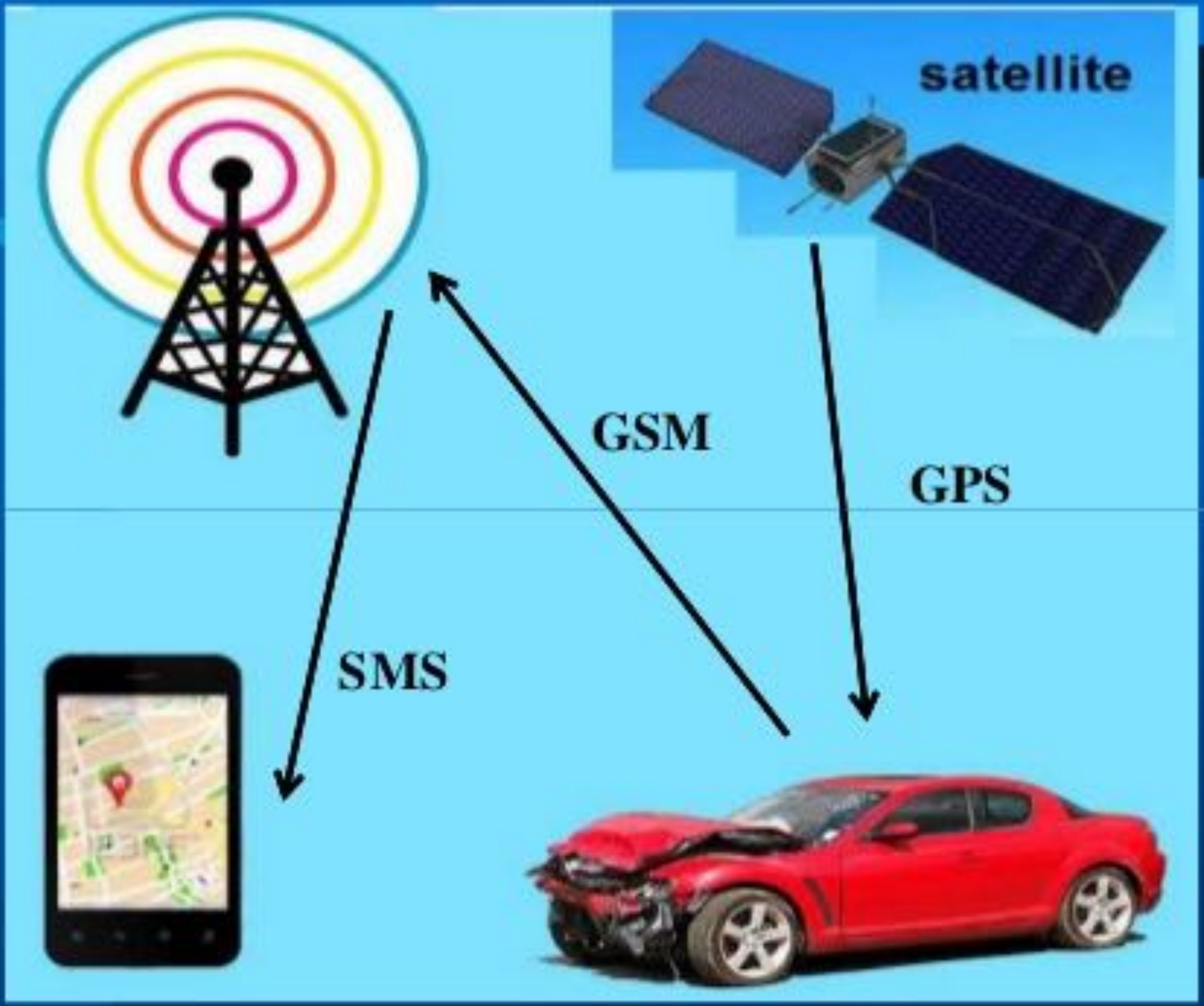
PROJECT REPORT ON :

ACCIDENT ALERTNESS IN VEHICLES

INTRODUCTION :-)

- ARDUINO based vehicle accident alertness system is made using GPS , GSM , accelerometer .
- Accelerometer detects the sudden change in the axes of vehicles.
- GSM module sends the alert message on your mobile phone with the location of the accident.
- Location of accident is sent in the form of google map link , derived from the latitude and longitude from GPS module.The message also contains the speed of vehicle in knots.


satellite
GSM
GPS
SMS

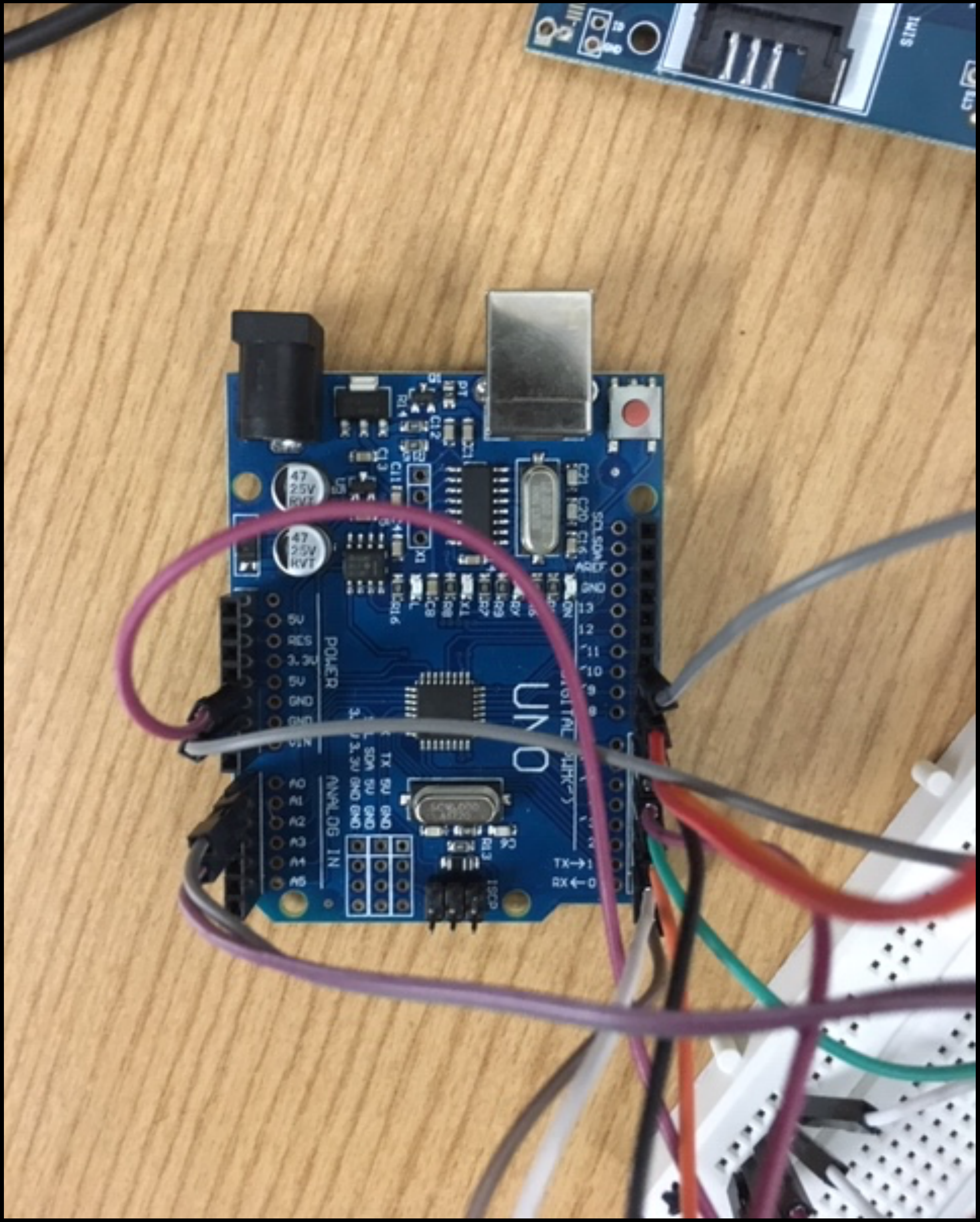
COMPONENTS REQUIRED :-)

- ARDUINO UNO and Cable
- Breadboard
- 12 V Adapters
- 16x2 LCD
- ADXL 335 Accelerometer
- M to M ; M to F ; F to F cables
- Cutter
- Soldering iron and Solder wire
- GPS + External Antenna
- Sim 800A1

ARDUINO UNO

- Arduino is a single-board micro controller meant to make the application more accessible which are interactive objects and its surroundings.

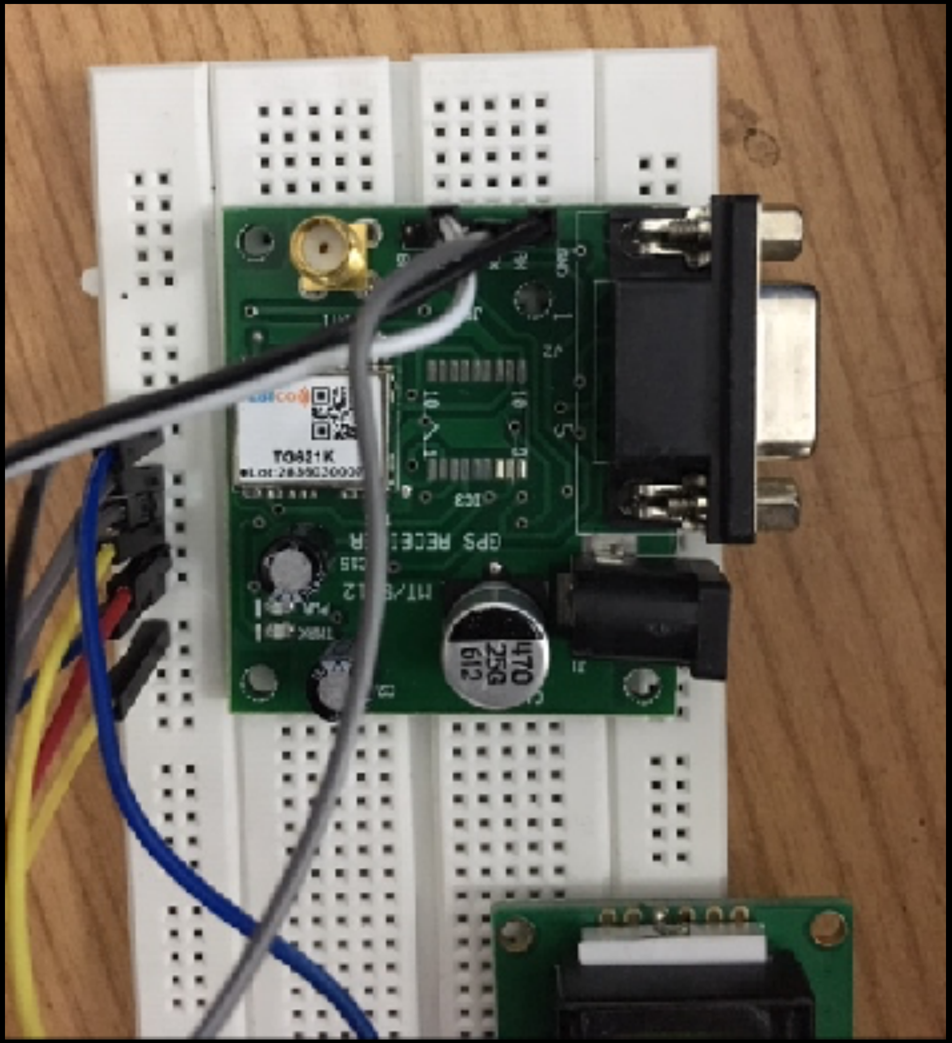
- The hardware features with an open-source hardware board designed around an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. The current models consists a USB interface, 6 analog input pins and 14 digital I/O pins that allows the user to attach various extension boards.



GPS

(Global Positioning System)

- It is used to detect latitude and longitude on the east with exact UTC time .
- GPS module is used to track the location of the accident .
- **GPS module** sends the data related to tracking position in real time, and it sends so many data in NMEA format.



GSM

(Global System for Mobile communications)

- The SIM800A Quad-Band GSM/GPRS Module with RS232 Interface is a complete Quad-band GSM/GPRS solution in an LGA(Land grid array) type which can be embedded in the customer applications.

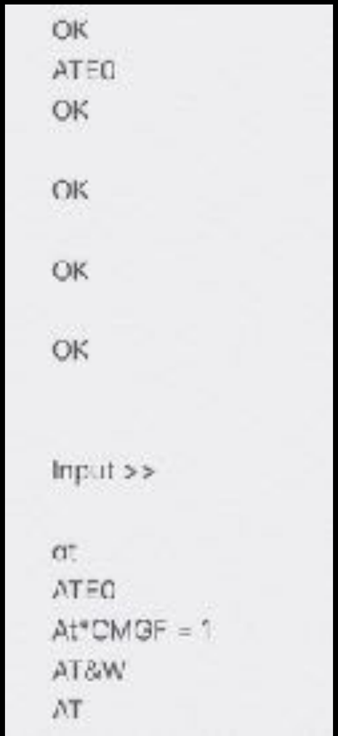
- SIM800A support Quad-band 850/900/1800/1900 MHz, it can transmit Voice, SMS and data information with low power consumption .

- The SIM800A modem has a SIM800A GSM chip and RS232 interface while enables easy connection with the computer or laptop using the USB to the Serial connector or to the micro-controller using the RS232 to TTL converter.

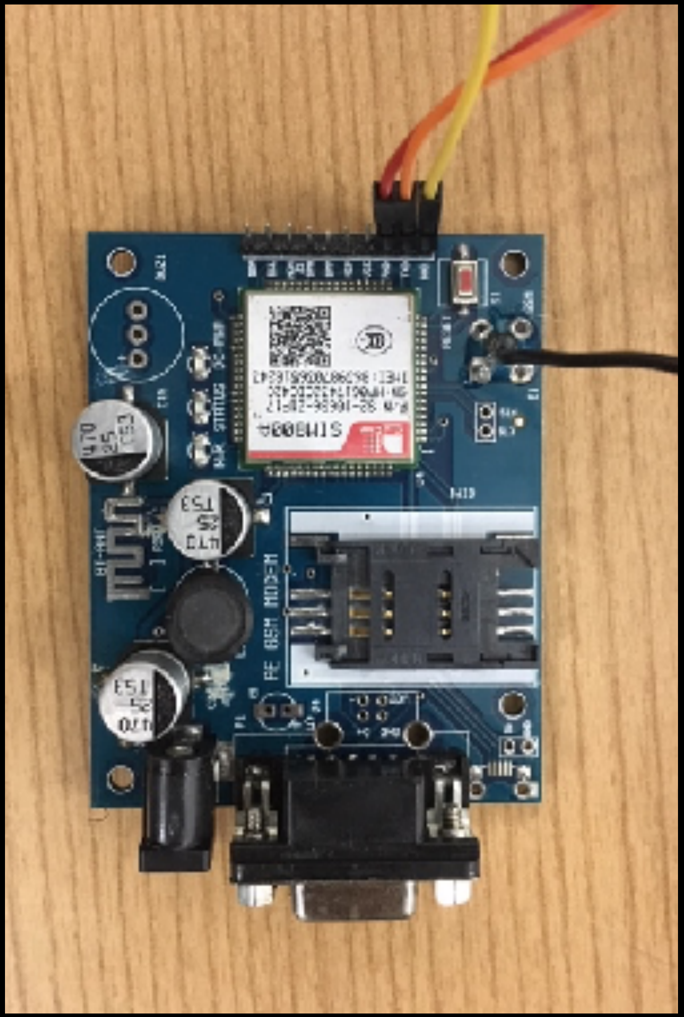
- Once you connect the SIM800A modem using the USB to RS232 connector, you need to find the correct COM port from the Device Manager of the USB to Serial Adapter.

- AT means ATTENTION. This command is used to control GSM module.

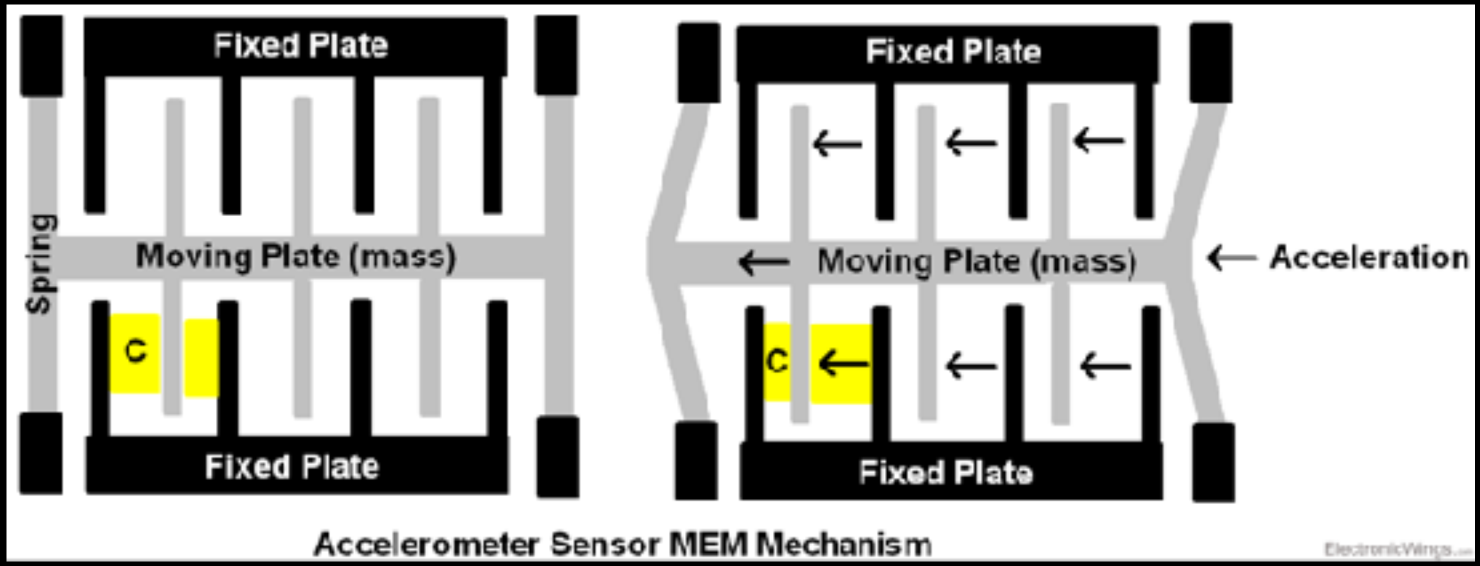
After receiving AT Command GSM Module respond with OK. It means GSM module is working fine. Below is **some AT commands** we used here in this project:



```
OK
ATE0
OK

OK

OK

OK


Input >>

at
ATE0
At*CMGF = 1
AT&W
AT
```



Accelerometer - ADXL 345



Accelerometer Sensor MEM Mechanism

- An accelerometer is an electromechanical device that will measure acceleration force.

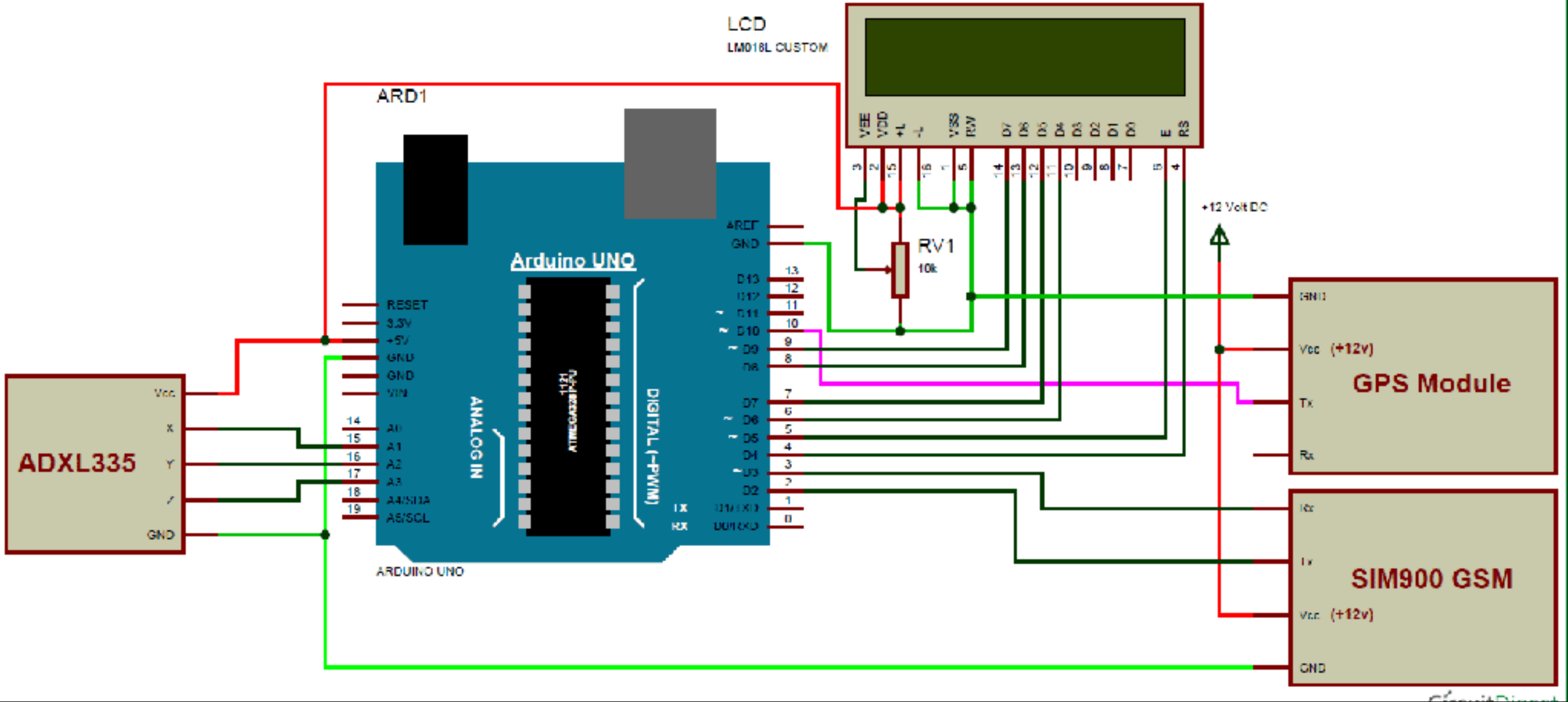
- It shows acceleration, only due to cause of gravity i.e. g force. It measures acceleration in g unit. On the earth, 1g means acceleration of 9.8 m/s2 is present. On moon, it is 1/6th of earth and on mars it is 1/3rd of earth.

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

—> Pin Description of accelerometer:

1. Vcc -5 volt supply should connect at this pin.
2. X-OUT -This pin gives an Analog output in x direction
3. Y-OUT -This pin give an Analog Output in y direction
4. Z-OUT -This pin gives an Analog Output in z direction
5. GND - Ground
6. ST -This pin used for set sensitivity of sensor.

CIRCUIT DIAGRAM :-)



CIRCUIT EXPLANATION

- Circuit Connections of this **Vehicle Accident Alert System Project** is simple. Here Tx pin of **GPS module** is directly connected to digital pin number 10 of Arduino.

- By using Software Serial Library here, we have allowed serial communication on pin 10 and 11, and made them Rx and Tx respectively and left the Rx pin of GPS Module open.

- By default , Pin 0 and 1 of Arduino are used for serial communication ; but by using the SoftwareSerial library, we can allow serial communication on other digital pins of the Arduino.

- 12 Volt supply is used to power the GPS Module. **GSM module's** Tx and Rx pins of are directly connected to pin D2 and D3 of Arduino.

- For GSM interfacing, here we have also used software serial library.

GSM module is also powered by 12v supply.

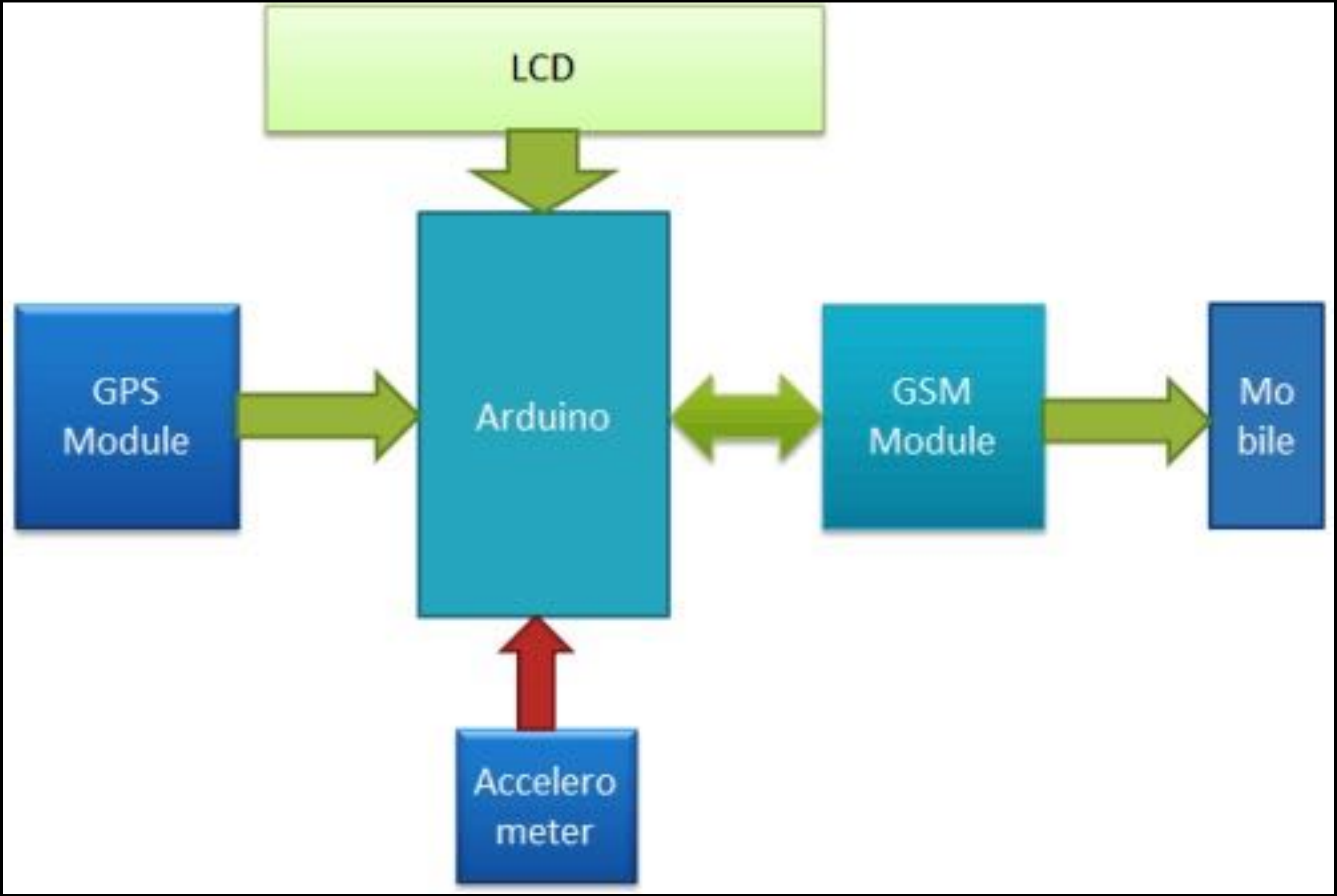
- An **optional LCD's** data pins D4, D5, D6, and D7 are connected to pin number 6, 7, 8, and 9 of Arduino. Command pin RS and EN of LCD are connected with pin number 4 and 5 of Arduino and RW pin is directly connected with ground.

- A Potentiometer is also used for setting contrast or brightness of LCD.

- An **Accelerometer** is added in this system for detecting an accident and its x, y, and z-axis .

- ADC output pins are directly connected to Arduino ADC pin A1, A2, and A3.

BLOCK DIAGRAM :-)



WORKING EXPLANATION :-)

- In this project, Arduino is used for controlling whole the process with a **GPS Receiver and GSM module**.

- GPS Receiver is used for detecting coordinates of the vehicle .

- GSM module is used for sending the alert SMS with the coordinates and the link to Google Map.

- **Accelerometer namely ADXL345** is used for detecting accident or sudden change in any axis.

- An optional 16x2 LCD is also used for displaying status messages or coordinates.

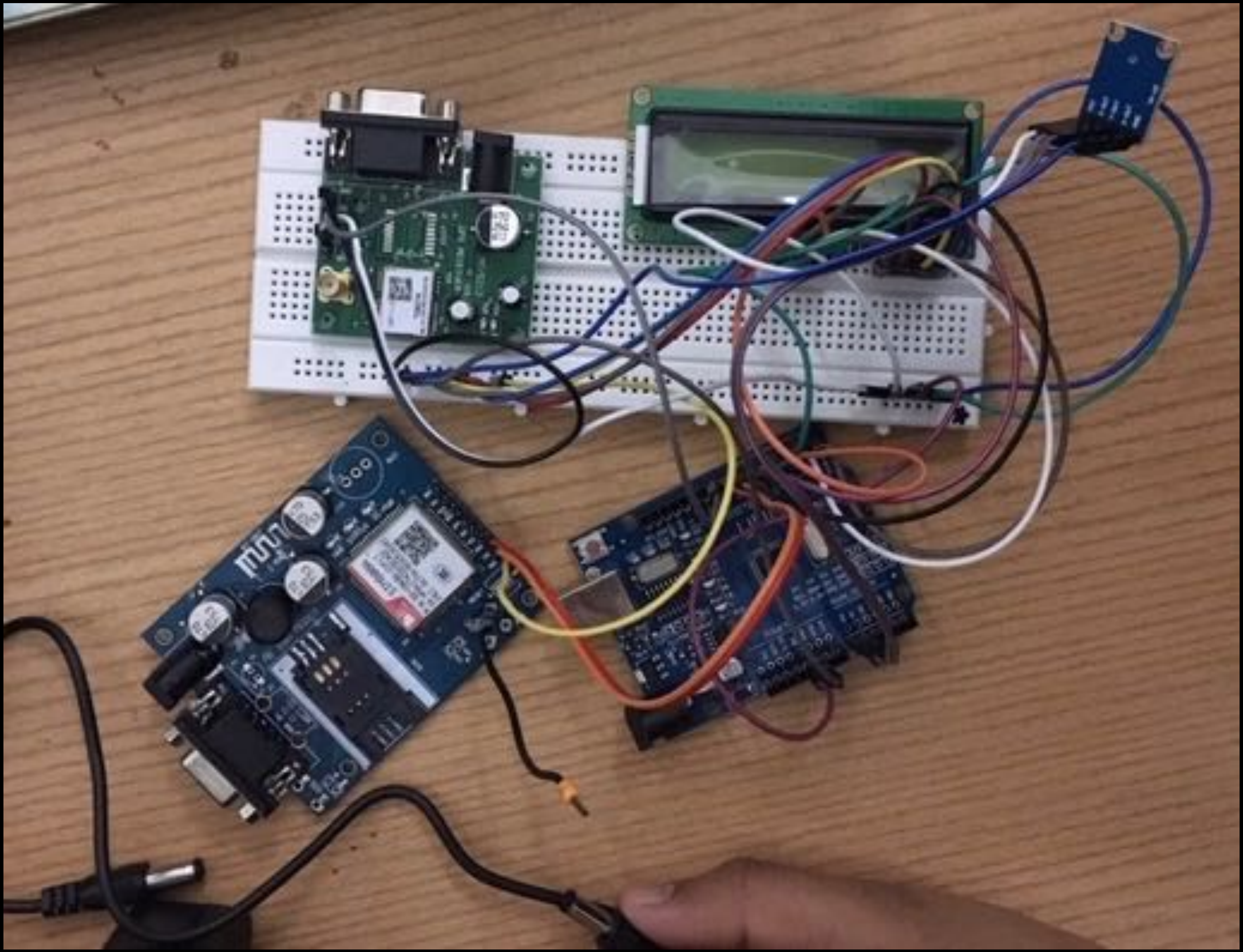
PROGRAM CODE :-)

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);

int led=13;

int dev1=8;
int dev2=9;

int temp=0,i=0,x=0,k=0;
char str[100],msg[32];

void setup()
{
  lcd.begin(16,2);
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  pinMode(dev1, OUTPUT);
  pinMode(dev2, OUTPUT);

  digitalWrite(led, HIGH);


lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Home Device  ");
  lcd.setCursor(0,1);
  lcd.print(" Control-GSM ");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("GSM Initilizing...");
  gsm_init();
  delay(1000);
  Serial.println("AT+CNMI=2,2,0,0,0");
  delay(500);
  Serial.println("AT+CMGF=1");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("GSM Initialized");
  delay(2000);
 lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("System Ready");
  digitalWrite(led, LOW);
}
```

```
void loop()
{
 for(unsigned int t=0;t<60000;t++)
 {
  serialEvent();
 if(temp==1)
 {
  x=0,k=0,temp=0;
  while(x<i)
  {
   while(str[x]=='*')
   {
    x++;
    while(str[x]!='#')
    {
     msg[k++]=str[x++];
    }
   }
   x++;
  }
  msg[k]='\0';
  lcd.clear();
  lcd.print(msg);
  delay(1000);
  temp=0;
  i=0;
  x=0;
  k=0;
  if(!strcmp(msg,”Device1on"))
{
   lcd.clear();
  lcd.print("Device 1 ON");
  digitalWrite(dev1,HIGH);
   delay(2000);
  lcd.clear();
 lcd.setCursor(0,0);
 lcd.print(" Home Device  ");
 lcd.setCursor(0,1);
 lcd.print(" Control-GSM ");
 Serial.println("AT+CMGD=1");
 delay(1000);
  }
  if(!strcmp(msg,"Device1off"))
 {
     lcd.clear();
    lcd.print("Device 1 OFF");
    digitalWrite(dev1,LOW);
      delay(2000);
    lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Home Device  ");
  lcd.setCursor(0,1);
  lcd.print(" Control-GSM ");
    Serial.println("AT+CMGD=1");
  delay(1000);
    }
    if(!strcmp(msg,"Device2on"))
  {
    lcd.clear();
   lcd.print("Device 2 ON");
    digitalWrite(dev2,HIGH);
      delay(2000);
    lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Home Device  ");
  lcd.setCursor(0,1);
  lcd.print(" Control-GSM ");
    Serial.println("AT+CMGD=1");
  delay(1000);
    }
    if(!strcmp(msg,”Device2off"))

      {
 lcd.clear();
   lcd.print("Device 2 OFF");
    digitalWrite(dev2,LOW);
      delay(2000);
      lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Home Device  ");
  lcd.setCursor(0,1);
  lcd.print(" Control-GSM ");
    Serial.println("AT+CMGD=1");
  delay(1000);
    }

  }
  }

}
void serialEvent()
```

```
{
  while(Serial.available())
  {
    char ch=(char)Serial.read();
    str[i++]=ch;
    if(ch == '*')
    {
      temp=1;
    lcd.clear();
     lcd.print("Message Received");
      delay(1000);
    }
  }
}

void gsm_init()
{
  lcd.clear();
  lcd.print("Finding Module..");
  boolean at_flag=1;
  while(at_flag)
  {
    Serial.println("AT");
    while(Serial.available()>0)
    {
      if(Serial.find("OK"))
      at_flag=0;
    }
    delay(1000);
  }

 lcd.clear();
  lcd.print("Module Connected..");
  delay(1000);
  lcd.clear();
  lcd.print("Disabling ECHO");
  boolean echo_flag=1;
  while(echo_flag)
  {
    Serial.println("ATE0");
    while(Serial.available()>0)
    {
      if(Serial.find("OK"))
      echo_flag=0;
    }
    delay(1000);
  }
  lcd.clear();
  lcd.print("Echo OFF");

  delay(1000);
  lcd.clear();
  lcd.print("Finding Network..");
  boolean net_flag=1;
  while(net_flag)
  {
    Serial.println("AT+CPIN?");
    while(Serial.available()>0)
    {
      if(Serial.find("+CPIN: READY"))
      net_flag=0;
    }
    delay(1000);
  }
  lcd.clear();
  lcd.print("Network Found..");
  delay(1000);
  lcd.clear();
}
```

Final view of the model :-)



CONCLUSION :-)

- When we are ready with our hardware and software , we can install it in our vehicle and power it up.
- Now whenever there is an accident, the vehicle gets tilt and accelerometer changes his axis values. These values are read by the Arduino and it checks if any changes are occurs in any axes.
- If any changes are occurred , then the Arduino reads coordinates by extracting $GPGGA String from GPS module data and sends SMS to the predefined number (to the police or ambulance or family member) with the location coordinates of accident place.
- The message also contains a Google Map link of the accident location, so the location can be easily tracked. When the predefined number receives the message , they need to click the link and they will be redirected to the Google map .
- The exact location of the vehicle , Speed of Vehicle [in knots (1.852 KPH)] , is also sent in the SMS and there is a display on the LCD panel.

THANK YOU !!!