|  |
| --- |
| AUTOMATIC ACCIDENT ALERT SYSTEM |
| BY TEAM GRYFFINDOR |

SAHITHI GUDI

SAMHITHA REDDY V

RAJASHRI KATTA

KAJA NIHARIKA

ABSTRACT

Everyone needs a safe and secured travelling. The advancement of technology also plays a significant role. With the improvement of the growth of traffic and thus road accident count has reached to an enormous scale. Now a days it became very difficult to know that an accident has occurred and to locate the position where it has happened. And there is no system to identify it. The main cause of the death is due to lack of immediate medical facility provided to the victim. The main intention of automatic vehicle accident is to find the accident happened at any place and intimating it to the medical rescue team, control station and family members through GPS. If the person is not in a position to control the vehicle then the accident occurs. Once the accident occurs to the vehicle, this system will send information to registered mobile number.no This System comprises of Arduino UNO, Node MCU, GPS Module, Accelerometer.

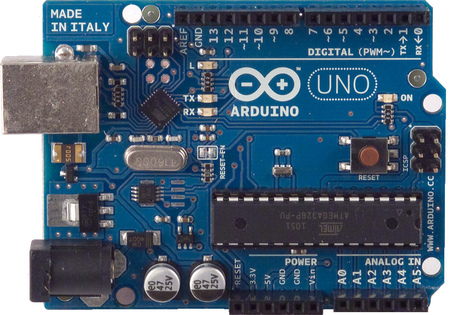
PROBLEM STATEMENT

Accidents not only bring loss to life and property but also for our economy. Transportation has evolved greatly over time. With modern technology, the automobile industry has obtained new heights with respect to speed, security, efficiency and comfort. But with this improvement in technology, there has also been an increase in the rate of accidents and sometimes these accidents even lead to death. Majorly these deaths are causing due to the delay of the proper medical aid provide to the road accident victims

HARDWARE COMPONENTS

ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.



NODE MCU

NodeMCU is an open source [LUA](https://www.lua.org/start.html) based firmware developed for ESP8266 Wi-Fi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board. Since NodeMCU is open source platform, their hardware design is open for edit/modify/build.

NodeMCU Dev Kit/board consist of ESP8266 Wi-Fi enabled chip. The **ESP8266** is a low-cost [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer [ESP8266 Wi-Fi Module](http://www.electronicwings.com/sensors-modules/esp8266-wifi-module).

NodeMCU Dev Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board.

It supports serial communication protocols i.e. UART, SPI, I2C etc.



Accelerometer ADXL335

An accelerometer is an electromechanical device that will measure acceleration force. It shows acceleration, only due to cause of gravity i.e. g force. It measures acceleration in g unit.

Accelerometer can be used for tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

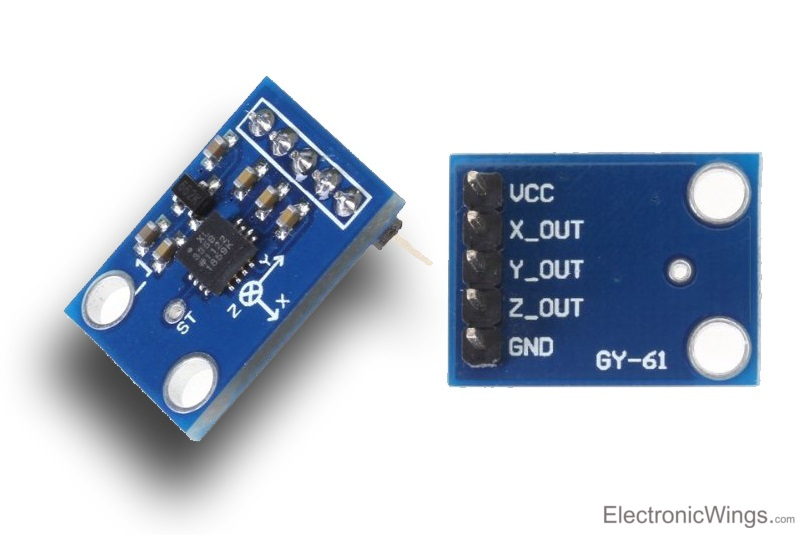
**ADXL335 module**

-  The ADXL335 gives complete 3-axis acceleration measurement.

-  This module measures acceleration within range ±3 g in the x, y and z axis.

-  The output signals of this module are analog voltages that are proportional to the acceleration.

-  It contains a polysilicon surface-micro machined sensor and signal conditioning circuitry



GPS MODULE

To find the location on the earth the whole is divided into some coordinates where the location can be easily captured by a module called GPS module. Here the GPS used is NEO-6M. This GPS module will find the location of the vehicle and the information fetched by the GPS receiver is received through the coordinates and the received data is first send to Arduino and the information is transmitted to the saved contact through GSM module. The frequency is operated in the range of 1575.42 MHz and the output of GPS module is in NMEA format which includes data like location in real time



WORKING

In this project we are using Node MCU, Arduino UNO, Accelerometer, GPS module to store the location of the vehicle and to send the immediate alert message to the nearby hospitals, police stations…etc. Here we use serial communication to send the information from Arduino to Node MCU.

The Arduino receives the latitude and longitude values of the vehicle from the gps module. The accelerometer reads the xyz axis values. These values are read by Arduino. The xyz values are compared with the condition given in the code to check whether the vehicle is met with an accident or not.

If the vehicle is met with an accident, latitude and longitude values are copied to the NodeMCU and message is sent to the nearby police stations, hospitals using msg91 application. The gps location of the vehicle is continuously stored in the IBM cloud.

ARDUINO CODE

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

TinyGPSPlus gps;

SoftwareSerial ss(5,6);

SoftwareSerial mySerial(7,8); //D2-RX, D3-TX

String value,url1;

float latitude;

float longitude;

int a=A1;

int b=A3;

void setup()

{

Serial.begin(9600);

mySerial.begin(9600);

ss.begin(9600);

}

void loop() {

int x,y,z;

y=analogRead(a);

z=analogRead(b);

Serial.println("y value");

Serial.println(y);

Serial.println(" z value");

Serial.println(z);

delay(1000);

if (y<=395||y>=400 ||z<=395||z>=400)

{

latitude=gps.location.lat();

longitude=gps.location.lng();

Serial.println(latitude);

delay(1000);

String p="http://maps.google.com/maps?&z=15&mrt=yp&t=k&q=";

String url1="$";

url1 += p;

url1 += String(latitude,DEC);

url1 += String(longitude,DEC)

url1 +="!";

mySerial.println(url1);

Serial.println(url1)

}

if (millis() > 5000 && gps.charsProcessed() < 10)

Serial.println(("No GPS data received: check wiring"));

delay(2000);

}

NODE MCU CODE

#include <SoftwareSerial.h>

#include <PubSubClient.h>

#include <ESP8266WiFi.h>

#define ORG "gabjnx"

#define DEVICE\_TYPE "niharika"

#define DEVICE\_ID "2016"

#define TOKEN "17251A0468"

float latitude ;

float longitude;

String url;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/evt/Data/fmt/json";//its the topic that is created for a mqtt protocol since it has publish and subscribe method of data access

char authMethod[] = "use-token-auth";

char token[] = TOKEN;//this is defined in pre processor commands above

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

const char\* ssid = "Wi-Fi";

const char\* password = "sahithi1234";

const char\* host = "api.msg91.com";

SoftwareSerial mySerial(D2,D7); // D2=RX, D3=TX,ARDUINO

String u;

String url2;

WiFiClient wificlient;

PubSubClient client(server, 1883,wificlient);

void setup() {

Serial.begin(9600);

mySerial.begin(9600);

Serial.println();

Serial.println("hi");

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

}

void loop() {

if (mySerial.find("$"))

{

url2=mySerial.readStringUntil('!');

Serial.println(url2);

u = "/api/sendhttp.php?route=4&sender=TESTIN&mobiles=8790438259&authkey=280146ArgtqFMG5cfb6063&message=accident occured"+url2;

Serial.print("Requesting URL: ");

Serial.println(u);

url=u;

msg();

}

else{

Serial.println("no data from arduino1");

}

delay(5000);

PublishData(latitude, longitude);

delay(2000);

}

void PublishData(float lati, float longi){

if (!!!client.connected()) {//if client is connected or not

Serial.print("Reconnecting client to ");

Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

//if client is not connected to the above credentials then the dots are printed

Serial.print(".");

delay(500);

}

Serial.println();

}

String payload = "{\"d\":{\"latitude\":";

payload += lati;

payload+="," "\"longitude\":";

payload += longi;

payload += "}}";

Serial.print("Sending payload: ");

Serial.println(payload);

if (client.publish(topic, (char\*) payload.c\_str())) {

Serial.println("Publish ok");

} else {

Serial.println("Publish failed");

}

}

void msg() {

const int httpPort = 80;

if (!wificlient.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}

Serial.print("Requesting URL: ");

Serial.println(url);

wificlient.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

delay(10);

while(wificlient.available()){

String line = wificlient.readStringUntil('\r');

Serial.print(line);

}

Serial.println();

Serial.println("closing connection");

}