

IBM – NALAYATHIRAN PROJECT

Problem Statement :

**IoT-Based Smart Crop Protection System for
Agriculture**

Domain :

Internet of Things

Assignment 3 :

To build a python code, for blinking LED and traffic lights for Raspberry Pi.

SUBMITTED BY:

HARINI AANANTHI K S - 917719D026

KEERTHIGA R M – 917719D040

SHENBAGA THENDRAL B – 917719D090

SNEHA S R – 917719D094

CODE:

LED blinking:

```
import time
```

```
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BOARD)
```

```
led = 5 # GPIO pin number is 5 and name is GPIO3
```

```
GPIO.setup(led, GPIO.OUT, initial = 0) # Setup LED and  
set it initially to OFF
```

```
while(True):
```

```
    GPIO.output(led, GPIO.HIGH) # Set LED to ON
```

```
    time.sleep(2) # Wait for 2 seconds
```

```
    GPIO.output(led, GPIO.LOW) # Set LED to OFF
```

```
    time.sleep(2)
```

Traffic light:

```
import RPi.GPIO as GPIO
```

```
import time
```

```
# Using physical pin locations
```

```
GPIO.setmode(GPIO.BOARD)
```

```
# Pin header IDs for LEDs and button
```

```
YellowLed = 35
```

```
RedLed = 33
```

```
GreenLed = 37
```

```
safeCrossing = 38
```

```
button = 11
```

```
# Set up LEDs
```

```
GPIO.setup(RedLed, GPIO.OUT)
```

```
GPIO.setup(YellowLed, GPIO.OUT)
```

```
GPIO.setup(GreenLed, GPIO.OUT)
```

```
GPIO.setup(safeCrossing, GPIO.OUT)
```

```
GPIO.output(RedLed, GPIO.HIGH)
```

```
GPIO.output(YellowLed, GPIO.HIGH)
```

```
GPIO.output(GreenLed, GPIO.HIGH)
```

```
# Set up button
```

```
GPIO.setup(button,GPIO.IN,pull_up_down=GPIO.PUD_
DOWN)
```

```
def cycleLights ():
```

```
    print ('Traffic: GREEN off, AMBER on')
```

```
    GPIO.output(GreenLed, GPIO.HIGH)
```

```
    GPIO.output(YellowLed, GPIO.LOW)
```

```
    time.sleep(1)
```

```
    print ('Traffic: AMBER off, RED on')
```

```
    GPIO.output(YellowLed, GPIO.HIGH)
```

```
    GPIO.output(RedLed, GPIO.LOW)
```

```
    time.sleep(1)
```

```
    print ('Pedestrian: Safe to cross on')
```

```
    GPIO.output(safeCrossing, GPIO.LOW)
```

```
time.sleep(5)
```

```
print ('Pedestrian: Safe to cross flashing')
```

```
for flash in range(0, 5):
```

```
    GPIO.output(safeCrossing, GPIO.HIGH)
```

```
    time.sleep(0.8)
```

```
    GPIO.output(safeCrossing, GPIO.LOW)
```

```
    time.sleep(0.8)
```

```
print ('Pedestrian: Safe to cross off')
```

```
GPIO.output(safeCrossing, GPIO.HIGH)
```

```
time.sleep(1)
```

```
print ('Traffic: AMBER and RED on')
```

```
GPIO.output(YellowLed, GPIO.LOW)
```

```
time.sleep(1.5)
```

```
print ('Traffic: AMBER and RED off, GREEN on')
```

```
GPIO.output(RedLed, GPIO.HIGH)
```

```
GPIO.output(YellowLed, GPIO.HIGH)
```

```
GPIO.output(GreenLed, GPIO.LOW)
```

```
print ('Pedestrian button blocked to let traffic flow')
```

```
time.sleep(4)
```

```
print ('Pedestrian button unblocked')
```

```
return
```

```
def teardown ():
```

```
    GPIO.output(RedLed, GPIO.HIGH)
```

```
    GPIO.output(YellowLed, GPIO.HIGH)
```

```
    GPIO.output(GreenLed, GPIO.HIGH)
```

```
    GPIO.cleanup()
```

```
    return
```

```
try:
```

```
    while True:
```

```
        ButtonPress = False
```

```
        # Lights start with the green traffic light on
```

```
        # and the pedestrian light off
```

```
        GPIO.output(GreenLed, GPIO.LOW)
```

```
GPIO.output(safeCrossing, GPIO.HIGH)
```

```
# Wait until button is presses
```

```
print ('Waiting for a padestrian to press the  
button', end="")
```

```
while not ButtonPress:
```

```
    # Check every 2 seconds for a press
```

```
    print ('.', end="")
```

```
    time.sleep(1)
```

```
    ButtonPress = GPIO.input(button)
```

```
print ('\nPadestrian button press detected!')
```

```
cycleLights()
```

```
except KeyboardInterrupt:
```

```
    teardown()
```

