

ASSIGNMENT 3

LED BLINKING USING RASPBERRY PI

PYTHON CODE:

```
import RPi.GPIO as GPIO # RPi.GPIO can be referred to as GPIO from now
import time

ledPin = 22    # pin22

def setup():
    GPIO.setmode(GPIO.BOARD)    # GPIO Numbering of Pins
    GPIO.setup(ledPin, GPIO.OUT) # Set ledPin as output
    GPIO.output(ledPin, GPIO.LOW) # Set ledPin to LOW to turn Off the LED

def loop():
    while True:
        print 'LED on'
        GPIO.output(ledPin, GPIO.HIGH) # LED On
        time.sleep(1.0)                # wait 1 sec
        print 'LED off'
        GPIO.output(ledPin, GPIO.LOW)  # LED Off
        time.sleep(1.0)                # wait 1 sec

def endprogram():

    GPIO.output(ledPin, GPIO.LOW)  # LED Off
    GPIO.cleanup()                # Release resources

if __name__ == '__main__':        # Program starts from here
    setup()
    try:
        loop()
    except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the destroy() will be executed.
        endprogram()
```

CODE EXPLANATION: In the Python program, first, we have imported two packages *RPi.GPIO* and *time*. The package *RPi.GPIO* will help us in controlling the GPIO Pins of the Raspberry Pi.

The first important function of the *RPi.GPIO* Module is the *set mode()*. Using *GPIO.setmode()*, we can select either GPIO Numbering of the Pins or Physical Numbering. By using ***GPIO.setmode(GPIO.BOARD)***, we are selecting the Physical Numbering Scheme.

The next function is the *setup(pin, mode)*. This function will allow us to set the pin as either input (GPIO.IN) or as output (GPIO.OUT). In the program, I have set the ledPin as output by using ***GPIO.setup(ledPin, GPIO.OUT)***.

After setting the LED Pin as OUTPUT, now we need to set the state of this OUTPUT i.e. HIGH (GPIO.HIGH) or LOW (GPIO.LOW). For this, we need to use the function *output(pin, state)*. So, in our program, we need to use ***GPIO.output(ledPin, GPIO.HIGH)*** for turning ON the LED and ***GPIO.output(ledPin, GPIO.LOW)*** for turning it OFF.

The last function is the ***GPIO.cleanup()***. With the help of this function, we can make a clean exit from the program as it will clean all the resources that are used in the program.

TRAFFIC LIGHT using RASPBERRY PI

PYTHON CODE:

```
from gpiozero import LED
from time import sleep
red=LED(22)
orange=LED(27)
green=LED(17)
while True:
    red.on()
    sleep(1)
    orange.on(1)
    sleep(1)
    green.on()
    sleep(1)
    red.off()
    sleep(1)
    orange.off()
    sleep(1)
    green.off()
```

CODE EXPLANATION:

The Traffic lights interface takes three GPIO pin numbers, one for each pin: red, orange, and green. While loop is used to control the traffic lights.

SUBMITTED BY: MADHUMITHA.G