ASSINGMENT-3

|  |  |
| --- | --- |
| DATE | 22 September 2022 |
| TEAM ID | PNT2022TMID37462 |
| NAME | Syed Aakib |
| PROJECT NAME | Industry-Specific Intelligent Fire  management System |

**Python code for Blinking Led using Raspberry pi**

#!/usr/bin/python

Import RPi.GPIO import time

class Led(object):

'''

Represents a physical LED. It uses the RPi GPIO naming scheme and NOT

Broadcomm's ''' def \_\_init\_\_(self, pin\_number):

'''

Set up the hardware connection

Params: pin\_number of type int - Follow RPi GPIO naming scheme

'''

self.pin\_number = pin\_number self.\_\_setup\_gpio\_\_()

def \_\_setup\_gpio\_\_(self):

RPi.GPIO.setmode(RPi.GPIO.BOARD)

RPi.GPIO.setup(self.pin\_number, RPi.GPIO.OUT)

def clean\_up(self):

'''

Reset the GPIO header to its initial state.

'''

RPi.GPIO.cleanup(self.pin\_number)

def on\_light(self):

'''

Switch on the LED

'''

RPi.GPIO.output(self.pin\_number, True)

def off\_light(self):

'''

Switch off the LED

'''

RPi.GPIO.output(self.pin\_number, False)

def blink(self, drift\_time=0.2): self.on\_light() time.sleep(float(drift\_time)) self.off\_light()

time.sleep(float(drift\_time))

def blinkn(self, number\_times):

'''

Blink for any number of times

Param: number\_times of type int

''' for i in range(0, int(number\_times)):

self.blink()

def blink\_non\_stop(self): while True: self.blink()

# Python Code For Traffic LED using Raspberry pi

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) 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)