ACCIDENT DETECTION SYSTEM

A PROJECT REPORT

Submitted by

DEVARSH CHANDIWADE

OMKAR PATIL

NINAD GAWADE

SHUBHAM KARNE

In Partial Fulfillment for the Award Of

DIPLOMA

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

2022

GOVERNMENT POLTECHNIC MUMBAI

CERTIFICATE

This is to certify that the project entitled "**ACCIDENT DETECTION SYSTEM**" is the bonafide that work of **"DEVARSH CHANDIWADE (FS20IF009), OMKAR PATIL (FS20IF066), NINAD GAWADE (FS20IF058), SHUBHAM KARNE (FS20IF051)"** submitted in partial fulfillment of the requirements for the award of Diploma in Information Technology of Government Polytechnic Mumbai.

HEAD OF DEPARTMENT	**PROJECT MENTOR**
Mr. Hemant Pardeshi	Mrs. Namrata A. Wankhade
PRINCIPAL	**EXTERNAL EXAMINER**
Dr. D.K.Gupta Sir	

DECLARATION

We hereby declare that the project entitled **“ACCIDENT DETECTION SYSTEM”** being submitted by us towards the partial fulfillment of the requirements for the award of Diploma in Information Technology is a project work carried by us under the supervision of Ms. Namrata A. Wankhade Mam and have not been submitted anywhere else.

We will be solely responsible if any kind of plagiarism is found.

Date:

Student Name	Enrollment No
Devarsh Chandiwade	FS20IF009
Omkar Patil	FS20IF066
Ninad Gawade	FS20IF058
Shubham Karne	FS20IF051

ACKNOWLEDGEMENT

Project is a huge team effort. My team extends our deepest gratitude and thanks to the following people to have helped us to achieve our work. I would like to thank Mr. Kunal Deokar Sir, Mrs.Deepali Gosavi Ma'am, Mrs.Namrata Wankhede Ma'am, Mrs.Pradnya Natekar Ma'am, and Mrs.Swapna Chavan Ma'am for guiding us and helping in the time of need.

My team and I extend thanks to other faculties of our college whom we have approached for the academic help with regards to our project.

DEVARSH CHANDIWADE (FS20IF009)

OMKAR PATIL (FS20IF066)

NINAD GAWADE (FS20IF058)

SHUBHAM KARNE (FS20IF051)

ABSTRACT

Road accidents rates are very high nowadays, especially two wheelers. Timely medical aid can help in saving lives. This system aims to alert the nearby medical center about the accident to provide immediate medical aid. The attached accelerometer in the vehicle senses the tilt of the vehicle and the heartbeat sensor on the user's body senses the abnormality of the heartbeat to understand the seriousness of the accident. Thus the systems will make the decision and sends the information to the smartphone, connected to the accelerometer through 'GSM' and 'GPS' modules. The Android application in the mobile phone will send text messages to the nearest medical center and friends. Application also shares the exact location of the accident and it can save time.

Keyword: Accident detection, alert system, GPS, GSM, Accelerometer, Arduino, Node MCU (esp 8266)

INDEX

- **Results**

- **Future Scope**

- **Conclusion**

INTRODUCTION

Nowadays, the rate of accidents has increased rapidly. Due to employment, the usage of vehicles like cars, bikes have increased, because of this reason the accidents can happen due to over speed. People are going under risk because of their over speed, due to unavailability of advanced techniques, the rate of accidents can’t be decreased. To reduce the accident rate in the country this paper introduces a solution. Automatic accident detection and alert systems are introduced. The main objective is to control the accidents by sending a message to the registered mobile, hospital and police station using wireless communications techniques. When an accident occurs in a city or any place, the message is sent to the registered mobile through GSM module in less time. Arduino is the heart of the system which helps in transferring the message to different devices in the system. Vibration sensor will be activated when the accident occurs and the information is transferred to the registered number through the GSM module. The GPS system will help in finding the location of the accident spot. The proposed system will check whether an accident has occurred and notify nearest medical centers and registered mobile numbers about the place of accident using GSM modules. The location can be sent through a tracking system to cover the geographical coordinates over the area. The accident can be detected by a vibration sensor which is used as a major module in the system.

LITERATURE REVIEW

In this project we will create one website where the data about the Weather will be displayed. This data can be utilized for various purposes. This would result in better efficiency in daily life events.

- This website will let you get updates about the current Weather.
- Farmers can plan their agricultural activities accordingly.
- Airlines can schedule their flights with respect to weather reports and ensure safe trip for passengers.
- Sports events can be scheduled without interruption of rains and other seasonal effects.
- Helps understand the community life according to climate change.
- Helps control Air Pollution and keep the Air Quality Index in control.

Apps Referred

Mostly we referred the in-built weather applications present in the mobile devices or other websites or web applications. All the vulnerabilities, short comings and defects we taken note of and we made our best efforts to address them all within our project. We try solving these problems with a combination of hardware and software integration. More accurate sensors and microcontrollers help bring out precise data.

SYSTEM ANALYSIS

I. Existing System

The GPS detects the latitude and longitudinal position of a vehicle. The latitudes and longitude position of the vehicle is sent as message through the GSM. The phone number is pre-saved in the EEPROM. Whenever an accident has occurred the position is detected and a message has been sent to the pre save number.

This idea proposal has been introduced at the start of the modern age of mobile phones. With the introduction of GPS sensors in the mobile, security applications based on GPS were
proposed. Then they proposed special hardware devices which can be linked with mobile
phones. Though, it had the disadvantage of actually buying extra hardware with more money.
With the massive development of mobile phones in the last decade and new sensors added
with the development, the extra hardware can be avoided. The present application of this
paper is present in a very few countries and providing the information with the relatives and
friends with the emergency services the efficiency of the application can be increased massively.

II. Proposed System

We plan on developing the entire ecosystem required to get high accuracy data about the weather and publish live updates. We plan to do so with using microcontroller, sensors, website, and a network connection. With the help of the entire chain, we aim on providing the proper information and outstand form the existing systems or our competition. Easy available portal within the LAN network to ensure data credibility and Integrity. Sole focus on accuracy, availability, integrity and understandability.

III. Objective of the Project

The main objective of the project is to make precise data about the particular area available to the users in an very user friendly manner.

- **Automation** – All the processes are managed by the backend hence reducing the stress on the user.
- **Accuracy** – Provides high accuracy due to powerful sensors and microcontroller integrated system all in one.
- **User Friendly**–Easy to understand UI which makes our system user friendly.
- **Availability**– Data is available whenever needed at just a click of the link. Just enter the IP to access the data.

SYSTEM REQUIREMENTS

I. Hardware Requirements

Microcontroller	Arduino UNO
Wireless Module	NODEMCU ESP 8266
Sensors	GPS
Power	5V (via Power Bank)
Sensor	GSM
Sensor	ADXL 335
POWER	12V(External Power supply)
Display	LCD
Variable resistor for display	10k POT
Others	Bread Board, Jumper cables etc

II. Software Requirements

Browser	Chrome, Brave, Edge, Firefox, etc.
Network	Enough bandwidth wireless connectivity
Arduino IDE	Embedded C, HTML5 & CSS3, JavaScript.
GUI	Adobe Creative Cloud, Microsoft Paint, etc.
Sms receiver	Message App
Location and Service	Google Maps

III. Feasibility Study

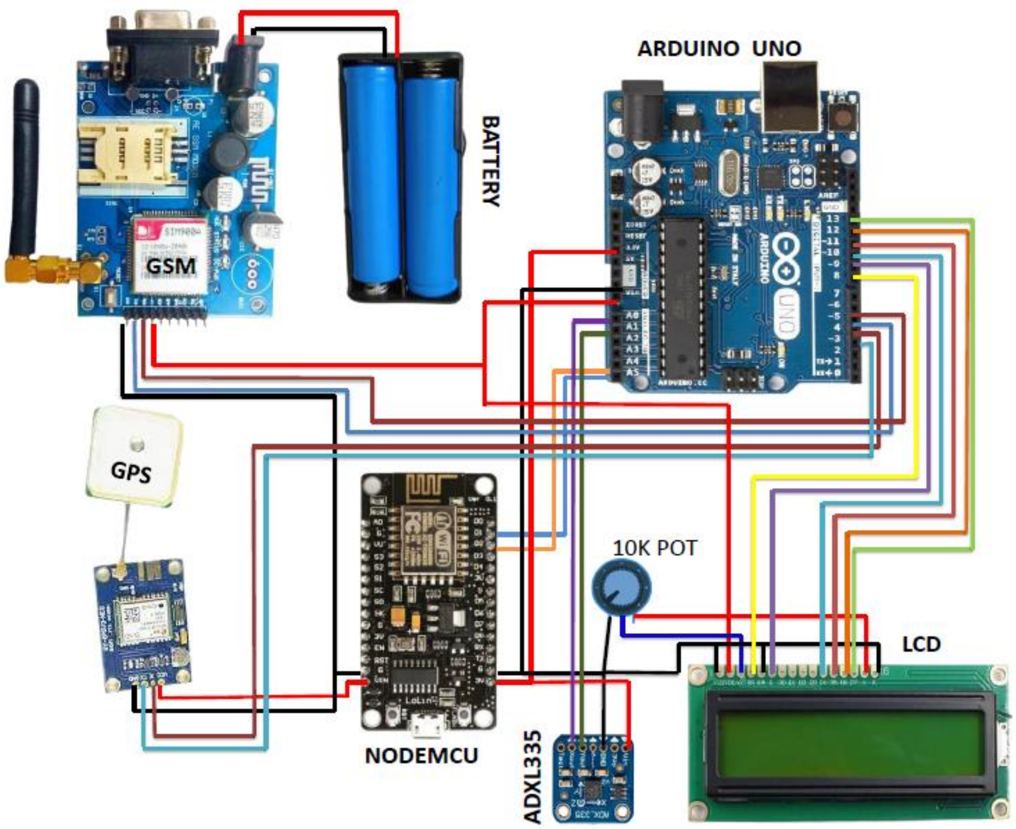
A feasibility study is an assessment of the practicality of a project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation. A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

SYSTEM DESIGN

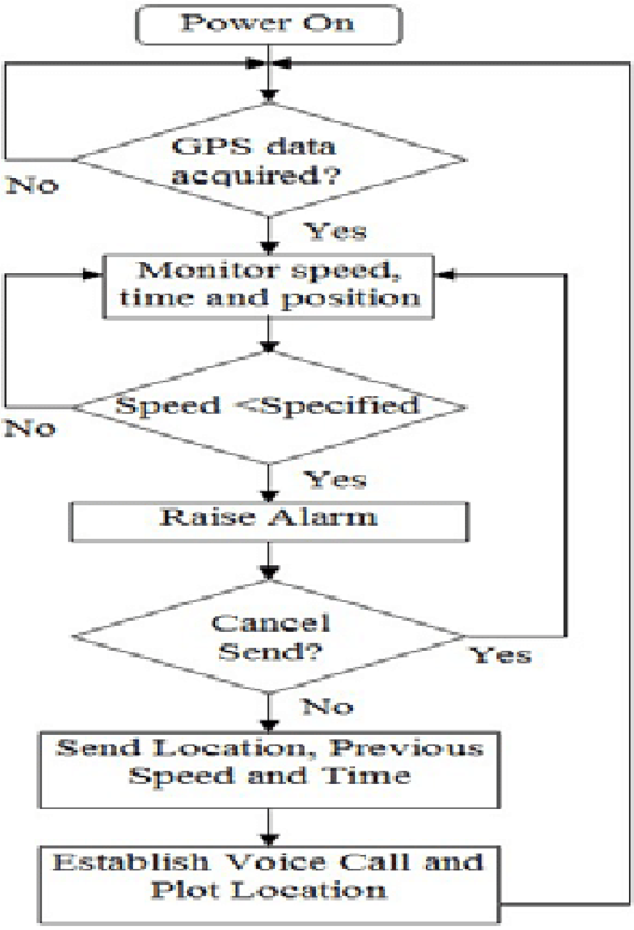
I. Use Case Diagram

Use case diagram is a graph of actors, a hard and fast of use instances enclosed by means of a device boundary, conversation associations among the actor and the use case. The use case diagram describes how a gadget interacts with out of doors actors; each use case represents a bit of functionality that a machine provides to its users. The use instances are used at some point of the evaluation phase of a task to pick out and partition system capability. They separate the device into actors and use case. Actors represent roles which might be played by using person of the system. Those users may be people, different computer systems, portions of hardware, or maybe other software structures.



II. Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system to show data inputs, outputs, storage points and routes between each destination.


Power On
GPS data acquired?
No
Yes
Monitor speed, time and position
Speed <Specified
No
Yes
Raise Alarm
Cancel Send?
Yes
No
Send Location, Previous Speed and Time
Establish Voice Call and Plot Location

TECHNOLOGY STACK

I. Front End

Message app

A messaging application is a mobile-phone-based software program that allows users to send and receive information using their phone's internet connection.

Messaging apps can transmit or receive a much wider range of data types than Short Message Service (SMS) or Multimedia Messaging Service (MMS). In addition to voice calls, video calls and text, messaging-app users can send and receive files, images, audio, location data, emojis and (in some cases) documents.

Messaging apps were primarily designed for private communication between individuals or small groups, but are increasingly being used in new ways, including:

Broadcast or bulk messaging. The capacity to send messages or other content to a large number of people.

Encryption. End-to-end encryption means that the content of a message can be viewed only by the people sending and receiving messages. It cannot be decrypted and read by the company itself.

Bots or "chatbots". A piece of code that performs automated functions within an app (often in natural language like a human − hence "chatbot"), such as replying to users' questions with short pieces of information.

By using the message app the information which is send by the GSM Module (Vehicle).

As we know that the main moto of a software is completely dependent on the frontend of the entire project

This was one of the biggest advantage of our project that today every person is active on the message. So due to this reason the chances that the person will visit the message is very high.

Microsoft Paint

Microsoft Paint is a simple raster graphics editor that has been included with all versions of Microsoft Windows. The program opens and saves files in Windows bitmap (BMP), JPEG, GIF, PNG, and single-page TIFF formats. The program can be in color mode or two-color black-and-white, but there is no gray scale mode. For its simplicity and that it is included with Windows, it rapidly became one of the most used applications in the early versions of Windows, introducing many to painting on a computer for the first time. It is still widely used for simple image manipulation tasks.

In July 2017, Microsoft added Paint to the list of deprecated features of Windows 10 and announced that it would become a free standalone application in Microsoft Store.

II. Back End

Embedded C

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems.

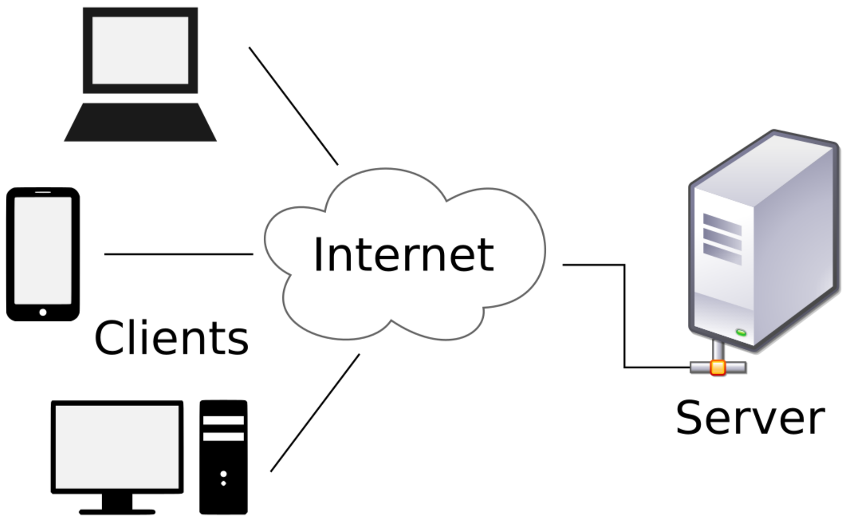
Embedded C programming typically requires nonstandard extensions to the C language in order to support enhanced microprocessor features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. The C Standards Committee produced a Technical Report, most recently revised in 2008 and reviewed in 2013, providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as fixed-point arithmetic, named address spaces and basic I/O hardware addressing. Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

III. Network

Client

In computing, a client is a piece of computer hardware or software that accesses a service made available by a server as part of the client–server model of computer networks. The server is often (but not always) on another computer system, in which case the client accesses the service by way of a network.

A client is a computer or a program that, as part of its operation, relies on sending a request to another program or a computer hardware or software that accesses a service made available by a server (which may or may not be located on another computer). For example, web browsers are clients that connect to web servers and retrieve web pages for display. Email clients retrieve email from mail servers. Online chat uses a variety of clients, which vary on the chat protocol being used. Multiplayer video games or online video games may run as a client on each computer. The term "client" may also be applied to computers or devices that run the client software or users that use the client software.



Server

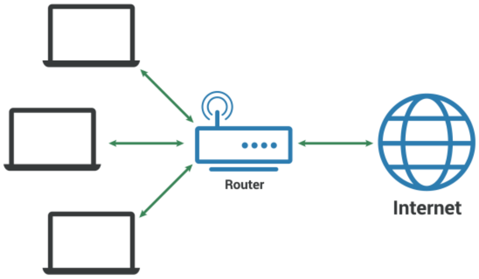
In computing, a server is a piece of computer hardware or software (computer program) that provides functionality for other programs or devices, called "clients". This architecture is called the client–server model. Servers can provide various functionalities, often called "services", such as sharing data or resources among multiple clients, or performing computation for a client. A single server can serve multiple clients, and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device. Typical servers are database servers, file servers, mail servers, print servers, web servers, game servers, and application servers.

LAN

A local area network (LAN) is a computer network that interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building. By contrast, a wide area network (WAN) not only covers a larger geographic distance, but also generally involves leased telecommunication circuits.

Ethernet and Wi-Fi are the two most common technologies in use for local area networks. Historical network technologies include ARCNET, Token Ring and AppleTalk.

In a wireless LAN, users have unrestricted movement within the coverage area. Wireless networks have become popular in residences and small businesses, because of their ease of installation. Most wireless LANs use Wi-Fi as wireless adapters are typically integrated into smartphones, tablet computers and laptops. Guests are often offered Internet access via a hotspot service.



IP Address

An Internet Protocol address (IP address) is a numerical label such as *192.0.2.1* that is connected to a computer network that uses the Internet Protocol for communication. An IP address serves two main functions: network interface identification and location addressing.

Internet Protocol version 4 (IPv4) defines an IP address as a 32-bit number. However, because of the growth of the Internet and the depletion of available IPv4 addresses, a new version of IP (IPv6), using 128 bits for the IP address, was standardized in 1998. IPv6 deployment has been ongoing since the mid-2000s.

IP addresses are written and displayed in human-readable notations, such as *192.0.2.1* in IPv4, and *2001:db8:0:1234:0:567:8:1* in IPv6. The size of the routing prefix of the address is designated in CIDR notation by suffixing the address with the number of significant bits, e.g., *192.0.2.1/24*, which is equivalent to the historically used subnet mask *255.255.255.0*.

GSM MODEM

GSM stands for Global System for Mobile Communication. GSM is an open and digital cellular technology used for mobile communication. It uses 4 different frequency bands of 850 MHz, 900 MHz, 1800 MHz and 1900 MHz . It uses the combination of FDMA and TDMA. This article includes all the concepts of GSM architecture and how it works.

GSM is having 4 different sizes of cells are used in GSM :

Macro : In this size of cell, Base Station antenna is installed.

Micro : In this size of cell, antenna height is less than the average roof level.

Pico : Small cells' diameter of few meters.

Umbrella : It covers the shadowed (Fill the gaps between cells) regions.

Features of GSM are :

Supports international roaming

Clear voice clarity

Ability to support multiple handheld devices.

Spectral / frequency efficiency

Low powered handheld devices.

Ease of accessing network

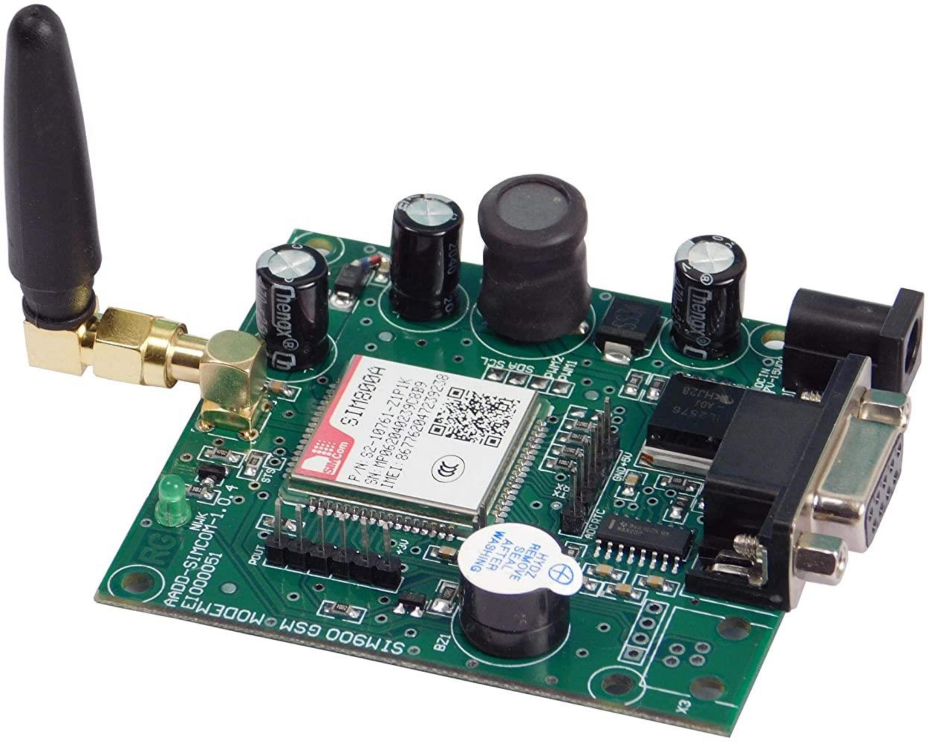
International ISDN compatibility.

GSM is nothing but a larger system which is divided into further 3 subsystems.

Base Station Subsystem (BSS.)

Network Switching Subsystem (NSS)

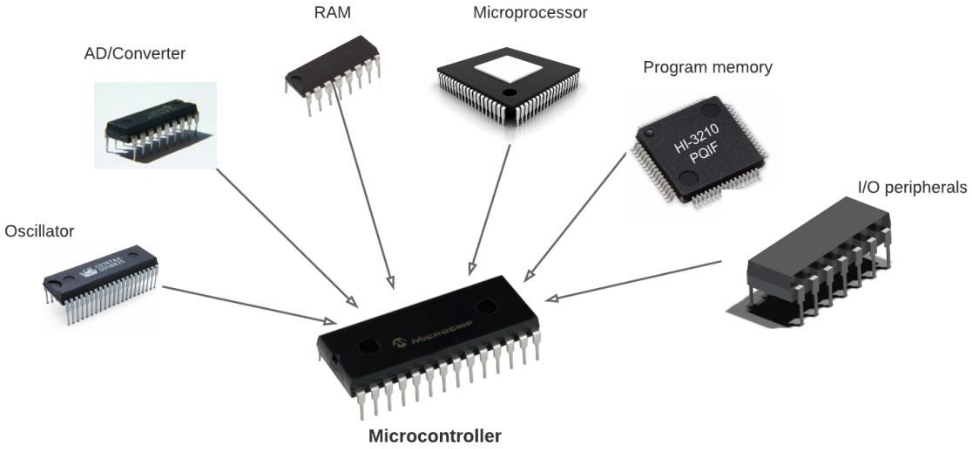
Network Management Subsystem (NMS)


SIM900A
SIM900 GSM MODEM

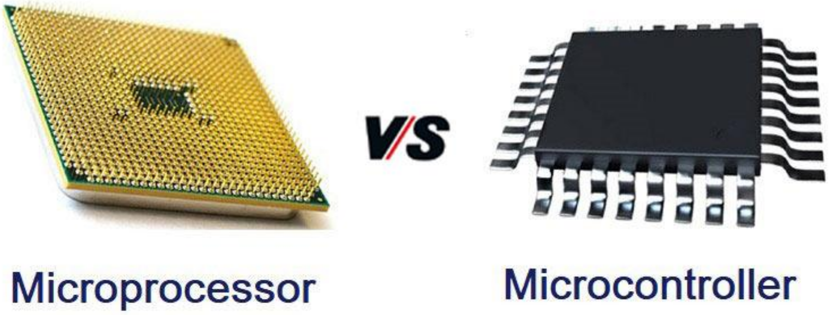
Microcontroller

A microcontroller (MCU for *microcontroller unit*) is a small computer on a single VLSI integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

In modern terminology, a microcontroller is similar to, but less sophisticated than, a system on a chip (SoC). An SoC may connect the external microcontroller chips as the motherboard components, but an SoC usually integrates the advanced peripherals like graphics processing unit (GPU) and Wi-Fi interface controller as its internal microcontroller unit circuits.



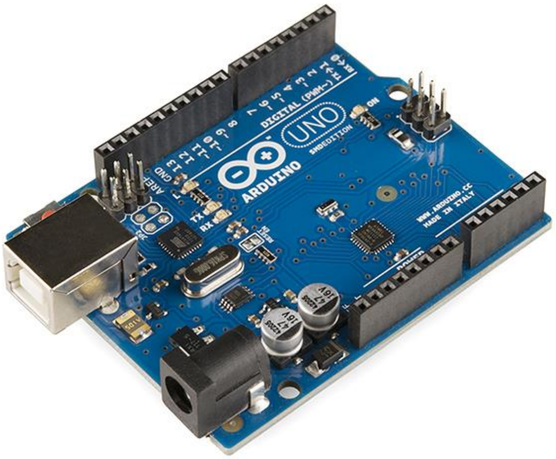
Microprocessor VS Microcontroller



Microprocessor	**Microcontroller**
Heart of the system.	Heart of the embedded system.
Externally connected with input-output components.	input-output components are embedded.
The circuit may be large depending upon usage.	The circuit is very small.
Not cost-effective.	Cost-effective.
The total consumption of power is high.	Total consumption of power is less.
Power saving mode is not generally available.	Power saving mode is generally offered.
Used in PC.	Used in MP3 players, washing machines, etc.
Memories like RAM and ROM are absent.	Carries RAM, ROM, etc.
Runs at a very high speed.	Runs at a relatively lower speed.
It is complex and costly.	Simple and cheap.
Example: DEC Alpha 21164, IBM RS6000, etc	Example: Intel 8031/8051, PIC1x, etc.

Arduino

Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its hardware products are licensed under a CC BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.



NODEMCU ESP 8266

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added. NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). Strictly speaking, the term "NodeMCU" refers to the firmware rather than the associated development kits.

Both the firmware and prototyping board designs are open source. NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub.

GPS Module

The Global Positioning System (GPS) is a satellite based navigation system that provides location and time information. The system is freely accessible to anyone with a GPS receiver and unobstructed line of sight to at least four of GPS satellites. A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites. GPS is nowadays widely used and also has become an integral part of smart phones.

The GTPA010 module is easy to use, having RS232 as well as USB interface. It operates over 3.2 to 5V supply range thus enabling interfacing with microcontrollers with 3.3V as well as 5V. The module outputs GPS data in NMEA0183 format. Each of message string starts with ‘$’ and then the message identifier. Each parameter is separated using a comma so that the message can be parse with the help of the commas.

Features

MediaTek MT3329 Chipset, L1 Frequency, C/A code, 66 Channels

3m position accuracy

Jammer detection and reduction

Data output Baud rate: 9600 bps(Default)

Low Power Consumption: 55mA @ acquisition, 40mA @ tracking

High Sensitivity, -165 dBm, TCXO Design , superior urban performances

Patch antenna

High sensitivity

DGPS(WAAS/EGNOS/MSAS/GAGAN) support

One of the global positioning system (GPS) devices ultilizes data from satellites to locate a specific point on the Earth in a process named trilateration. Meanwhile, a GPS receiver measures the distances to satellites using radio signals to trilaterate. And trilateration is similar to triangulation, which measures angles, depicted in this illustration (Tim Gunther, 2020). GPS modules contain tiny processors and antennas that directly receive data sent by satellites through dedicated RF frequencies. From there, it'll receive timestamp from each visible satellites, along with other pieces of data. If the module's antenna can spot 4 or more satellites, it's able to accurately calculate its position and time.

ADXL 335

Accelerometer Sensor

Accelerometers are widely used in low-power, low-cost motion and tilt sensing applications such as mobile devices, gaming systems, disk drive protection, image stabilization, and sports and health devices.

cube is in outer space, where everything is weightless, the ball will simply float in the center of the cube.

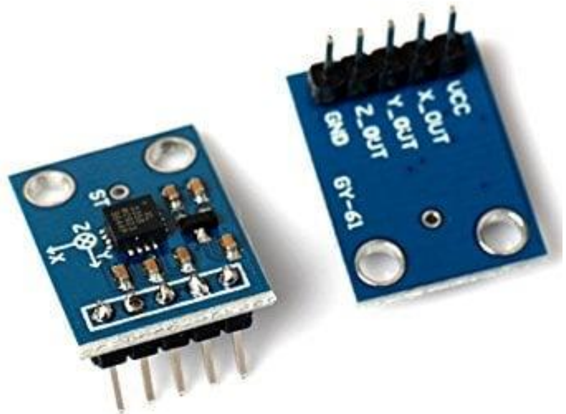
Now assume that each wall represents a specific axis.

If we suddenly move the box to the left with acceleration 1g (a single G-force 1g is equivalent to gravitational acceleration 9.8 m/s2), the ball will undoubtedly hit the wall X. If we measure the force the ball exerts on wall X, we can obtain an output value of 1g along the X axis.

In this case, the box isn't moving, but we still get a 1g reading on the Z axis. This is because gravity (which is actually a form of acceleration) is pulling the ball downward with a force of 1g.

While this model does not exactly represent how a real-world accelerometer sensor is built, it is often useful in understanding why an accelerometer's output signal is typically specified in ±g, or why an accelerometer reads 1g in the z-axis at rest, or what accelerometer readings you can expect at different orientations.

In the real world, accelerometers are based on Micro-Electro-Mechanical Systems (MEMS fabrication technology). So, let's find out how a MEMS accelerometer works.

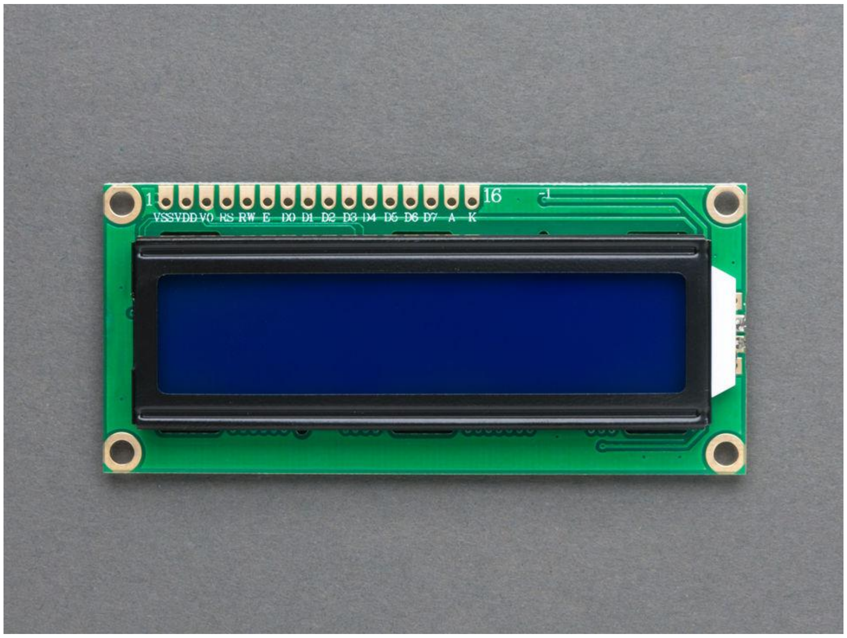


Display For Output (16*2)

Use the 16x2 standard alphanumeric LCD display, they are extremely common and is a fast way to have your project show status messages.

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

Command register stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. In your arduino project Liquid Crystal Library simplifies this for you so you don't need to know the low-level instructions. Contrast of the display can be adjusted by adjusting the potentiometer to be connected across VEE pin.

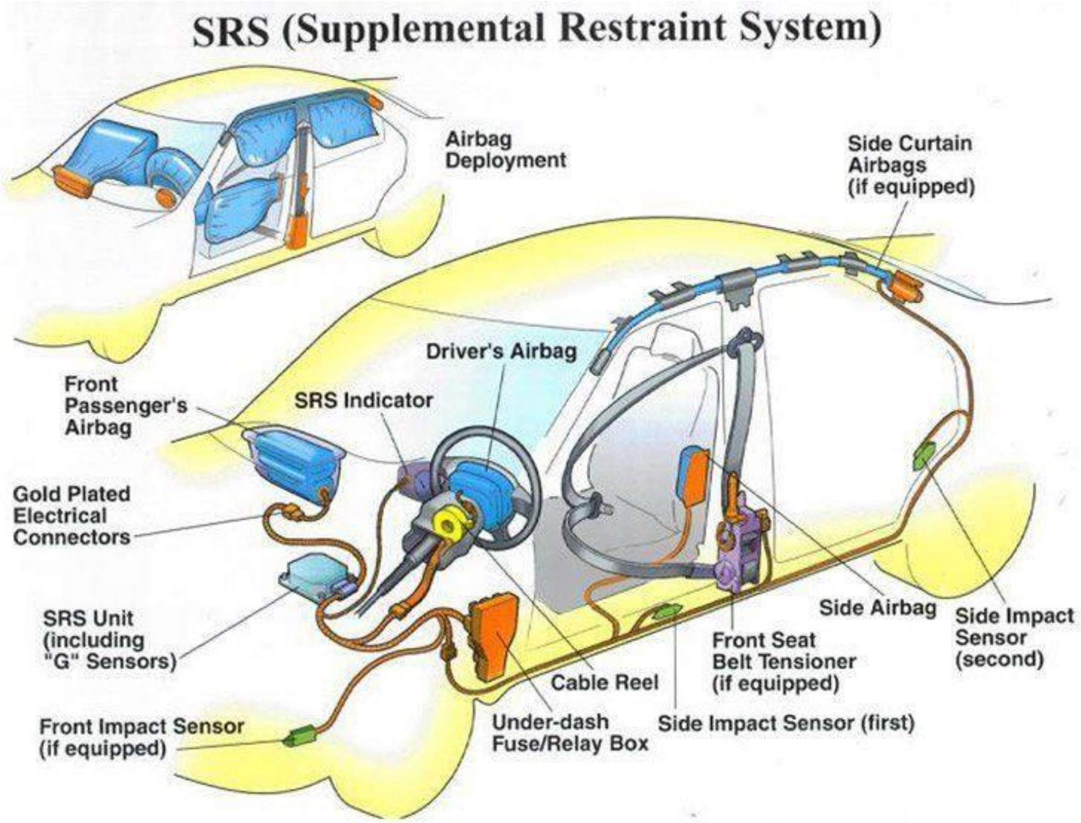


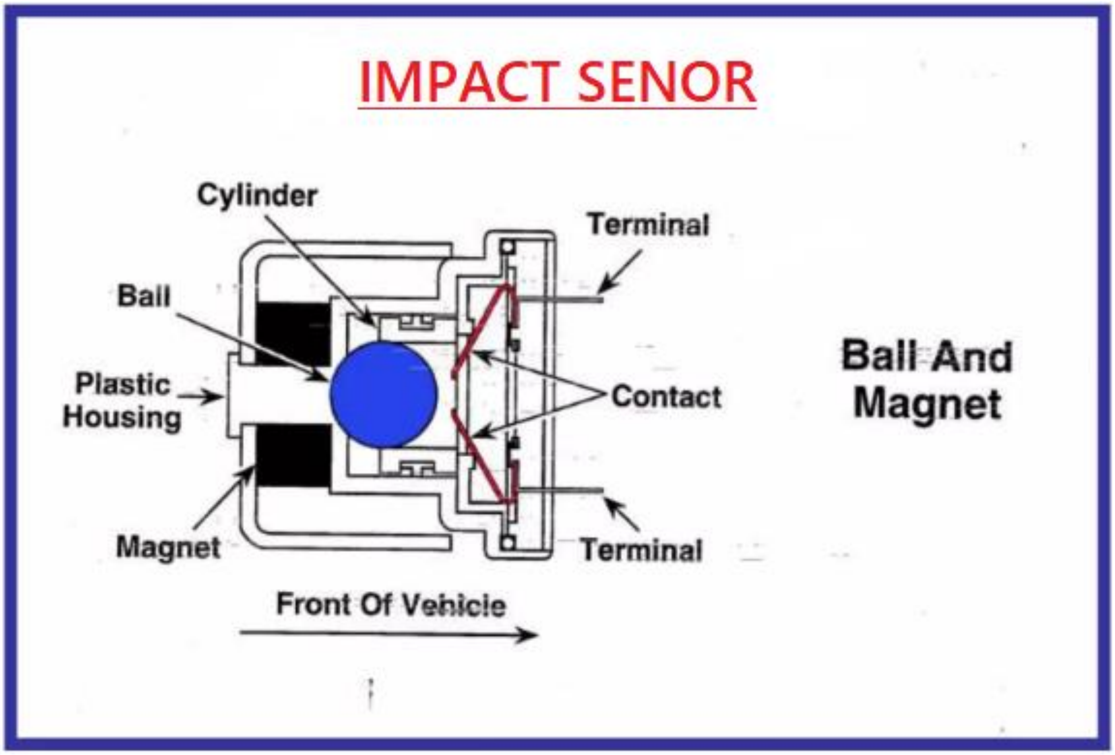
SYSTEM IMPLEMENTATION

IN REAL LIFE-

Let’s image the scenario of the accident in car . At this situation the car airbags generally activates when the front impact sensor, side impact sensor, back impact sensor receives any accident, it send the positive signal to the SRS Unit .

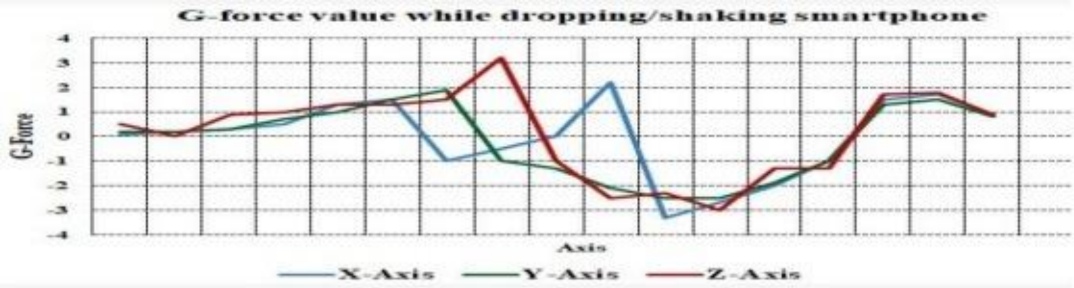
Here’s the tricky point.

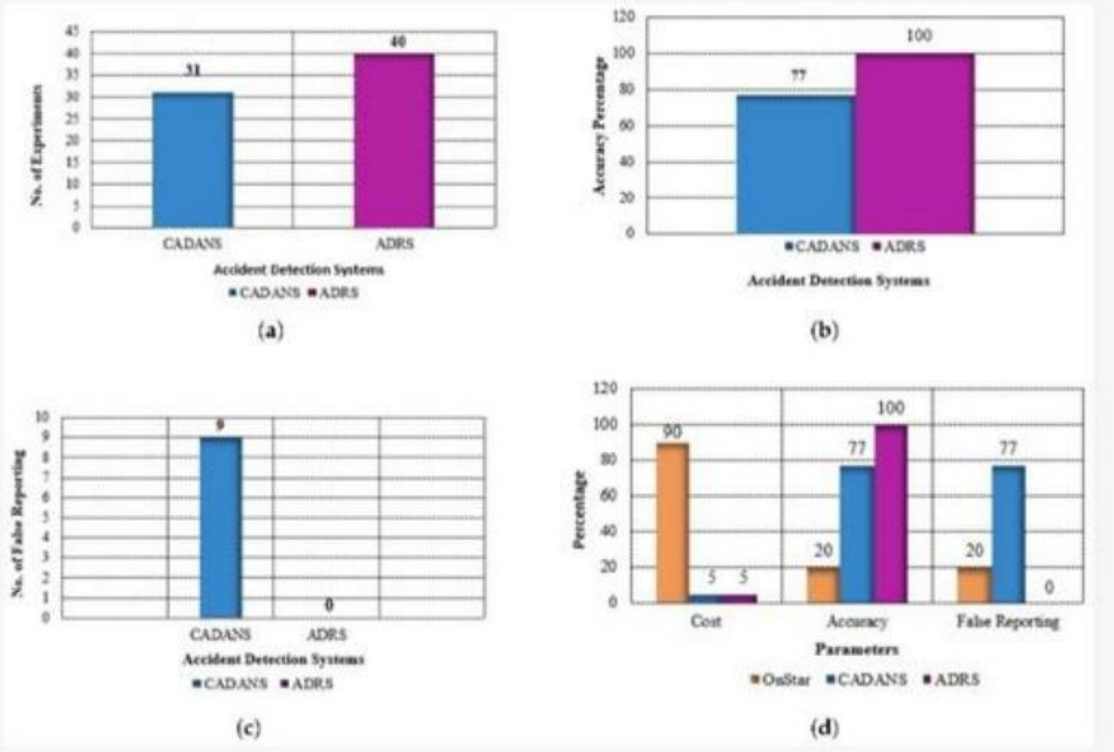

SRS (Supplemental Restraint System)
Airbag Deployment
Side Curtain Airbags (if equipped)
Driver's Airbag
Front Passenger's Airbag
SRS Indicator
Gold Plated Electrical Connectors
SRS Unit (including "G" Sensors)
Front Impact Sensor (if equipped)
Under-dash Fuse/Relay Box
Cable Reel
Side Impact Sensor (first)
Front Seat Belt Tensioner (if equipped)
Side Airbag
Side Impact Sensor (second)



If any of the positive signal is sensed by the any of the impact sensor . the SRS system open's the airbag and using the same reading the Arduino detects the Accidents . After which it sends that message to the SOS Contact Number . This is the working of Our prototype in the real Life

As to demonstrate it here we have used the Adxl sensor the adxl sensor works as the impact detection system. Just in some way . But due to this our job for the prototype is done. This is the working model of our system



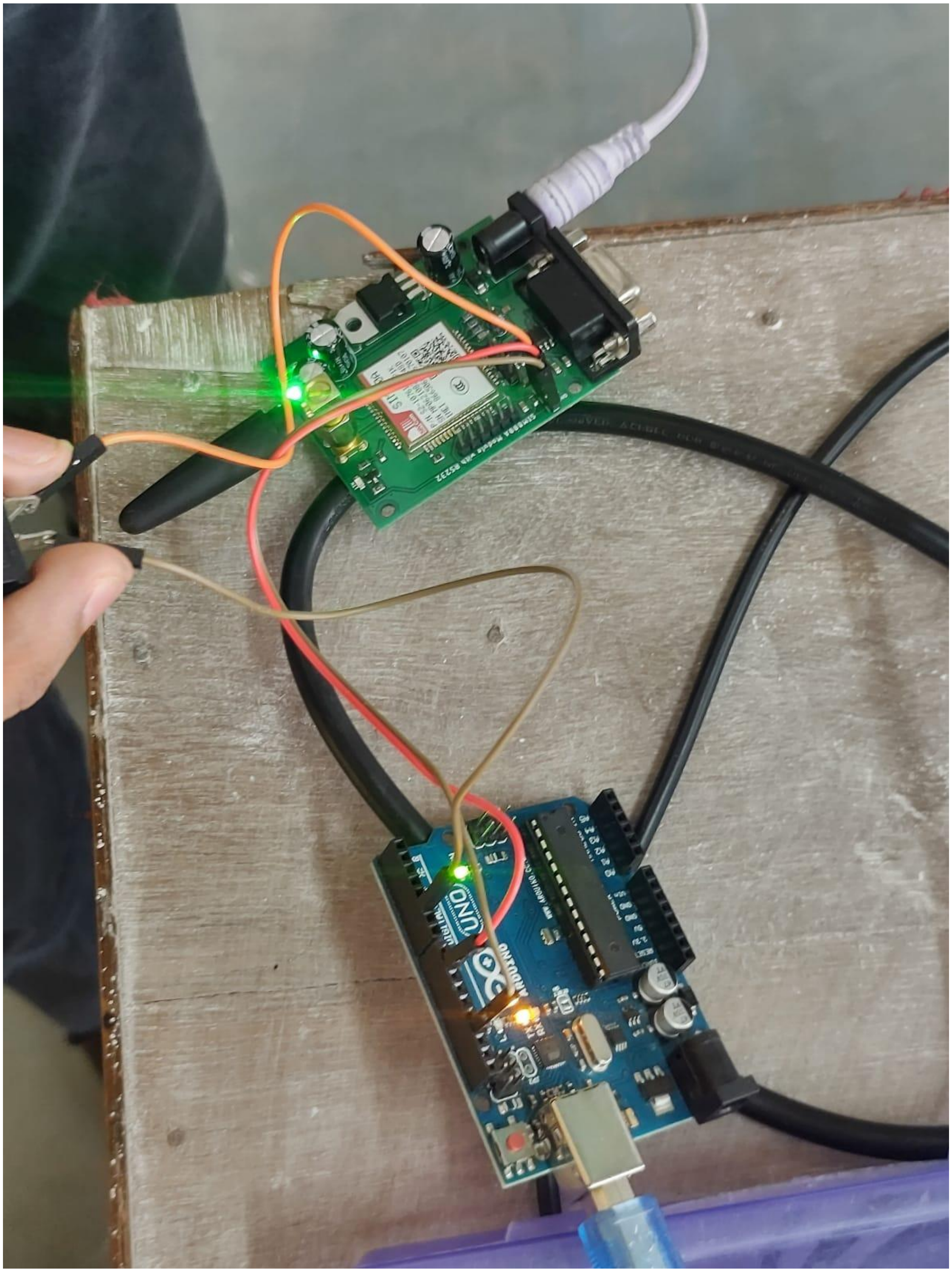

No. of Experiments
45
40
35
30
25
20
15
10
5
0
31
40
CADANS
ADRS
Accident Detection Systems
■CADANS ■ADRS
Accuracy Percentage
120
100
80
60
40
20
0
77
100
■CADANS ■ADRS
Accident Detection Systems
No. of False Reporting
10
9
8
7
6
5
4
3
2
1
0
9
0
CADANS
ADRS
Accident Detection Systems
■CADANS ■ADRS
Percentage
120
100
80
60
40
20
0
90
5
5
20
77
100
20
77
0
Cost
Accuracy
False Reporting
Parameters
■OnStar ■CADANS ■ADRS

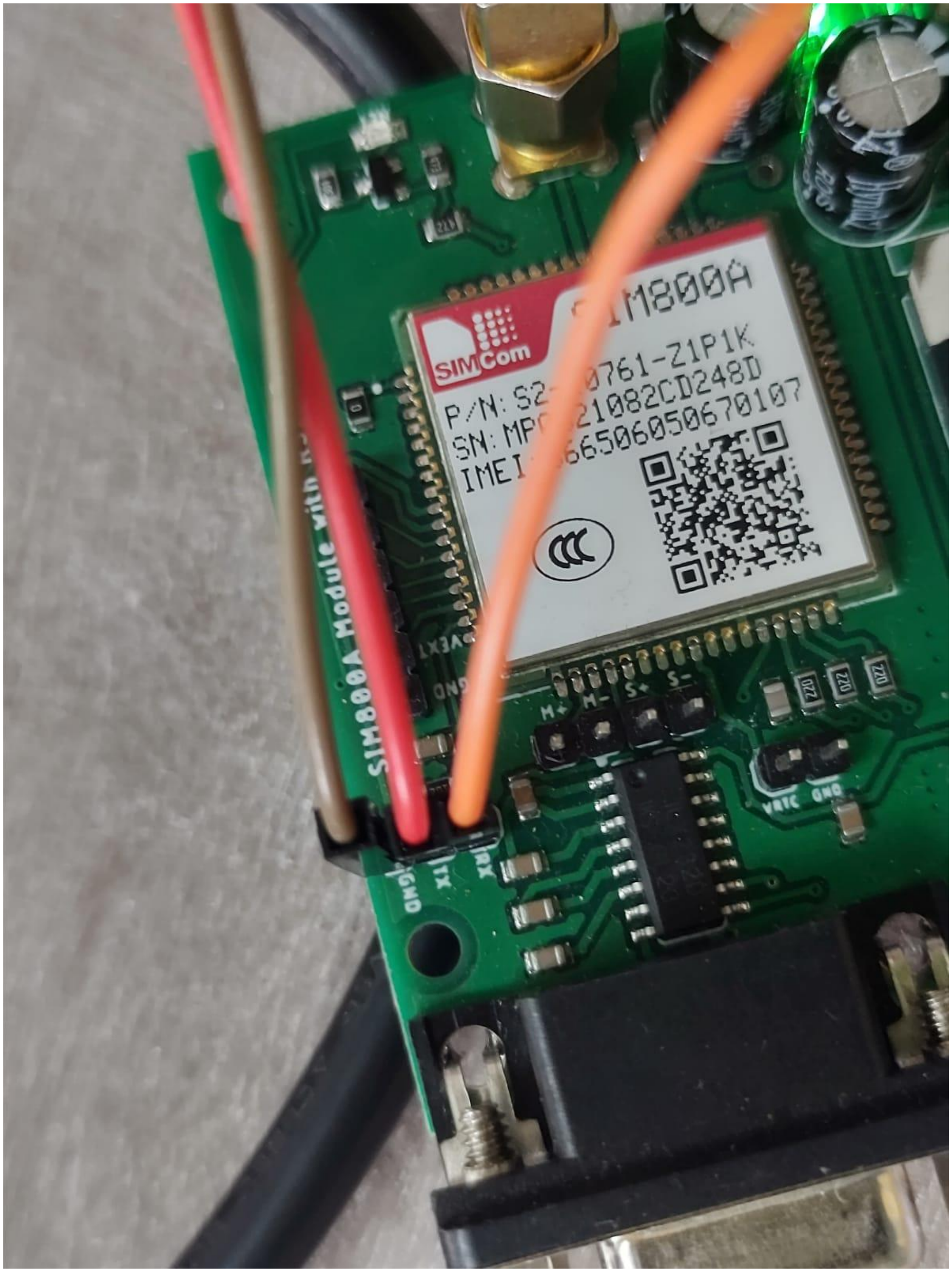
(a)

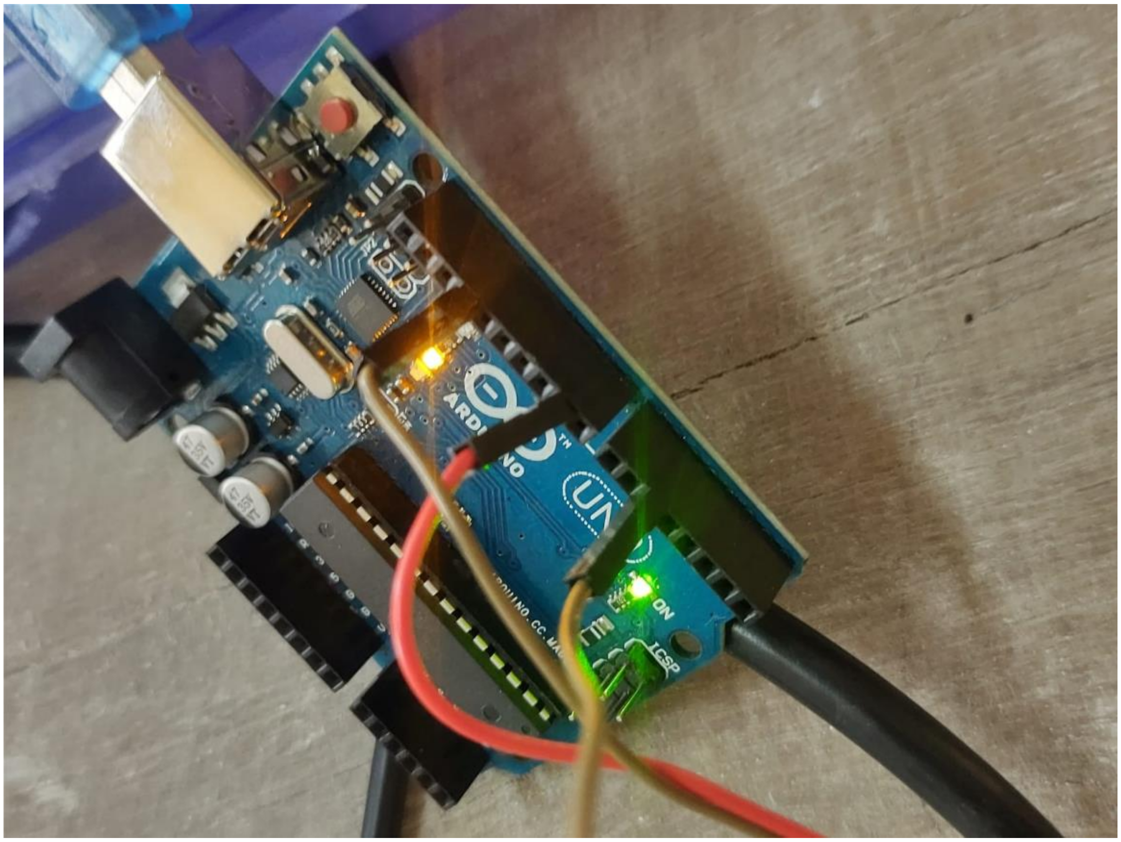
(b)

(c)

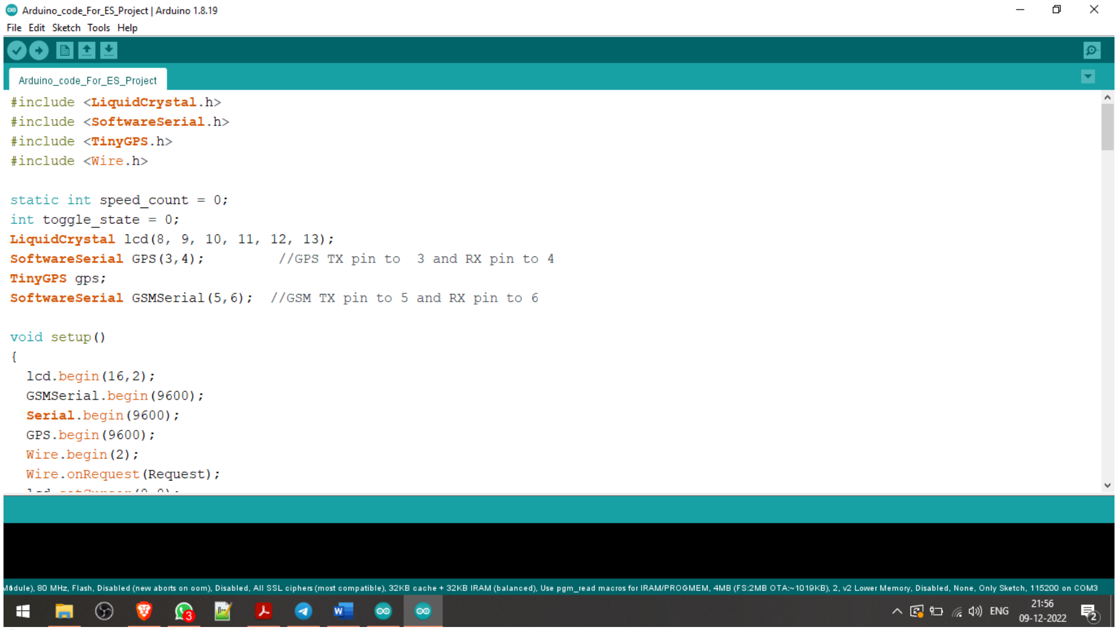
(d)




SIM800A
SIMCom
P/N: S2-
0761-Z1P1K
SN: MP
21082CD248D
IMEI
6650605067010
SIM800A Module with RS
M+ M- S+ S-
VRTC GND
GND TX RX

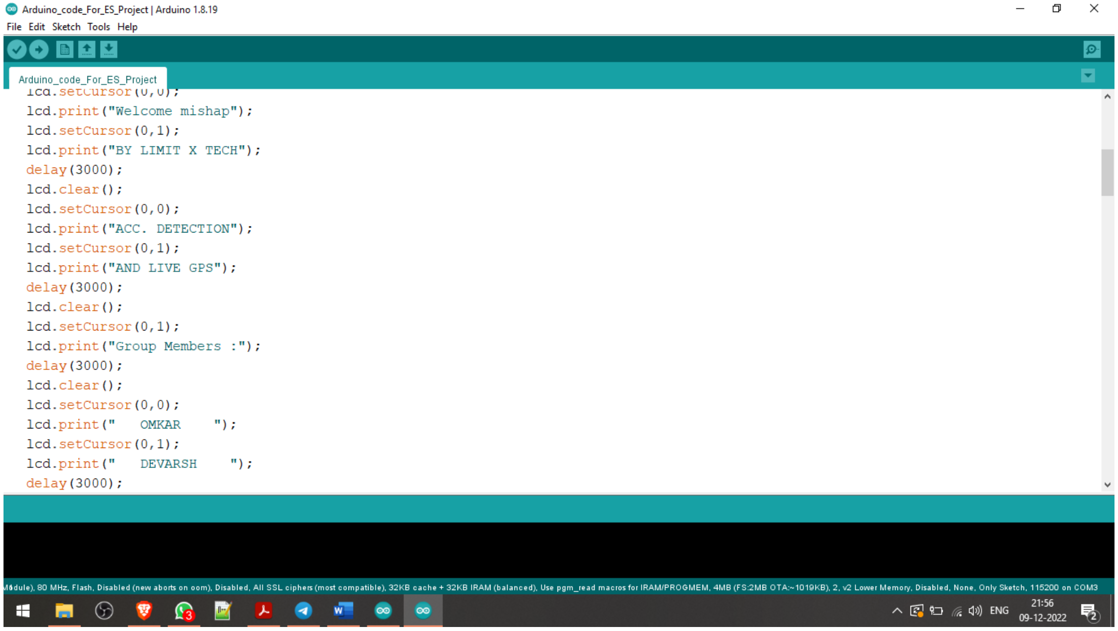

ARDUINO
UNO
ON
ICSP

Screenshorts of the code prepared earlier


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
21:56
09-12-2022
ENG

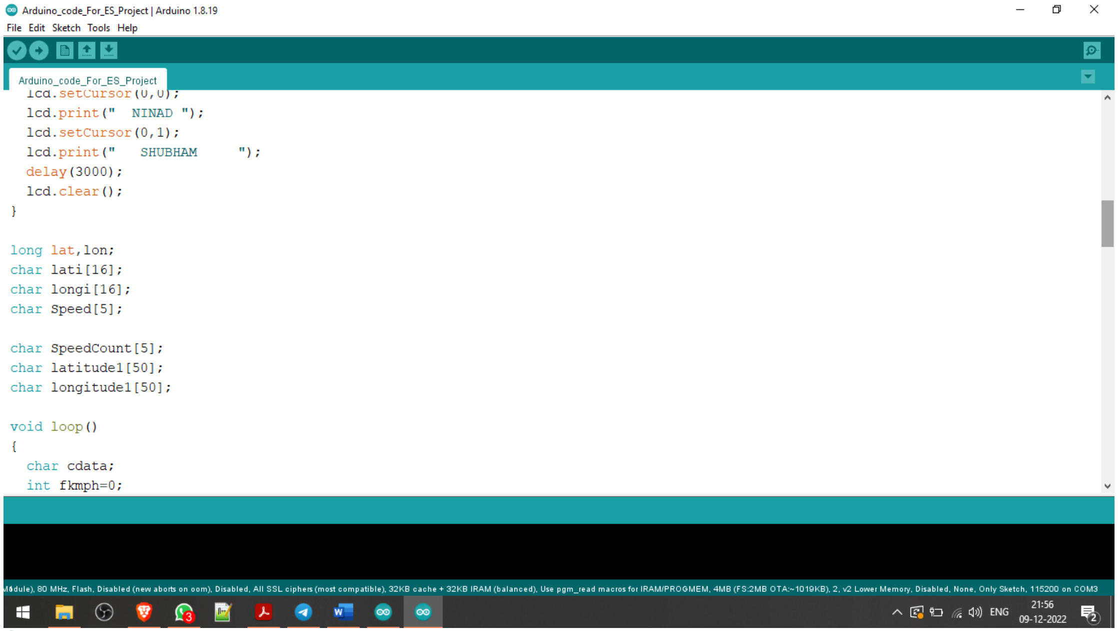
```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <Wire.h>

static int speed_count = 0;
int toggle_state = 0;
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
SoftwareSerial GPS(3,4);          //GPS TX pin to  3 and RX pin to 4
TinyGPS gps;
SoftwareSerial GSMSerial(5,6);  //GSM TX pin to 5 and RX pin to 6

void setup()
{
  lcd.begin(16,2);
  GSMSerial.begin(9600);
  Serial.begin(9600);
  GPS.begin(9600);
  Wire.begin(2);
  Wire.onRequest(Request);
  lcd.setCursor(0,0);
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
21:56
09-12-2022
ENG

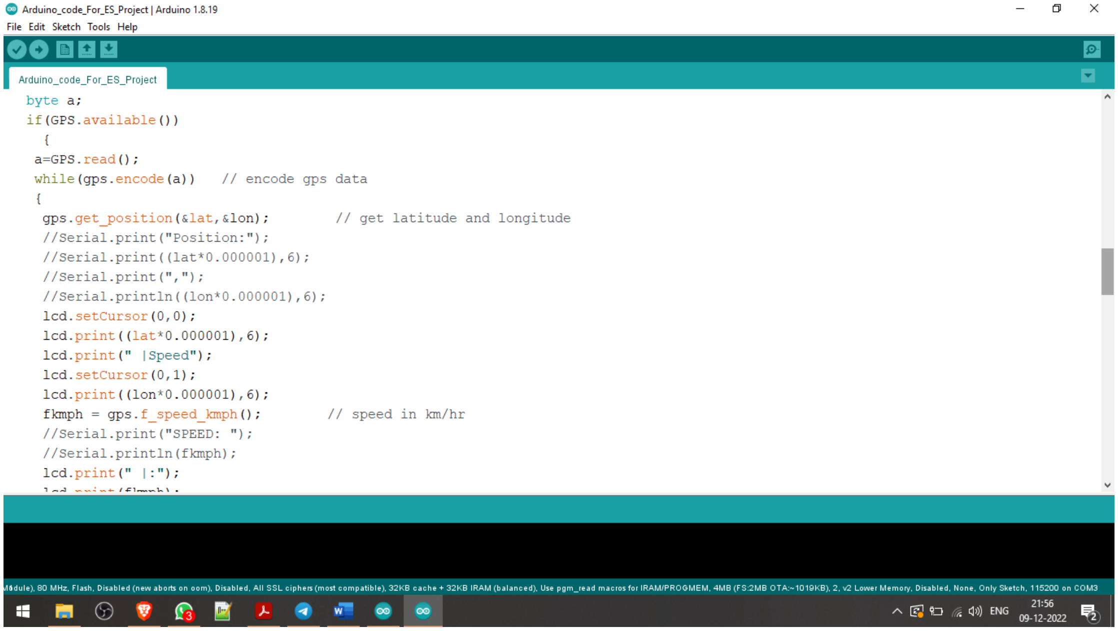
```
  lcd.setCursor(0,0);
  lcd.print("Welcome mishap");
  lcd.setCursor(0,1);
  lcd.print("BY LIMIT X TECH");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("ACC. DETECTION");
  lcd.setCursor(0,1);
  lcd.print("AND LIVE GPS");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("Group Members :");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("   OMKAR     ");
  lcd.setCursor(0,1);
  lcd.print("   DEVARSH     ");
  delay(3000);
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
21:56
09-12-2022
ENG

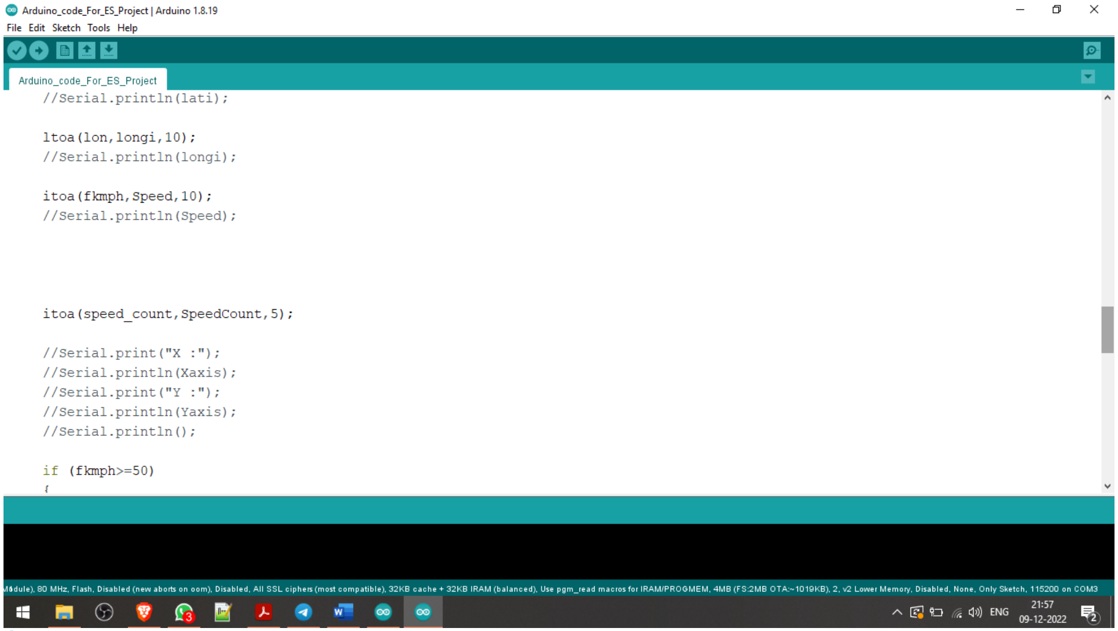
```
  lcd.setCursor(0,0);
  lcd.print("  NINAD ");
  lcd.setCursor(0,1);
  lcd.print("   SHUBHAM     ");
  delay(3000);
  lcd.clear();
}

long lat,lon;
char lati[16];
char longi[16];
char Speed[5];

char SpeedCount[5];
char latitude1[50];
char longitude1[50];

void loop()
{
  char cdata;
  int fkmph=0;
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
21:56
09-12-2022
ENG

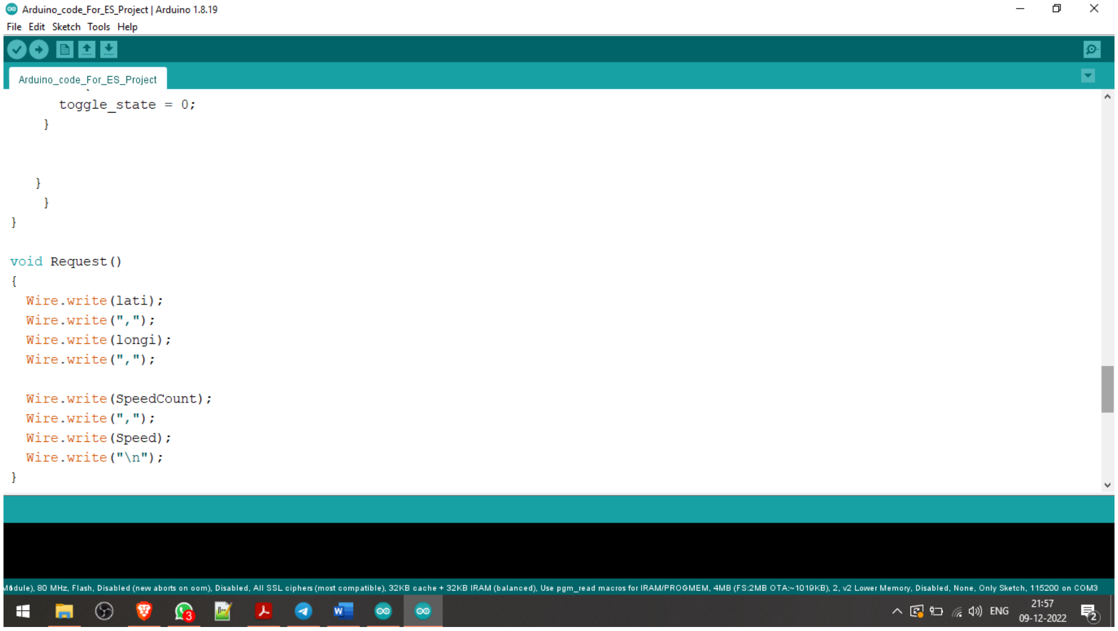
```
  byte a;
  if(GPS.available())
   {
   a=GPS.read();
   while(gps.encode(a))    // encode gps data
   {
    gps.get_position(&lat,&lon);        // get latitude and longitude
    //Serial.print("Position:");
    //Serial.print((lat*0.000001),6);
    //Serial.print(",");
    //Serial.println((lon*0.000001),6);
    lcd.setCursor(0,0);
    lcd.print((lat*0.000001),6);
    lcd.print("  |Speed");
    lcd.setCursor(0,1);
    lcd.print((lon*0.000001),6);
    fkmph = gps.f_speed_kmph();        // speed in km/hr
    //Serial.print("SPEED: ");
    //Serial.println(fkmph);
    lcd.print("  |:");
    lcd.print(fkmph);
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
ENG
21:57
09-12-2022

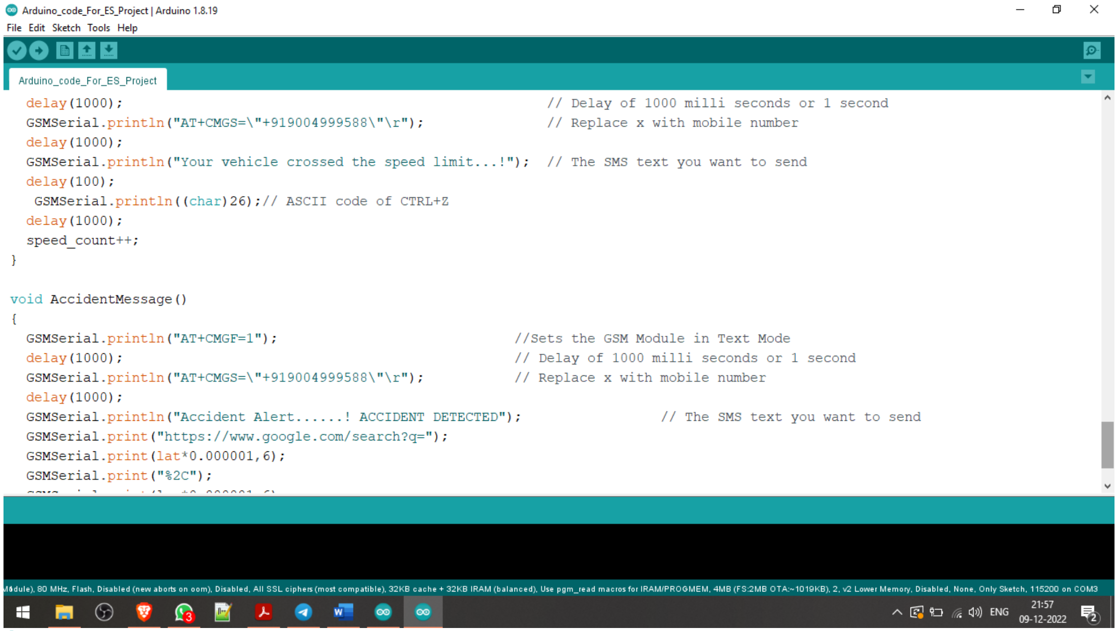
```
    //Serial.println(lati);

    ltoa(lon,longi,10);
    //Serial.println(longi);

    itoa(fkmph,Speed,10);
    //Serial.println(Speed);



    itoa(speed_count,SpeedCount,5);

    //Serial.print("X :");
    //Serial.println(Xaxis);
    //Serial.print("Y :");
    //Serial.println(Yaxis);
    //Serial.println();

    if (fkmph>=50)
    {
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
ENG
21:57
09-12-2022

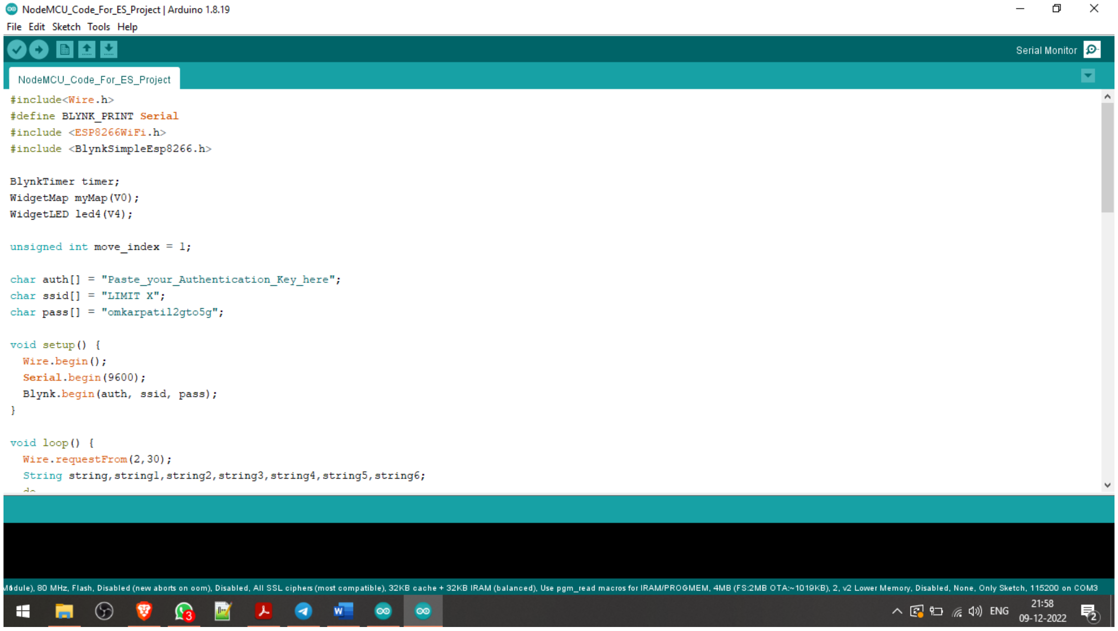
```
      toggle_state = 0;
    }


   }
    }
}

void Request()
{
  Wire.write(lati);
  Wire.write(",");
  Wire.write(longi);
  Wire.write(",");

  Wire.write(SpeedCount);
  Wire.write(",");
  Wire.write(Speed);
  Wire.write("\n");
}
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


Arduino_code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Arduino_code_For_ES_Project
21:57
09-12-2022
ENG

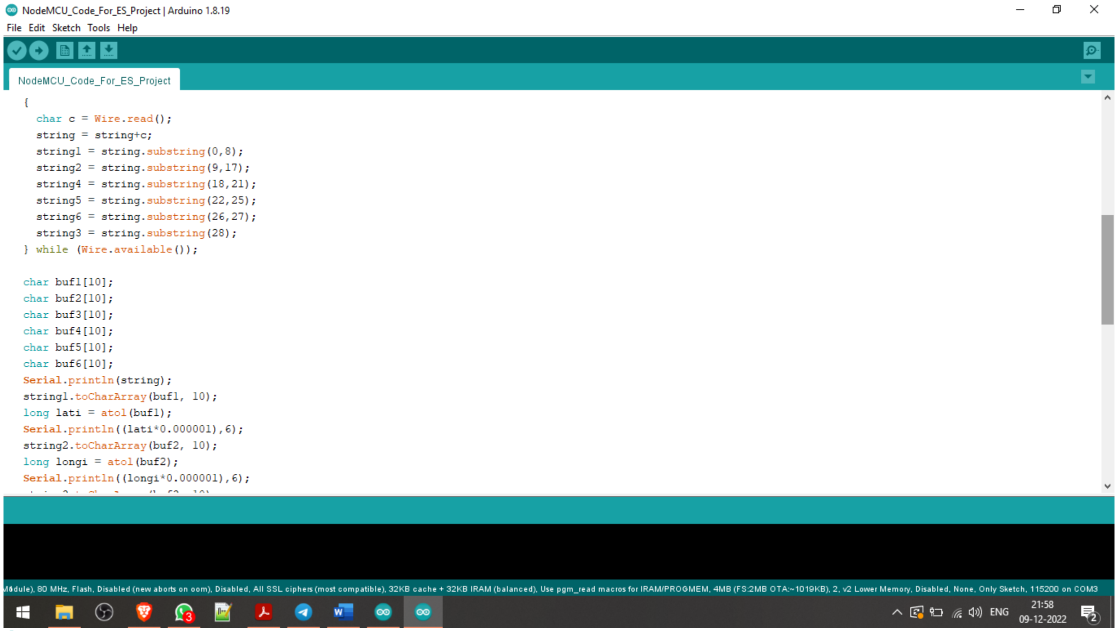
```
  delay(1000);                                              // Delay of 1000 milli seconds or 1 second
  GSMSerial.println("AT+CMGS=\"+919004999588\"\r");         // Replace x with mobile number
  delay(1000);
  GSMSerial.println("Your vehicle crossed the speed limit...!");  // The SMS text you want to send
  delay(100);
   GSMSerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
  speed_count++;
}

void AccidentMessage()
{
  GSMSerial.println("AT+CMGF=1");                       //Sets the GSM Module in Text Mode
  delay(1000);                                          // Delay of 1000 milli seconds or 1 second
  GSMSerial.println("AT+CMGS=\"+919004999588\"\r");     // Replace x with mobile number
  delay(1000);
  GSMSerial.println("Accident Alert......! ACCIDENT DETECTED");                  // The SMS text you want to send
  GSMSerial.print("https://www.google.com/search?q=");
  GSMSerial.print(lat*0.000001,6);
  GSMSerial.print("%2C");
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3


NodeMCU_Code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
Serial Monitor
NodeMCU_Code_For_ES_Project
21:58
09-12-2022
ENG

```
#include<Wire.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;
WidgetMap myMap(V0);
WidgetLED led4(V4);

unsigned int move_index = 1;

char auth[] = "Paste_your_Authentication_Key_here";
char ssid[] = "LIMIT X";
char pass[] = "omkarpatil2gto5g";

void setup() {
  Wire.begin();
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}

void loop() {
  Wire.requestFrom(2,30);
  String string,string1,string2,string3,string4,string5,string6;
```

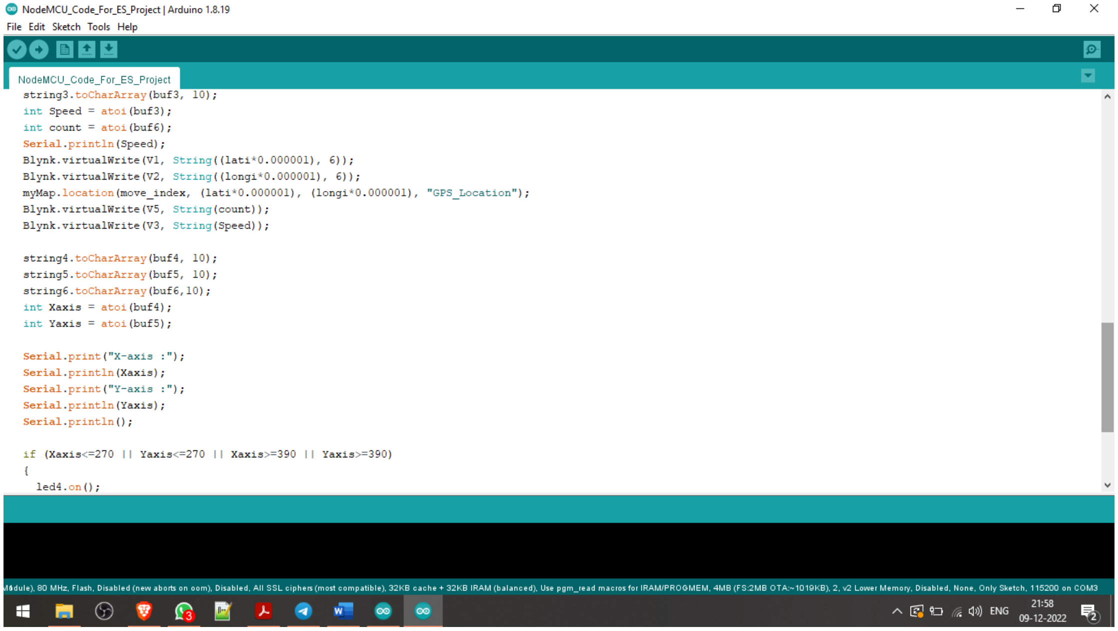
Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3



NodeMCU_Code_For_ES_Project | Arduino 1.8.19

```
{
  char c = Wire.read();
  string = string+c;
  string1 = string.substring(0,8);
  string2 = string.substring(9,17);
  string4 = string.substring(18,21);
  string5 = string.substring(22,25);
  string6 = string.substring(26,27);
  string3 = string.substring(28);
} while (Wire.available());

char buf1[10];
char buf2[10];
char buf3[10];
char buf4[10];
char buf5[10];
char buf6[10];
Serial.println(string);
string1.toCharArray(buf1, 10);
long lati = atol(buf1);
Serial.println((lati*0.000001),6);
string2.toCharArray(buf2, 10);
long longi = atol(buf2);
Serial.println((longi*0.000001),6);
```

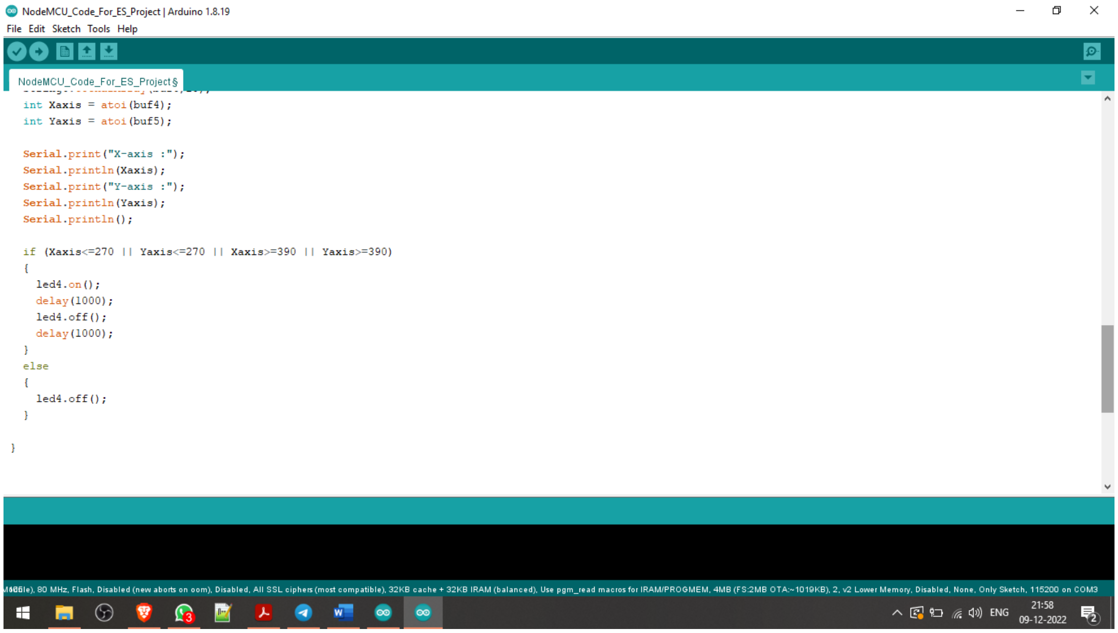
Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3



NodeMCU_Code_For_ES_Project | Arduino 1.8.19

```
string3.toCharArray(buf3, 10);
int Speed = atoi(buf3);
int count = atoi(buf6);
Serial.println(Speed);
Blynk.virtualWrite(V1, String((lati*0.000001), 6));
Blynk.virtualWrite(V2, String((longi*0.000001), 6));
myMap.location(move_index, (lati*0.000001), (longi*0.000001), "GPS_Location");
Blynk.virtualWrite(V5, String(count));
Blynk.virtualWrite(V3, String(Speed));

string4.toCharArray(buf4, 10);
string5.toCharArray(buf5, 10);
string6.toCharArray(buf6,10);
int Xaxis = atoi(buf4);
int Yaxis = atoi(buf5);

Serial.print("X-axis :");
Serial.println(Xaxis);
Serial.print("Y-axis :");
Serial.println(Yaxis);
Serial.println();

if (Xaxis<=270 || Yaxis<=270 || Xaxis>=390 || Yaxis>=390)
{
  led4.on();
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

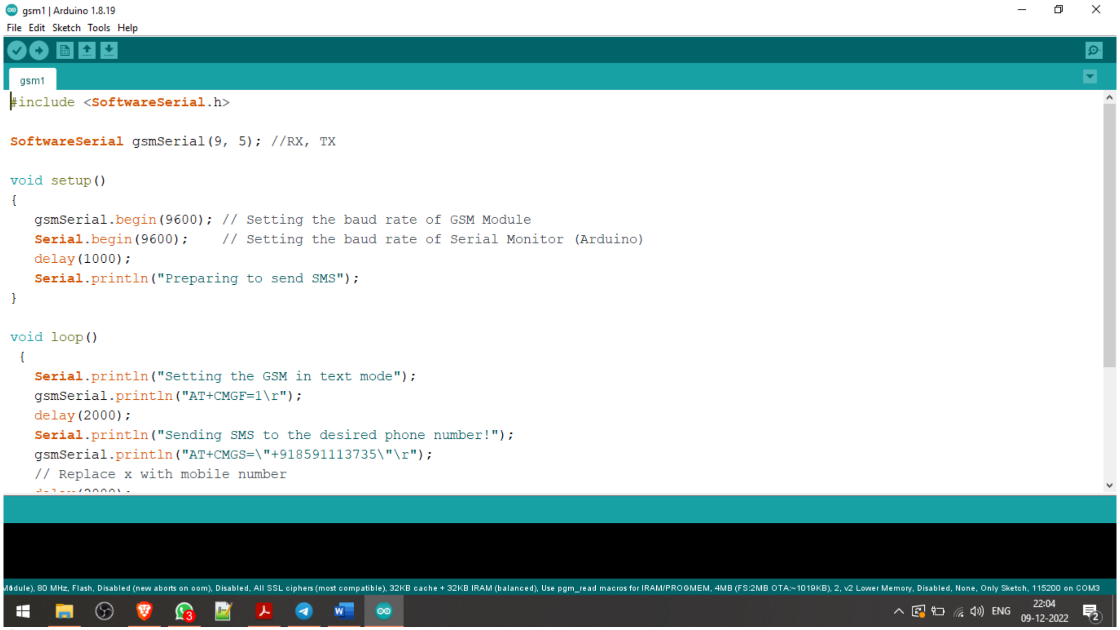


NodeMCU_Code_For_ES_Project | Arduino 1.8.19
File Edit Sketch Tools Help
NodeMCU_Code_For_ES_Project§

```
  int Xaxis = atoi(buf4);
  int Yaxis = atoi(buf5);

  Serial.print("X-axis :");
  Serial.println(Xaxis);
  Serial.print("Y-axis :");
  Serial.println(Yaxis);
  Serial.println();

  if (Xaxis<=270 || Yaxis<=270 || Xaxis>=390 || Yaxis>=390)
  {
    led4.on();
    delay(1000);
    led4.off();
    delay(1000);
  }
  else
  {
    led4.off();
  }

}
```

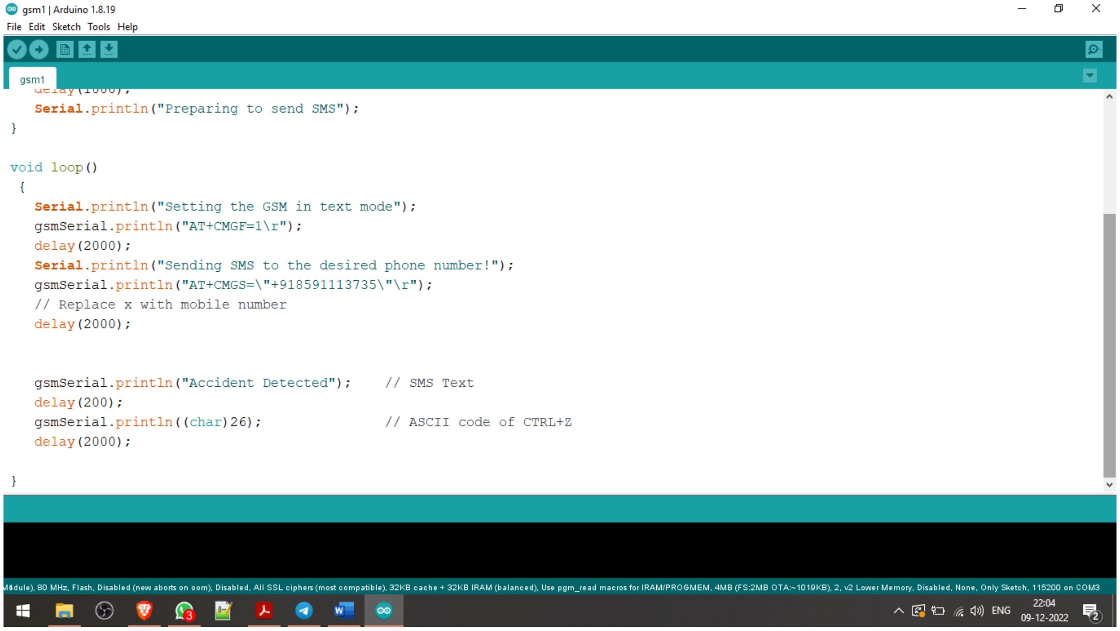
```
M1dule), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3
```

Code Screenshot of actual project



```
#include <SoftwareSerial.h>

SoftwareSerial gsmSerial(9, 5); //RX, TX

void setup()
{
   gsmSerial.begin(9600); // Setting the baud rate of GSM Module
   Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)
   delay(1000);
   Serial.println("Preparing to send SMS");
}

void loop()
 {
   Serial.println("Setting the GSM in text mode");
   gsmSerial.println("AT+CMGF=1\r");
   delay(2000);
   Serial.println("Sending SMS to the desired phone number!");
   gsmSerial.println("AT+CMGS=\"+918591113735\"\r");
   // Replace x with mobile number
   delay(2000);
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3



```
   delay(1000);
   Serial.println("Preparing to send SMS");
}

void loop()
 {
   Serial.println("Setting the GSM in text mode");
   gsmSerial.println("AT+CMGF=1\r");
   delay(2000);
   Serial.println("Sending SMS to the desired phone number!");
   gsmSerial.println("AT+CMGS=\"+918591113735\"\r");
   // Replace x with mobile number
   delay(2000);


   gsmSerial.println("Accident Detected");    // SMS Text
   delay(200);
   gsmSerial.println((char)26);               // ASCII code of CTRL+Z
   delay(2000);

}
```

Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

SYSTEM TESTING & IMPLEMENTATION

I. Software Testing

Software testing is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but not necessarily limited to:

- Analyzing the product requirements for completeness and correctness in various contexts like industry perspective, business perspective, feasibility and viability of implementation, usability, performance, security, infrastructure considerations, etc.
- Reviewing the product architecture and the overall design of the product
- Working with product developers on improvement in coding techniques, design patterns, tests that can be written as part of code based on various techniques like boundary conditions, etc.
- Executing a program or application with the intent of examining behavior
- Reviewing the deployment infrastructure and associated scripts and automation
- Take part in production activities by using monitoring and observability techniques

White Box Testing

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which also known as glass box is testing, structural testing, clear box testing, open box testing and transparent box testing. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

White box testing technique

1) Statement Coverage:- This technique requires every possible statement in the code to be tested at least once during the testing process. In statement coverage, every node must be traversed at least once.

2) Branch Coverage - This technique checks every possible path (if-else and other conditional loops) of a software application. Every edge must be traversed at least once. As per flowchart all edges must be traversed at least once.

3) Path Coverage This technique is used to ensure that every possible path (each statement and branch) is executed and tested.

Types of White Box Testing

White box testing can take several forms:

Unit testing — tests written as part of the application code, which test that each component is working as expected.

Mutation testing — a type of unit testing that checks the robustness and consistency of the code by defining tests, making small, random changes to the code and seeing if the tests still pass. Mutation testing is generally conducted to re-check any kind software bugs in the system. Mutation testing is often used to discover the best coding techniques to use for expanding a software solution.

Integration testing — tests specifically designed to check integration points between internal components in a software system, or integrations with external systems.

Penetration Testing: In this testing, the tester/developer has full information of the application's source code, detailed network information, IP addresses involved and all server information the application runs on. The aim is to attack the code from several angles to expose security threats.

Dynamic Testing-

Under Dynamic Testing, a code is executed. It checks for functional behavior of software system, memory/cpu usage and overall performance of the system. Hence the name "Dynamic" • The main objective of this testing is to confirm that the software product works in conformance with the business requirements. This testing is also called an Execution technique or validation testing. • Dynamic testing executes the software and validates the output with the expected outcome. Dynamic testing is performed at all levels of testing and it can be either black or white box testing.

Structural Testing:

- Structure-based testing, therefore, can be defined as a type of software testing that tests the code’s structure and intended flows. For example, verifying the actual code for aspects like the correct implementation of conditional statements, and whether every statement in the code is correctly executed. It is also known as structure-based testing.

● To carry out this type of testing, we need to thoroughly understand the code. This is why this testing is usually done by the developers who wrote the code as they understand it best.

● Structural testing is the type of testing carried out to test the structure of code. It is also known as White Box testing or Glass Box testing. This type of testing requires knowledge of the code, so, it is mostly done by the developers. It is more concerned with how system does it rather than the functionality of the system. It provides more coverage to the testing. For ex, to test certain error message in an application, we need to test the trigger condition for it, but there must be many trigger for it. It is possible to miss out one while testing the requirements drafted in SRS. But using this testing, the trigger is most likely to be covered since structural testing aims to cover all the nodes and paths in the structure of code.

● Structural testing is a type of software testing which uses the internal design of the software for testing or in other words the software testing which is performed by the team which knows the development phase of the software, is known as structural testing.

● Structural testing is basically related to the internal design and implementation of the software i.e. it involves the development team members in the testing team. It basically tests different aspects of the software according to its types. Structural testing is just the opposite of behavioral testing.

● The knowledge of the code's internal executions and how the software is implemented is a necessity for the test engineer to implement the structural testing.

● Throughout the structural testing, the test engineer intends on how the software performs, and it can be used at all levels of testing. The intention behind the testing process is finding out how the system works not the functionality of it. To be more specific, if an error message is popping up in an application there will be a reason behind it. Structural testing can be used to find that issue and fix it Advantages of Structural Testing: · Forces test developer to reason carefully about implementation Reveals errors in "hidden" code · Spots the Dead Code or other issues with respect to best programming practices.

Disadvantages of Structural Box Testing:

Expensive as one has to spend both time and money to perform white box testing. Every possibility that few lines of code is missed accidentally.

In-depth knowledge about the programming language is necessary to perform white box testing.

Functional Testing -

FUNCTIONAL TESTING is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Functional testing mainly involves black box testing and it is not concerned about the source code of the application. This testing checks User Interface, APIs, Database, Security, Client/Server communication and other functionality of the Application Under Test. The testing can be done either manually or using automation.

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on simulation of actual system usage but does not develop any system structure assumptions.

Functional Testing: It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application. Functional testing also called as black-box testing, because it focuses on application specification rather than actual code. Tester has to test only the program rather than the system. How to do Functional Testing Following is a step by step process on

How to do Functional Testing :

✔ Understand the Functional Requirements

✔ Identify test input or test data based on requirements

✔ Compute the expected outcomes with selected test input values

✔ Execute test cases

✔ Compare actual and computed expected results.

Code Coverage :

Code coverage is a software testing metric that determines the number of lines of code that is successfully validated under a test procedure, which in turn, helps in analyzing how comprehensively a software is verified. Code coverage is a software testing metric or also termed as a Code Coverage Testing which helps in determining how much code of the source is tested which helps in accessing quality of test suite and analyzing how comprehensively a software is verified. Actually in simple code coverage refers to the degree of which the source code of the software code has been tested. This Code Coverage is considered as one of the form of white box testing As we know at last of the development each client wants a quality software product as well as the developer team is also responsible for delivering a quality software product to the customer/client. Where this quality refers to the product's performance, functionalities, behavior, correctness, reliability, effectiveness, security, and maintainability. Where Code Coverage metric helps in determining the performance and quality aspects of any software. Code coverage is one such software testing metric that can help in assessing the test performance and quality aspects of any software. Such an insight will equally be beneficial to the development and QA team. For developers, this metric can help in dead code detection and elimination. On the other hand, for QA, it can help to check missed or uncovered test cases. They can track the health status and quality of the source code while paying more heed to the uncaptured parts of the code.

Code Coverage testing is determining how much code is being tested. It can be calculated using the formula:

Code Coverage = (Number of lines of code executed)/(Total Number of lines of code in a system component) * 100

Code Coverage Criteria

1.Statement coverage: how many of the statements in the program have been executed.

2.Branches coverage: how many of the branches of the control structures (if statements for instance) have been executed. Condition coverage: how many of the boolean sub-expressions have been tested for a true and a false value.

3.Line coverage: how many of lines of source code have been tested. These metrics are usually represented as the number of items actually tested, the items found in your code, and a coverage percentage (items tested / items found)

4.Function coverage: how many of the functions defined have been called.

Code Complexity Testing

Cyclomatic Complexity in Software Testing is a testing metric used for measuring the complexity of a software program. It is a quantitative measure of independent paths in the source code of a software program. Cyclomatic complexity can be calculated by using control flow graphs or with respect to functions, modules, methods or classes within a software program. • independent path is defined as a path that has at least one edge which has not been traversed before in any other paths. • Cyclomatic Complexity is software metric useful for structured or White Box Testing. It is mainly used to evaluate complexity of a program. If the decision points are more, then complexity of the program is more. If program has high complexity number, then probability of error is high with increased time for maintenance and trouble shoot • For example, if source code contains no control flow statement then its cyclomatic complexity will be 1 and source code contains a single path in it. Similarly, if the source code contains one if condition then cyclomatic complexity will be 2 because there will be two paths one for true and the other for false. • This metric was developed by Thomas J. McCabe in 1976 and it is based on a control flow representation of the program. Control flow depicts a program as a graph which consists of Nodes and Edges • In the graph, Nodes represent processing tasks while edges represent control flow between the nodes.

Uses of Cyclomatic Complexity:

Cyclomatic Complexity can prove to be very helpful for developers and testers.

Helps developers and testers to determine independent path executions

Developers can assure that all the paths have been tested atleast once

Helps us to focus more on the uncovered paths

Improve code coverage in Software Engineering

Evaluate the risk associated with the application or program

Black Box Testing

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing. Black box testing is a type of software testing in which the functionality of the software is not known. The testing is done without the internal knowledge of the products. • It is also known as specification based testing • The black box testing is also known as an opaque, closed box, function-centric testing. It emphasizes on the behavior of the software. Black box testing checks scenarios where the system can break •

Types of Black Box Testing

There are many types of Black Box Testing but the following are the prominent ones –

Functional testing - This black box testing type is related to the functional requirements of a system; it is done by software testers.

Non-functional testing - This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.

Regression testing – Regres is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code. Black Box Testing Techniques In order to systematically test a set of functions, it is necessary to design test cases. Testers can create test cases from the requirement specification document using the following Black Box Testing techniques.

Equivalence Partitioning

Boundary Value Analysis

Decision Table Testing

State Transition Testing

Error Guessing

Graph-Based Testing Methods

1.Boundary Value Analysis(BVA)

Boundary Value Technique is used to test boundary values, boundary values are those that contain the upper and lower limit of a variable. It tests, while entering boundary value whether the software is producing correct output or not.

Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.

Boundary value analysis is one of the widely used case design technique for black box testing. It is used to test boundary values because the input values near the boundary have higher chances of error.Whenever we do the testing by boundary value analysis, the tester focuses on, while entering boundary value whether the software is producing correct output or not.Boundary values are those that contain the upper and lower limit of a variable. Assume that, age is a variable of any function, and its minimum value is 18 and the maximum value is 30, both 18 and 30 will be considered as boundary values. The basic assumption of boundary value analysis is, the test cases that are created using boundary values are most likely to cause an error

Equivalence partitioning is a technique of software testing in which input data divided into partitions of valid and invalid values, and it is mandatory that all partitions must exhibit the same behavior. • This technique is also known as Equivalence Class

Partitioning (ECP). In this technique, input values to the system or application are divided into different classes or groups based on its similarity in the outcome. Hence, instead of using each and every input value we can now use any one value from the group/class to test the outcome. In this way, we can maintain the test coverage while we can reduce a lot of rework and most importantly the time spent.

Decision Table- Decision Table Technique is a systematic approach where various input combinations and their respective system behavior are captured in a tabular form. It is appropriate for the functions that have a logical relationship between two and more than two inputs. Decision Table is aka Cause-Effect Table. This test technique is appropriate for functionalities which has logical relationships between inputs (if-else logic). In Decision table technique, we deal with combinations of inputs. To identify the test cases with decision table, we consider conditions and actions. We take conditions as inputs and actions as outputs. • In some instances, the inputs combinations can become very complicated for tracking several possibilities.Such complex situations rely on decision tables, as it offers the testers an organized view about the inputs combination and the expected output.

II. Test Plan

A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.

The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.

The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager. The test plan is prepared by the Test Lead (60%), Test Manager(20%), and by the test engineer(20%).

III. Test Cases

Test ID	Case Name	Case Description	Expected Value	Actual Value	Result
01	Adxl	When moved by x axis message must be printed	Must send message	Must send message	Pass
02	Adxl 2	When moved by y Axis messge must be printed	Must send message	Must send message	Pass
03	Adxl 3	When moved by z Axis messge must be printed	Must not send message	Must not send message	Pass
04	Gps	Gps must send latitude and longtitude	Must send both the values	Not sending as sensor is failed	failed
05	Gsm	Gsm must send messge to given emergency contact	Must send Message	Sending the message	pass

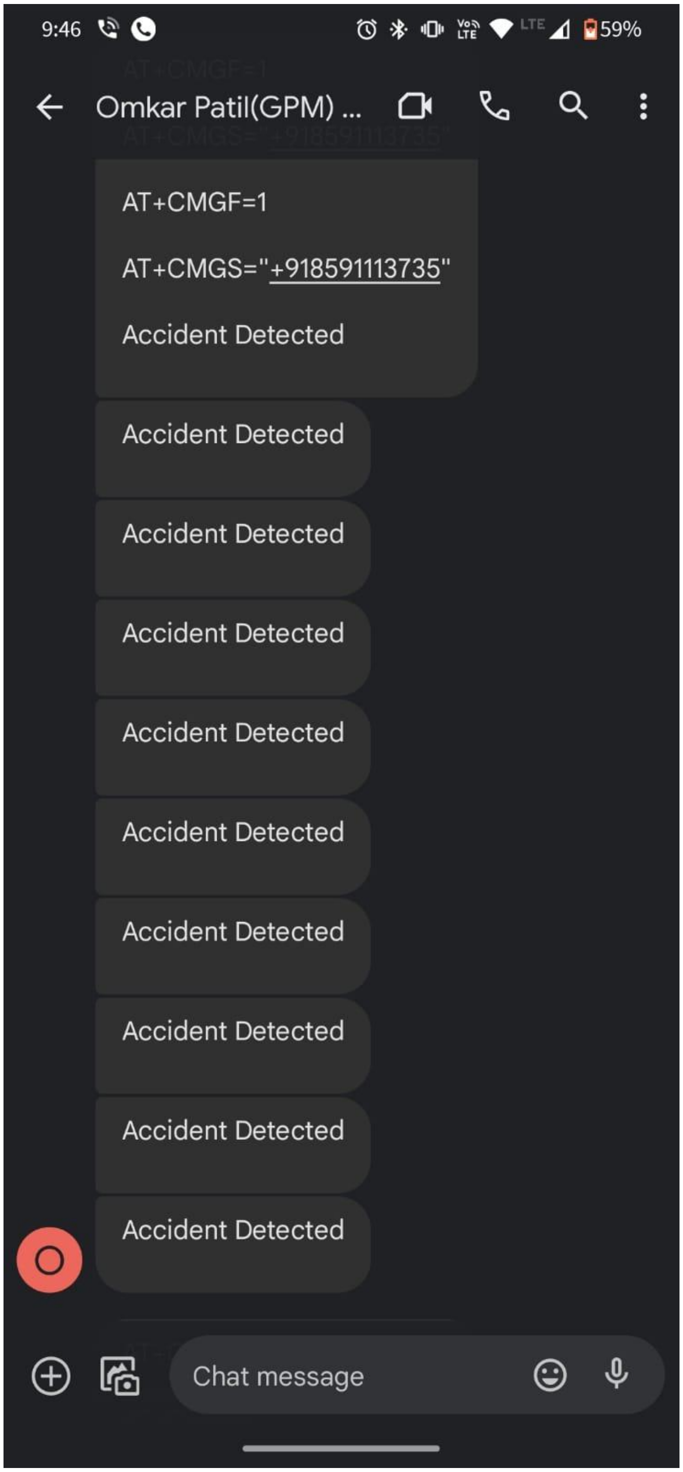
Changes due to Failure of Components

The main 3 components of our projects were GPS, GSM, Node Mcu ,

But because the Gps sensor as well as adxl sensor was defective we need to change the entire project idea. So here only the GSM modem is used

Due to this the node mcu in the entire Project was of no use so we also altered this microcontroller.

RESULT



AT+CMGF=1

AT+CMGS="+918591113735"

Accident Detected

Accident Detected

Accident Detected

Accident Detected

Accident Detected

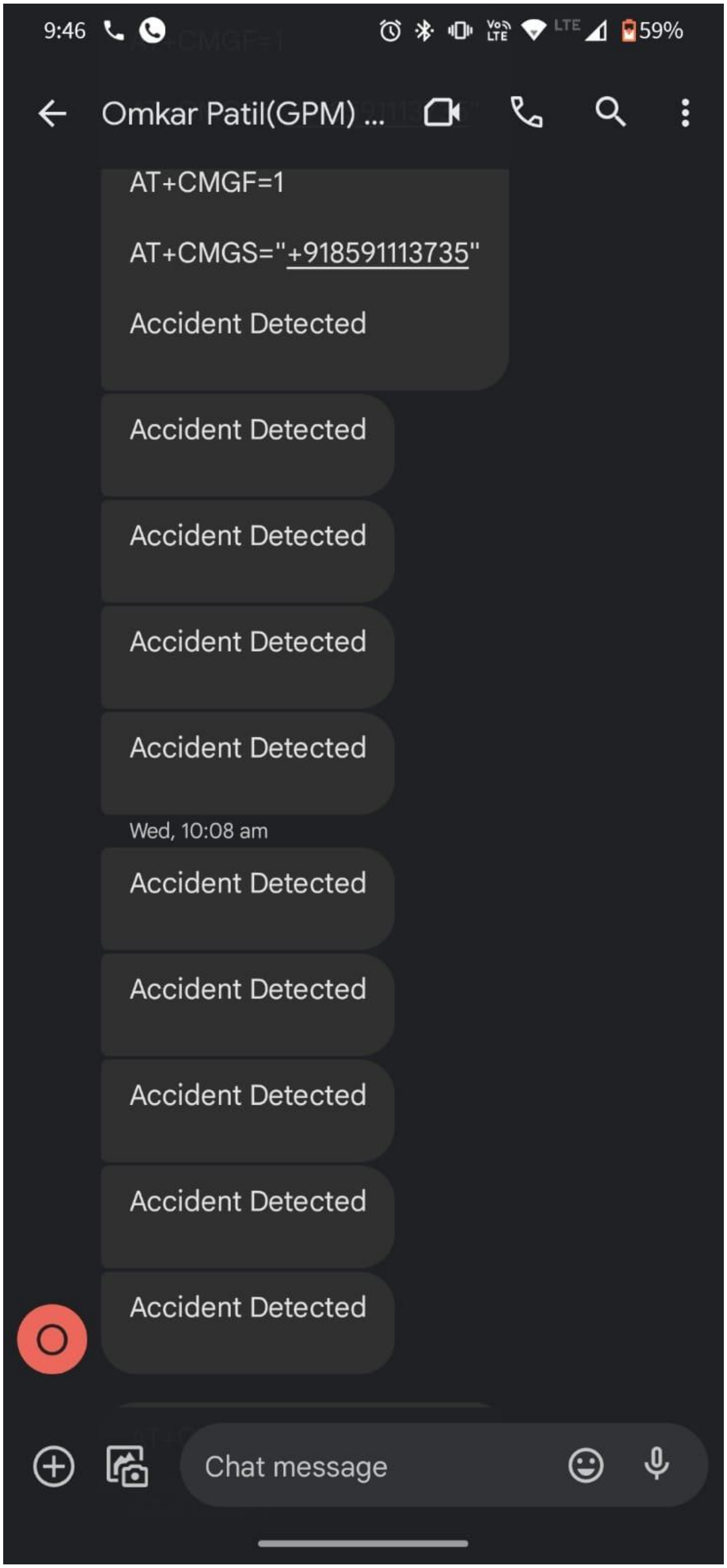
Accident Detected

Accident Detected

Accident Detected

Accident Detected

Accident Detected



AT+CMGF=1

AT+CMGS="+918591113735"

Accident Detected

Accident Detected

Accident Detected

Accident Detected

Accident Detected

Wed, 10:08 am

Accident Detected

Accident Detected

Accident Detected

Accident Detected

Accident Detected

FUTURE SCOPE

The future scope of this project is very bright:

- The victim can get the First Aid as Early as Possible
- As the cost of the project is very less more and more people can afford it due to which the life of people
- Gdp loss due the accidents will be reducded
- Helps understand the community life according to life save
- Helps control to keep speed within limits .
- May help individuals to make their lives easier.
- Alerts could be issued way early.
- Useful for entire india system

- **Conclusion**

The name of our Project Initially was Car Accident Detection and Live location Detection, but due to the failure of Gps modem and Adxl 335 sensor we need to change our project name to Car accident detection and sending Emergency SOS message.

During the development of this there was a lot of ups and downs. During this development we actually learned more detailed about Embedded C, components working, node mcu uses and other languages. We studied hardware integration and

The Mini Project was all about gaining knowledge and transforming our knowledge into work or project. We had a great team effort, planning and execution. This helped developed various skills and sharpened them.