import time

from pygame import mixer

import serial

import string

import pynmea2

import praytimes

from datetime import date

import math

import RPi.GPIO as GPIO

#======================Compass Libraries

#import smbus

#from time import sleep

#======================Compass Configurations

#Register\_A = 0

#Register\_B = 0x01

#Register\_mode = 0x02

#X\_axis\_H = 0x03

#Z\_axis\_H = 0x05

#Y\_axis\_H = 0x07

#declination = 4.8

#pi = 3.14159265359

#def Magnetometer\_Init():

# bus.write\_byte\_data(Device\_Address, Register\_A, 0x70)

# bus.write\_byte\_data(Device\_Address, Register\_B, 0xa0)

# bus.write\_byte\_data(Device\_Address, Register\_mode, 0)

#def read\_raw\_data(addr):

# high = bus.read\_byte\_data(Device\_Address, addr)

# low = bus.read\_byte\_data(Device\_Address, addr+1)

# value = ((high << 8) | low)

# if(value > 32768):

# value = value - 65536

# return value

#bus = smbus.SMBus(1)

#Device\_Address = 0x0d

#Magnetometer\_Init()

#GPIO.setmode(GPIO.BOARD)

#Default Coordinates

lat = 21.50

lng = 39.10

#Makkah Coordinates

mkh\_lat=21.42

mkh\_lng=39.82

#Variable Definiton

#x\_Axis=0

#y\_Axis=0

#quarter=0

#thetaRad=0

#thetaDeg=0

#thetaDirection=0

#LED Definition

#LED1=32

#LED2=36

#LED3=38

#LED4=40

#LED5=31

#LED6=33

#LED7=35

#LED8=37

#GPIO.setup(LED1,GPIO.OUT)

#GPIO.setup(LED2,GPIO.OUT)

#GPIO.setup(LED3,GPIO.OUT)

#GPIO.setup(LED4,GPIO.OUT)

#GPIO.setup(LED5,GPIO.OUT)

#GPIO.setup(LED6,GPIO.OUT)

#GPIO.setup(LED7,GPIO.OUT)

#GPIO.setup(LED8,GPIO.OUT)

while (1):

#Get GPS Coordinates

port="/dev/ttyAMA0"

ser=serial.Serial(port, baudrate=9600, timeout=0.5)

dataout = pynmea2.NMEAStreamReader()

newdata=ser.readline()

if newdata[0:6] == "$GPRMC":

newmsg=pynmea2.parse(newdata)

lat=newmsg.latitude

lng=newmsg.longitude

gps = "Latitude=" + str(format(lat,'.6f')) + " Longitude=" + str(format(lng,'.06f'))

print('Current GPS Coordinates: ' + gps)

lat=round(lat,2)

lng=round(lng,2)

if lat == 0:

lat=21.50

lng=39.10

#Calculate Athan Time

tmm = praytimes.PrayTimes().getTimes(date.today(),[lat,lng],3)

FAJR=tmm['fajr']

DHUHR=tmm['dhuhr']

ASR=tmm['asr']

MAGHRIB=tmm['maghrib']

ISHA=tmm['isha']

print("FAJR--DHUHR--ASR--MAGHRIB--ISHA")

print(FAJR+'--'+DHUHR+'--'+ASR+'--'+MAGHRIB+'--'+ISHA)

#Get System Time

tempT= time.strftime(str('%H'))

currTime= tempT

tempT= time.strftime(str('%M'))

currTime= currTime +':'+ tempT

print("Current date and time")

print(currTime)

print(time.asctime())

print(" ")

if currTime == FAJR:

print("FAJR ALARM!!!!!!!!")

mixer.init()

mixer.music.load('/home/pi/Downloads/1.mp3')

mixer.music.play()

while mixer.music.get\_busy() == True:

continue

if currTime == DHUHR:

print("DUHUR ALARM!!!!!!!!")

mixer.init()

mixer.music.load('/home/pi/Downloads/1.mp3')

mixer.music.play()

while mixer.music.get\_busy() == True:

continue

if currTime == ASR:

print("ASR ALARM!!!!!!!!")

mixer.init()

mixer.music.load('/home/pi/Downloads/1.mp3')

mixer.music.play()

while mixer.music.get\_busy() == True:

continue

if currTime == MAGHRIB:

print("MAGHRIB ALARM!!!!!!!!")

mixer.init()

mixer.music.load('/home/pi/Downloads/1.mp3')

mixer.music.play()

while mixer.music.get\_busy() == True:

continue

if currTime == ISHA:

print("ISHA ALARM!!!!!!!!")

mixer.init()

mixer.music.load('/home/pi/Downloads/1.mp3')

mixer.music.play()

while mixer.music.get\_busy() == True:

continue

# #=====================Get Compass Coordinates

# print (" Reading Heading Angle")

#

# #Read Accelerometer raw value

# x = read\_raw\_data(X\_axis\_H)

# z = read\_raw\_data(Z\_axis\_H)

# y = read\_raw\_data(Y\_axis\_H)

#

# heading = math.atan2(y, x) + declination

#

#

# #Due to declination check for >360 degree

# if(heading > 2\*pi):

# heading = heading - 2\*pi

#

# #check for sign

# if(heading < 0):

# heading = heading + 2\*pi

#

# #convert into angle

# heading\_angle = int(heading \* 180/pi)

#

# print ("Heading Angle = %dDegree" %heading\_angle)

#

#

# #Calculate our position in reference to Makkah

# x\_Axis=lng-mkh\_lng

# y\_Axis=lat-mkh\_lat

#

# if x\_Axis >= 0 and y\_Axis >=0:

# quarter =1

# if x\_Axis < 0 and y\_Axis >=0:

# quarter =2

# if x\_Axis < 0 and y\_Axis <0:

# quarter =3

# if x\_Axis >= 0 and y\_Axis <0:

# quarter =4

#

# x\_Axis= abs(x\_Axis)

# y\_Axis= abs(y\_Axis)

#

# thetaRad= math.atan((y\_Axis/x\_Axis))

# thetaDeg= math.degrees(thetaRad)

#

# if quarter ==1:

# thetaDeg=thetaDeg

# if quarter ==2:

# thetaDeg=thetaDeg+90

# if quarter ==3:

# thetaDeg=thetaDeg+180

# if quarter ==4:

# thetaDeg=thetaDeg+270

#

# thetaDirection= ((heading\_angle-thetaDeg)+360)-90

#

# if thetaDirection >=350 and thetaDirection <370:

# GPIO.output(LED1,HIGH)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=370 and thetaDirection <430:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,HIGH)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=430 and thetaDirection <470:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,HIGH)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=470 and thetaDirection <510:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,HIGH)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=510 and thetaDirection <550:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,HIGH)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=550 and thetaDirection <630:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,HIGH)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=0 and thetaDirection <305:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,HIGH)

# GPIO.output(LED8,LOW)

#

# if thetaDirection >=305 and thetaDirection <350:

# GPIO.output(LED1,LOW)

# GPIO.output(LED2,LOW)

# GPIO.output(LED3,LOW)

# GPIO.output(LED4,LOW)

# GPIO.output(LED5,LOW)

# GPIO.output(LED6,LOW)

# GPIO.output(LED7,LOW)

# GPIO.output(LED8,HIGH)