**BLINKING LED AND TRAFFIC LIGHTS FOR RASPBERRY PI**

**BLINKING LED**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(21,GPIO.OUT)

while(1):

GPIO.output(21,GPIO.HIGH)

time.sleep(1)

GPIO.output(21,GPIO.LOW)

time.sleep(1)

**TRAFFIC LIGHTS**

import sys

import RPi.GPIO as GPIO

from threading import Timer

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

aLights = {

"north": [36, 38, 40],

"east": [33, 31, 29]

}

for i in aLights["north"]:

GPIO.setup(i, GPIO.OUT)

for i in aLights["east"]:

GPIO.setup(i, GPIO.OUT)

iLightDelay = 2

iGreenTime = 8

bOrangeBeforeGreen = False

sGreen = "north"

def init():

changeLightTo("north", "red")

changeLightTo("east", "red")

Timer(iLightDelay, startUp).start()

def startUp():

if bOrangeBeforeGreen == True:

changeLightTo("north", "redorange")

Timer(iLightDelay, changeNorthToGreen).start()

else:

changeNorthToGreen()

def changeNorthToGreen():

changeLightTo("north", "green")

switchLights()

def switchLights():

Timer(iGreenTime, switchLightsTimed).start()

def switchLightsTimed():

global sGreen

if sGreen == "north":

s1 = "north"

s2 = "east"

else:

s1 = "east"

s2 = "north"

changeLightTo(s1, "yellow")

if bOrangeBeforeGreen == True:

changeLightTo(s2, "redorange")

Timer(iLightDelay, switchLightsFinal, (s1, s2)).start()

def switchLightsFinal(s1, s2):

global sGreen

changeLightTo(s1, "red")

changeLightTo(s2, "green")

sGreen = s2

switchLights()

def changeLightTo(sLight, sColor):

turnAllOff(sLight)

if sColor == "red":

setLed(aLights[sLight][0], "on")

elif sColor == "yellow":

setLed(aLights[sLight][1], "on")

elif sColor == "green":

setLed(aLights[sLight][2], "on")

elif sColor == "redorange":

setLed(aLights[sLight][0], "on")

setLed(aLights[sLight][1], "on")

def turnAllOff(sLight):

for i in aLights[sLight]:

setLed(i, "off")

def setLed(iLed, sState):

if sState == "on":

GPIO.output(iLed, True)

else:

GPIO.output(iLed, False)

init()