A

PROJECT REPORT

ON

“VEHICLE SAFETY SYSTEM FOR ALCOHOL DETECTION & ACCIDENT AWARENESS”

Submitted in partial fulfilment of the requirement for the degree of

Bachelor of Engineering
In
Electronics and Electrical Engineering

Guided by:

MS. RAKHEE

(Department of ECE)

Submitted by:

IJAYRAJ SINGH (20R/41085)

AKSHITA VYAS (20R/41059)

AKSHITA VYAS (20R/41074)

GARGI MISHRA (20R/43609)

Department of Electronics and Communication Engineering
M.B.M. University, Jodhpur 342001
2023-2024

CERTIFICATE

This is to certify that this project report titled **“VEHICLE SAFETY SYSTEM FOR ALCOHOL DETECTION & ACCIDENT AWARENESS”** has been submitted by **Ijay Raj Singh (21UEEE9018), Akshita Vyas (21UEEE9048), Akshita Vyas (21UEEE9049), Gargi Mishra (21UEEE9053)** in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Electronics and Electrical Engineering of the MBM University, Jodhpur during the academic year 2023-24 and is a record of work carried out by them under my guidance and supervision.

Date:

Ms. Rakhee
Department of Electronics & Communication
M.B.M University, Jodhpur

ACKNOWLEDGEMENT

It gives me immense pleasure to present this project report. I would like to take the opportunity to thank the people, without whom this would not have been possible. Firstly, I would like to express my deep gratitude to **Shri Kapil Parihar, Head of ECE department, MBM University,** Jodhpur for providing me an opportunity to accomplish the project.

It is my profound privilege to express my deep sense of gratitude to my guide **Ms. Rakhee** for her precious guidance, constructive encouragement and support throughout the project period, without whom things could not have proceeded as smoothly as they did.

I express my regards towards **Mr. KK ARORA** for all the technical and administrative support in carrying out the project.

I would, also, like to express my deepest appreciation to my fellow team members of the training for their valuable co-operation.

IJAY RAJ SINGH (20R/41085)

AKSHITA VYAS (20R/41059)

AKSHITA VYAS (20R/41074)

GARGI MISHRA (20R/43609)

Department of Electronics & Communication
M.B.M University, Jodhpur

ABSTRACT

The primary objective of this project is to address the critical issues of road accidents, which has become a prevalent concern in the ever developing world. With an alarming increase in accidents specially caused due to drunk driving worldwide, installing alcohol detectors in vehicles has become a crucial safety measure to protect occupants and prevent fatalities.

And hence keeping these issues in mind, we have designed a system which in the event of an accident, will automatically shut off the engine and activates an alarm, ensuring immediate attention to the situation. Furthermore, to address the challenges of delayed emergency response times, the system incorporates advanced technology such as the MPU6050 sensor for accident detection and powered by the ESP8266 and GPS module (Neo-6M) for precise location tracking. Upon detecting an accident, the GPS module swiftly transmits the location coordinates to preconfigured emergency contacts, such as the police or medical services, facilitating rapid assistance and potentially saving lives.
The system also has the additional facilities of detecting presence of alcohol in the breath of the driver, on the occasion that alcohol is detected it would deactivate the engine.

This design not only detects accidents but also streamlines the emergency response process, thereby reducing the severity of injuries and fatalities. The integration of alcohol detection and accident detection systems in vehicles represents a significant advancement in road safety technology, contributing to the overarching goal of preserving human lives and promoting safer transportation environments

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER: 1
INTRODUCTION

Automobiles play a vital role in enabling us to commute to workplaces, connect with family and friends, and facilitate the delivery of goods. However, they also contribute to significant disasters. Accidents, defined as unforeseen and unintentional events, frequently occur on roads, often resulting from human error.

The prevalent scenario highlights that a significant proportion of road accidents stem from drunk driving. Intoxicated drivers, lacking stability, often engage in reckless driving on highways, posing grave risks to their lives and those of others on the road. Despite existing laws in India prohibiting drinking and driving, strict enforcement remains a challenge for law enforcement due to the limitations in monitoring every driver's state in real-time.

1.1 BACKGROUND

An alarming number of road accidents were reported in 2016 by the Indian Ministry of Statistics, with speed violation cited as a primary cause. However, it can be inferred that the majority of these cases result from drivers operating under the influence of alcohol. Studies, including one conducted by the World Health Organization in 2008, indicate that approximately 50%-60% of traffic accidents are linked to drink-driving.

Moreover, global WHO data on road traffic fatalities in 2013 revealed a staggering 1.25 million deaths, with higher fatality rates observed in low- and middle-income countries. Many commercial vehicle drivers in India admitted to consuming alcohol during workdays, highlighting the prevalence of drink-driving among drivers, including those operating heavy-duty trucks.

India has set a legal blood alcohol concentration (BAC) limit of 30mg/100mL, beyond which driving is deemed unlawful. Different levels of BAC impair a driver's cognitive and physical abilities, with severe impairment occurring at higher levels, rendering them incapable of safe operation.

Despite numerous initiatives by governments and NGOs worldwide to promote safe driving practices, accidents persist with alarming frequency. Many lives could potentially be saved if emergency services could promptly receive crash information.

Therefore, the implementation of an efficient automatic accident detection system, coupled with automatic notification to emergency services, becomes imperative to safeguard precious human lives. Furthermore, there is a crucial need for a system capable of curbing drunk driving. Such a system should incorporate advanced alcohol detection technology to effectively identify intoxicated drivers and prevent them from operating vehicles.

This project would leverage the capabilities of GPS receivers to monitor vehicle speed and detect accidents based on monitored data. Utilizing a microcontroller to process GPS data and communicate via the GSM network, the system would swiftly relay accident location and time to the pre fed numbers, enabling prompt emergency response. Additionally, to deal with the accidents corresponding to drunk driving, this system would focus on preventive measures by enabling sensors that detect the presence of alcohol, and by doing that the vehicle will not operate henceforth eliminating any chance of an accident caused by the same.

1.2 INTRODUCTION TO EMBEDDED SYSTEMS

An **embedded system** is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electrical system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints. Embedded systems control many devices in common use today. Ninety- eight percent of all microprocessors manufactured are used in embedded systems.

Modern embedded systems are often based on microcontrollers (i.e. microprocessors with integrated memory and peripheral interfaces), but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also common, especially in more complex systems. In either case, the processor(s) used may be types ranging from general purpose to those specialized in a certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP). Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have real-time performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems consist of small parts within a larger device that serves a more general purpose. Similarly, an embedded system in an automobile provides a specific function as a subsystem of the car itself. The program instructions written for embedded systems are referred to as firmware, and are stored in read-only memory or flash memory chips. They run with limited computer hardware resources: little memory, small or non- existent keyboard or screen.

1.3 PROBLEM STATEMENT

The escalating number of road accidents attributed to drivers under the influence of alcohol presents a pressing concern for road safety. Despite legislative measures and public awareness campaigns, the incidence of drunk driving persists, resulting in tragic consequences and loss of lives. Adding to this issue is the fact that accidents often to go undetected, further increasing the number of lives that are lost.

The problem statement highlights the rising trend of road accidents caused by drivers consuming alcohol. This underscores the urgency of addressing this pervasive issue to prevent further loss of life and property damage.

Emphasizing the tendency for accidents to go unnoticed underscores the need for improved detection and intervention mechanisms. In short, there is a critical need for a comprehensive solution to combat drunk driving and undetected accidents on our roads.

1.4 OBJECTIVE

The primary objective of this project is to develop and implement a comprehensive vehicle safety system equipped with advanced alcohol detection capabilities and accident detection functionalities. By embedding various functionalities within vehicles, the system aims to address the dual challenges of drunk driving and undetected accidents on our roadways.

The foremost goal is to accurately detect the presence of alcohol in the driver's breath and automatically immobilize the vehicle's engine if the alcohol concentration exceeds legal limits, thereby preventing accidents caused by impaired driving. Additionally, the system will prioritize fast and accurate detection of crashes, enabling timely intervention to mitigate the severity of injuries and minimize potential fatalities.

An integral aspect of the project is to establish seamless communication channels for notifying relevant authorities or loved ones in the event of a detected crash, facilitating prompt assistance and emergency response efforts.

Ultimately, the objective of this device is to safeguard the lives of vehicle occupants and other road users by proactively addressing the root causes of road accidents associated with drunk driving and undetected collisions.

1.5 SCOPE OF THE PROJECT

This project aims to utilize electronic devices, particularly mobile phones equipped with accelerometer and GPS sensors, to mitigate casualties resulting from the lack of timely medical assistance during accidents. In addition to detecting accidents, these devices will promptly transmit the exact location to both emergency services and pre-designated contacts, such as friends and relatives stored by the user. The scope of this project encompasses the development and implementation of a system capable of automatically alerting emergency responders and personal contacts simultaneously in the event of a road accident.

CHAPTER: 2
LITERATURE REVIEW

Here is a literature survey on alcohol detection with engine locking system using GPS and Wi-Fi module:

By Ramu S, Dinesh R, Naveen S
Published in: USRD-International Journal for Scientific Research & Development-2018
This system presents the detection of alcohol using alcohol detector connected to Printed circuit board (PCB) such that when the level of alcohol crosses a permissible limit, the vehicle ignition system will turn off. Alcohol Detector in Car project is designed for the safety of the people seating inside the car. Alcohol breath analyzer project should be installed inside the vehicle.

By Sandeep G, Tejaswink, Shweta Hk, Nagaraj, Proff. Yathish Babu Am
Published in: International Journal of Creative Research Thoughts (JCRT)-2022
This project presents the design and implementation of an Alcohol Detection with Engine Locking for cars using the Ultrasonic Sensor and Arduino UNO. The system will continuously monitor level of alcohol concentration in alcohol detection sensor and thus turn off the engine of vehicle if the alcohol concentration is above threshold level. The project provides an efficient solution to control accidents due to drunk driving.

By Dr. Pavan Shukla, Utkarsh Srivastava, Sridhar Singh, Rishabh Tripathi, Rakesh Raushan Sharma
Published in: International Journal of Engineering Research & Technology (IJERT) 2020
This project presents the design and implementation of an Alcohol Detection with Engine locking for cars using the Ultrasonic Sensor and Arduino UNO as the MCU (Master Control Unit). The system will continuously monitor level of alcohol concentration in alcohol detection sensor and thus turn off the engine of vehicle if the alcohol concentration is above threshold level. The model will also send the message of whereabouts of the vehicle through SIM900A.

By Ravi Shankar, P. Naga Surendra, P. Neelima, M. Siddhartha
Published in: International Conference on Recent Trends in Engineering & Technology-2023 (ICRTET-3)
This paper presents a system in which the alcohol is detected by alcohol sensor in the vehicle. If the sensor sense that driver is drunk, it sends the signals to the Arduino board. Then Arduino board sends the signals to the GPS. The GPS sends a location to pre-registered mobile. Led is indicated red and then the engine will be locked. In this way pre-registered person know the exact location of the vehicle through google maps

By Aniket Vilas, Prof. S. R. Jagtap.
Published in: 2014 International Conference on communication and Signal
Processing. GPS. The GPS sends a location to pre-registered mobile. Led is indicated red and then the engine will be locked. In this way pre-registered person know the exact

location of the vehicle through google maps

By Aniket Vilas, Prof. S. R. Jagtap.
Published in: 2014 International Conference on communication and Signal Processing.
This paper presents a system which gives an idea about what can be done to provide medical help and other facilities after the accident as soon as possible. A flex sensor and accelerometer can be used to detect an accident, while the location of the accident will be told to desired persons, such as the nearest hospital, police, and owner of the vehicle through SMS sent using GSM modem containing coordinates obtained from GPS along with the time of the accident and vehicle number. The camera located inside the vehicle will transmit real-time video to see the current situation of passengers. inside the vehicle. Thus this paper emphasizes the post-accident system for detecting and informing about it. Simulation result on the hyper terminal is also presented in this paper.

By Nicky Kattukkaran et. Al
Published in: 2017 International Conference on Computer Communication and Informatics (ICCCI)
This system aims to alert the nearby centre about the accident to supply medical care. The attached accelerometer within the vehicle senses the lean of the vehicle and therefore the heartbeat sensor on the user's body senses the abnormality of the heartbeat to know the seriousness of the accident. Thus the systems will make the choice and send the knowledge to the smartphone, connected to the accelerometer and heartbeat sensor, through Bluetooth. The Android application on the mobile phone will be sent a text message to the nearest medical center and friends. The application also shares the exact location of the accident that can save time.

By Bhandari Prachi, Dalvi Kasturi, Chopade Priyanka,
Published in: 2014 International Journal of Science & Technology Research
This paper presents a system with GPS and GSM module in the concerned 2/3 vehicle which will send the location of the accident to the main server and an ambulance will rush from a nearest hospital to the accident spot. Along with this there would be control of traffic light signals in the path of the ambulance using RF communication. This will minimize the time of ambulance to reach the hospital. A patient monitoring system in the ambulance will send the vital parameters of the patient to the concerned hospital.

CHAPTER: 3

THEORY AND DESIGN

3.1 BLOCK DIAGRAM

This block diagram represents the flow of information and control in an Embedded-system based monitoring and detection system for alcohol consumption and accidents.

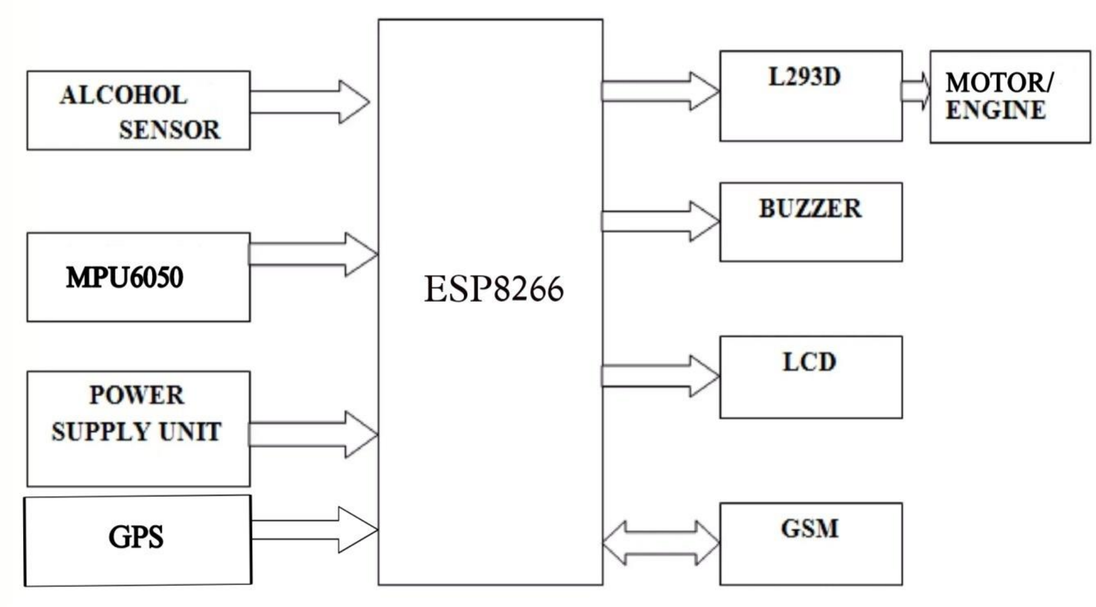


Fig 3.1 Block diagram explaining the working of the circuit

The block diagram illustrates the major components used in our project.

- The ESP8266 microcontroller (MCU) acts as the main control unit of the system. It is responsible for handling the overall operation and communication with other components. The ESP8266 MCU is known for its built-in Wi-Fi capabilities, which can be utilized for connecting to the internet. It collects the data given by the sensors and forwards it to the GSM SIM 900A Module.
- This project includes usage of sensors namely: **MPU6050** and **MQ3**
- The MPU 6050 gyro sensor is an integrated motion sensor that combines a gyroscope and an accelerometer. It is used to detect any tilting or movement of the vehicle. The gyroscope measures angular velocity, while the accelerometer measures acceleration.
- The MQ3 gas sensor is alcohol sensor which is used to detect the alcohol concentration on your breath. This sensor provides an analog resistive output based on alcohol concentration.
- The GSM SIM900A module provides GSM network connectivity to the system. It enables the system to send notifications or alerts to the user.
- GPS provides the system with the ability to realize the location related information.

3.2 WORKING

Our system comprises two integral components **Alcohol Sensing and Engine Locking**, alongside **Accident Detection**. Elaborating on each part:

1. Alcohol Sensing and Engine Locking Operation:

In this segment of the project, we employ an alcohol sensor namely MQ3 to ascertain the concentration of alcohol in the breath of the vehicle's operator. The data collected by the alcohol sensor is relayed to the NodeMCU microcontroller, serving as the central processing unit of our system. If the alcohol concentration exceeds the legal or predetermined limit, the engine is promptly immobilized. In practical automotive applications, this would entail the microcontroller controlling the ignition coil. However, for demonstrative purposes, in our project model, a battery-operated DC motor is utilized to simulate engine locking. The motor's activation and deactivation are managed by a motor driver.

2. Accident Detection:

In the event of an accident, regardless of the cause, our system relies on the MPU6050 sensor. This sensor integrates both gyroscope and accelerometer functionalities, enabling it to detect any significant changes in the vehicle's angular position or acceleration indicative of a collision. Upon detecting an accident, pertinent data is transmitted to the ESP8266 microcontroller. Subsequently, the microcontroller leverages the GPS module to pinpoint the precise location of the crash. Once the location data is obtained, the system dispatches pertinent notifications to predefined recipients, be they authorities or designated individuals, utilizing the GSM module. These notifications include comprehensive details regarding the accident's location.

In this manner, our system not only reduces accidents resulting from alcohol consumption but also ensures swift response and dissemination of critical information in the event of any unforeseen mishaps. By adhering to these objectives, we strive to minimize casualties and enhance overall road safety.

3.3 COMPONENTS USED

Components used while designing the device are given as follows:

- Node MCU ESP8266
- GSM SIM 900A
- MPU6050 Sensor
- MQ3 Sensor
- NEO 6M GPS
- LCD Module
- Buzzer
- LM7805 Voltage Regulator IC
- L293 Motor Driver

3.3.1 ESP 8266 Microcontroller:

The Node MCU (Node Microcontroller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK. The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibratedRF allowing it to work under all operation conditions, and requires no external RF parts.

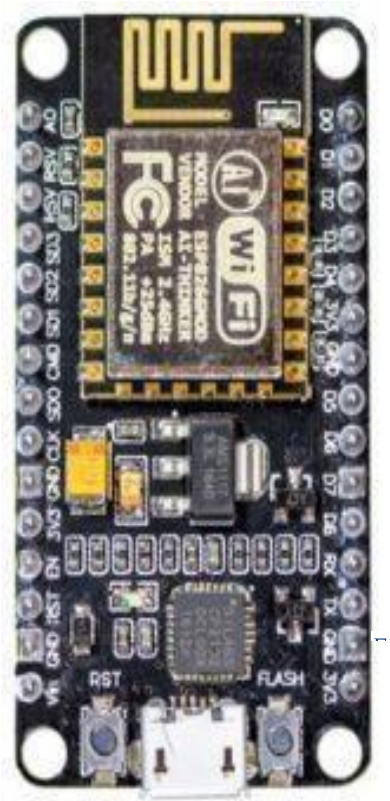


Fig 3.2 NodeMCU ESP 8266

3.3.1.1 Specifications:

The NodeMCU is available in various package styles. Common to all the designs is the base ESP8266 core. Designs based on the architecture have maintained the standard 30-pin layout. Some designs use the more common narrow (0.9″) footprint, while others usea wide (1.1″) footprint.

The most common models of the NodeMCU are the Amica (based on the standard narrow pin-spacing) and the LoLin which has the wider pin spacing and larger board. The open-source design of the base ESP8266 enables the market to design new variants of the NodeMCU continually.

3.3.1.2 Technical Specifications:

Table 3.1: Technical Specifications of NodeMCU ESP8266

SPECIFICATIONS	**OfficialNodeMCU**
Microcontroller	ESP 8266 32-bit
NodeMCU Model	Amica
NodeMCU Size	49mm x 26 mm
Carrier Board Size	n/a
Pin Spacing	22.86 mm
Clock Speed	80 MHz
USB to Serial	CP2102
USB Connector	Micro USB
Operating Voltage	3.3V
Input Voltage	4.5V-10V
Flash Memory/SRAM	4 MB / 64 KB
Digital I/O Pins	11
Analog In Pins	1
ADC Range	0 – 0.3V
UART/SPI/I2C	1 / 1 / 1
Wi-Fi Built-in	802.11 b/g/n
Temperature Range	-40C – 125C

3.3.1.3 NodeMCU Pinout and Functions:

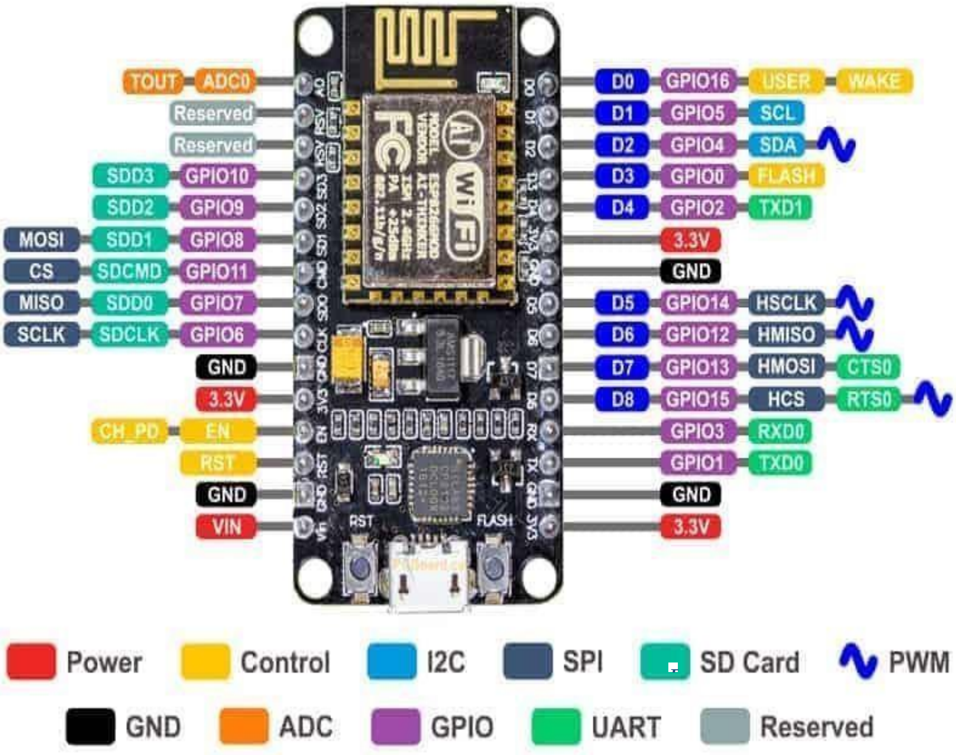


Fig 3.3 NodeMCU Pinout

- Power Pins: There are four power pins. VIN pin and three 3.3V pins.

- VIN can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the NodeMCU module – we can also supply 5V regulated to the VIN pin

- 3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components.

- GND are the ground pins of NodeMCU/ESP8266.

- I2C Pins are used to connect I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

- GPIO Pins: NodeMCU/ESP8266 has 17 GPIO pins which can be assigned to functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

- ADC Channel: The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

- UART Pins: NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) canbe used for communication. However, UART1 (TXD1 pin) features only data transmit signal it is usually used for printing log.

- SPI Pins: NodeMCU/ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the various general-purpose SPI features, like timing modes of the SPI format transfer, upto 80 MHz and the divided clocks of 80 MHzUp to 64-Byte FIFO.

- SDIO Pins: NodeMCU/ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and4-bit 50 MHz SDIO v2.0 are supported.

- PWM Pins: The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs (100 Hz and 1 kHz).

- Control Pins: These are used to control the NodeMCU/ESP8266. These pins include ChipEnable pin (EN), Reset pin (RST) and WAKE pin.
  - EN: The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
  - RST: RST pin is used to reset the ESP8266 chip.
  - WAKE: Wake pin is used to wake the chip from deep-sleep.

3.3.2 GSM SIM 900A MODULE:

A GSM module, also known as a GSM modem or GSM/GPRS module, is a hardware device that allows communication over the Global System for Mobile Communications (GSM) network. It enables devices to send and receive data, voice, and SMS (Short Message Service) messages over cellular networks.

GSM modules are commonly used in various applications that require wireless communication, including remote monitoring, IoT devices, security systems, vehicle tracking, and industrial automation.

GSM modules typically include a UART (Universal Asynchronous Receiver-Transmitter) interface, which allows them to connect to microcontrollers or other devices. The UART interface facilitates serial communication and enables sending and receiving data between the module and the connected device.

GSM modules require a SIM (Subscriber Identity Module) card to establish a connection with the GSM network. The SIM card contains essential information, such as the mobilenetwork operator, phone number, and authentication credentials. The module has a SIM card slot to insert and securely hold the SIM card.

GSM modules support communication over the GSM network, which is a standard for cellular communication. They operate on different frequency bands, including 2G.

SIM900A GSM Module is the smallest and cheapest module for GPRS/GSM communication. It is common with Arduino and microcontroller (NodeMCU in this case) in most of embedded application. The module offers GPRS/GSM technology for communication with the uses of a mobile SIM. It uses a 900 and 1800MHz frequency band and allows users to receive/send mobile calls and SMS. The keypad and display interface allows the developers to make the customize application with it.

3.3.2.1 SIM900A Pin Configuration:

The Module SIM900A looks like a single chip but it has a bunch of features that can help to build almost many commercial applications. Although, there are a total of 68 pins on SIM900A and using these pins helps to build the applications.

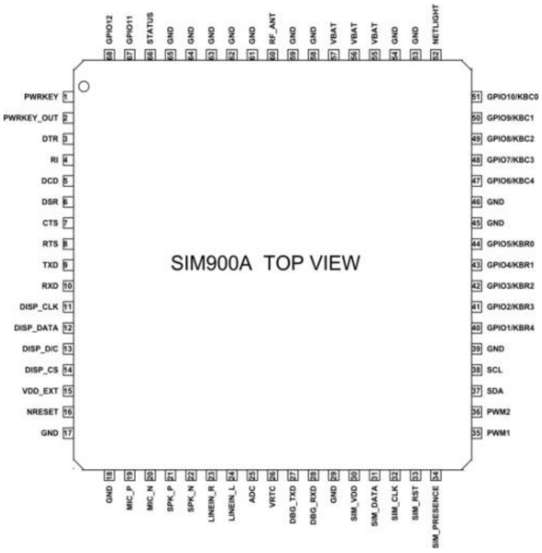


Fig 3.4 Pin Diagram of GSM SIM 900A module

3.3.2.2 SIM900A GSM Module Block Diagram:

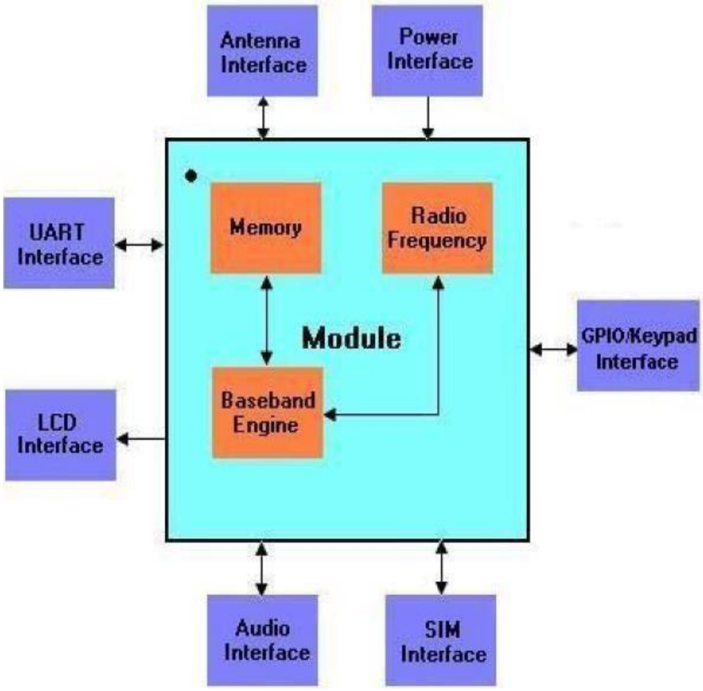


Fig 3.5 Block diagram of GSM SIM 900A Module

3.3.2.3 GSM SIM 900A Module Features:

Table 3.2: Specifications of GSM SIM 900A Module

FEATURES	DETAIL
Power Input	3.4V to 4.5V
Operating Frequency	EGSM900 and DCS1800
Transmitting Power Range	2V for EGSM900 and 1W for DCS1800
Data Transfer Link	Download: 85.6kbps, Upload:42.8kbps
SMS	MT, MO, CB, Text and PDU mode.
Antenna Support	Available
Audio Input/Output	Available
Serial Port	I2C and UART
Serial Debug Port	Available

3.3.3 MPU 6050 GYRO SENSOR:

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3- axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package.

It has Auxiliary I2C bus to communicate with other sensor devices.

The MPU-6050 uses micro-electromechanical systems technology to measure angular velocity (rotational motion) and linear acceleration along three axes. It communicates with other devices, such as microcontrollers or computers, using protocolslike I2C (Inter-Integrated Circuit) or SPI (Serial Peripheral Interface).

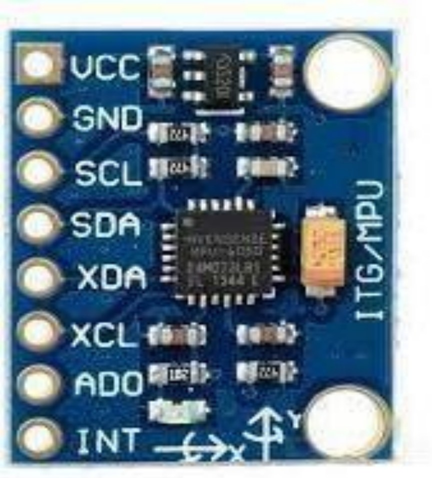


Fig 3.6 MPU6050 Gyro Sensor

3.3.3.1-Axis Gyroscope:

The MPU-6050 integrates a 3-axis gyroscope, which measures rotational motion in three dimensions. The gyroscope provides angular velocity data around the X, Y, and Z axes, allowing for the detection and tracking of changes in orientation and rotational movement. This information is valuable in applications such as motion sensing, stabilization algorithms, and motion controlled devices, where the gyroscope data can be used to precisely monitor and respond to rotational changes in real-time.

It works in a procedure given below:

- When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.
- The resulting signal is amplified, demodulated, and filtered to produce a voltage that isproportional to the angular rate.
- This voltage is digitized using 16-bit ADC to sample each axis.
- The full-scale range of output is +/- 250, +/- 500, +/- 1000, +/- 2000.
- It measures the angular velocity along each axis in degree per second unit.

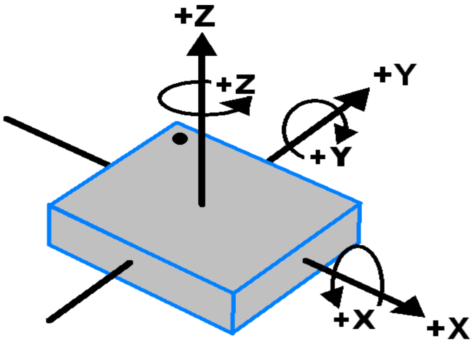


Fig 3.7 Gyroscope Orientation and Polarity Rotation

3.3.3.2-Axis Accelerometer:

The MPU-6050 also incorporates a 3-axis accelerometer, which measures linear acceleration along the X, Y, and Z axes. This accelerometer enables the detection of changes in velocity and acceleration in three-dimensional space. By monitoring acceleration in different directions, the MPU-6050 can provide information about the device's movement, tilt, and orientation. This data is commonly used in applications such as motion tracking, gesture recognition, and balancing systems, where precise acceleration measurements are crucial for accurately interpreting and responding to physical movements.

3.3.3.3 MPU 6050 Module Pinout:

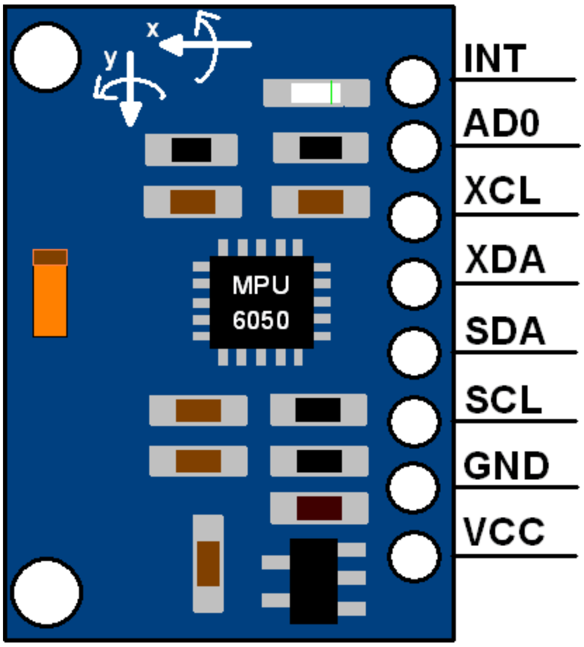


Fig 3.8 Pinout Diagram of MPU 6050 Gyro Sensor

Pin Description:

- INT: Interrupt digital output pin.
- AD0: I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.
- XCL: Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.
- XDA: Auxiliary Serial Data pin. This pin is used to connect other I2C interface.
- SCL: Serial Clock pin. Connect this pin to microcontrollers SCL pin
- SDA: Serial Data pin. Connect this pin to microcontrollers SDA pin.
- GND: Ground pin. Connect this pin to ground connection.
- VCC: Power supply pin. Connect this pin to +5V DC supply.
- MPU-6050 module has Slave address (When AD0 = 0, i.e., it is not connected to VCC) . Slave Write address: (SLA+W): 0xD0

3.3.4 Alcohol Sensor:

The alcohol sensor named MQ-3, detects ethanol in the air. Typically, it is used as a part of the breathalyzers or breath testers for the detection of ethanol in the human breath.
It is a low cost semiconductor sensor which can detect the presence of alcohol gases at concentrations from 0.05 mg/L to 10 mg/L. The sensitive material used for this sensor is SnO2, whose conductivity is lower in clean air. It's conductivity increases as the concentration of alcohol gases increases. It has high sensitivity to alcohol and has a good resistance to disturbances due to smoke, vapor and gasoline. This module provides both digital and analog outputs. MQ3 alcohol sensor module can be easily interfaced with microcontrollers, Arduino Boards, Raspberry Pi etc.

This alcohol sensor is suitable for detecting alcohol concentration on a person's breath. It has a high sensitivity and fast response time. Sensor provides an analog resistive output based on alcohol concentration. The drive circuit is very simple; all it needs is one resistor. A simple interface could be a 0-3.3V ADC.

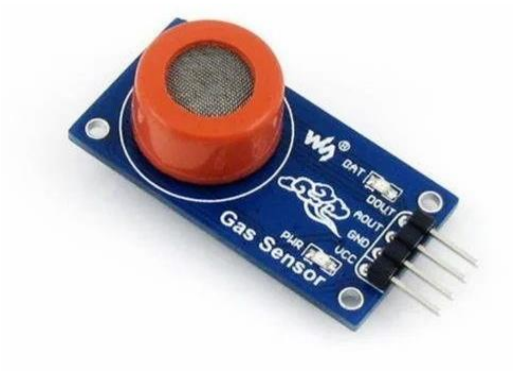


Fig 3.9 MQ3 alcohol sensor

3.3.4.1 Technical Data

- Concentration : 0.05 mg/L ~ 10 mg/L Alcohol
- Operating Voltage : 5V ±0.1
- Current Consumption : 150mA
- Operation Temperature : -10°C ~ 70°C

3.3.4.2 Pin OUT

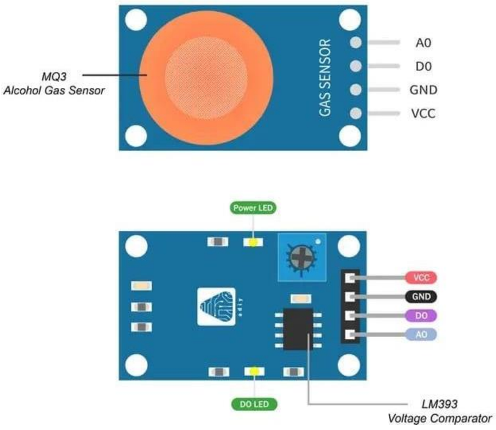


Fig 3.10 Pin out of MQ3

- VCC – Input Power Supply
- GND – Supply Ground
- DO – Digital Output
- AO – Analog Output

3.3.4.3 Ethanol

Ethanol is also called alcohol, ethyl alcohol and drinking alcohol is chemical compound simple alcohol with chemical formula C2HSOH. Its formulae can be written also as C2H5-OH, and is often abbreviated as EtOH. Ethanol is a volatile, flammable, colourless liquid with a slightcharacteristic odour.

Ethanol is naturally produced by the fermentation of sugars by yeasts or via petrochemical process and is most commonly consumed as a applications as an antiseptic and disinfectant.

3.3.5 NEO-6M GPS

The NEO-6M module is designed for various positioning and navigation applications. It's commonly used in projects where accurate positioning information is needed, such as in drones, robotics, vehicle tracking systems, and location-based services.

3.3.5.1 Pinout

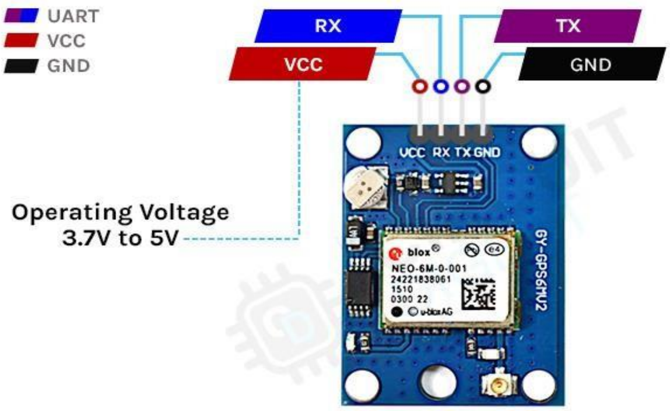


Fig 3.11 Pinout of NEO 6M GPS

- GND is the ground pin of the GPS Module and it should be connected to the ground pin of theESP32.
- TX is the transmit pin of the GPS module that needs to connect to the RX pin of the ESP32.
- RX is the receive pin of the GPS module that needs to connect to the TX pin of the ESP32.
- VCC is the power pin of the GPS module and needs to connect to the 3.3V of your pin.

3.3.5.2 NEO-6M GPS Module – Parts

The NEO-6M module is a ready-to-use GSM module that can be used in many different applications. The NEO-6M GPS module has five major parts on the board, the first major part is the NEO-6M GPS chip in the heart of the PCB. Next, we have a rechargeable battery and a serial EEPROM module. An EEPROM together with a battery helps retain the clock data, latest position data (GNSS orbit data), and module configuration but it's not meant for permanent data storage. Without the battery, the GPS always cold-starts so the initial GPS lock takes more time. The battery is automatically charged when power is applied and maintains data for up to two weeks without power. Next, we have our LDO, because of the onboard LDO, the module can be powered from a 5V supply. Finally, we have our UFL connector where we need to connect an external antenna for the GPS to properly work. The parts on the NEO-6M GPS module are shown below-

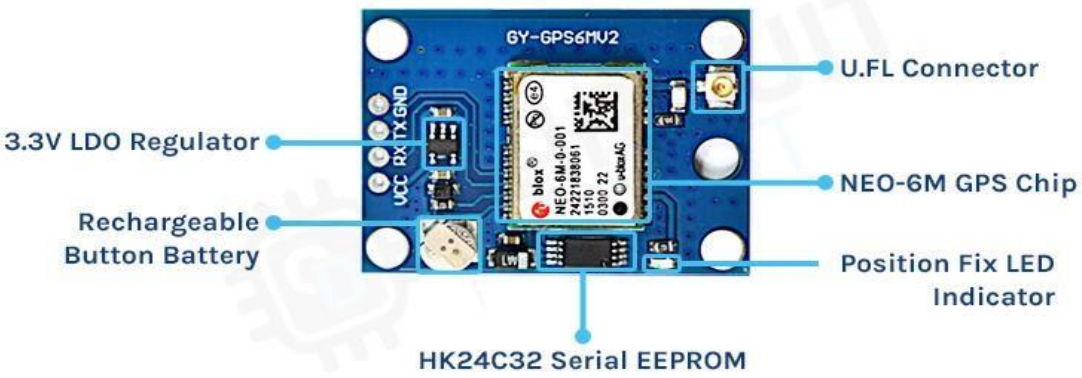


Fig 3.12 Parts of NEO-6M Module

3.3.5.3 Overview of the NEO-6M GPS Module

The Global Positioning System (GPS) is a system consisting of 31 satellites orbiting earth. We can know their exact location because they are constantly transmitting position information with time through radio signals. At the heart of the breakout board, there is the NEO-6M GPS module that is designed and developed by u-blox. This is very small but it packs a lot of features. It can track up to 22 satellites over 50 channels while consuming only 45mA of current and has an operating voltage of 2.7V ~ 3.6V. One of the most interesting features of this module is its power-saving mode, which allows a reduction in system power consumption. With power-saving mode on, the current consumption of the module reduces to 11MA only.

3.3.5.4 Position Fix LED Indicator

The NEO-6M GPS module board, has a small LED (Light Emitting Diode) which is used to indicate that the GPS module is able to communicate with the satellites.

- No blinking – it is searching for satellites.
- Blink every second– Position Fix is found (the module can see enough satellites).

3.3.5.5 Antenna

The module comes with a -161 dBm sensitive patch antenna that can receive radio signals from GPS satellites. The antenna can be connected to a small UFL connector.

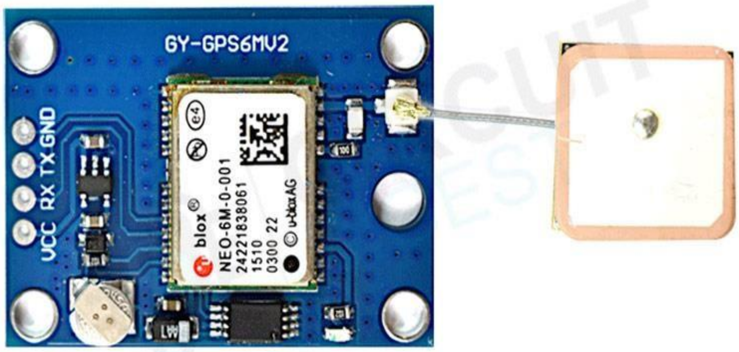


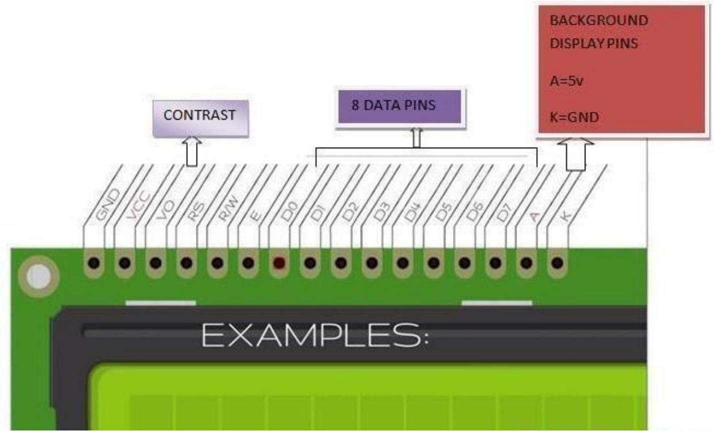
Fig 3.13 NEO-6M Module with its antenna

3.3.6 LCD MODULE

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendliness.

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have(16×2=32) 32 characters in total and each character will be made of 5×8 pixel dots.

Each character has (5×8=40) 40 Pixels and for 32 Characters we will have (32×40) 1280 Pixels. Further, the LCD should also be instructed about the Position of the Pixels. Hence it will be a hectic task to handle everything with the help of MCU, hence an Interface IC like HD44780 is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the Commands and Data from the MCU and process



.

Figure-3.14 Pin description of LCD

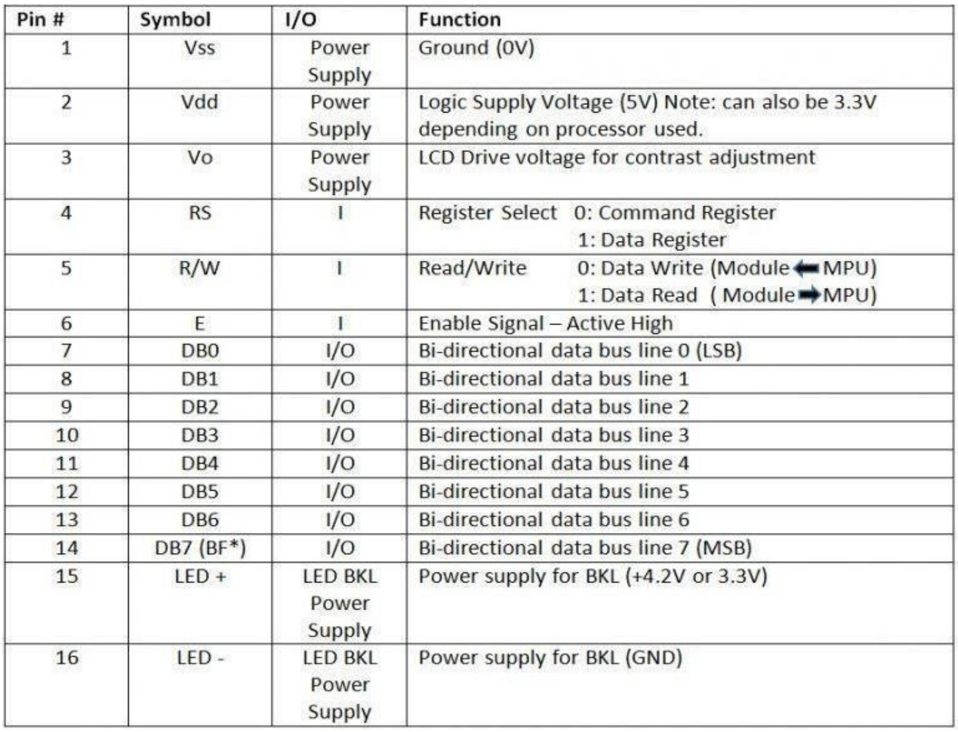
Register configuration of LCD is given:

- RS (Register select): A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

- Command Register: The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. Processing for commands happens in the command register.

- Data Register: The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected.

Table.3.3: 8 bit parallel in description



Pin #	Symbol	I/O	Function
1	Vss	Power Supply	Ground (0V)
2	Vdd	Power Supply	Logic Supply Voltage (5V) Note: can also be 3.3V depending on processor used.
3	Vo	Power Supply	LCD Drive voltage for contrast adjustment
4	RS	I	Register Select 0: Command Register 1: Data Register
5	R/W	I	Read/Write 0: Data Write (Module⬅MPU) 1: Data Read (Module➡MPU)
6	E	I	Enable Signal – Active High
7	DB0	I/O	Bi-directional data bus line 0 (LSB)
8	DB1	I/O	Bi-directional data bus line 1
9	DB2	I/O	Bi-directional data bus line 2
10	DB3	I/O	Bi-directional data bus line 3
11	DB4	I/O	Bi-directional data bus line 4
12	DB5	I/O	Bi-directional data bus line 5
13	DB6	I/O	Bi-directional data bus line 6
14	DB7 (BF*)	I/O	Bi-directional data bus line 7 (MSB)
15	LED +	LED BKL Power Supply	Power supply for BKL (+4.2V or 3.3V)
16	LED -	LED BKL Power Supply	Power supply for BKL (GND)

3.3.7 BUZZER

The buzzer is a piezo circuit that produces a sharpsound. When provided with a suitable amount of voltage. When a voltage is being applied across the poles the metal plate vibrates due to the electromagnetic effect. Thus the vibration of metal plate causes a sound.

Fig 3.15 A piezo electric buzzer

3.3.8 7805 Voltage Regulator:

The 7805 is a popular linear voltage regulator integrated circuit (IC) that provides a constant, regulated output voltage of +5 volts. It accepts an unregulated input voltage typically between 7V and 35V and effectively regulates it to a stable +5V output. The 7805 IC incorporates a voltage reference, error amplifier, and power transistor to maintain a constant output voltage, even when the input voltage or load conditions fluctuate. It is commonly used in electronic circuits to provide a reliable and stable power supply for microcontrollers, digital logic circuits, and various low-power electronic devices.

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 Voltage Regulator, a member of the 78xx seriesof fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC).The xx in 78xx indicates the output voltage it provides.

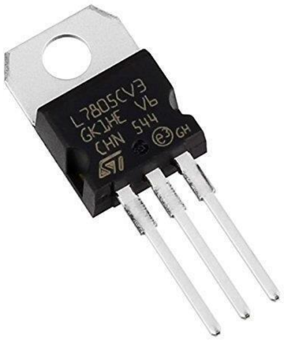


Fig 3.16 7805 Voltage Regulator

3.3.8.1 7805 Voltage Regulator Pinout:

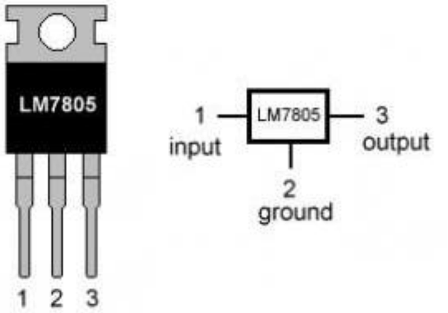


Fig 3.17 Pinout Diagram of 7805 Voltage Regulator

Table 3.4 Description of Pins of 7805 Voltage Regulator

Pin	**Function**	**Description**
INPUT	Input voltage (7V-35V)	In this pin of the IC positive unregulated voltage is given in the regulation.
GROUND	Ground (0V)	In this pin where the ground is given. This pin is neutral for equally the input and output.
OUTPUT	Regulated output; 5V (4.8V-5.2V)	The output of the regulated 5V is taken out at this pin of the IC regulator.

There is a significant difference between the input voltage & the output voltage of the voltage regulator. This difference between the input and output voltage is released as heat. The greater the difference between the input and output voltage, more the heat is generated.

If the regulator does not have a heat sink to dissipate this heat, it can get destroyed and malfunction. Hence, the voltage is limited to a maximum of 2-3 volts above the output voltage.

We now have 2 options. Either we can design our circuit so that the input voltage going into the regulator is limitedto 2-3 volts above the output regulated voltage or place an appropriate heat sink, that can efficiently dissipate heat.

3.3.8.2 Capacitors with 7805 Voltage Regulator:

If the voltage regulator is situated more than 25cm (10 inches) from the power supply, capacitors are needed to filter residual AC noise. Voltage regulators work efficiently on a clean DC signal being fed. The bypass capacitorshelp reduce AC ripple.

Essentially, they short AC noise from the voltage signal and allow only DC voltage into the regulator.

Capacitors will be beneficial in this case as they are good at maximizing voltage regulation. The values of capacitors can also be changed slightly.

3.3.9 L293D DUAL H-Bridge Motor Driver IC:

L293D motor Driver IC is an integrated circuit that can drive two motors simultaneously and is usually used to control the motors in an autonomous system. This motor driver IC enables us to drive a DC motor in either direction and also control the speed of the motor. L293D is a dual H-bridge motor driver IC. H-bridge is the simplest circuit for controlling a low current- rated motor. One H-bridge is capable to drive a DC motor bidirectional. L293D is a current enhancing IC. It can also act as a switching device. The L293D is a 16-pin Integrated circuit, with eight pins, on each side, dedicated to the controlling of amotor. There are 2 input pins, 2 output pins and 1 enable pin for each motor. The L293D IC is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. It is designed to drive inductive loads such as relays, solenoids, DC & bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. L293D motor Driver IC is one of the most popular drivers in the market. Because of several reasons such as cheap price (compared to other drivers), easy control, proper shape and size, no need for protective circuit and diodes, no need for heat sinks, and good resistance to temperature and high-speed variations, L293D motor driver is mostly preferred driver to the user.

3.3.9.1 Features and Characteristics of L293D IC

The followings are the main features and specifications of the L293D Motor Driver IC.
Wide supply voltage range: 4.5V to 36V

- Output current 600mA per channel
- Peak Output Current 1.2A per channel
- High-Noise-Immunity Inputs
- Separate Input-Logic Supply
- Internal Electrostatic Discharge (ESD) Protection

3.3.9.2 L293D Pin Configuration & Description:

The pin configuration diagram and real image of the L293D Motor Driver IC are shown below:

- Pin 1 (Enable1,2): When this pin is given High or Logic 1, the left part of the IC will work and when it is low, the left part doesn’t work. So, this pin is the Master Control pin for the left part of the IC.

- Pin 2 (Input 1): When this pin is High or Logic 1, output 1 becomes high. i.e. the current will flowthrough output 1
- Pin 3 (Output 1): This pin is connected to one of the terminal of motor 1
- Pin 4, Pin 5 (GND): These pins should be connected to the circuit's ground
- Pin 6 (Output 2): This pin is connected to one of the terminal of motor 1
- Pin 7 (Input 2): When this pin is given High or Logic 1, output 2 becomes high i.e. the current willflow through output 2.
- Pin 8 (VCC 2): This is the voltage required to run the motor. It can be greater than IC voltage Vcc 1. If we are driving 12 V DC motors then make sure that this pin is supplied with 12 V .
- Pin 9 (Enable 3, 4): When this pin is given High or Logic 1, the right part of the IC will work, and when it is low the right part doesn't work. So, this pin is the Master Control pin for the right part of the IC.
- Pin 10 (Input 3): When this pin is given High or Logic 1, output 3 becomes high i.e; the current will flow through output 3.
- Pin 11 (Output 3): This pin is connected to one of the terminal of motor 2
- Pin 12,13 (GND): These pins should be connected to the circuit's ground
- Pin 14 (Output 4): This pin is connected to one of the terminals of motor 2
- Pin 15 (Input 4): When this pin is given High or Logical 1, output 4 becomes High i.e. the current will flow through output 4.
- Pin 16 (VCC1): This pin provides power to the IC. So, this pin should be supplied with a 5V.

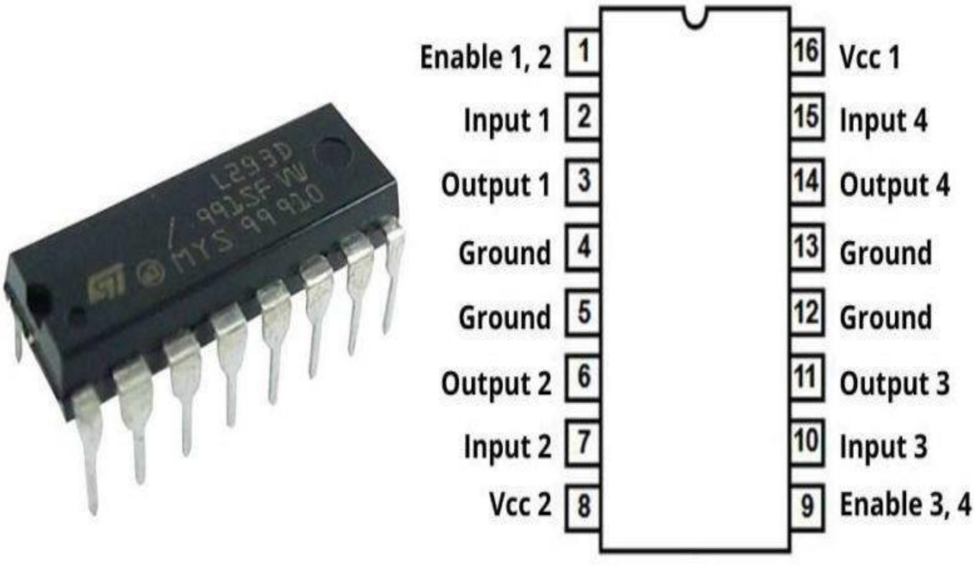


Fig 3.18 Pin Configuration of L293D

Table 3.5 Output of motor driver for different inputs.

Input 1	Input 2	Enable 1,2	Result
0	0	1	Stop
0	1	1	Anti- clockwise rotation
1	0	1	Clockwise rotation
1	1	1	Stop
0	1	50% duty cycle	Anti- clockwise rotation with half speed
1	0	50% duty cycle	Clockwise rotation with half speed

3.3.10 DC MOTOR

DC motors are configured in many types and sizes, including brush less,servo, and gear motor types. A motor consists of a rotor and a permanent magnetic field stator. The magnetic field is maintained using either permanent magnets or electromagnetic windings. DC motors are most commonly used in variable speed and torque.

Motors are the devices that provide the actual speed and torque in a drive system. This family includes AC motor types (single and multiphase motors, universal, servo motors, induction, synchronous, and gear motor) and DC motors (brush less, servo motor, and gear motor) as well as linear, stepper and air motors, and motor contactors and starters.

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field,it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. Opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

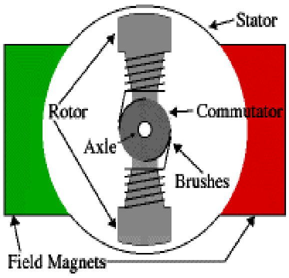


Fig 3.19 A DC motor

In the figure a simple 2-pole DC electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).motors (brush less, servo motor, and gear motor) as well as linear, stepper and air motors, and motor contactors and starters.

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that Beamers will see), the external magnetic field is produced by high-strength permanent magnets. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnet.

3.4 CIRCUIT AND WORKING

The diagram explaining the circuit of the project is given below:

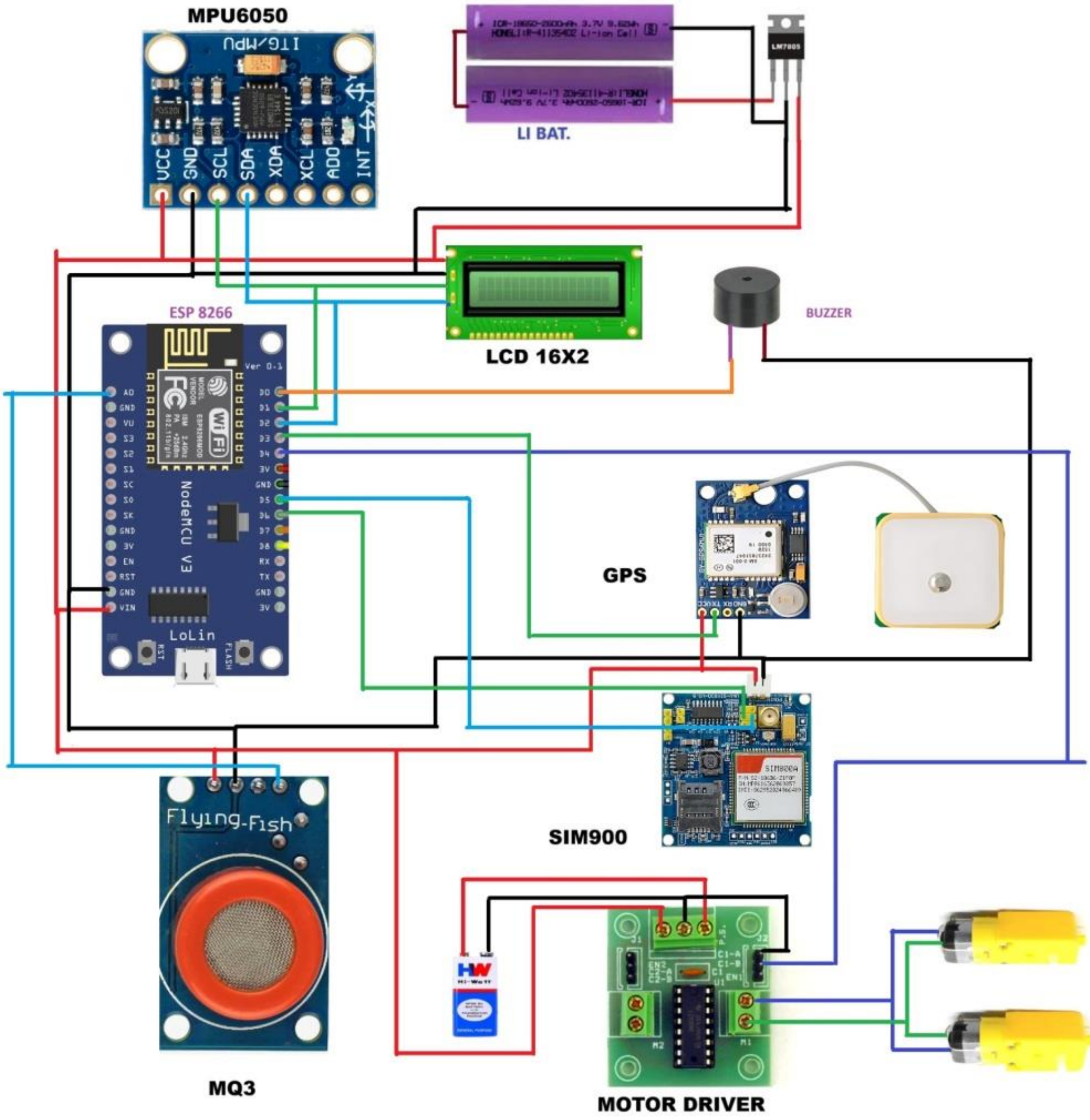


Fig 3.20 Circuit diagram for Vehicle Safety System for Alcohol Detection and Accident Awareness

The circuit design for the project "Vehicle Safety System for Alcohol Detection and Accident Awareness” which can be implemented as follows:

3.4.1 Components:

- Node MCU ESP8266
- GSM SIM 900A
- MPU6050 Gyro Sensor
- MQ3 Alcohol Sensor
- NEO 6M GPS
- LCD Module
- Buzzer
- LM7805 Voltage Regulator IC
- L293 Motor Driver
- 18650 Li-Po Batteries (or a suitable Li-Po battery pack
- Other required passive components (resistors, capacitors, etc.)

3.4.2 Circuit connections and working:

1. **Li-Po Batteries**:

In the proposed circuit configuration, power will be provided by a series connection of two lithium-ion batteries, each boasting a nominal voltage of 3.7 volts. Cumulatively, these batteries deliver approximately 8 volts. Li-Po batteries, such as 18650 cells or a Li-Po battery pack, are used as the power source for our circuit. Li-Po batteries are commonly used in portable and IoT applications due to their high energy density and rechargeable nature.

Connect the positive terminal of the Li-Po batteries to the input pin of the 7805 voltage regulator. The positive terminal is usually denoted by the "+" sign or a red wire.

Connect the negative terminal of the Li-Po batteries to the ground (GND) pin of the voltage regulator. The negative terminal is usually denoted by the "-" sign or a black wire.

2. **Voltage Regulators**:

To regulate this voltage to the required 5 volts for the circuit, a three-pin LM7805 voltage regulator integrated circuit (IC) will be employed. This IC features an input pin for receiving the incoming DC voltage, a ground pin for establishing the reference ground, and an output pin to supply the positive five volts. The 7805 is a popular linear voltage regulator that converts the input voltage to a stable 5V output voltage required by the components in the circuit.

Connect a 2200μF capacitor between the output pin (Vout) and ground (GND) pins of the voltage regulator. This capacitor helps stabilize the output voltage by filtering out any high-frequency noise. The regulated 5V output (Vout) of the voltage regulator is connected to the VCC pins of the components that require 5V power.

3. **Ground:**

Connect the GND pins of all the components together, including the ESP8266, GSM module, MQ3 sensor, MPU6050 sensor, and the voltage regulator. Ground serves as a reference point for all electrical signals in the circuit, ensuring a common voltage reference for all components.

4. **ESP 8266 Microcontroller:**

- Connect the VCC pin of the ESP8266 to the regulated 5V output from the voltage regulator.
- Connect the GND pin of the ESP8266 to the ground.
- Connect the Tx pin of the ESP8266 to the Rx pin of the GSM SIM900A module.

- Connect the Rx pin of the ESP8266 to the Tx pin of the GPS NEO-6M module.

5. MPU6050 and LCD

The MPU6050 sensor and the LCD display operate on the I2C protocol;
I2C stands for Inter-Integrated Circuit. It is a bus interface connection protocol incorporated into devices for serial communication. It is a widely used protocol for short-distance communication. It is also known as Two Wired Interface (TWI).

Working of I2C Communication Protocol:
It uses only 2 bi-directional open-drain lines for data communication called SDA and SCL. Both these lines are pulled high.

- Serial Data (SDA) – Transfer of data takes place through this pin.
- Serial Clock (SCL) – It carries the clock signal.

SDA serves as the conduit for transmitting data between the master and slave devices, while SCL carries the clock signal, synchronizing communication between devices bit by bit.
Each of the MPU6050 and LCD modules' SCL and SDA pins will be individually connected to the corresponding SCL and SDA pins of the ESP8266 microcontroller. As previously discussed, D1 will be assigned for the SCL connection, while D2 will facilitate the SDA connection.

6. MQ3

The MQ-3 sensor's output pin generates an analog voltage signal correlated with the alcohol concentration; the higher the alcohol concentration, the greater the voltage output. This analog signal is linked to the A0 pin of the ESP8266 microcontroller, the sole pin equipped for analog-to-digital conversion.

7. GSM SIM 900A

In our system, the GSM module's RX and TX pins play a pivotal role in facilitating bidirectional communication between the GSM module and the ESP8266 microcontroller. Ordinarily, the TX pin of the GSM module would be utilized to transmit data to an external device, such as the ESP8266, upon receiving a message or call. However, in our specific configuration, the TX pin of the GSM module remains inactive, serving no transmitting function.
Instead, the RX pin of the GSM module is connected to a GPIO pin of the ESP8266, which has been designated for transmitting data (coded as TX). Through this configuration, data flows from the ESP8266 microcontroller to the GSM module, enabling seamless communication between the two components. This approach ensures efficient handling of incoming messages or calls by the GSM module, as it receives instructions and relevant data from the ESP8266 microcontroller via the designated RX pin.
Through this arrangement, our system achieves optimal data exchange and coordination, enhancing overall performance and functionality.

8. GPS NEO 6M

The project achieves location transmission capabilities through the integration of a GPS module. Operating on the trilateration principle, the GPS module determines the device's coordinates upon establishing communication with a minimum of three satellites. With the introduction of a third satellite, it is possible to pinpoint your exact location at the place where all three circles intersect.

This process involves the reception of signals from orbiting satellites, enabling the GPS module to calculate its precise position. An LED indicator on the Neo 6M GPS module serves as a visual cue for the status of the position fix: a steady illumination indicates an ongoing search for satellites, while a static LED signifies a successful position fix, ensuring accurate location data retrieval.

To facilitate the transmission of location information, the GPS module's TX pin is connected to a GPIO pin of the ESP8266 microcontroller, which has been configured to receive data (coded as RX). This connection enhances the project's capabilities, enabling seamless communication between the GPS module and the ESP8266 microcontroller. Through this synergy, the project gains the ability to transmit real-time location data, thereby enhancing its utility and effectiveness in various applications.

9. Motor Driver

In this setup, a DC motor serves as a visual indicator to demonstrate the response to the presence of alcohol in the driver's system. The motor is powered by a 9-volt battery and controlled via an L293D motor driver, which interfaces with the NodeMCU microcontroller. The motor driver receives its power from a separate 9-volt battery.

- The NodeMCU microcontroller facilitates motor control by connecting its GPIO pin to the input of the motor driver.
- Both output terminals of the motor driver are linked to the respective positive and negative terminals of the motor.
- The enable pin of the motor driver is connected to the Vcc pin of the NodeMCU.

The direction of motor rotation is dictated by the logic levels applied to the direction controlling pins, denoted as pins A and B.

- If pin A receives a logic 0 and pin B receives a logic 1, the motor rotates in the anticlockwise direction.
- If pin A receives a logic 1 and pin B receives a logic 0, the motor would rotate clockwise direction.
- When both pins receive the same logic level, the motor remains stationary.

In this particular setup, pin B is grounded, resulting in a logic 0 state. Pin A is linked to a GPIO pin of the microcontroller.Thus, if pin A receives a logic 1 from the controller, the motor rotates clockwise; whereas, if pin A receives a logic zero, the motor ceases rotation.

This arrangement enables the motor to visually convey the system's response to alcohol detection, with its direction of rotation indicating the specific condition detected. Such a design ensures clear and intuitive feedback, enhancing user understanding and safety in alcohol detection scenarios.

10. Buzzer

The Vcc pin of the buzzer is connected to one of the GPIO pin of the ESP8266 and the GD pin is connected to the common ground.

3.5 CIRCUIT SIMULATION

The circuit of the proposed system was simulated by the use of proteus, while designing the circuit.

3.5.1 Introduction to Proteus:

542package that provides a virtual environment for designing, simulating, and testing electronic circuits and systems. It offers a comprehensive set of tools for schematic capture, simulation, PCB (Printed Circuit Board) design, and microcontroller simulation. With Proteus, users can create and connect components on a virtual breadboard or schematic diagram, simulate their functionality, analyze circuit behavior, and debug designs before manufacturing the physical prototype. The software supports a wide range of microcontroller families, sensors, actuators, and other electronic components, allowingusers to emulate and verify the behavior of complex embedded systems. Proteus provides an intuitive user interface and extensive libraries of components, making it suitable for both beginners and experienced engineers in various fields such as electronics, robotics, and embedded systems development.

- By using proteus we can make two-dimensional circuits designs as well.
- With the use of this engineering software, you can construct and simulate different electrical and electronic circuits on your personal computers or laptops.
- There are numerous benefits to simulate circuits on proteus before make them practically.
- Designing of circuits on the proteus takes less time than practical construction ofthe circuit.
- The possibility of error is less in software simulation such as loose connection that takes a lot of time to find out connection problems in a practical circuit.
- There is zero possibility of burning and damaging of any electronic component in proteus.
- The electronic tools that are very expensive can easily get in proteus such as an oscilloscope.
- Using proteus you can find different parents of circuits such as current, a voltage value of any component and resistance at any instant which is very difficult in a practical circuit.

3.5.1.1 Features of Proteus:

1. Schematic Capture: Proteus allows users to create circuit schematics using a comprehensive library of electronic components. Users can easily connect components, define their properties, and create complex circuit designs.
2. Simulation: The software provides robust simulation capabilities, allowing users to verify the functionality of their circuits before prototyping. It supports both analog and digital simulations, enabling analysis of circuit behavior, voltage and current waveforms, and various performance parameters.
3. Virtual Testing: Proteus includes virtual testing features, such as virtual instruments and oscilloscopes, which enable users to interact with their circuits and observe real-time responses. This helps in debugging and refining designs without the need for physical hardware.
4. PCB Design: Proteus offers tools for designing professional-quality PCB layouts. Users can transfer their circuit designs to the PCB layout environment, define footprints for components, route traces, and perform design rule checks. It also supports Gerber file generation for manufacturing the physical PCB.
5. Microcontroller Simulation: Proteus includes a microcontroller simulation module that allows users to simulate and debug microcontroller-based projects. Itsupports a wide range of microcontroller families and provides a virtual environment for code execution and peripheral interaction.
6. 3D Visualization: Proteus features 3D visualization capabilities that allow users to view their circuit designs in a three-dimensional representation. This helps in

visualizing the spatial arrangement of components and can aid in identifying potential design issues.

7. Library Components: Proteus offers an extensive library of components, including electronic components, microcontrollers, sensors, actuators, and communication modules. This library provides a vast collection of pre-designed components thatcan be easily integrated into circuit designs.
8. Co-simulation: Proteus supports co-simulation with popular microcontroller development environments, such as Keil, allowing users to seamlessly integrate their software code with the simulation environment for thorough testing and debugging.
9. Integration with ARES: Proteus integrates with ARES, a powerful PCB layout editor, enabling a smooth transition from circuit design to PCB layout and fabrication.
10. Educational Resources: Proteus offers a range of educational resources, tutorials,and examples that assist users in learning and mastering the software's features and electronic design principles.

3.5.1.2 Proteus Layout

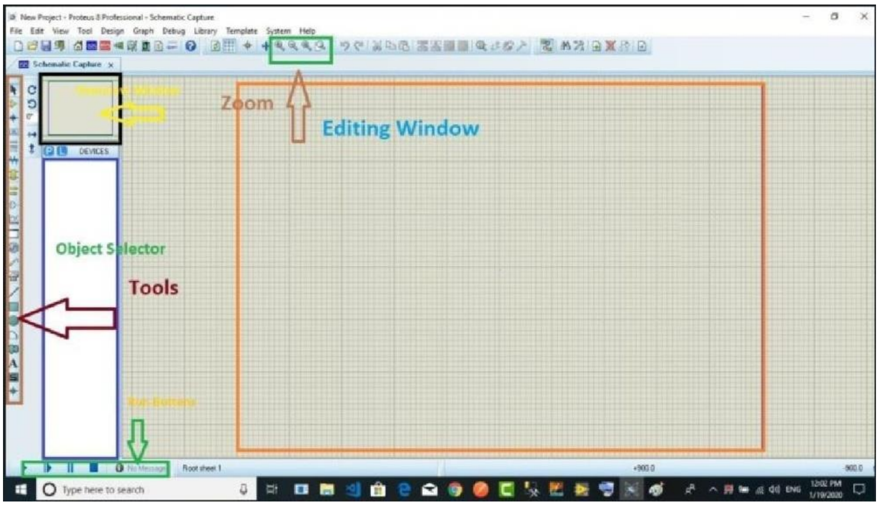


Fig 3.21 Proteus Layout

- Editing Window: In the above figure, you can see that the dotted portion is calledan editing window. This is a drawing portion of proteus where you simulate your engineering circuits and projects.
- Overview Window: In the overview window, you see the complete view of your complete design.
- Object Selector: This section has 2 buttons P and E. P is used to select different components and shown in this box. The E button is for edit something for exampleyou want to vary any value of components than you can use this edit button.
- Zoom Option: By using this option you can easily zoom in and zoom out your layout and can observe complete simulation very clearly.
- Tool Option: By using this option you can select different devices like voltmeter, ammeter, oscilloscope, etc.
- Run Buttons: At the left bottom there are 4 buttons Run, stop, pause and stop. These buttons arelike the remote control and on and off your circuit.

3.5.2 PCB Design:

Start by creating a schematic diagram in Proteus. Add the components required for your project, such as microcontrollers, sensors, communication modules, voltage regulators, capacitors, and other necessary components. Connect the components according to their electrical connections and interconnections.

Component Selection and Footprint Design:

In Proteus, we can choose components from the built-in component libraries or create custom components with specific footprints. Ensure that the footprints accurately represent the physical dimensions, pin locations, and mounting holes of the components you are using.

PCB Layout:

Once the schematic is complete, you can proceed to the PCB layout stage in Proteus. The software allows you to switch from the schematic view to the PCB layout view seamlessly. In the PCB layout view, you can place the components on the board, taking into account factors like component size, orientation, thermal considerations, and signal routing requirements.

Routing:

Use Proteus's routing tools to connect the components with traces on the PCB. Route the traces according to the schematic and consider factors like signal integrity, power distribution, and noise immunity. Proteus provides various routing options, including manual routing and auto-routing, to simplify the process.

Power and Ground Planes:

Design dedicated power and ground planes in Proteus to ensure proper power distribution and minimize noise. These planes help provide stable and low impedance paths for power and ground signals.

Design Rule Check (DRC):

Proteus includes a Design Rule Check feature that allows you to check the PCB layout for errors and violations. Run the DRC to identify issues such as trace spacing violations, clearance violations, overlapping components, and other design rule violations.

3D Visualization:

Proteus offers a 3D visualization feature that allows you to view the PCB layout in a realistic 3D representation. This feature helps you visualize the final design and identify any physical conflicts or issues with the component placement.

Gerber Generation:

Once you are satisfied with the PCB layout, you can generate Gerber files in Proteus. Gerber files are the industry-standard files that describe the PCB design and are used by PCB manufacturers for fabrication. Proteus provides options to generate Gerber files for different layers, solder mask, silkscreen, drill holes, and other necessary details.

The figure shown below is the layout designed by us for the PCB that would be required to complete the circuit in our project.

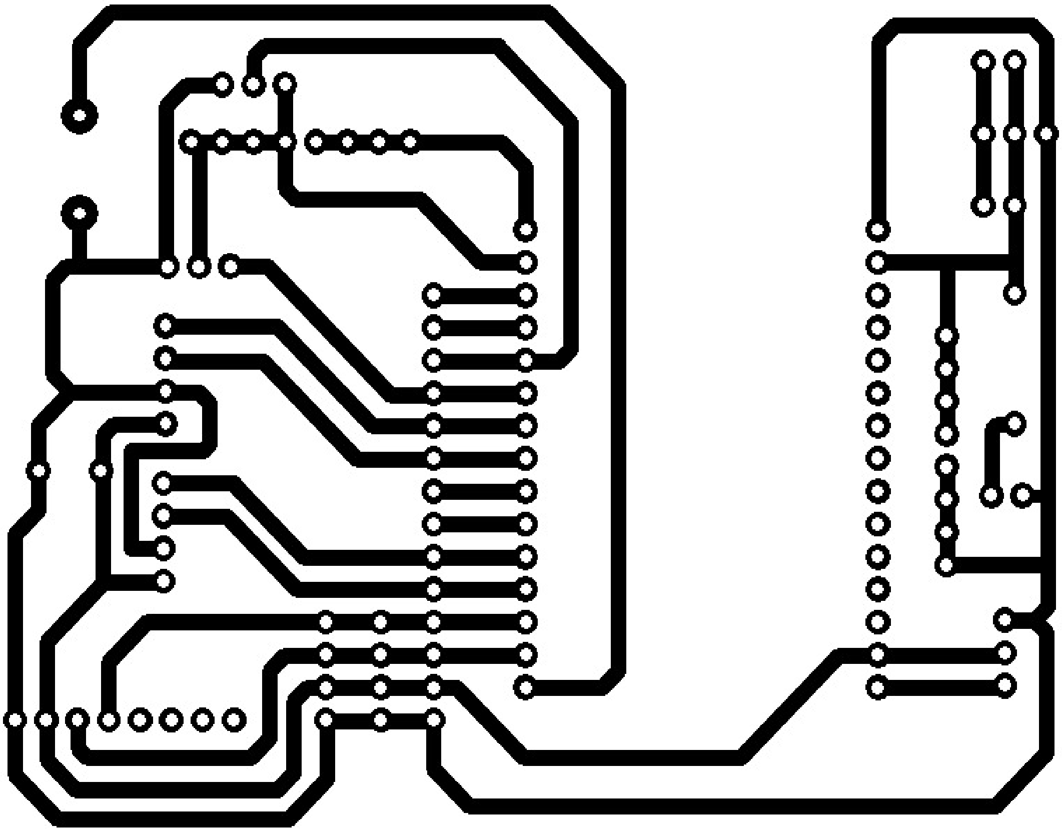


Fig 3.22 PCB layout of the circuit

3.6 PROGRAMMING NODEMCU ON ARDUINO IDE

Step 1: Connect the NodeMCU to the computer
Connect the computer cable to the NodeMCU by using a USB cable, blue onboard LED flicker when powered up, but they will not stay lit.
Step 2: Install the COM/Serial Port Driver
In order to upload code to the ESP8266 and use the serial console, connect any data-capable micro USB cable to ESP8266 IOT Boardand the other side to your computer's USB port.

The new version NodeMCUv1.0 comes with the CP2102 serial chip download and install the driver from:https://www.silabs.com/products/development-tools/
The NodeMCUv0.9 comes with the CH340 serial download and install the driver from:https://github.com/nodemcu/nodemcu- devkit/tree/mas

Step 3: Install the Arduino IDE 1.6.4 or greater

Download Arduino IDE from Arduino.cc (1.6.4 or greater).

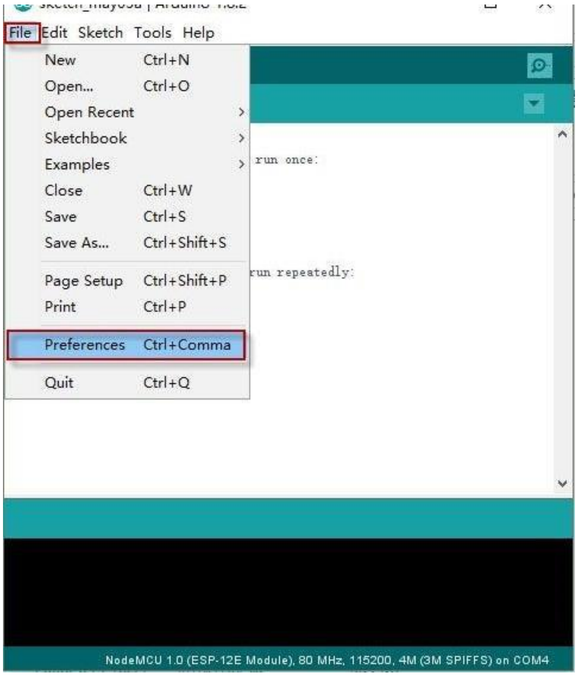


Fig 3.23 Step 3 Install IDE

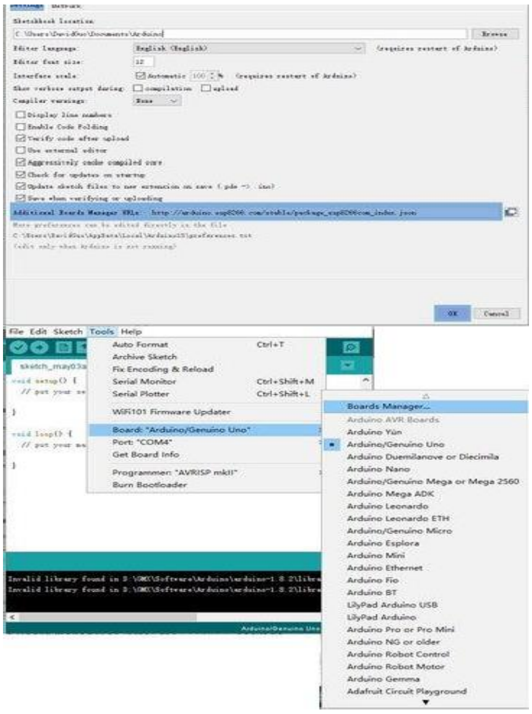

Sketchbook location:
Browse
Editor language: English (English)
Editor font size: 12
Interface scale: ☑ Automatic 100 %
Show verbose output during: □ compilation □ upload
Compiler warnings: None
□ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
□ Use external editor
☑ Aggressively cache compiled core
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading
Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json
OK
Cancel
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM4"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground

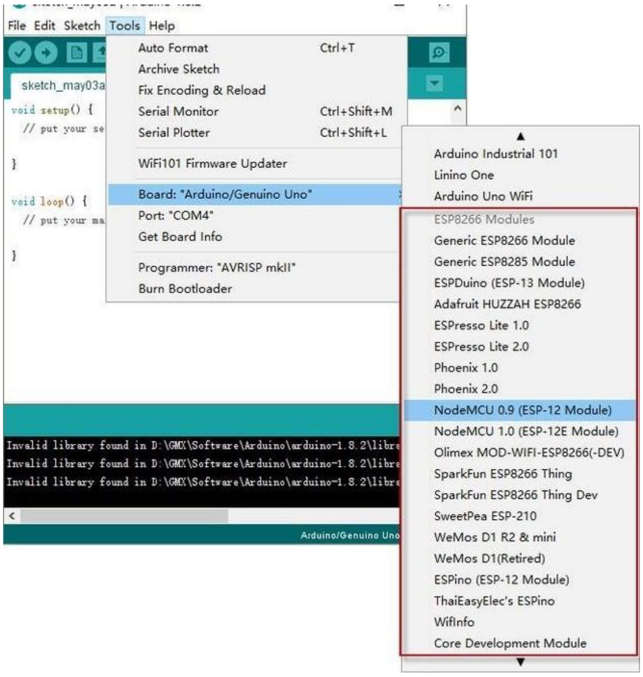
Fig 3.24 Step 4 Install Board Package

Step 4: Install the ESP8266 Board Package

Enter http://arduino.esp8266.com/stable/package_esp8266 into Additional Board Manager URLs field in the Arduino v1.6.4+ preferences (Open Arduino IDE–>File–>Perferences–>Settings). Enter the link and click "OK" to save your changes. Next, use the Board Manager to install the ESP8266 package Enter the Boards Manager and find the board type as below: Scroll the Broads Manager screen down to the bottom, there will be a module called "esp8266 by esp8266 Community" (see following picture), select the latest version and click "Install". The ESP8266 package has been installed

successfully.
Close the ArduinoIDE and restart it again.

Step 5: Setup ESP8266 Support

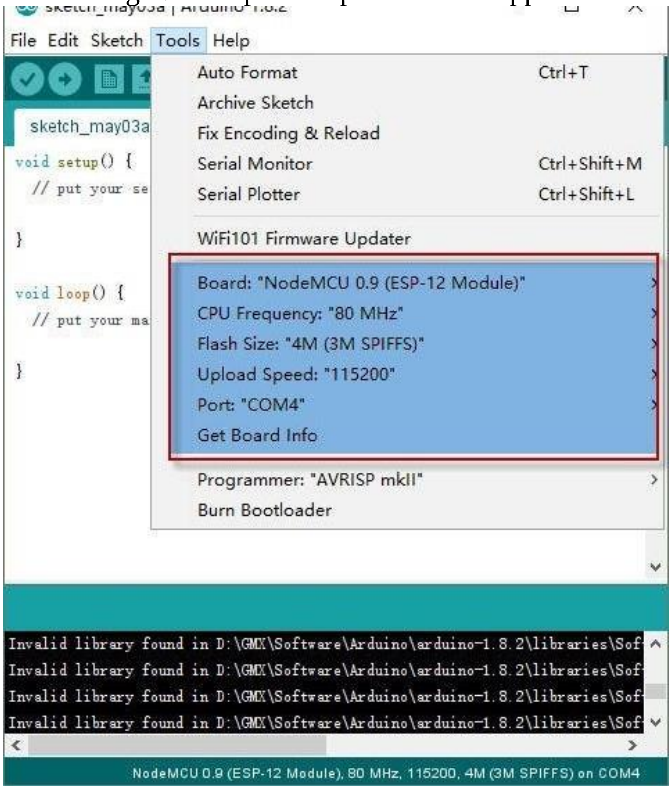


Board: "Arduino/Genuino Uno"
Port: "COM4"
Programmer: "AVRISP mkII"

```
void setup() {
  // put your se

}

void loop() {
  // put your ma

}
```

```
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libre
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libre
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libre
```

Fig 3.25 Step 5 Setup ESP8266 Support



Board: "NodeMCU 0.9 (ESP-12 Module)"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "115200"
Port: "COM4"
Programmer: "AVRISP mkII"

```
void setup() {
  // put your se

}

void loop() {
  // put your ma

}
```

```
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libraries\Sof
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libraries\Sof
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libraries\Sof
Invalid library found in D:\GMX\Software\Arduino\arduino-1.8.2\libraries\Sof
```

NodeMCU 0.9 (ESP-12 Module), 80 MHz, 115200, 4M (3M SPIFFS) on COM4

Fig 3.26 Configuring the board menu

After restart, select NodeMCU 0.9 (or NodeMCU 1.0) from the Tools->Board dropdown> configure the Board menu and choose the right Port for your device. CPU Frequency： 80MHz, Flash Size：4M（3M SPIFFS）

Upload Speed：115200Now just proceed as the Arduino.

3.7 CODE FOR THE PROJECT

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <Wire.h>//I2c
#include <LiquidCrystal_I2C.h>
#include <I2Cdev.h>
#include <MPU6050.h>

MPU6050 mpu;
intsensorValue = 0;
int16_t ax, ay, az; // raw accelerometer data
int16_tgx, gy, gz; // raw gyroscope data
floataccelMagnitude; // accelerometer magnitude
floatimpactThreshold = 5000; // set the impact threshold
LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2 line display

#include <EEPROM.h>
//sender phone number with country code
const String PHONE = "+916375699008";
constinttotalPhoneNo = 5;
String phoneNo[totalPhoneNo] = {"","","","",""};
intoffsetPhone[totalPhoneNo] = {0,13,26,39,52};
String tempPhone = "";

//GPS Module TX pin to NodeMCU D1
//GPS Module RX pin to NodeMCU D2
#define rxGPS D3 //D1
#define txGPS D4 //D2
SoftwareSerialneogps(rxGPS,txGPS);
TinyGPSPlusgps;

//GSM Module TX pin to NodeMCU D3
//GSM Module RX pin to NodeMCU D4
#define rxGSM D6 //D3
#define txGSM D5 //D4
SoftwareSerial sim800(rxGSM,txGSM);

String smsStatus;
String senderNumber;
String receivedDate;
String msg;
boolean DEBUG_MODE = 1;
/* setup function */
voidsetup() {
  lcd.init();   // initialize the lcd
  lcd.init();

   Wire.begin();//I2c
```

```
//Serial.begin(9600);
mpu.initialize();
mpu.setFullScaleAccelRange(MPU6050_ACCEL_FS_16);
mpu.setFullScaleGyroRange(MPU6050_GYRO_FS_2000);
pinMode(D0, OUTPUT);
 digitalWrite(D0, LOW);  // turn the LED on (HIGH is the voltage level)
delay(200);                     // wait for a second
digitalWrite(D0, HIGH);   // turn the LED off by making the voltage LOW
delay(200);
digitalWrite(D0, LOW);
delay(200);                     // wait for a second
digitalWrite(D0, HIGH);   // turn the LED off by making the voltage LOW
delay(1000);

// Print a message to the LCD.
lcd.backlight();
pinMode(D0, OUTPUT);pinMode(D4, OUTPUT);
 digitalWrite(D0, LOW);  // turn the LED on (HIGH is the voltage level)
delay(1000);                     // wait for a second
digitalWrite(D0, HIGH);   // turn the LED off by making the voltage LOW
delay(1000);
digitalWrite(D0, LOW);
  delay(1000);                     // wait for a second
digitalWrite(D0, HIGH);   // turn the LED off by making the voltage LOW
delay(1000);

lcd.clear();
lcd.setCursor(5, 0);
lcd.print("WELCOME");

Serial.begin(115200);
Serial.println("NodeMCU USB serial initialize");
sim800.begin(9600);
Serial.println("SIM800L serial initialize");
neogps.begin(9600);
Serial.println("NEO6M serial initialize");
sim800.listen();
neogps.listen();


EEPROM.begin(512);

Serial.println("List of Registered Phone Numbers");
for (inti = 0; i<totalPhoneNo; i++){
  phoneNo[i] = readFromEEPROM(offsetPhone[i]);
  if(phoneNo[i].length() != 13)
    {phoneNo[i] = "";Serial.println(String(i+1)+": empty");}
  else
    {Serial.println(String(i+1)+": "+phoneNo[i]);}
}
```

```
  smsStatus = "";
  senderNumber="";
  receivedDate="";
  msg="";

  delay(9000);
  sim800.println("AT+CMGF=1"); //SMS text mode
  delay(1000);
  sim800.println("AT+CLIP=1"); //Enable Caller ID
  delay(500);

      phoneNo[0] = "+916375699008";
      phoneNo[1] = "+917014689809";
      phoneNo[2] = "+917627029203";
      phoneNo[3] = "+916378638280";
      phoneNo[4] = "+916375699008";

       writeToEEPROM(offsetPhone[0],phoneNo[4]);
       writeToEEPROM(offsetPhone[1],phoneNo[4]);
       writeToEEPROM(offsetPhone[2],phoneNo[4]);
       writeToEEPROM(offsetPhone[3],phoneNo[4]);
       writeToEEPROM(offsetPhone[4],phoneNo[4]);
   digitalWrite(D4, HIGH);
delay(20000);

}

voidloop() {
  //sim800.listen();
    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    sensorValue = analogRead(A0);
             lcd.setCursor(0, 1);
  lcd.print("MQ3  ");  lcd.print(sensorValue);
    if(sensorValue>1000){//MQ3
     digitalWrite(D0, LOW);
          lcd.setCursor(0, 1);
  lcd.print("Alcohol Detected");
       String phoneNumber1 = "+916375699008";
       digitalWrite(D4, LOW);
    sendLocation2(phoneNumber1);
    lcd.clear();
    }
    Serial.println(sensorValue);
  accelMagnitude = sqrt(ax*ax + ay*ay + az*az);
               lcd.setCursor(0, 0);
  lcd.print("Impect  ");  lcd.print(accelMagnitude);
   if (accelMagnitude>impactThreshold) {       digitalWrite(D4, LOW);
    Serial.println("Impact detected!");
    digitalWrite(D0, LOW);  delay(200);
    lcd.setCursor(0, 0);
  lcd.print("Accident Aleart");
  String phoneNumber1 = "+916375699008";
    sendLocation1(phoneNumber1);
    delay(1500);
      lcd.clear();
digitalWrite(D0, HIGH);
  lcd.clear();
  }

  delay(10);

while(sim800.available()){
  parseData(sim800.readString());
}

while(Serial.available())  {
  sim800.println(Serial.readString());
}

}

voidparseData(String buff){
  Serial.println(buff);

  if(buff.indexOf("RING") > -1)
  {
    boolean flag = 0;
    String callerID = "";

    //+CLIP: "03001234567",129,"",0,"",0
    if(buff.indexOf("+CLIP:")){
     unsignedint index, index1;

      index = buff.indexOf("\"");
      index1 = buff.indexOf("\"", index+1);
      callerID = buff.substring(index+2, index1);
      callerID.trim();
      if(callerID.length() == 13){

        flag = comparePhone(callerID);
      }

      elseif(callerID.length() == 10){

        flag = compareWithoutCountryCode(callerID);
        callerID = "0" + callerID;
      }
   }
    if(flag == 1){
```

```
    sim800.println("ATH");
    delay(1000);
    sendLocation(callerID);
  }
  else{
    sim800.println("ATH");
    debugPrint("The phone number is not registered.");
  }
  return;
}

unsignedintlen, index;
 index = buff.indexOf("\r");
buff.remove(0, index+2);
buff.trim();

if(buff != "OK"){
  index = buff.indexOf(":");
  String cmd = buff.substring(0, index);
  cmd.trim();

  buff.remove(0, index+2);

  if(cmd == "+CMTI"){
    index = buff.indexOf(",");
    String temp = buff.substring(index+1, buff.length());
    temp = "AT+CMGR=" + temp + "\r";
    sim800.println(temp);
  }

  elseif(cmd == "+CMGR"){
    extractSms(buff);

    if(msg.equals("r") &&phoneNo[0].length() == 13) {
      writeToEEPROM(offsetPhone[0],senderNumber);
      phoneNo[0] = senderNumber;
      String text = "Number is Registered: ";
      text = text + senderNumber;
      debugPrint(text);
      Reply("Number is Registered", senderNumber);
    }

    if(comparePhone(senderNumber)){
      doAction(senderNumber);
      //sendLocation();
    }
  }
}
else{
}
```

```
}

voiddoAction(String phoneNumber){

  if(msg == "send location"){
      sendLocation(phoneNumber);
  }

  if(msg == "r2="){

      Serial.println(offsetPhone[1]);
      writeToEEPROM(offsetPhone[1],tempPhone);
      phoneNo[1] = tempPhone;
      String text = "Phone2 is Registered: ";
      text = text + tempPhone;
      debugPrint(text);
      Reply(text, phoneNumber);
  }

  elseif(msg == "r3="){
      writeToEEPROM(offsetPhone[2],tempPhone);
      phoneNo[2] = tempPhone;
      String text = "Phone3 is Registered: ";
      text = text + tempPhone;
      Reply(text, phoneNumber);
  }

  elseif(msg == "r4="){
      writeToEEPROM(offsetPhone[3],tempPhone);
      phoneNo[3] = tempPhone;
      String text = "Phone4 is Registered: ";
      text = text + tempPhone;
      Reply(text, phoneNumber);
  }

  elseif(msg == "r5="){
      writeToEEPROM(offsetPhone[4],tempPhone);
      phoneNo[4] = tempPhone;
      String text = "Phone5 is Registered: ";
      text = text + tempPhone;
      Reply(text, phoneNumber);
  }

  elseif(msg == "list"){
      String text = "";
      if(phoneNo[0])
        text = text + phoneNo[0]+"\r\n";
      if(phoneNo[1])
        text = text + phoneNo[1]+"\r\n";
      if(phoneNo[2])
      text = text + phoneNo[2]+"\r\n";
    if(phoneNo[3])
      text = text + phoneNo[3]+"\r\n";
    if(phoneNo[4])
      text = text + phoneNo[4]+"\r\n";

    debugPrint("List of Registered Phone Numbers: \r\n"+text);
    Reply(text, "+916375699008");
}
elseif(msg == "del=1"){
    writeToEEPROM(offsetPhone[0],"");
    phoneNo[0] = "";
    Reply("Phone1 is deleted.", phoneNumber);
}
elseif(msg == "del=2"){
    writeToEEPROM(offsetPhone[1],"");
    phoneNo[1] = "";
    debugPrint("Phone2 is deleted.");
    Reply("Phone2 is deleted.", phoneNumber);
}
elseif(msg == "del=3"){
    writeToEEPROM(offsetPhone[2],"");
    phoneNo[2] = "";
    debugPrint("Phone3 is deleted.");
    Reply("Phone3 is deleted.", phoneNumber);
}
elseif(msg == "del=4"){
    writeToEEPROM(offsetPhone[3],"");
    phoneNo[3] = "";
    debugPrint("Phone4 is deleted.");
    Reply("Phone4 is deleted.", phoneNumber);
}
elseif(msg == "del=5"){
    writeToEEPROM(offsetPhone[4],"");
    phoneNo[4] = "";
    debugPrint("Phone5 is deleted.");
    Reply("Phone5 is deleted.", phoneNumber);
}

if(msg == "del=all"){
    writeToEEPROM(offsetPhone[0],"");
    writeToEEPROM(offsetPhone[1],"");
    writeToEEPROM(offsetPhone[2],"");
    writeToEEPROM(offsetPhone[3],"");
    writeToEEPROM(offsetPhone[4],"");
    phoneNo[0] = "";
    phoneNo[1] = "";
    phoneNo[2] = "";
    phoneNo[3] = "";
    phoneNo[4] = "";
```

```
      offsetPhone[0] = NULL;
      offsetPhone[1] = NULL;
      offsetPhone[2] = NULL;
      offsetPhone[3] = NULL;
      offsetPhone[4] = NULL;
      debugPrint("All phone numbers are deleted.");
      Reply("All phone numbers are deleted.", phoneNumber);
  }
  smsStatus = "";
  senderNumber="";
  receivedDate="";
  msg="";
  tempPhone = "";
}
voidextractSms(String buff){
   unsignedint index;

    index = buff.indexOf(",");
    smsStatus = buff.substring(1, index-1);
    buff.remove(0, index+2);

    senderNumber = buff.substring(0, 13);
    buff.remove(0,19);

    receivedDate = buff.substring(0, 20);
    buff.remove(0,buff.indexOf("\r"));
    buff.trim();

    index =buff.indexOf("\n\r");
    buff = buff.substring(0, index);
    buff.trim();
    msg = buff;
    buff = "";
    msg.toLowerCase();

    String tempcmd = msg.substring(0, 3);
    if(tempcmd.equals("r1=") || tempcmd.equals("r2=") ||
       tempcmd.equals("r3=") || tempcmd.equals("r4=") ||
       tempcmd.equals("r5=")){

        tempPhone = msg.substring(3, 16);
        msg = tempcmd;

    }
}

voidReply(String text, String Phone)
{
  //return;
  sim800.print("AT+CMGF=1\r");
  delay(1000);
  sim800.print("AT+CMGS=\""+Phone+"\"\r");
  delay(1000);
  sim800.print(text);
  delay(100);
  sim800.write(0x1A); //ascii code for ctrl-26 //sim800.println((char)26); //ascii code for
ctrl-26
  delay(1000);
  Serial.println("SMS Sent Successfully.");
\

  smsStatus = "";
  senderNumber="";
  receivedDate="";
  msg="";
  tempPhone = "";
}

voidwriteToEEPROM(intaddrOffset, const String &strToWrite)
{
  byte len = 13; //strToWrite.length();
   for (inti = 0; i<len; i++)
  {
    EEPROM.write(addrOffset + i, strToWrite[i]);
  }

  EEPROM.commit();

}

String readFromEEPROM(intaddrOffset)
{
  intlen = 13;
  chardata[len + 1];
  for (inti = 0; i<len; i++)
  {
    data[i] = EEPROM.read(addrOffset + i);
  }
  data[len] = '\0';
  return String(data);
}


booleancomparePhone(String number)
{
  boolean flag = 0;
  //--------------------------------------------------
  for (inti = 0; i<totalPhoneNo; i++){
    phoneNo[i] = readFromEEPROM(offsetPhone[i]);
    if(phoneNo[i].equals(number)){
```

```
   //Serial.println(phoneNo[i]);
   flag = 1;
   break;
  }
 }
 return flag;
}


booleancompareWithoutCountryCode(String number)
{
 boolean flag = 0;
 for (inti = 0; i<totalPhoneNo; i++){
  phoneNo[i] = readFromEEPROM(offsetPhone[i]);
  //remove first 3 digits (country code)
  phoneNo[i].remove(0,3);
  //Serial.println("meee1: "+phoneNo[i]);
  if(phoneNo[i].equals(number)){
   //Serial.println(phoneNo[i]);
   flag = 1;
   break;
  }
 }
 return flag;
}

 voiddebugPrint(String text){
 if(DEBUG_MODE == 1)
  Serial.println(text);
}

voidsendLocation(String phoneNumber)
{
 // Can take up to 60 seconds
 booleannewData = false;
 for (unsignedlong start = millis(); millis() - start <2000;)
 {
  while (neogps.available())
  {
   if (gps.encode(neogps.read()))
    {newData = true;break;}
  }
 }

 //If newData is true
 if(newData)
 {
  newData = false;
  String latitude = String(gps.location.lat(), 6);
  String longitude = String(gps.location.lng(), 6);
```

```
    //String speed = String(gps.speed.kmph());

    String text = "Latitude= " + latitude;
    text += "\n\r";
    text += "Longitude= " + longitude;
    text += "\n\r";
    text += "Speed= " + String(gps.speed.kmph()) + " km/h";
    text += "\n\r";
    text += "Altitude= " + String(gps.altitude.meters()) + " meters";
    //text += "\n\r";
    //text += "Connected Satellites= " + String(gps.satellites.value());
    text += "\n\r";
    text += "http://maps.google.com/maps?q=loc:" + latitude + "," + longitude;

    debugPrint(text);
    //delay(300);
    Reply(text, phoneNumber);
  }
}

void sendLocation1(String phoneNumber)
{
  Serial.print("aaya");
  // Can take up to 60 seconds
  booleannewData = false;
  for (unsignedlong start = millis(); millis() - start <2000;)
  {
    while (neogps.available())
    {
      if (gps.encode(neogps.read()))
        {newData = true;break;}
    }
  }

  //If newData is true
  if(newData)
  {
    newData = false;
    String latitude = String(gps.location.lat(), 6);
    String longitude = String(gps.location.lng(), 6);
    //String speed = String(gps.speed.kmph());

      String text1 = "Emergency.... Accident alert " +    text1 +=
"http://maps.google.com/maps?q=loc:" + latitude + "," + longitude;

          Serial.println(text1);
    //debugPrint(text1);
    //delay(300);
    Reply(text1, phoneNumber);
  }
```

```
}

void sendLocation2(String phoneNumber)
{

  // Can take up to 60 seconds
  booleannewData = false;
  for (unsignedlong start = millis(); millis() - start <2000;)
  {
    while (neogps.available())
    {
      if (gps.encode(neogps.read()))
        {newData = true;break;}
    }
  }

  //If newData is true
  if(newData)
  {
    newData = false;
    String latitude = String(gps.location.lat(), 6);
    String longitude = String(gps.location.lng(), 6);
    //String speed = String(gps.speed.kmph());

      String text1 = "Alcohol Detected " +    text1 +=
"http://maps.google.com/maps?q=loc:" + latitude + "," + longitude;

          Serial.println(text1);
    //debugPrint(text1);
    //delay(300);
    Reply(text1, phoneNumber);
  }
}
```

3.7 BILL OF MATERIAL

Table 3.6 Bill of Material

Sr. No.	COMPONENTS	PRICE (In Rs.)
1	ESP8266 NodeMCU	500
2	MPU6050	150
3	MQ3 alcohol sensor	180
4	GSM SIM 900A	900
5	NEO 6M GPS	450
6	9V BO DC Motor	200
7	L293D motor driver	250
8	16X2 LCD	100
9	Interface IC HD44780	39
10	LM7805 voltage regulator	15
11	Buzzer	50
12	Supply	150
13	Resister & capacitors	30
14	Zero Torque Motor	200
15	Miscellaneous	300
	TOTAL	**3,514**

CHAPTER: 4
RESULT AND DISCUSSION

4.1 RESULT:

The project successfully implemented a comprehensive system aimed at enhancing road safety through alcohol sensing, engine locking operation, and accident detection functionalities. Utilizing a range of components including the Node MCU ESP8266, GSM SIM900A, gyro sensor, MQ3 alcohol sensor, and NEO 6M GPS, the system effectively monitored alcohol levels in the vehicle operator's breath and detected accidents in real-time.

Upon detecting alcohol concentrations surpassing legal limits, the system swiftly immobilized the engine, simulating engine locking by activating and deactivating a battery-operated DC motor managed by a motor driver. Simultaneously, in the event of an accident, the MPU6050 sensor identified changes in the vehicle's angular position or acceleration indicative of a collision.

The ESP8266 microcontroller utilized the GPS module to pinpoint the precise crash location and dispatched comprehensive notifications via the GSM module to predefined recipients, providing vital accident details.

4.2 TESTING:

Table 4.1 Testing table

ACTION	SCENARIO	OUTCOME	CORRECTIVE MEASURE (IF APPLICABLE)
Switching the system ON	All the devices start to calibrate.	Same as expected	None
Booting up of the system	20 seconds after the device has been turned ON values of MQ3 and MPU6050 start to display on the LCD	MQ3 sensor giving false detection of presence of alcohol	Value of the alcohol detector were varied accordingly.
Head on collision	In case the system collides head ON the MPU6050 detects collision, LCD displays “accident detected”	Same as expected	None
Message sending	When accident is detected, the message is sent to the pre-fed numbers with the location of the device	Same as expected	None

Alcohol detection	The alcohol sensor detects the presence of alcohol in the air. The engine is locked.	Same as expected	None
Message sending	When alcohol is detected, message containing the location is sent to pre-fed number.	Same as expected	None

4.3 DISCUSSION:

The successful implementation of this project underscores its pivotal role in significantly enhancing road safety by addressing critical issues such as intoxicated driving and rapid accident response. By making use of the alcohol sensing feature with engine turn OFF mechanisms and accident detection functionalities, the system provides a proactive approach to reduce accidents resulting from alcohol consumption and ensures swift response in emergency situations.

This system not only detects alcohol levels and accidents but also actively intervenes by immobilizing the vehicle's engine, potentially saving lives by preventing accidents caused by impaired driving. Additionally, the swift dissemination of critical accident information to authorities or designated individuals enables prompt response and assistance, further reducing the severity of accidents and minimizing casualties on the road.

Moving forward, the project could undergo further optimizations in sensor accuracy, response time, and user interaction to enhance its practicality and effectiveness in real-world scenarios. Moreover, considerations such as regulatory compliance, user acceptance, and scalability should be addressed to facilitate widespread adoption and deployment of the system, ultimately contributing to the overarching goal of saving lives and fostering safer road environments.

4.4 PROJECT PHOTOS

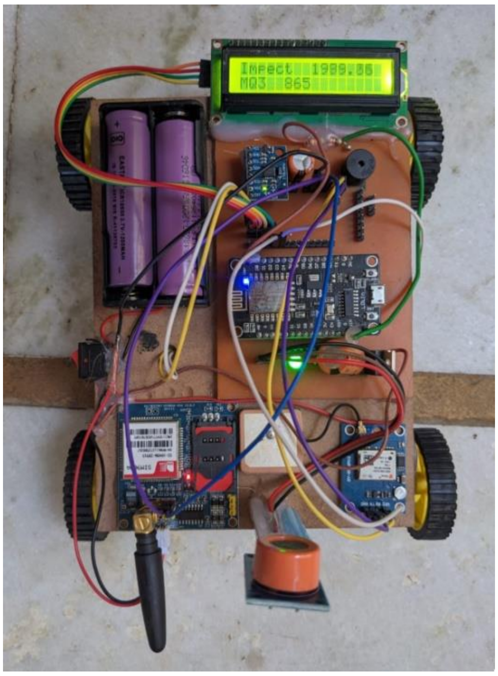


Fig 4.1 Project photograph

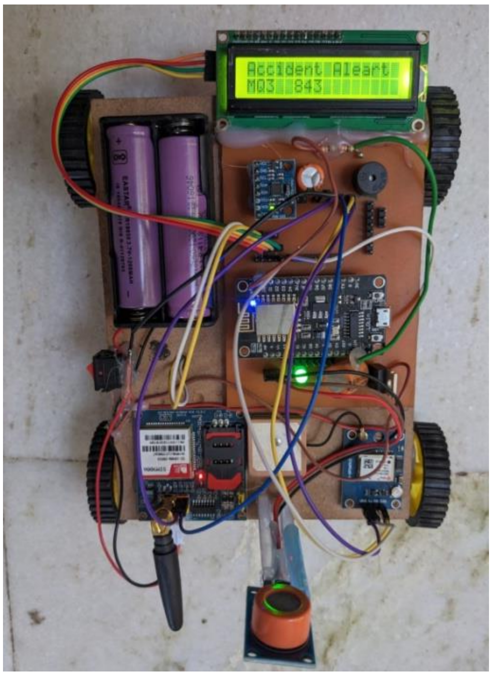
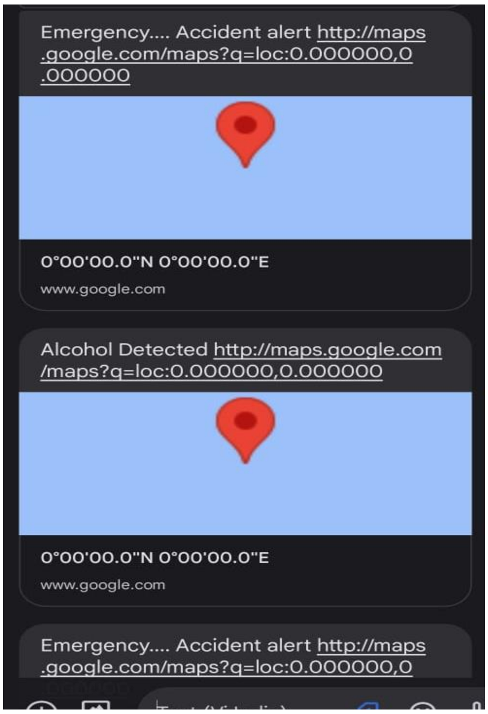


Fig 4.2 Accident detected


Emergency.... Accident alert http://maps.google.com/maps?q=loc:0.000000,0.000000

0°00'00.0"N 0°00'00.0"E

www.google.com

Alcohol Detected http://maps.google.com/maps?q=loc:0.000000,0.000000

0°00'00.0"N 0°00'00.0"E

www.google.com

Emergency.... Accident alert http://maps.google.com/maps?q=loc:0.000000,0

Fig 4.3 Message sent

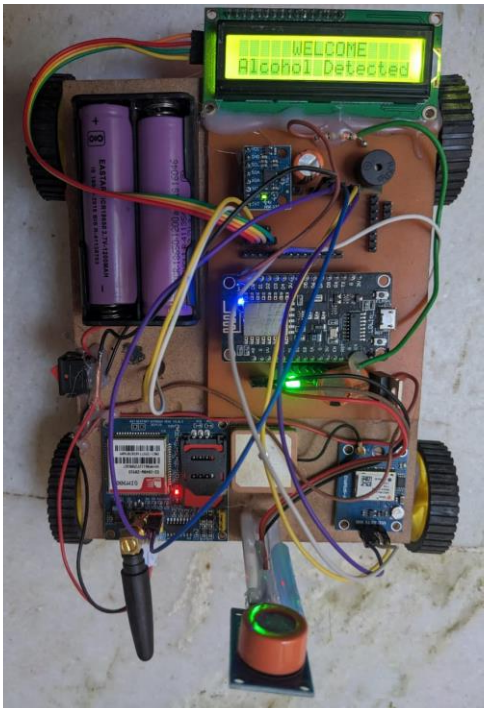


Fig 4.4 Alcohol detected

4.5 LIMITATIONS

1. Accuracy and Reliability of Sensors: The accuracy and reliability of sensors like the MQ3 alcohol sensor and MPU6050 gyro sensor can vary, impacting the system's ability to accurately detect alcohol levels and accidents. Calibration and periodic maintenance may be necessary to ensure consistent performance.

2. Integration and Compatibility Issues: Integrating multiple components from different manufacturers can lead to compatibility issues or challenges in software integration. Ensuring seamless communication between components and robust software design is crucial for the system's overall functionality.

3. Power Consumption: Some components, such as the GSM module and GPS module, consume significant power. Managing power consumption efficiently is essential to prevent draining the vehicle's battery and ensure continuous operation, especially in emergency situations.

4. Legal and Regulatory Considerations: The project may face legal and regulatory hurdles, particularly regarding the implementation of engine immobilization based on alcohol levels. Adhering to relevant laws and ensuring the system's compliance with regulations is essential for its acceptance and legality.

5. User Acceptance and Privacy Concerns: Some drivers may be hesitant to adopt a system that monitors their alcohol levels and location at all times due to privacy concerns. Ensuring user acceptance and addressing privacy issues through transparent policies and safeguards is crucial for the project's success.

6. Power Source and Battery Life: The project relies on a 18650 Li-po battery as the power source. However, Li-po batteries have limited capacity, and the battery life may vary depending on the usage and frequency of data transmission. The system may require frequent battery replacements or recharging, which can be inconvenient and may interrupt the continuous. monitoring of manholes.

7. Connectivity and Network Limitations: The system relies on network connectivity to transmit data to the cloud platform or send alerts through the GSM module. Network outages or weak signal strength can disrupt the system's functionality and lead to delayed or missed notifications. Moreover, in areas with limited or no network coverage, the system may not be able to operate effectively.

8. Sensor Accuracy and Calibration: The accuracy of the sensor readings can vary depending on factors such as sensor quality, calibration, and environmental conditions. Regular calibration and maintenance of the sensors are necessary to ensure accurate and reliable data. Without proper calibration, there is a risk of false readings or inaccurate measurements, which could lead to erroneous decisions or ineffective monitoring.

4.6 CONCLUSION

The implementation of a comprehensive system aimed at mitigating road fatalities resulting from alcohol consumption and reckless driving represents a crucial step towards enhancing road safety. By integrating ignition control mechanisms to prevent intoxicated individuals from operating vehicles and implementing continuous speed monitoring with alerts to authorities, the proposed solution not only safeguards drivers but also pedestrians. Through this proactive approach, the project effectively addresses two primary causes of road accidents.

Moreover, the proposed system extends its impact beyond accident detection by incorporating measures for immediate response and assistance. By providing on-the-spot medication to accident victims, it demonstrates a commitment to reducing the severity of injuries and potentially saving lives. Additionally, through the utilization of advanced alert systems capable of preemptively halting vehicles to prevent accidents, the project showcases a forward-thinking approach to accident prevention.

In conclusion, the proposed system presents a multifaceted approach to enhancing road safety by addressing the root causes of accidents, implementing proactive measures for accident detection and response, and leveraging advanced technologies to prevent accidents altogether. Through these concerted efforts, the project aims to significantly reduce road fatalities and promote safer transportation environments for all road users.

4.7 FUTURE SCOPE

1. Enhanced Data Analytics: Implementing advanced data analytics techniques can enhance the system's capabilities in detecting patterns and trends related to alcohol-related accidents. Machine learning algorithms could be employed to analyze historical data and identify risk factors, enabling proactive measures to prevent accidents.

2. Integration with Autonomous Vehicles: As autonomous vehicle technology advances, integrating your system with autonomous vehicles could provide additional safety benefits. For example, if an autonomous vehicle detects alcohol consumption or an accident, it could autonomously reroute or seek assistance, improving overall safety on the road.

3. Collaboration with Insurance Companies: Partnering with insurance companies to incentivize the adoption of your system could lead to reduced premiums for drivers who install it in their vehicles. Insurance companies could leverage the data collected by the system to offer personalized insurance rates based on driving behavior and risk factors.

4. Integration with Emergency Services: Enhancing the system to directly communicate with emergency services, such as dispatching ambulances to the scene of an accident, could further improve response times and potentially save lives.

5. Expansion to Public Transportation: Adapting the system for use in public transportation vehicles, such as buses and taxis, could extend its benefits to a broader population and help reduce alcohol-related accidents in the transportation sector. This could involve modifications to accommodate larger vehicles and passenger volumes.

6. Continuous Research and Development: Investing in ongoing research and development efforts to improve sensor accuracy, reduce system costs, and enhance user experience will be crucial for the long-term success and sustainability of the project.

7. By implementing Iot: This system can also be further improved by the use of Iot, throught which a system can be developed which would be able to sense the hospitals or emergency care unit, and send messages to those specific hospitals, which would decrease the response time by a huge margin.

REFRENCES

[1] Zhu Yi-duo The drunk-driving measurement and control system based on multisensory [J] Journal of Wuhan Jiao tong Polytechnic_2011_13 1_78-80

[2] Pham Hoang Oat, Micheal Drieberg and Nguyen Chi Cuong. "Development of Vehicle Tracking system using GPS and GSM", 2013 IEEE Conference on Open Systems (ICOS).

[3] Bmugilag, MuthuLakshmi, M, Santhiya. K, Prof. Divya “Smart Helmet System Using Alcohol Detection For Vechicle Protection [International Journal of Innovative Research in Science Engineering and Technology (IJIRTSE) ISSN: 2395-5619, Volume 2, Issue-7. July 2016].

[4] Dhivya M and Kathiravan S, Dept. of ECE, Kalaignar Karunanidhi Institute of Technology- Driver Authentication and Accident Avoidance System for Vehicles (Smart Computing Review, vol. 5, no. 1, February 2015].

[5] Al Wadhahi, N.T.S., Hussain, S.M., Yosof, K.M., Hussain, S.A. and Singh, A.V., 2018, August. "Accidents Detection and Prevention System to reduce Traffic Hazards using IR Sensors. In 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 737-741), IEEE.

[6] Kota, V.K., Mangali, N.K., Kanakurthi, T.K., Kumar, A.R. and Velayutham, T., 2017, March, "Automated accident detection and rescue system". In 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 1437-1441) IEEE.

[7] Khalil, U., Javid, T. and Nasir, A., 2017, November." Automatic road accident detection techniques": A brief survey. In 2017 International Symposium on Wireless Systems and Networks (ISWSN) (pp. 1-6). IEEE.

[8] Anil. B.S., Vilas, K.A. and Jagtap, S.R., 2014, April." Intelligent system for vehicular accident detection and notification". In 2014 International Conference on Communication and Signal Processing (pp: 1238-1240). IEEE.

[9] Ali, A. and Eid, M., 2015, May." An automated system for accident detection". In 2015 IEEE International Instrumentation and Measurement Technology Conference. (12MTC) Proceedings (pp. 1608-1612), IEEE.

[10] Meena, A., lyer, S., Nimje, M., Joglekar, S., Jagtap, S. and Rahman, M., 2014, May." Automatic Accident Detection and reporting framework for two wheelers". In 2014 IEEEInternational Conference on Advanced Communications, Control and Computing Technologies (pp. 962-967), IEEE.

APPENDIX

18650 Li-Po Batteries:

A lithium-ion (Li-ion) 18650 battery is a specific rechargeable cylindrical battery commonly used in various electronic devices and applications. The term “18650” represents the battery’s dimensions: 18mm in diameter and 65.0mm in length (hence, the numbers 18-65-0).
When purchasing high-capacity 18650 batteries, make sure to buy from reputable manufacturers and sellers to ensure you receive genuine, high-quality products. Be cautious of counterfeit or low-quality batteries that may claim higher capacities than they can actually deliver, as these can pose safety risks or provide subpar performance.
Li-ion Battery 18650 or we say 18650 Li-ion batteries are popular due to their high energy density, rechargeability, and relatively low cost.

Applications of Li-Po Batteries:

1. Laptop batteries: Many laptop battery packs contain multiple Li-ion Battery 18650 cells connected in series or parallel to achieve the desired voltage and capacity.

2. Power banks: Portable power banks for charging smartphones and other devices often use Li-ion Battery 18650 batteries.

3. Flashlights: High-performance LED flashlights frequently use Li-ion Battery 18650 batteries due to their high capacity and output capabilities.

4. Electric vehicles: Some electric bicycles, scooters, and even early Tesla electric vehicle models utilized Li-ion Battery 18650 batteries in their battery packs.

5. Vaping devices: Electronic cigarettes and vaping devices may use Li-ion Battery 18650 batteries as their power source.

6. DIY projects: Li-ion Battery 18650 batteries are popular among hobbyists for various DIY projects and custom-built devices that require a reliable and compact power source.

When using Li-ion Batteries 18650 or cylindrical Li-ion batteries, following safety precautions and ensuring compatibility with our device is essential. Be mindful of the battery’s voltage (usually 3.6V or 3.7V), capacity (measured in mAh), and discharge rate (C-rating) to prevent potential damage or safety risks. Additionally, it is crucial to use a dedicated lithium-ion charger to charge these batteries properly.

How to test the capacity of the battery?

To test the capacity of a 18650 lithium-ion battery, you will need a specialized battery tester or charger with a capacity testing function. These devices discharge the battery at a controlled current and measure the amount of charge it can deliver before reaching its cutoff voltage.
Here’s a step-by-step guide on how to test the capacity of a 18650 battery:

1. Choose a tester: Purchase or obtain a reliable battery tester or charger that is specifically designed to test the capacity of lithium-ion batteries. Ensure that the device is compatible with 18650 batteries and has a capacity testing function.

2. Charge the battery: Before testing, fully charge the 18650 battery using a dedicated lithium-ion charger. This ensures that the battery is at its maximum capacity when you start the test.

3. Insert the battery: Place the 18650 battery into the battery tester or charger, ensuring the positive (+) and negative (-) terminals correctly align with the device's contacts.

4. Select the test mode: Set the device to its capacity testing mode, which may be labeled as "discharge," "test," or "capacity test" on the device or in its user manual.

5. Set the discharge current: Choose an appropriate discharge current for the test. A lower discharge current (e.g., 0.5A or 1A) will provide a more accurate representation of the battery's capacity in low-drain applications, while a higher discharge current (e.g., 2A or more) can simulate the battery's performance in high-drain applications. Keep in mind that using a higher discharge current will result in a shorter runtime and may produce a slightly lower measured capacity.

6. Start the test: Initiate the capacity test on the device. The tester will discharge the battery at the selected current until the battery reaches its cutoff voltage (usually around 2.5V to 3.0V). During the test, the device will measure the total amount of charge (mAh) that the battery delivers.

7. Record the results: Once the test is complete, the device will display the measured capacity of the 18650 batteries in milliampere-hours (mAh). Record this value and compare it to the battery's rated capacity to assess its performance and health.

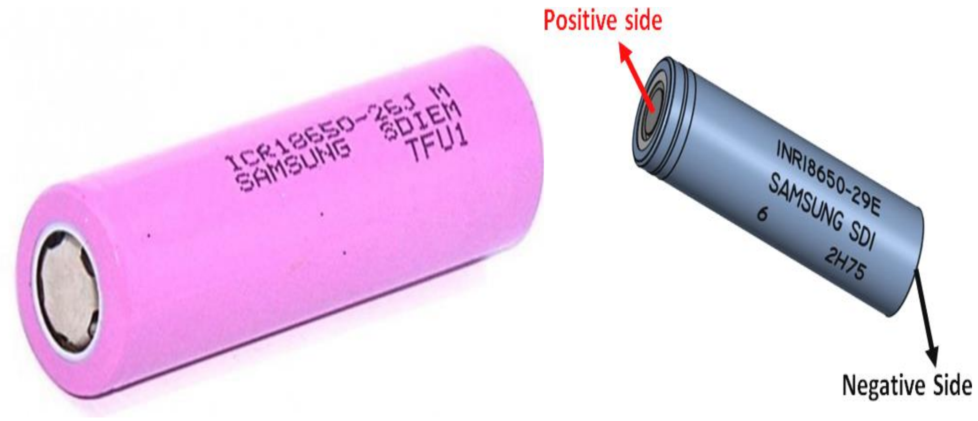


Fig A- 1 Li-Po Battery and Pin Diagram

18650 Features and Technical Specifications:

- Nominal Voltage: 3.6V
- Nominal Capacity: 2,850 mAh

- Minimum Discharge Voltage: 3V
- Maximum Discharge current: 1C
- ChargingVoltage: 4.2V (maximum)
- Charging current: 0.5C
- Charging Time: 3 hours (approx)
- Charging Method: CC and CV
- Cell Weight: 48g (approx)
- Cell Dimension: 18.4mm (dia) and 65mm (height)

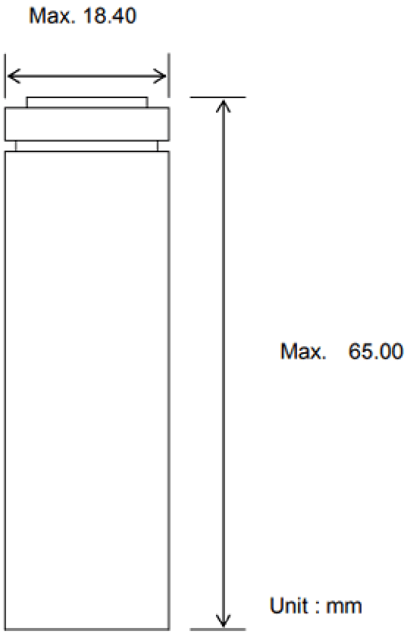


Fig A 2- 2-D Model of a 18650 Li-Po Battery

Types of NodeMCU ESP8266 Microcontroller:

The most common models of the NodeMCU are the Amica (based on the standard narrow pin-spacing) and the LoLin which has the wider pin spacing and larger board.

1. Official Amica NodeMCU:Amica NodeMCU measures 49mm x 26mm with a standard pin space of 0.1″ between pins and 0.9″ between rows. The Amica NodeMCU is approximately 25% smaller in size than a closely compatible LoLin style NodeMCU

2. Official Amica NodeMCU on Carrier Board: Amico NodeMCU mounted to a 102mm x 51mm carrier board with dual DB-09 male/female connectors

3. Lolin NodeMCU: Lolin NodeMCU measures 58mm x 32mm with a pin spacing of 0.1″ between pins and 1.1″ between rows.

Comparison of specifications of different types of NodeMCU:

Specifications	**Official NodeMCU**	**NodeMCU Carrier Board**	**LoLin NodeMCU**
Microcontroller	ESP 8266 32-bit	ESP 8266 32-bit	ESP 8266 32-bit
NodeMCU Model	Amica	Amica	Clone LoLin
NodeMCU Size	49mm x 26 mm	49 mm x 26 mm	58 mm x 32 mm
Carrier Board Size	n/a	102 mm x 51 mm	n/a
Pin Spacing	22.86 mm	22.86 mm	27.94 mm
Clock Speed	80 MHz	80 MHz	80 MHz
USB to Serial	CP2102	CP2102	CH340G
USB Connector	Micro USB	Micro USB	Micro USB
Operating Voltage	3.3V	3.3V	3.3V
Input Voltage	4.5V-10V	4.5V-10V	4.5V-10V
Flash Memory/SRAM	4 MB / 64 KB	4 MB / 64 KB	4 MB / 64 KB
Digital I/O Pins	11	11	11
Analog In Pins	1	1	1
ADC Range	0 – 0.3V	0 – 0.3V	0 – 0.3V
UART/SPI/I2C	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1
Wi-Fi Built-in	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Temperature Range	-40C – 125C	-40C – 125C	-40C – 125C