ROLL NUMBER:- 160116733079 NAME: G SAI CHANDANA

IOT LAB INTERNAL -II

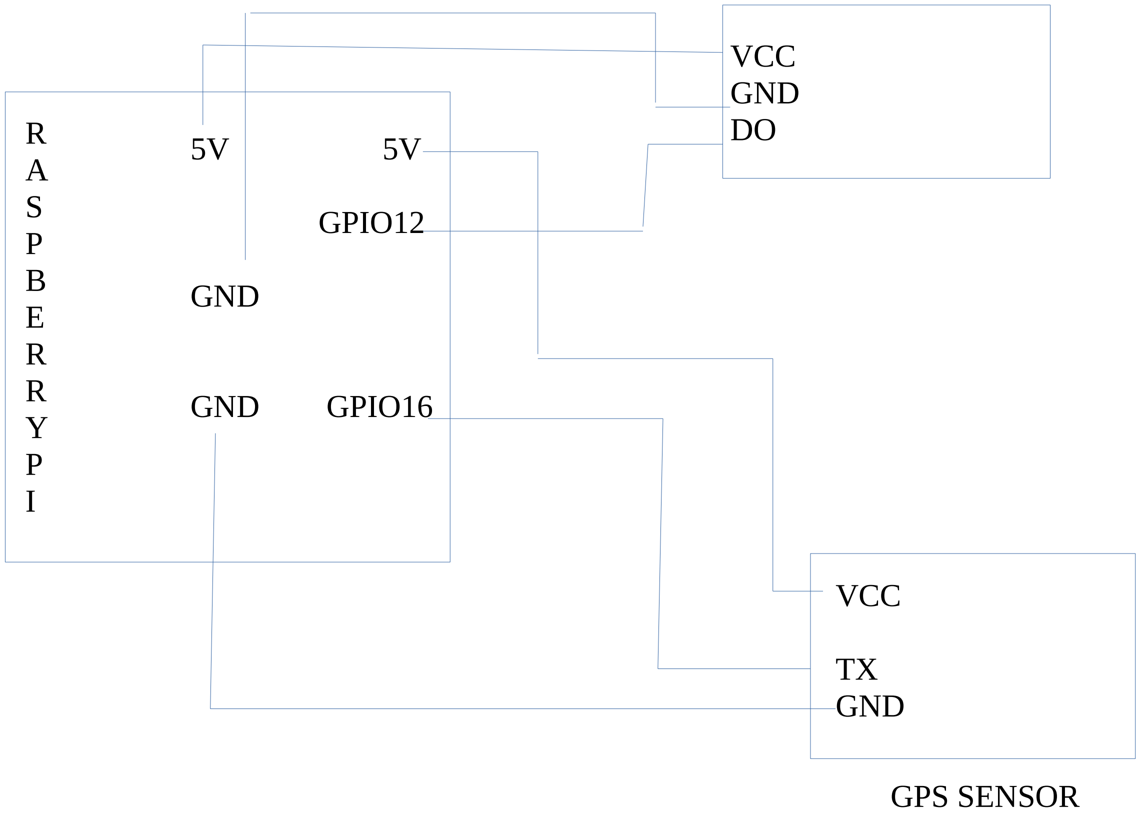
AIM : To Design IOT based Fire Alerting system to give alert message to the fire department.

DESCRIPTION : The Fire Alerting system detects the presence of flame and sends alerts to mobile when a fire occurs. Raspberry Pi is used to develop this fire alarm system . A feature of the system is the ability to remotely send an alert when a fire is detected. The system will report the event to the Fire Department using text message. Using this system we can reduce the possibility of false alert reported to the Fire Department.

HARDWARE AND SOFTWARE REQIREMENTS:

- RaspberryPi
- Thonny Python IDE
- Jumper wires
- GPS Sensor
- LED
- Flame Sensor

CIRCUIT DIAGRAM:



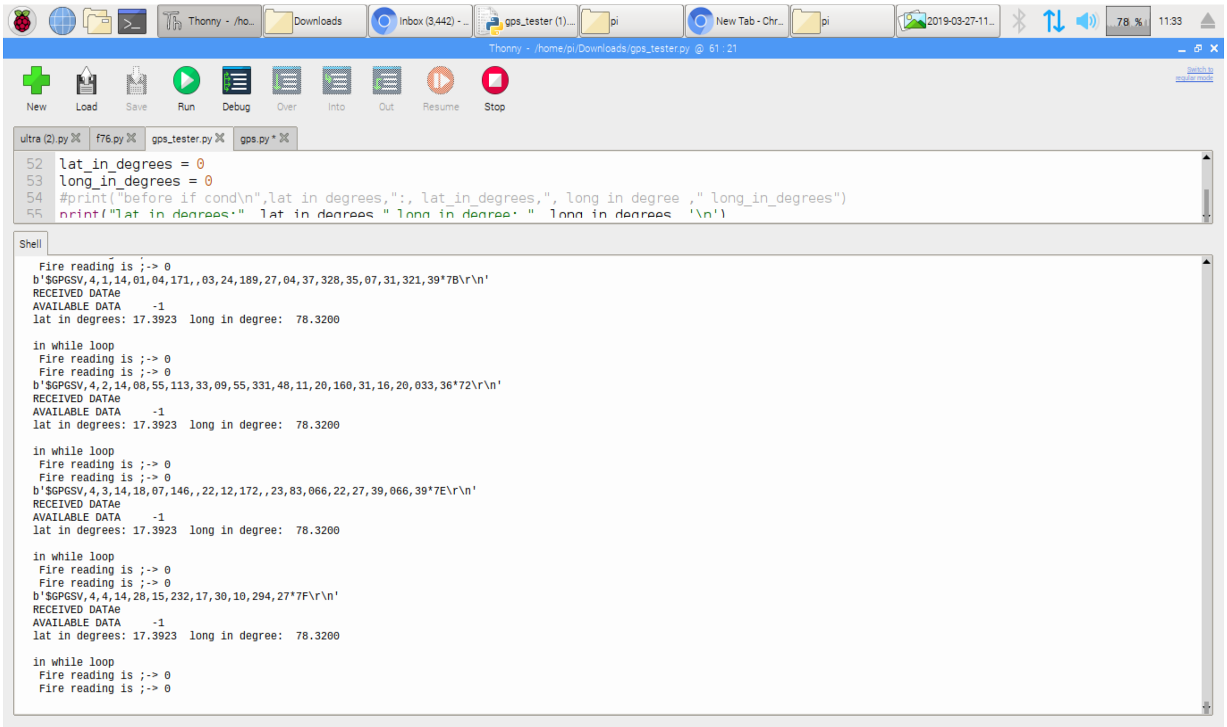
CODE:

```
#!/usr/bin/python

import serial, os, time, sys, datetime, csv
import RPi.GPIO as GPIO
from gpiozero import Buzzer, InputDevice
from time import sleep
import webbrowser
import sys
```

```
port1=2
GPIO.setmode(GPIO.BCM)
GPIO.setup(2,GPIO.IN)
GPIO.setup(23, GPIO.OUT,initial=GPIO.LOW)




def GPS_Info():
    global NMEA_buff
    global lat_in_degrees
    global long_in_degrees
    nmea_time = []
    nmea_latitude = []
    nmea_longitude = []
    nmea_time = NMEA_buff[0]                    #extract time from GPGGA string
    nmea_latitude = NMEA_buff[1]                #extract latitude from GPGGA string
    nmea_longitude = NMEA_buff[3]               #extract longitude from GPGGA string

    print("NMEA Time: ", nmea_time,'\n')
    print ("NMEA Latitude:", nmea_latitude,"NMEA Longitude:", nmea_longitude,'\n')

    lat = float(nmea_latitude)                  #convert string into float for calculation
    longi = float(nmea_longitude)               #convertr string into float for calculation

    lat_in_degrees = convert_to_degrees(lat)    #get latitude in degree decimal format
    long_in_degrees = convert_to_degrees(longi) #get longitude in degree decimal format


#convert raw NMEA string into degree decimal format
def convert_to_degrees(raw_value):
    decimal_value = raw_value/100.00
    degrees = int(decimal_value)
    mm_mmmm = (decimal_value - int(decimal_value))/0.6
    position = degrees + mm_mmmm
    position = "%.4f" %(position)
    return position



gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyS0")              #Open port with baud rate
GPGGA_buffer = 0
NMEA_buff = 0
lat_in_degrees = 0
long_in_degrees = 0
#print("before if cond\n",lat in degrees,":, lat_in_degrees,", long in degree ," long_in_degrees")
print("lat in degrees:", lat_in_degrees," long in degree: ", long_in_degrees, '\n')
try:
    while True:
        print("in while loop")
        result1=GPIO.input(2)
        print(" Fire reading is ;->",result1)
        if result1==0:
            print(" Fire reading is ;->",result1)
        received_data = (str)(ser.readline())
        sleep(2)

        print(received_data)#read NMEA string received
        print("RECEIVED DATA  ",)
        GPGGA_data_available = received_data.find(gpgga_info)
#check for NMEA GPGGA string
        print("AVAILABLE DATA    ",GPGGA_data_available)

        print("lat in degrees:", lat_in_degrees," long in degree: ",
long_in_degrees, '\n')
        if (GPGGA_data_available>=0):
            GPGGA_buffer = received_data.split("$GPGGA,",1)[1]
#store data coming after "$GPGGA," string
            NMEA_buff = (GPGGA_buffer.split(','))               #store
comma separated data in buffer
            GPS_Info()                                          #get time, latitude,
longitude

            print("latitude in degrees:", lat_in_degrees," longitude in
degrees: ", long_in_degrees, '\n')
            map_link = 'http://maps.google.com/?q=' +
str(lat_in_degrees) + ',' + str(long_in_degrees)    #create link to
plot location on Google map
            print("<<<<<<<<press ctrl+c to plot location on google
maps>>>>>>\n")               #press ctrl+c to plot on map and exit

print("--------------------------------------------------------\n")
            print("after latitude in degrees:", str(lat_in_degrees),"
longitude in degrees: ", str(long_in_degrees), '\n')
except KeyboardInterrupt:
    webbrowser.open(map_link)        #open current position
information in google map
```

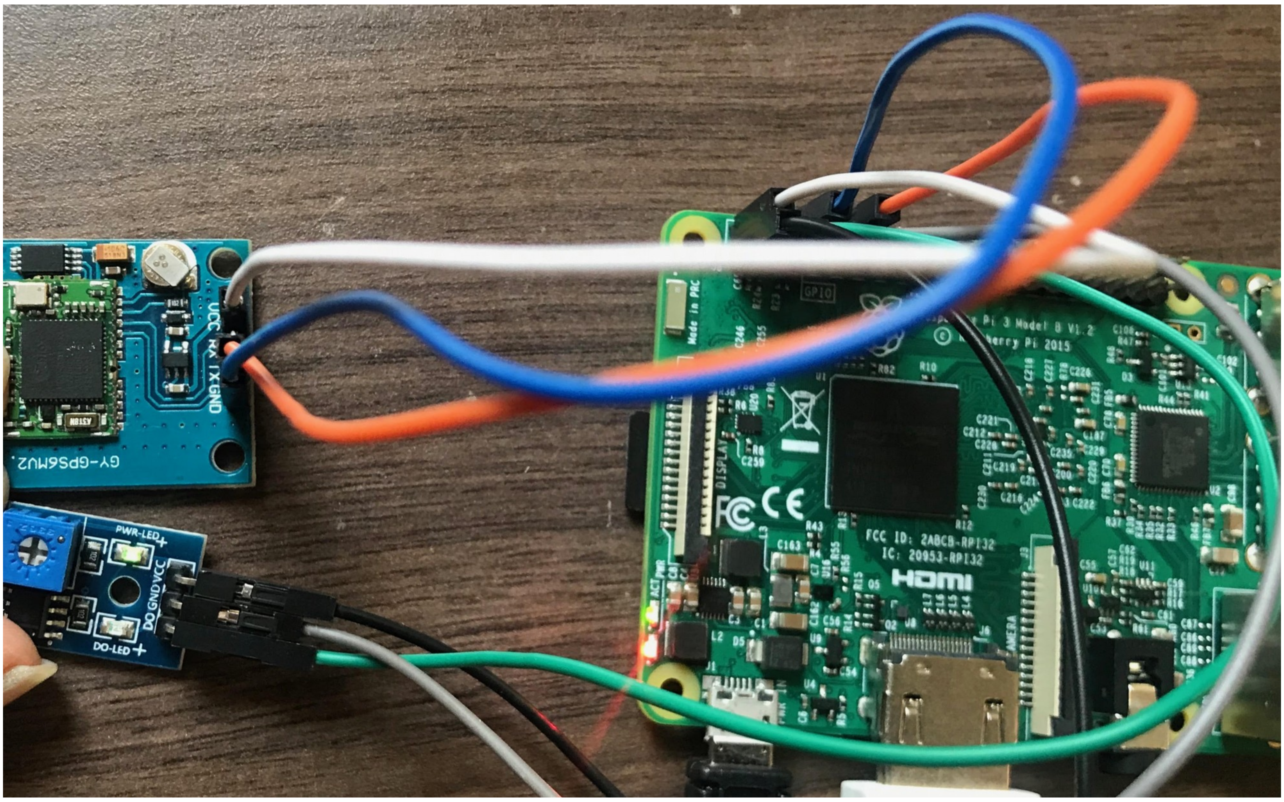
```
sys.exit(0)
```

OUTPUT:



```
lat_in_degrees = 0
long_in_degrees = 0
#print("before if cond\n",lat in degrees,":, lat_in_degrees,", long in degree ," long_in_degrees")
print("lat in degrees:", lat in degrees ," long in degree: ", long in degrees, '\n')
```

```
 Fire reading is ;-> 0
b'$GPGSV,4,1,14,01,04,171,,03,24,189,27,04,37,328,35,07,31,321,39*7B\r\n'
RECEIVED DATAe
AVAILABLE DATA     -1
lat in degrees: 17.3923  long in degree:  78.3200

in while loop
 Fire reading is ;-> 0
 Fire reading is ;-> 0
b'$GPGSV,4,2,14,08,55,113,33,09,55,331,48,11,20,160,31,16,20,033,36*72\r\n'
RECEIVED DATAe
AVAILABLE DATA     -1
lat in degrees: 17.3923  long in degree:  78.3200

in while loop
 Fire reading is ;-> 0
 Fire reading is ;-> 0
b'$GPGSV,4,3,14,18,07,146,,22,12,172,,23,83,066,22,27,39,066,39*7E\r\n'
RECEIVED DATAe
AVAILABLE DATA     -1
lat in degrees: 17.3923  long in degree:  78.3200

in while loop
 Fire reading is ;-> 0
 Fire reading is ;-> 0
b'$GPGSV,4,4,14,28,15,232,17,30,10,294,27*7F\r\n'
RECEIVED DATAe
AVAILABLE DATA     -1
lat in degrees: 17.3923  long in degree:  78.3200

in while loop
 Fire reading is ;-> 0
 Fire reading is ;-> 0
```


GY-GPS6MV2
VCC RX TX GND
PWR-LED
DO-LED
DO GND VCC
GPIO
Raspberry Pi 3 Model B V1.2
Raspberry Pi 2015
DISPLAY
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
CAMERA