ASSIGNMENT-3

Date	04 october 2022
Team ID	PNT2022TMID01040
Name	Gas Leakage monitoring & Alerting system for industries

PYTHON CODE FOR BLINKING LED AND TRAFFIC LIGHT FOR RASPBERRY PI

❖ LED BLINKING USING RASBERRY PI - PYTHON CODE

- To install the Python library open a terminal and execute the following:

```
$ sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

- With the library installed now open your favorite Python IDE:

Our script needs to do the following:

- Initialize the GPIO ports
- Turn the LED on and off in 1 second intervals

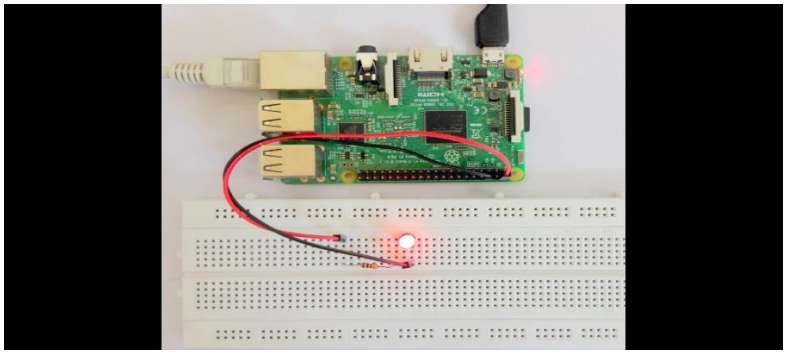
```
import RPi.GPIO as GPIO    # Import Raspberry Pi GPIO library

from time import sleep     # Import the sleep function from the time module


GPIO.setwarnings(False)    # Ignore warning for now

GPIO.setmode(GPIO.BOARD)   # Use physical pin numbering

GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)   # Set pin 8 to be an output pin and
 set initial value to low (off)
```

- Next we need to turn the LED on and off in 1 second intervals by setting the output pin to either high (on) or low (off). We do this inside a infinite loop so our program keep executing until we manually stop it.

```
while True: # Run forever

    GPIO.output(8, GPIO.HIGH) # Turn on

    sleep(1)                  # Sleep for 1 second

    GPIO.output(8, GPIO.LOW)  # Turn off

    sleep(1)                  # Sleep for 1 second
```

FINAL PROGRAM

```
import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library

from time import sleep # Import the sleep function from the time module



GPIO.setwarnings(False) # Ignore warning for now

GPIO.setmode(GPIO.BOARD) # Use physical pin numbering

GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 to be an output pin and set
initial value to low (off)

while True: # Run forever

 GPIO.output(8, GPIO.HIGH) # Turn on

 sleep(1) # Sleep for 1 second

 GPIO.output(8, GPIO.LOW) # Turn off

 sleep(1) # Sleep for 1 second
```



BLINKING TRAFFIC LIGHTS FOR RASBERRY PI (PYTHON CODE)

First, you need to install a couple of extra software packages needed to allow you to download my sample code, and to give Python access to the GPIO pins on the Pi. Enter the following at the command line:

```
$ sudo apt-get install python-dev python-rpi.gpio git
import RPi.GPIO as GPIO
import time
import signal
import sys
# Setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(9, GPIO.OUT)
GPIO.setup(10, GPIO.OUT)
GPIO.setup(11, GPIO.OUT)
# Turn off all lights when user ends demo
def allLightsOff(signal, frame):
    GPIO.output(9, False)
    GPIO.output(10, False)
    GPIO.output(11, False)
    GPIO.cleanup()
    sys.exit(0)
signal.signal(signal.SIGINT, allLightsOff)
```

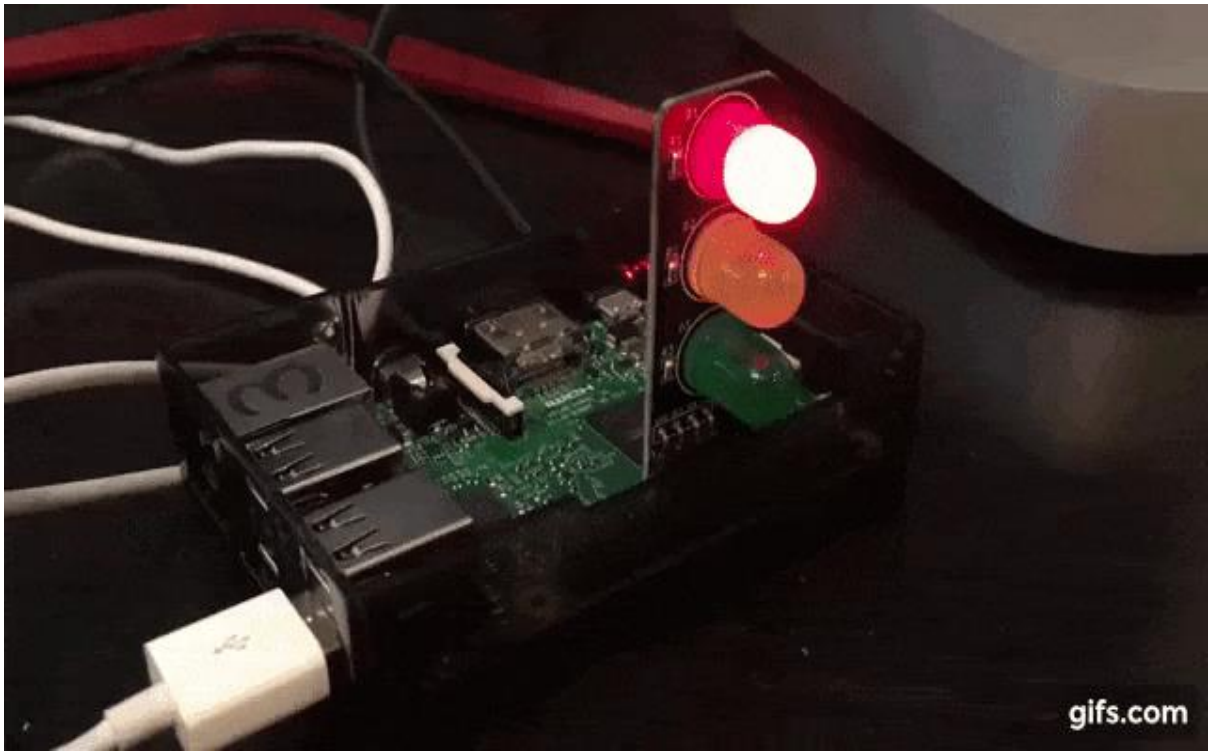
The main body of the code then consists of an infinite `while` loop that turns on the red light (pin 9), waits, turns on the amber light

(pin 10), waits, then cycles through the rest of the traffic light pattern by turning the appropriate LEDs on and off:

```
while True:
    # Red
    GPIO.output(9, True)
    time.sleep(3)
    # Red and amber
    GPIO.output(10, True)
    time.sleep(1)
    # Green
    GPIO.output(9, False)
    GPIO.output(10, False)
    GPIO.output(11, True)
```

```
    time.sleep(5)
    # Amber
    GPIO.output(11, False)
    GPIO.output(10, True)
    time.sleep(2)
    # Amber off (red comes on at top of loop)
    GPIO.output(10, False)
```

- When `Control-C` is pressed an interrupt signal `signal.SIGINT` is sent. This is handled by the `allLightsOff` function that switches all the lights off, tidies up the GPIO library state and exits cleanly back to the operating system.


gifs.com