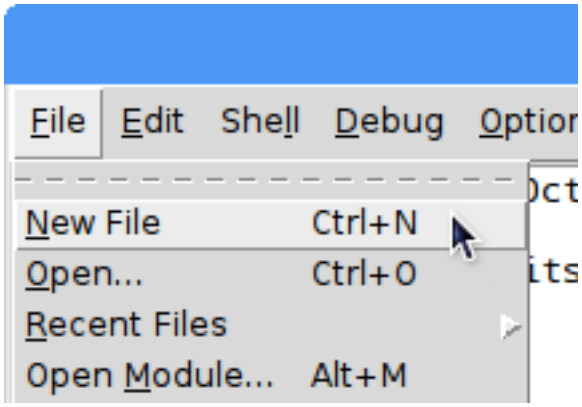
Some parts of the tutorials may look familiar as the code examples are written in a way that there is very little work needed and minor modifications to the previous code to get you up and running faster.

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)”

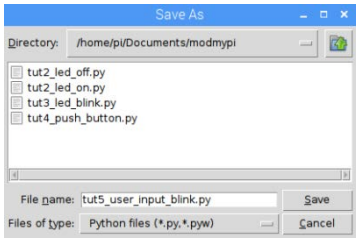


When IDLE has loaded, you will want to start working on a new file. You can do this by clicking on File and select “New File” or by pressing Ctrl+N



Save the Program

Now that we have IDLE running, first save a new file. First open up the File Manager by clicking on this icon on the taskbar and open the Documents folder.

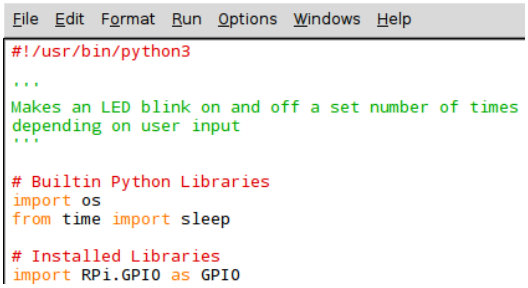


Go back to the IDLE and click on File and select “Save As”. Navigate to /home/pi/Documents/modmypi and enter tut5_user_input_blink.py for the filename then click Save.

Writing the Code

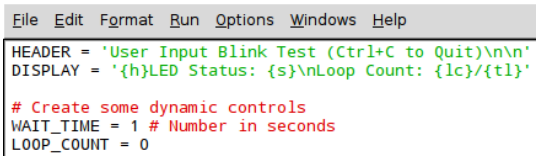
The first thing you should type is a shebang line, docstring and import your modules

```
#!/usr/bin/python3

'''
Makes an LED blink on and off a set number of times
depending on user input
'''

# 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 variables. The string variables will be used to display text in the terminal to let us know what the program is doing. The integer variables will be used for a sleep timer.

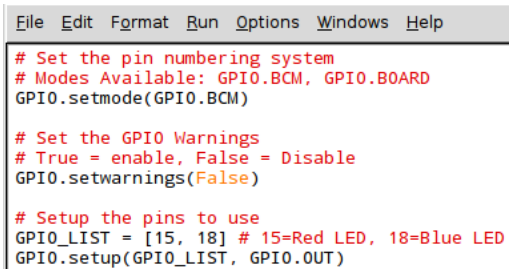



```
# Display Variables
HEADER = 'User Input Blink Test (Ctrl+C to Quit)\n\n'
DISPLAY = '{h}LED Status: {s}\nLoop Count: {lc}/{tl}'

# Create some dynamic controls
WAIT_TIME = 2 # Number in seconds
LOOP_COUNT = 0
```

Setting up the GPIO

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

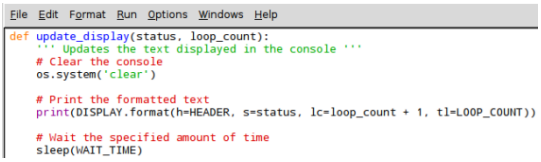
```
# Set the pin numbering system
# Modes Available: GPIO.BCM, GPIO.BOARD
GPIO.setmode(GPIO.BCM)

# 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)
```



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.

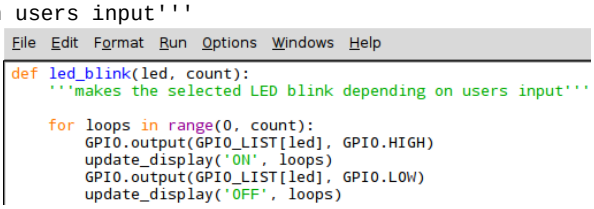
```
def update_display(status, loop_count):
    ''' Updates the text displayed in the console '''

    # Clear the console
    os.system('clear')
    # Print the formatted text to the console
    print(DISPLAY.format(h=HEADER, s=status, lc=loop_count + 1, tl=LOOP_COUNT))

    # Wait the specified amount of time
    sleep(WAIT_TIME)
```



LED Blink Method

This method is used to turn the LEDs on and off as well as update the console text. This method made from the action code from tutorial 3.

```
def led_blink(led, count):
    '''makes the selected LED blink depending on users input'''

    for loops in range(0, count):
        GPIO.output(GPIO_LIST[led], GPIO.HIGH)
        update_display('ON', loops)
        GPIO.output(GPIO_LIST[led], GPIO.LOW)
        update_display('OFF', loops)
```

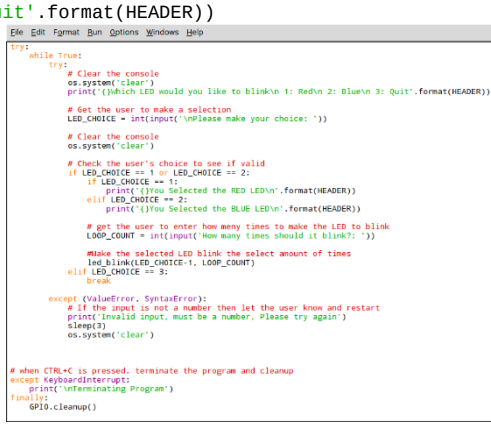


The Action Code – User Input

You will now need to add code to tell the program to turn on an LED when the push button is pressed and then reset after 2 seconds.

The the int statement allows you to convert a number string into an integer, however if you try and convert a string with letters or special characters then it will thow an exception error of either ValueError or SyntaxError. To prevent the exceptions crashing the program, we use a try/except statement to catch the errors and execute our own code, in this case, we tell the user to only enter numbers.

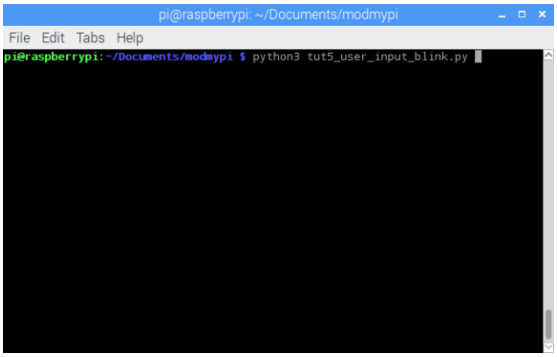
```
try:
    while True:
        try:
            # Clear the console
            os.system('clear')
            print('{}Which LED would you like to blink\n 1: Red\n 2: Blue\n 3: Quit'.format(HEADER))

            # Get the user to make a selection
            LED_CHOICE = int(input('\nPlease make your choice: '))

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

            # Check the user's choice to see if valid
            if LED_CHOICE == 1 or LED_CHOICE == 2:
                if LED_CHOICE == 1:
                    print('{}You Selected the RED LED\n'.format(HEADER))
                elif LED_CHOICE == 2:
                    print('{}You Selected the BLUE LED\n'.format(HEADER))

                # get the user to enter how many times to make the LED to blink
                LOOP_COUNT = int(input('How many times should it blink?: '))

                #Make the selected LED blink the select amount of times
                led_blink(LED_CHOICE-1, LOOP_COUNT)
            elif LED_CHOICE == 3:
                break

        except (ValueError, SyntaxError):
            # If the input is not a number then let the user know and restart
            print('Invalid input, must be a number, Please try again')
            sleep(3)
            os.system('clear')

# when CTRL+C is pressed, terminate the program and cleanup
except KeyboardInterrupt:
    print('\nTerminating Program')
finally:
    GPIO.cleanup()
```



Make sure to save your work by clicking File and select Save, or press Ctrl+S

Running the Program

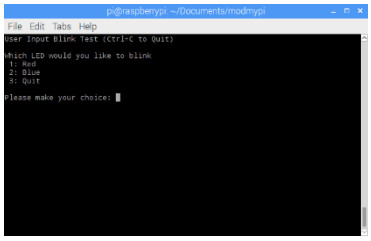
Save your work and it is time to run it so that you can make sure that it works as it should. Go back to the File Manager and open the modmypi folder you created. Next click on tools and select “Open Current Folder in Terminal” or press F4.

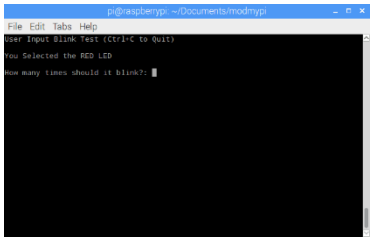


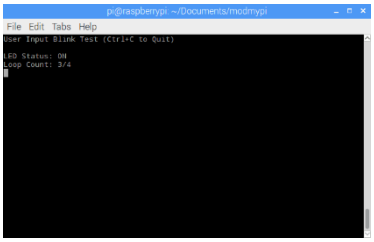
In the terminal, type
python3 tut5_user_input_blink.py
and press enter

Results

If everything is working correctly you should see in the console that it is asking you to select which LED you would like to blink, when you make your choice it then asks you how many time to blink the LED. In the console you should see something similar to this when running:







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.