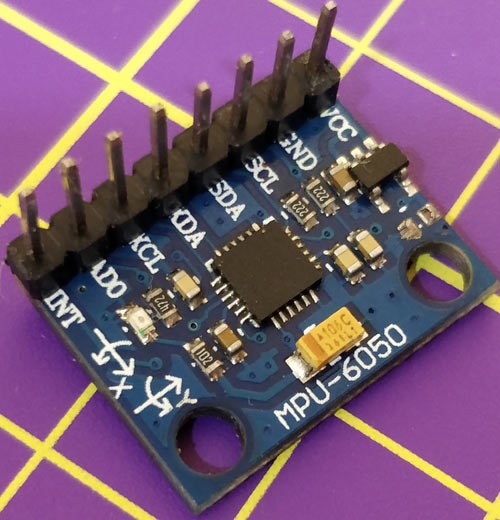
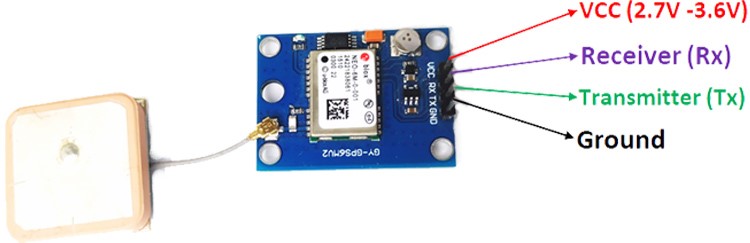
**IOT based vehicle accident notifier**

**Introduction**

 definitive and comprehensive vehicle accident and rash driving identification and alert system using NodeMCU. Unlike other projects across the internet, it has a fully functional and independent circuitry. It combines the features of Invensense’s **MPU6050, Ublox Neo-6M GPS module with NodeMCU** to a great effect. MPU6050 is a combination of the accelerometer and the gyroscope, with both the modules help the other with the data to overcome the shortcomings. Accelerometer records the acceleration across 3 axes, whereas the gyroscope records the rotational velocity across the axis. The GPS module encodes the data in the form NMEA format, which can be used to carve out the exact location of the misfortune. In addition to that, a mail is also sent to the registered mail id through an API call embedded with the google link. A service called IFTTT connects different services to help us send the mail. The email conveys the message and also **sends the location of the accident** to the user.

Accelerometer and gyroscope module Neo 6M GPS Module

Components

* NodeMCU ESP-01 module
* MPU6050 –Accelerometer and gyroscope module
* Neo 6M GPS Module
* 16\*2 LCD Display
* Bread Board
* Jumper Wires

Application

* Location — Determining a position.
* Navigation — Getting from one location to another.
* Tracking — Monitoring object or personal movement.
* Mapping — Creating maps of the world.
* Timing — Making it possible to take precise time measurements

Objective

During this activity ,you will help students to achieve following objectives

1. Understanding the principle and operation of NEO-6M GPS module.
2. Design algorithm and flowchart to get notified when accident happens
3. Programming NEO-6M GPS module using Arduino nano
4. Interfacing NEO-6M GPS module with Arduino nano

Flowchart

Program

#include <LiquidCrystal.h>  
#include <TinyGPS++.h>  
#include <Wire.h>  
#include <ESP8266WiFi.h>  
#define HOSTIFTTT "maker.ifttt.com"  
#define EVENTO "disturbance"  
#define IFTTTKEY "p7zBBT6FHzN\_7-6batwm3rg8v9T4gmrsjXSajuwaDJ1"  
TinyGPSPlus gps;  // The TinyGPS++ object  
// MPU6050 Slave Device Address  
const uint8\_t MPU6050SlaveAddress = 0x68;  
// Select SDA and SCL pins for I2C communication   
const uint8\_t scl = D1;  
const uint8\_t sda = D2;  
// sensitivity scale factor respective to full scale setting provided in datasheet   
const uint16\_t AccelScaleFactor = 16384;  
// MPU6050 few configuration register addresses  
const uint8\_t MPU6050\_REGISTER\_SMPLRT\_DIV   =  0x19;  
const uint8\_t MPU6050\_REGISTER\_USER\_CTRL    =  0x6A;  
const uint8\_t MPU6050\_REGISTER\_PWR\_MGMT\_1   =  0x6B;  
const uint8\_t MPU6050\_REGISTER\_PWR\_MGMT\_2   =  0x6C;  
const uint8\_t MPU6050\_REGISTER\_CONFIG       =  0x1A;  
const uint8\_t MPU6050\_REGISTER\_GYRO\_CONFIG  =  0x1B;  
const uint8\_t MPU6050\_REGISTER\_ACCEL\_CONFIG =  0x1C;  
const uint8\_t MPU6050\_REGISTER\_FIFO\_EN      =  0x23;  
const uint8\_t MPU6050\_REGISTER\_INT\_ENABLE   =  0x38;  
const uint8\_t MPU6050\_REGISTER\_ACCEL\_XOUT\_H =  0x3B;  
const uint8\_t MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET  = 0x68;  
int16\_t AccelX, AccelY, AccelZ, Temperature, GyroX, GyroY, GyroZ;  
int dcount=0;  
int mailcount=0;  
char temp[15];  
String location1;  
double MaxValue = 0.35;  
double MinValue = -0.35;  
int count=0;  
int gpscount=0;  
unsigned int timer = millis();  
float latitude , longitude;  
int year , month , date, hour , minute , second;  
static String date\_str , time\_str , lat\_str , lng\_str;  
int pm;  
const int RS = D0, EN = D3, d4 = D4, d5 = D5, d6 = D6, d7 = D7;     
LiquidCrystal lcd(RS, EN, d4, d5, d6, d7);  
const char\* ssid = "surya";   // Enter the namme of your WiFi Network.  
const char\* password = "123456789";  // Enter the Password of your WiFi Network.  
//char server[] = "mail.smtp2go.com";   // The SMTP Server  
//WiFiClient espClient;  
 WiFiClient client;  
void setup() {  
  Serial.begin(9600);                      
  lcd.begin(16, 2);  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("circuit digest");  
  Wire.begin(sda, scl);  
  MPU6050\_Init();  
  delay(1500);  
  lcd.clear();    
}  
void loop() {  
  double Ax, Ay, Az;  
  double xvalue,yvalue,zvalue;  
  double xvalue1,yvalue1,zvalue1;  
  double xvalue2,yvalue2,zvalue2;  
  double dx,dy,dz;  
  Read\_RawValue(MPU6050SlaveAddress, MPU6050\_REGISTER\_ACCEL\_XOUT\_H);  
    //divide each with their sensitivity scale factor  
  Ax = (double)AccelX/AccelScaleFactor;  
  Ay = (double)AccelY/AccelScaleFactor;  
  Az = (double)AccelZ/AccelScaleFactor;  
  xvalue = Ax -1.03;  
  yvalue = Ay +0.06;  
  zvalue = Az -0.07;  
if (dcount%2 ==0)  
{  
  xvalue1 = xvalue;  
  yvalue1 = yvalue;  
  zvalue1 = zvalue;  
  dcount++;  
}  
else  
{  
 xvalue2 = xvalue;  
 yvalue2 = yvalue;  
 zvalue2 = zvalue;  
 dcount++;  
 dx = xvalue2-xvalue1;  
 dy = yvalue2-yvalue1;  
 dz = zvalue2-zvalue1;  
 lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Normal Driving");  
delay(1100);  
  lcd.setCursor(0,1);  
  lcd.print(dx);  
  lcd.setCursor(6,1);  
  lcd.print(dy);  
  lcd.setCursor(11,1);  
  lcd.print(dz);  
  delay(500);  
 if(((dx < MinValue) || (dx > MaxValue)  || (dy < MinValue) || (dy > MaxValue)  || (dz < MinValue) || (dz > MaxValue)) && (timer+millis()>12000))  
 {count++;  
  if(count>10)  
  {if(mailcount<2)  
  {  
  delay(500);  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("High Disturbance");  
  wifi\_123();  
  gps\_123();   
  mail();   
  //byte ret = sendEmail();  
  }  
    mailcount++;   
 }  
}  
}  
}  
void I2C\_Write(uint8\_t deviceAddress, uint8\_t regAddress, uint8\_t data){  
  Wire.beginTransmission(deviceAddress);  
  Wire.write(regAddress);  
  Wire.write(data);  
  Wire.endTransmission();  
}  
// read all 14 register  
void Read\_RawValue(uint8\_t deviceAddress, uint8\_t regAddress){  
  Wire.beginTransmission(deviceAddress);  
  Wire.write(regAddress);  
  Wire.endTransmission();  
  Wire.requestFrom(deviceAddress, (uint8\_t)14);  
  AccelX = (((int16\_t)Wire.read()<<8) | Wire.read());  
  AccelY = (((int16\_t)Wire.read()<<8) | Wire.read());  
  AccelZ = (((int16\_t)Wire.read()<<8) | Wire.read());  
  Temperature = (((int16\_t)Wire.read()<<8) | Wire.read());  
  GyroX = (((int16\_t)Wire.read()<<8) | Wire.read());  
  GyroY = (((int16\_t)Wire.read()<<8) | Wire.read());  
  GyroZ = (((int16\_t)Wire.read()<<8) | Wire.read());  
}  
//configure MPU6050  
void MPU6050\_Init(){  
  delay(150);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SMPLRT\_DIV, 0x07);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_PWR\_MGMT\_1, 0x01);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_PWR\_MGMT\_2, 0x00);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_CONFIG, 0x00);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_GYRO\_CONFIG, 0x00);//set +/-250 degree/second full scale  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_ACCEL\_CONFIG, 0x00);// set +/- 2g full scale  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_FIFO\_EN, 0x00);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_INT\_ENABLE, 0x01);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET, 0x00);  
  I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_USER\_CTRL, 0x00);  
}  
void wifi\_123()  
{WiFi.begin(ssid, password);  
  while (WiFi.status() != WL\_CONNECTED)  
  {  
    delay(500);  
    Serial.print("\*");  
  }  
  Serial.println("");  
  Serial.println("WiFi Connected.");  
  Serial.print("IP address: ");  
  Serial.println(WiFi.localIP());  
}  
void gps\_123()  
{  
while ((Serial.available() > 0)&&(gpscount<2))  
{  
    gpscount++;  
    if (gps.encode(Serial.read()))  
    {  
      if (gps.location.isValid())  
      {  
        gpscount++;  
        latitude = gps.location.lat();  
        lat\_str = String(latitude , 6);  
        longitude = gps.location.lng();  
        lng\_str = String(longitude , 6);  
      }

      if (gps.date.isValid())  
      {  
        date\_str = "";  
        date = gps.date.day();  
        month = gps.date.month();  
        year = gps.date.year();  
        if (date < 10)  
          date\_str = '0';  
        date\_str += String(date);  
        date\_str += " / ";  
        if (month < 10)  
          date\_str += '0';  
        date\_str += String(month);  
        date\_str += " / ";  
        if (year < 10)  
          date\_str += '0';  
        date\_str += String(year);  
      }  
      if (gps.time.isValid())  
      {  
        time\_str = "";  
        hour = gps.time.hour();  
        minute = gps.time.minute();  
        second = gps.time.second();  
        minute = (minute + 30);  
        if (minute > 59)  
        {  
          minute = minute - 60;  
          hour = hour + 1;  
        }  
        hour = (hour + 5) ;  
        if (hour > 23)  
          hour = hour - 24;  
        if (hour >= 12)  
          pm = 1;  
        else  
          pm = 0;  
        hour = hour % 12;  
        if (hour < 10)  
          time\_str = '0';  
        time\_str += String(hour);  
        time\_str += " : ";  
        if (minute < 10)  
          time\_str += '0';  
        time\_str += String(minute);  
        time\_str += " : ";  
        if (second < 10)  
          time\_str += '0';  
        time\_str += String(second);  
        if (pm == 1)  
          time\_str += " PM ";  
        else  
          time\_str += " AM ";  
         Serial.print("Date= ");  
         Serial.println(date\_str);  
         Serial.print("Time= ");  
         Serial.println(time\_str);  
         Serial.print("Lat= ");  
         Serial.println(lat\_str);  
         Serial.print("Long= ");  
         Serial.println(lng\_str);   
         delay(500);  
         lcd.clear();  
         lcd.setCursor(0,0);  
         lcd.print(lat\_str);  
         lcd.setCursor(8,0);  
         lcd.print(lng\_str);           
      }  
    }  
}  
}  
void mail()  
{  
   if (client.connected())  
  {  
    client.stop();  
  }

  client.flush();  
  if (client.connect(HOSTIFTTT,80)) {  
    Serial.println("Connected");  
    // build the HTTP request  
    String toSend = "GET /trigger/";  
    toSend += EVENTO;  
    toSend += "/with/key/";  
    toSend += IFTTTKEY;  
    toSend += "?value1=";  
    toSend += lat\_str;  
    toSend += "&value2=";  
    toSend += lng\_str;  
    toSend += " HTTP/1.1\r\n";  
    toSend += "Host: ";  
    toSend += HOSTIFTTT;  
    toSend += "\r\n";  
    toSend += "Connection: close\r\n\r\n";  
    client.print(toSend);  
    delay(250);  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("Email Sent");  
  delay(1000);  
  }  
  client.flush();  
  client.stop();    
  }

Hardware

1.Connection of LCD

1. Vdd          -         5V
2. Vss          -         Ground
3. Vo            -          potentiometer output
4. RS           -         D0 pin of NodeMCU
5. E              -          D3 pin of NodeMCU
6. D4            -          D4 pin of NodeMCU
7. D5            -          D5 pin of NodeMCU
8. D6            -          D6 pin of NodeMCU
9. D7            -          D7 pin of NodeMCU
10. A              -          5V with 220ohm connected serially
11. K              -          Ground

2. connection of NEO-6M GPS module to ESP-01 module

|  |  |
| --- | --- |
| **NodeMCU** | **6M GPS Module** |
| Tx | Rx |
| Rx | Tx |
| 3.3V | Vcc |
| Gnd | Gnd |

3. connection of MPU-6050 and to ESP -01 module

|  |  |
| --- | --- |
| **NodeMCU** | **MPU6050** |
| D1 | SCL |
| D2 | SDA |
| 3.3V | Vcc |
| G | Gnd |

