Project On

OVER SPEED AND ALCOHOL DETECTION SMS ALERT SYSTEM

MSCIT (PART II) SEMESTER - III 2023-24

SUBMITTED BY

Suyog S. Fepade

Seat No. 2295262

Submitted in partial fulfillment of the requirement for

Qualifying

M.Sc. Part II Semester III Examination

2023-24

University of Mumbai

Department of Information Technology

Shailendra Degree College

S.V. Road, Dahisar (E) , Mumbai-68

Shailendra Education Society's

ARTS, COMMERCE & SCIENCE COLLEGE

NAAC Re-accreditated 'B++' grade with CGPA score 2.98 Shailendra Nagar, Dahisar (East), Mumbai - 400 068.

CERTIFICATE

This is to certify that Mr. /Miss. **Suyog S. Fepade** of M.Sc. (I.T.) Part II Semester III has Partially completed the project work in the title of **"Alocohol Detector"** during the academic year 2023- 24 under the guidance of Asst. Prof. Hemchandra Kumbhar being the partial requirement for the fulfillment of the curriculum of Master of Science in Information Technology, University of Mumbai.

Place: Mumbai
Date: / 03 /2024

Internal Examiner

External Examiner

M.Sc. (IT) Coordinator

College Seal

OVER SPEED AND ALCOHOL DETECTION SMS ALERT SYSTEM

ACKOWLEDGEMENT

I take this opportunity to express my profound feature gratitude to Prof. Hemchandra Kumbhar for giving us this opportunity to accomplish this project work.

I would like to thank the entire Information Technology department who directly or indirectly helped us in completion of this project and to my family and friends without whose support, motivation and encouragement this would not have been possible.

SUYOG S. FEPADE

ABSTRACT

With rising technologies and developed automobiles with high end motor vehicles with increase horsepower and accessible speed, it is easier than ever to come in contact with accidents. People are able to drive recklessly, affecting the safety of common people. Although speed limits and other prevention laws were implemented, road accidents continue to happen every day. Some of the main contributing factors leading to motor vehicle accidents are over speeding, rash driving and drunk driving**.**

The proposed work is to develop a device which controls accidents due to over speed, drunk driving, rash driving on motor vehicles. This device will be able to detect over speed and programmed to alert with a SMS. The device also includes an application which disables the starting of the vehicle engine when alcohol consumption is detected. The device model is designed using Arduino MEGA, Global system for Mobile communication (GSM), Global positioning system (GPS), alcohol sensors.

INDEX

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION TO EMBEDED 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. Ninetyeight 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. For example, the Gibson Robot Guitar features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music. 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.

1.1.1 EMBEDDED SYSTEM BLOCK DIAGRAM

Embedded system is the combination of hardware and software. Hardware consists of Peripherals and Processors. Peripherals are used to provide input and to obtain the output while Processor controls the operation of peripherals. Software is used by Processor to control the peripherals in required way.

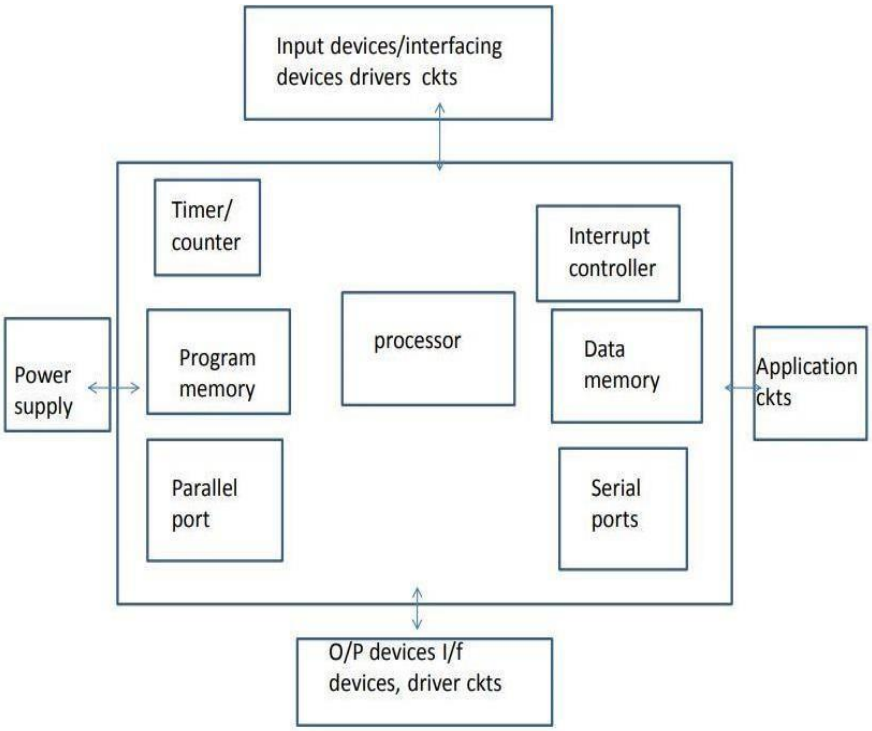

Fig. 1.1 Block diagram of Embedded Systems

1.2 PROCESSORS

Processors are the ones which performs some specific task or operation. These are divided into:

1. Microprocessors

2. Microcontrollers

1.2.1 MICROPROCESSORS

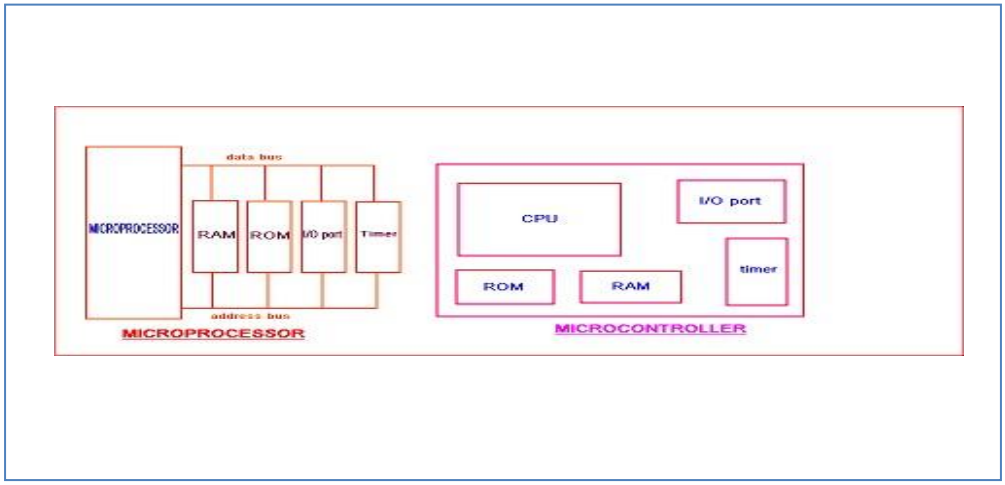

Fig.1.2 Block diagram of microprocessor and microcontroller

A microprocessor consists of input device, output device, arithmetic and logic unit, control unit, a register unit and memory unit. ALU (arithmetic and logic unit) performs all the arithmetic and logical operations, register unit consists of general purpose registers like B, C, D etc. Memory unit is the one which stores the data.

1.2.2 MICROCONTROLLERS

The functioning of microcontroller and microprocessor is similar but there are some inbuilt features like:

1. Electrically Erasable Programmable Read Only Memory(EPROM)
2. Universal Synchronous Asynchronous Receiver Transmitter(USART)
3. Analog to digital convertor(ADC)
4. Digital to analog convertor(DAC)
5. Oscillators
6. Timer

The output of microcontroller is always integer format only. It cannot provide accurate values of floating point values. Good example for Microcontroller is a Traffic Light Controller. Out of all these processors, we are using Microcontrollers as they are more efficient and accurate which can perform tasks easily and quickly. Coming to software point of view, we have many software languages and out of all we are using Assembly Level Language to load the program into the controller. We are using Atmel manufactured microcontrollers.

Atmel company manufactures different series of microcontrollers like AT89C, AT89S. Here, in this project we are using Arduino Mega which uses ATmega2560 processor.

1.3 INTRODUCTION

The project supervises the alcohol content and the speed of the driver. Traffic accidents caused by drunken drivers and due to rash driving not only represent a significant portion of all automobile accidents but they frequently involve fatalities. However, an effective apparatus that prevent such type of accidents is not yet developed. The objective of the apparatus is to initially check the alcohol content of the driver and not to start the ignition if it is more and to supervise the speed of the vehicle and if stays for three seconds in the excess speed it sends an alert to the respective authorities. Sweat sensors, straw like sensors are used to check the drunken condition of the drivers which may mislead because of wrong reading etc. Scientists developed a novel carbon nanotube based(CNT)-based alcohol sensor with particular focus on the response delay problem is presented in CNT based sensors.

1.4 OBJECTIVES

The main objectives of the project are:

- Alcohol detection
- Speed supervising
- Driver safety
- Decrease fatality due to accidents
- To provide information about the condition of the driver to respective authorities

1.5 BLOCK DIAGRAM

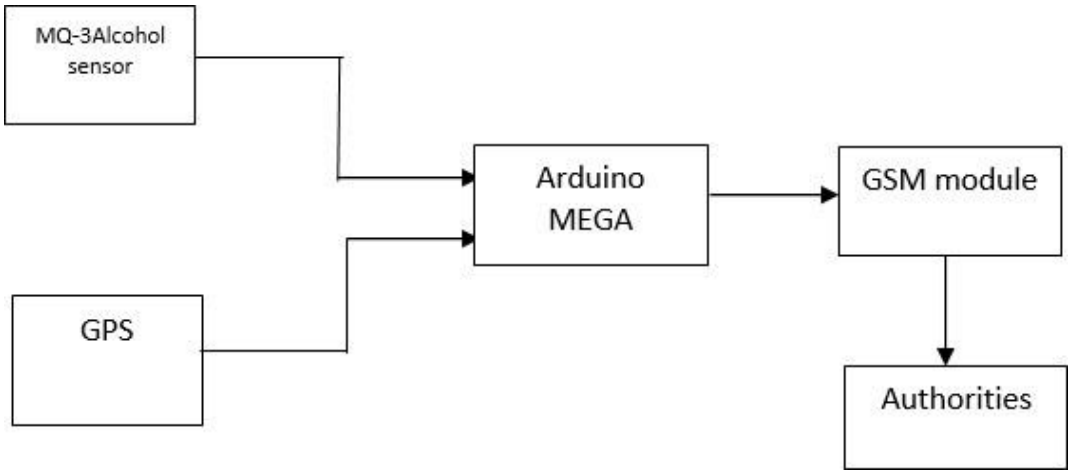

Fig 1.3 Block diagram of entire system

1.6 HARDWARE COMPONENTS USED

ALCOHOL SENSOR:

Alcohol sensor measures the amount of alcohol that is present in the user. There are contact and non-contact type sensors. As the output of the sensor is smaller in amplitude the signal power is also low therefore are used .

GPS MODULE:

GPS module is used to locate the user and also to calculate the speed. PS module works on the principle of trilateration. The speed is calculated by calculating the distance between the satellites and the time taken to travel the same distance.

GSM MODULE:

GSM module is used to send the message of the location and speed to the authorities.

LCD DISPLAY:

A 16x2 crystal LCD display is used to display the latitude longitude, status of the message.

RELAY:

A relay is simple electromechanical switch made up of an electromagnet and a set of contacts. Relays are found hidden in all sort of devices.

ARDUINO:

Arduino is an open source platform based on easy-to-use hardware and software.Arduino senses the environment by receiving inputs from many sensors and effects its surroundings.

1.7 SOFTWARE USED

As Arduino is the heart of the project we have use Arduino IDE.It is an open source software and is accessible to anyone.

CHAPTER-2

ARDUINO MEGA

2.1 WHAT IS ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. We can tell our board what to do by sending a set of instructions to the microcontroller on the board. To do so we use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT(internet of things) applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

2.2 WHY ARDUINO

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms.
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers.

2.3 WHAT DOES ARDUINO DO?

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their

contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

It works with sensors like push buttons, touchpads, variable resistors, motors, speakers etc.

2.4 ARDUINO USES

- It gives intelligence to our circuit or our design. It can take input from sensors, understand the data and do assigned tasks.
- It is used for interactive installation and rapid prototyping.

2.5 TYPES OF ARDUINO

There are many types of the USB Arduino.Some of them are:

- Arduino Uno
- Arduino Lenardo
- Arduino Mega
- Arduino Nano
- Arduino Mini
- Arduino Mini Pro
- Arduino BT

Table 2.1: Comparison of different Arduino board

Arduino Board	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KBSRAM, 512KB flash	54	12 input, 2 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	10	12 input, 0 output
Arduino Board	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KBSRAM, 512KB flash	54	12 input, 2 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	10	12 input, 0 output

2.6 ARDUINO MEGA

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560(datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs),16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USBconnection, a power jack, an ICSP header, and a reset button. It contains everythingneeded to support the microcontroller; simply connect it to a computer with a USB cable orpower it with a AC-to-DC adapter or battery to get started.

2.6.1 WHY ARDUINO MEGA?

- Big - lots of I/O for projects that need it

- Memory - if your code simply won't fit in the Uno, you've got much more room here • Same/Similar shape/layout - many shields made for the smaller boards will still work
- Nicely organized - I/O grouped by function
- The main reason for choosing Arduino MEGA is due to its number of serial ports available.

2.6.2 OPERATIONAL SPECIFICATIONS

Table 2.2 Operational Specifications

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	Pins54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloaderSRAM8
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2.7 ARDUINO FUNCTIONAL BLOCKS

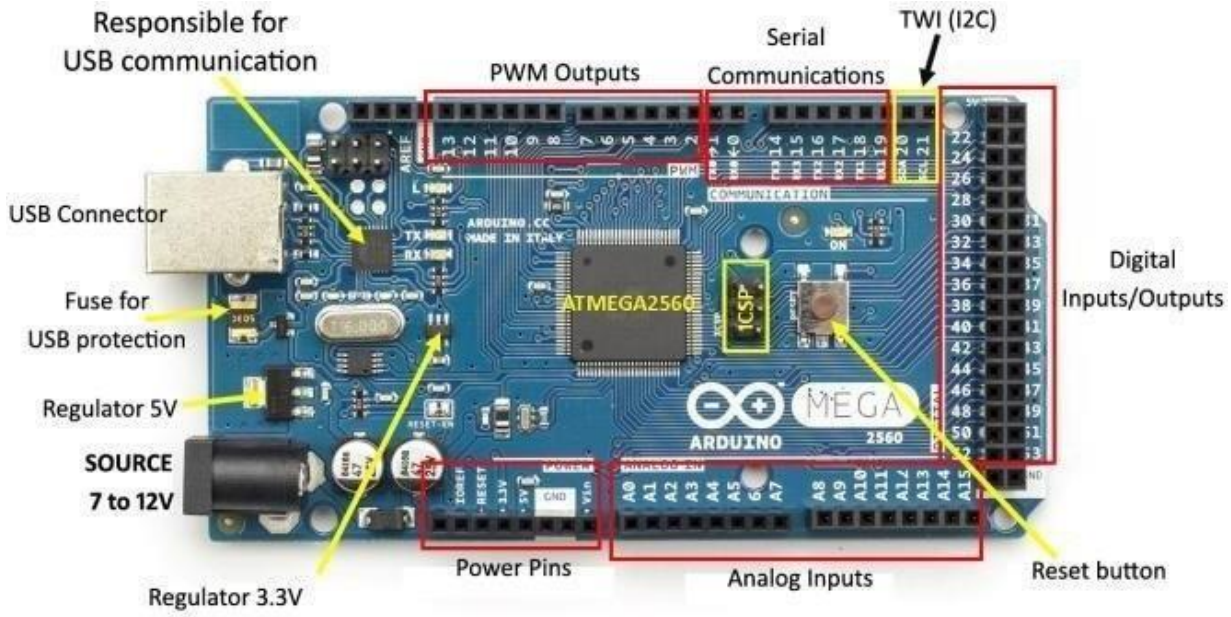

Fig 2.1 Block diagram of Arduino MEGA

2.7.1 POWER

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selectedautomatically.External (non-USB) power can come either from an AC-toDC adapter (wall-wart) or battery. The adapter can be connectedby plugging a 2.1mm centrepositive plug into the board's power jack. Leads from a battery can be inserted in the Gnd andVin pin headers of the POWER connector.The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supplyless than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damagethe board.

The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from theUSB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage viathe power jack, access it through this pin.
- 5V: The regulated power supply used to power the microcontroller and other components on the board. This can comeeither from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3.3V: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND: Ground pins.

2.7.2 MEMORY

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and4 KB of EEPROM (which can be read and written with the EEPROM library).

2.7.3 INPUT & OUTPUT

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead()functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor(disconnected by default) of 20-50 kΩ.

2.7.4 COMMUNICATION

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or othermicrocontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2(ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtualcommunication port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognizethe board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to besent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via

theATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega2560's digital pins. The ATmega2560 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use ofthe TWI bus; see the documentation for details. For SPI communication, use the SPI library.

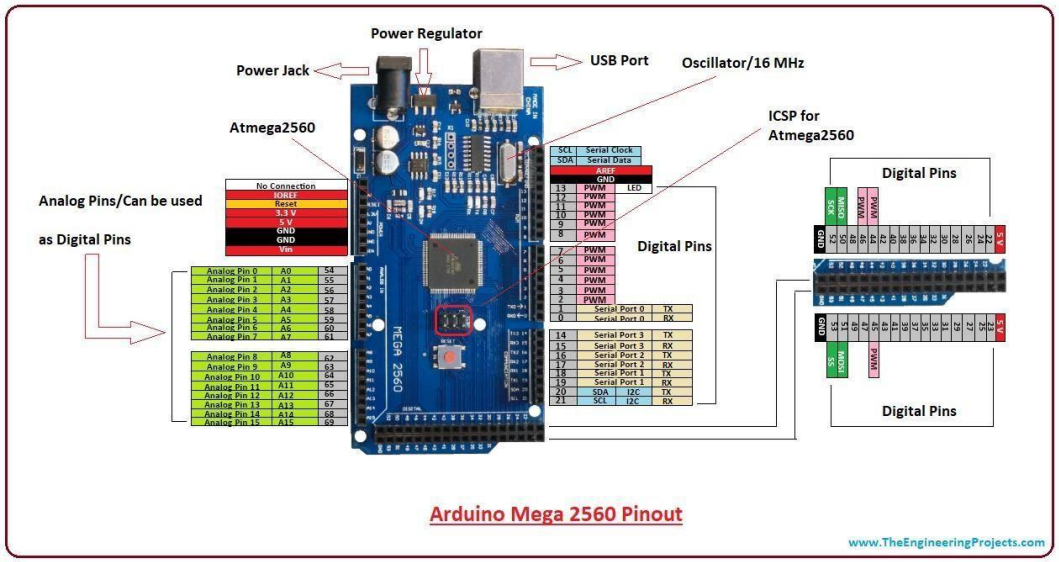

Fig 2.2 Pin diagram of Arduino MEGA

2.7.5 ANALOG PINS

The Mega has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

All analog pins do analog output using PWM. You can't set an analog pin to a specific voltage; PWM just adjusts the duty cycle of a square wave alternating between 0v and +5v so that the average over time appears between 0 and +5v.

The Analog input pins can also be used as Digital Input or Output pins. Just use **A0** through **A15** for a pin name where you would use a digital pin number.

2.7.6 SPECIALIZED FUNCTIONS

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the USB-to-TTL Serial chip.
- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM:** 2 to 13 and 44 to 46, provide 8-bit PWM output with the analogWrite() function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The **SPI** pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I2C/TWI:** 20 (SDA) and 21 (SCL). Support I2C aka IIC aka TWI communication using the Wire library. Note that these pins are not in the same location as the TWI/I2C pins on the Duemilanove or Diecimila.

2.7.7 OTHER PINS ON THE BOARD

- **AREF:** Reference voltage for the analog inputs. Used with analogReference().
- **RESET:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.8 WARNINGS

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is remove.

2.9 APPLICATIONS

Arduino was basically designed to make process of using electronics in multidisplinary projects more accessible. Some of the applications of Arduino MEGA are as follows:

- Developing 3D printer.
- Controlling and handling more than one motors.
- Interfacing of number of sensors.
- Sensing and detecting temperature.
- Water level detection projects.
- Home automation and security systems.
- Embedded Systems.
- IoT(Internet of applications)applications.

CHAPTER-3

HARDWARE COMPONENTS REQUIRED

3.1 ALCOHOL SENSOR:

The alcohol sensor named MQ-3, which detects ethanol in the air.It is one of the straight forward gas sensors and it works almost the same way like other gas sensors. Typically, it is used as part of the breathalyzers or breath testers for the detection of ethanol in the human breath.

3.1.1 INTERNAL DIAGRAM:

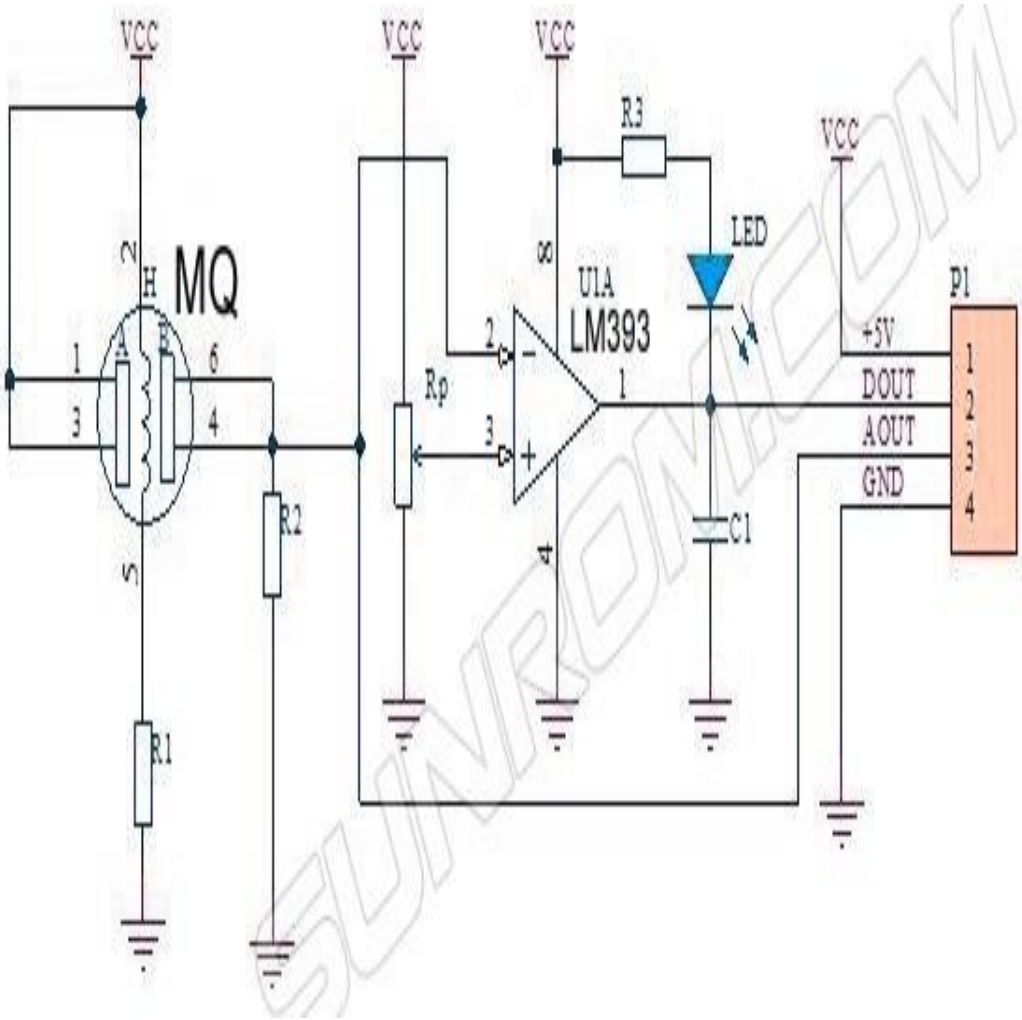

Fig-3.1 internal diagram of alcohol sensor

This module is made using Alcohol Gas Sensor MQ3. 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 your breath, just like your common breathalyzer. 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.

3.1.2 FEATURES

- 5V operation
- Simple to use
- LEDs for output and power
- Output sensitivity adjustable
- Analog output 0V to 5V
- Digital output 0V or 5V
- Low Cost
- Fast Response
- Stable and Long Life
- Good Sensitivity to Alcohol Gas
- Both Digital and Analog Outputs
- On-board LED Indicator

3.1.3 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

3.1.4 PIN OUT

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

3.1.5 APPLICATIONS

- Vehicle Alcohol Detector
- Portable Alcohol Detector

3.2 LCD MODULE

LCD modules are vey commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO's or calculators. The appearance and the pinouts have already been visualized above now let us get a bit technical.

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.

Now, we know that 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 them to display meaningful information onto our LCD Screen. You can learn how to interface an LCD using the above mentioned links. If you are an advanced programmer and would like to create your own library for interfacing your Microcontroller with this LCD module then you have to understand the HD44780 IC is working and commands which can be found its datasheet.The following is the diagram for LCD:

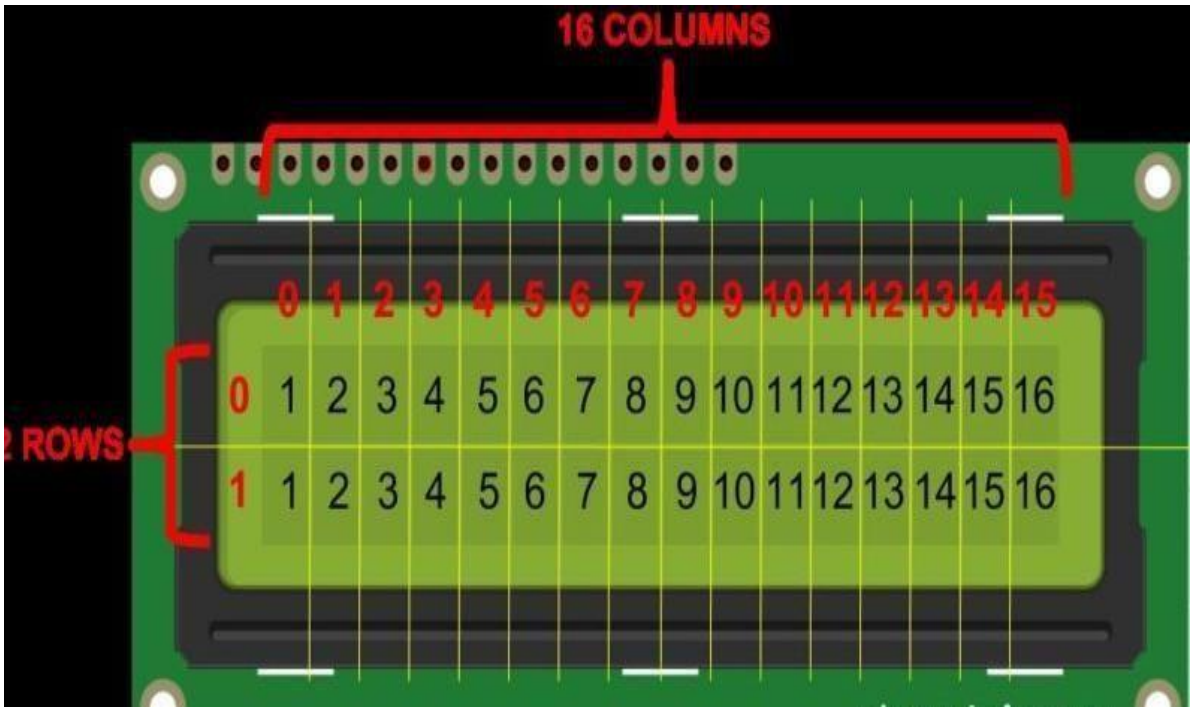

Fig.3.2 Pin description of LCD

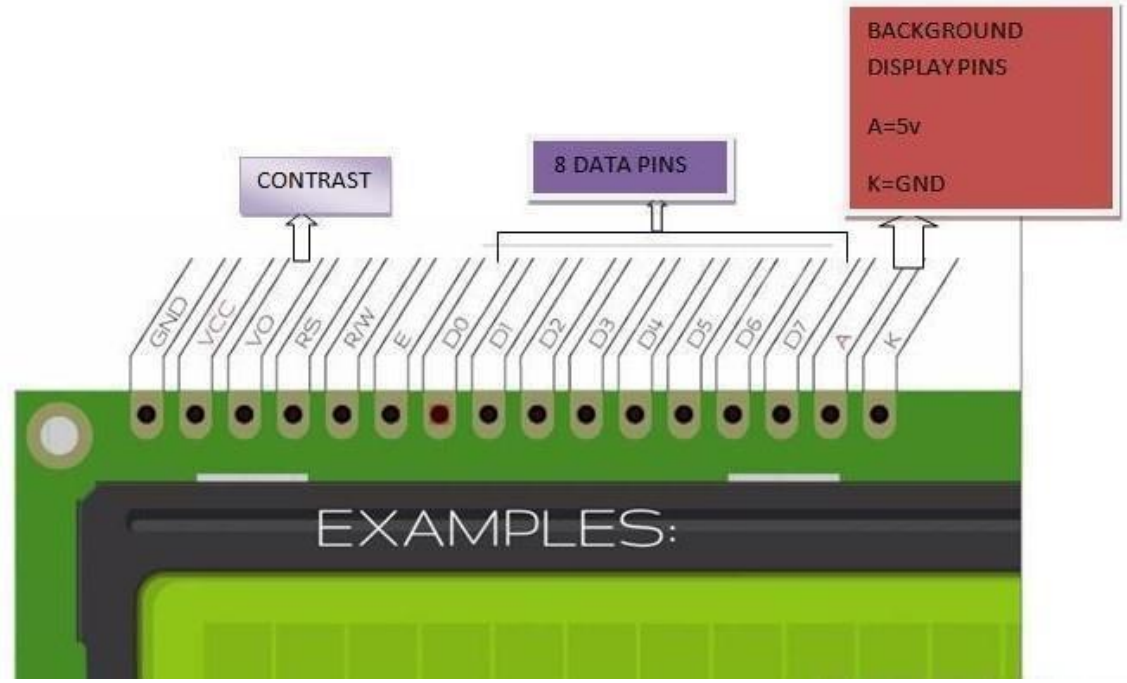

Figure-3.3Pin description of LCD

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.3.1 8-bit parallel pin description

8–Bit Parallel Pin Description

LCD Pin Assignments			
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)

Table.3.2 set of commands used in LCD

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or coursor position on display	0x80+add	128+add

Generating custom characters on LCD is not very hard. It requires the knowledge about custom generated random access memory (CG-RAM) of LCD and the LCD chip controller. Most LCDs contain Hitachi HD4478 controller. CG-RAM is the main component in making custom characters. It stores the custom characters once declared in the code. CG-RAM size is 64 byte providing the option of creating eight characters at a time. Each character is eight byte in size.

CG-RAM address starts from 0x40 (Hexadecimal) or 64 in decimal. We can generate custom characters at these addresses. Once we generate our characters at these addresses, now we can print them on the LCD at any time by just sending simple commands to the LCD. Character addresses and printing commands are below:

Table.3.3 Character addresses and printing commands

CG-RAM Characters	CG-RAM Address (Hexadecimal)	Commands to display Generated Characters
1st Character	0x40	0
2nd Character	0x48	1
3rd Character	0x56	2
4th Character	0x64	3
5th Character	0x72	4
6th Character	0x80	5
7th Character	0x88	6
8th Character	0x96	7

3.3 POTENTIOMETER

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

The potentiometer consists of a long resistive wire L made up of magnum or with constantan and a battery of known EMF V. This voltage is called as **driver cell voltage**. Connect the two ends of the resistive wire L to the battery terminals as shown below; let us assume this is a primary

circuit arrangement. One terminal of another cell (whose EMF E is to be measured) is at one end of the primary circuit and another end of the cell terminal is connected to any point on the resistive wire through a galvanometer G. Now let us assume this arrangement is a secondary circuit

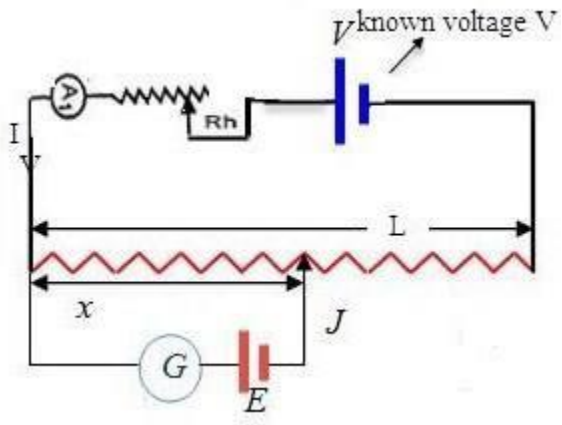

Figure.3.4 arrangement of the potentiometer

3.3.1 CHARACTERISTICS OF POTENTIOMETER

The following are the important characteristics of the potentiometer. The potentiometer is very accurate because its works on the comparing method rather than the deflection pointer method for determining the unknown voltages. It measures the null or balance point which does not require power for the measurement. The working of the potentiometer is free from the source resistance because no current flows through the potentiometer when it is balanced.

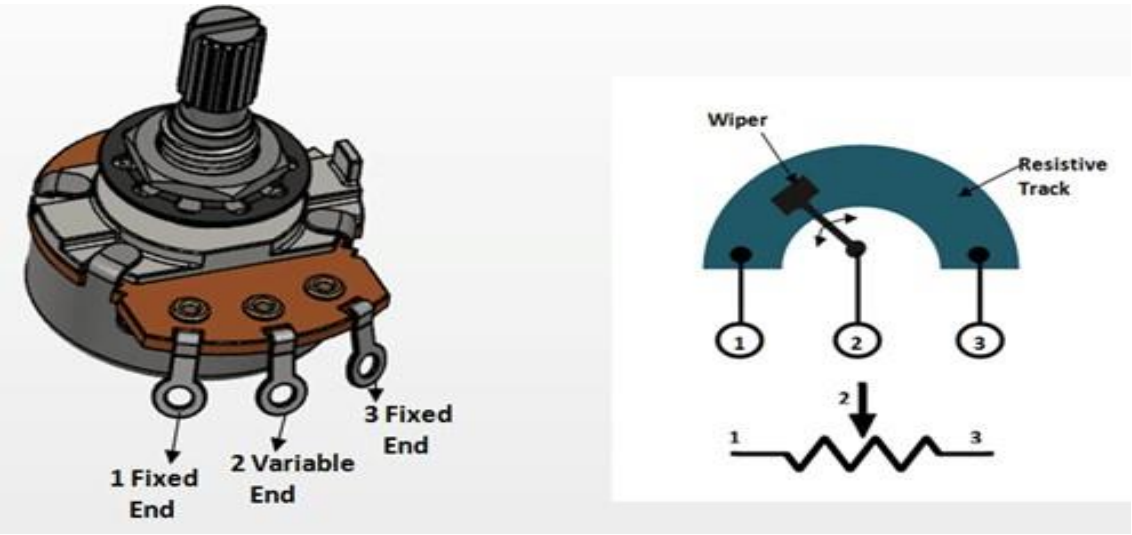

Figure.3.5 Diagrammatic representation of potentiometer

3.3.2 POTENTIOMETER PIN CONFIGURATION

Table.3.4 Pin configuration of potentiometer

PIN NO	PIN NAME	DESCRIPTION
1	Fixed End	This end is connected to one end of resistive track
2	Variable end	This end is connected to the wiper, to provide variable voltage
3	Fixed End	This end is connected to one end of resistive track

3.3.3 FEATURES

- Type: Rotary a.k.a Radio POT
- Available in different resistance values like 500Ω, 1K, 2K, 5K, 10K, 22K, 47K, 50K, 100K, 220K, 470K, 500K, 1 M.
- Power Rating: 0.3W
- Maximum Input Voltage: 200Vdc
- Rotational Life: 2000K cycles

3.3.4 APPLICATIONS

- Voltage and Current Control Circuits
- Used as volume control knobs in radios
- Tuning or controlling circuits
- Analog input control knob

3.4 RELAY

A **relay** is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof.

Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

3.4.1 BASIC DESIGN & OPERATION OF RELAY

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core (a solenoid), an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two contacts in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. The armature is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.
When an electric current is passed through the coil it generates a magnetic field that activates the armature, and the consequent movement of the movable contact(s) either makes or breaks

(depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduces arcing.

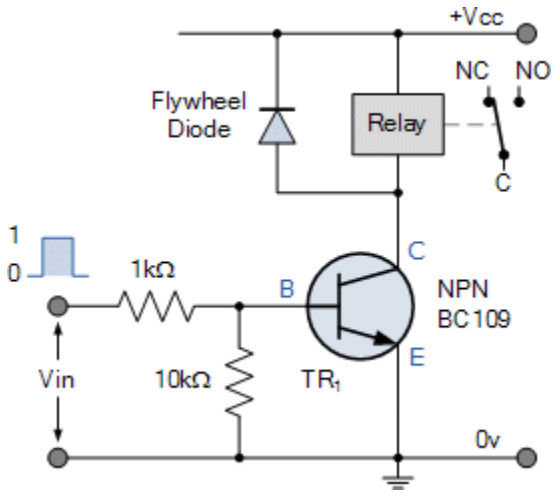

Fig.3.6 Relay internal diagram

3.4.2 INTERFACING OF RELAY WITH ARDUINO

Relays work on electromagnetism, When the Relay coil is energized it acts like a magnet and changes the position of a switch. The circuit which powers the coil is completely isolated from the part which switches ON/OFF, this provides electrical isolation. This is the reason we can control a relay using 5V's from an Arduino and the other end of it could be running an 230V appliance, the 230V end is completely isolated from the 5V Arduino circuitry.

The default state of the relay when the power is off for COMM (power) to be connected to NC (normally closed), this is the equivalent of setting the 4 Relay boards IN pin to HIGH (has +5v sent to it) It is a safety feature to not use the NC connector in-case you Arduino looses power it will automatically turns off all the devices connected to the relay. When you have something connected to the relays NO (Normally Open) connector and you set the corresponding IN pin to

LOW (0v), power will flow in from the COMM connector and out of the NO connect or powering your device.

3.4.3 USES OF RELAY

The main operation of a relay comes in places where only a low-power signal can be used to control a circuit. It is also used in places where only one signal can be used to control a lot of circuits. The application of relays started during the invention of telephones. They played an important role in switching calls in telephone exchanges. They were also used in long distance telegraphy. They were used to switch the signal coming from one source to another destination. After the invention of computers, they were also used to perform Boolean and other logical operations. The high end applications of relays require high power to be driven by electric motors and so on. Such relays are called contactors.

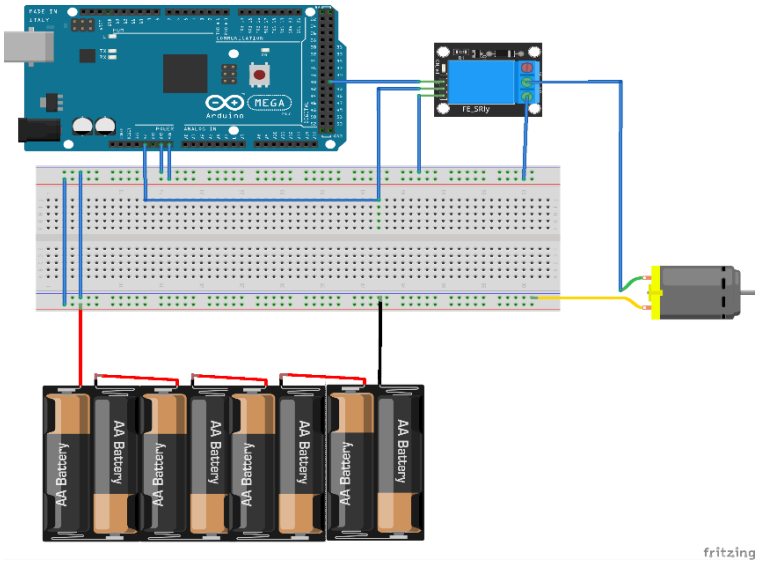

Fig.3.7 Relay interfacing with Arduino

3.5 GPS MODULE

In this project we have used NEO 6m GPS module. GPS receivers actually work by figuring out how far they are from a number of satellites. They are pre-programmed to know where the GPS satellites are at any given time. The function of the GPS module is to calculate the speed and location of the vehicle.

The satellites transmit information about their position and the current time in the form of radio signals towards the Earth. These signals identify the satellites and tell the receiver where they are located.

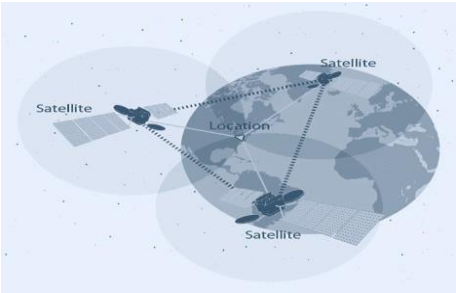

Fig.3.8 Trilateration method diagrammatic representation

The receiver then calculates how far away each satellite is by figuring out how long it took for the signals to arrive. Once it has information on how far away at least three satellites are and where they are in space, it can pinpoint your location on Earth. This process is known as Trilateration.

3.5.1 Hardware Overview of NEO-6M GPS Module

It can track up to 22 satellites on 50 channels and achieves the industry's highest level of sensitivity i.e. -161 dB tracking, while consuming only 45mA supply current.

Unlike other GPS modules, it can do up to 5 location updates a second with 2.5m Horizontal position accuracy. The u-blox 6 positioning engine also boasts a **Time-To-First-Fix** (TTFF) of under 1 second.

One of the best features the chip provides is Power Save Mode(PSM). It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just **11mA** making it suitable for power sensitive applications like GPS wristwatch.

The necessary data pins of NEO-6M GPS chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module

supports baud rate from 4800bps to 230400bps with default baud of 9600. Here are complete specifications:

Table.3.5 complete specifications of neo-6m GPS module

Receiver Type	50 channels, GPS L1(1575.42Mhz)
Horizontal Position Accuracy	2.5m
Navigation Update Rate	1HZ (5Hz maximum)
Capture Time	Cool start: 27sHot start: 1s
Navigation Sensitivity	-161dBm
Communication Protocol	NMEA, UBX Binary, RTCM
Serial Baud Rate	4800-230400 (default 9600)
Operating Temperature	-40°C ~ 85°C
Operating Voltage	2.7V ~ 3.6V
Operating Voltage	45mA
TXD/RXD Impedance	510Ω

3.5.2 Position Fix LED Indicator

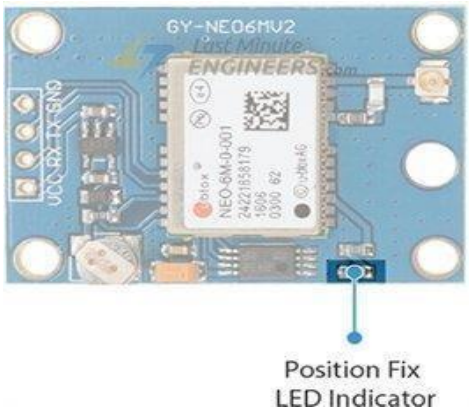

Fig.3.9 position fix led indicator

There is an LED on the NEO-6M GPS Module which indicates the status of Position Fix. It'll blink at various rates depending on what state it's in:

- **No Blinking** – It's searching for satellites. • **Blink every 1s** – Position Fix is found(The module can see enough satellites).

3.5.3 3.3V LDO Regulator

Fig.3.10 3.3V LDO Regulator

The operating voltage of the NEO-6M chip is from 2.7 to 3.6V. But the good news is that, the module comes with MIC5205 ultra-low dropout 3V3 regulator from MICREL.

The logic pins are also 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

3.5.4 Battery & EEPROM

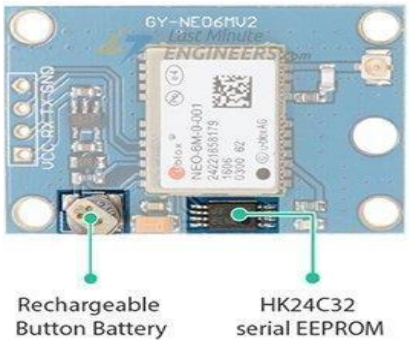

Fig.3.11 Battery & EEPROM

The module is equipped with an **HK24C32** two wire serial EEPROM. It is 4KB in size and connected to the NEO-6M chip via I2C.

The module also contains a rechargeable button battery which acts as a super-capacitor.

An EEPROM together with battery helps retain the **battery backed RAM** (BBR). The BBR contains clock data, latest position data(GNSS orbit data) and module configuration. But it's not meant for permanent data storage.

As the battery retains clock and last position, time to first fix (TTFF) significantly reduces to 1s. This allows much faster position locks.

Without the battery the GPS always cold-start 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.

3.5.5 ANTENNA

An antenna is required to use the module for any kind of communication. So, the module comes with a patch antenna having -161 dBm sensitivity.

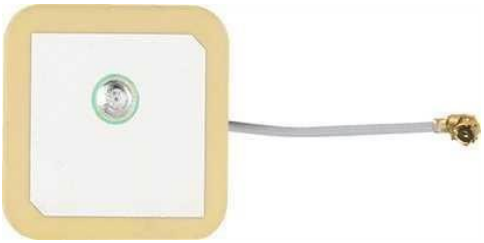

Fig.3.12 Antenna We can snap-fit this antenna to small U.FL connector located on the module.

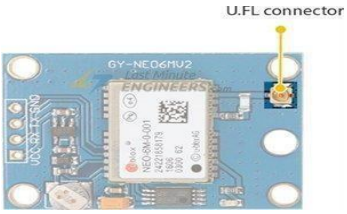

Fig.3.13 U.FL connector

Patch antenna is great for most projects. But if you want to achieve more sensitivity or put your module inside a metal case, you can also snap on any 3V active GPS antenna via the U.FL connector.

NOTE

U.FL connectors are small, delicate and are not rated for strain. To prevent damaging the U.FL connection, we recommend threading the U.FL cable through the mounting hole, then attach the U.FL connectors.

3.5.6 NEO-6M GPS Module Pinout

The NEO-6M GPS module has total 4 pins that interface it to the outside world. The connections are as follows:

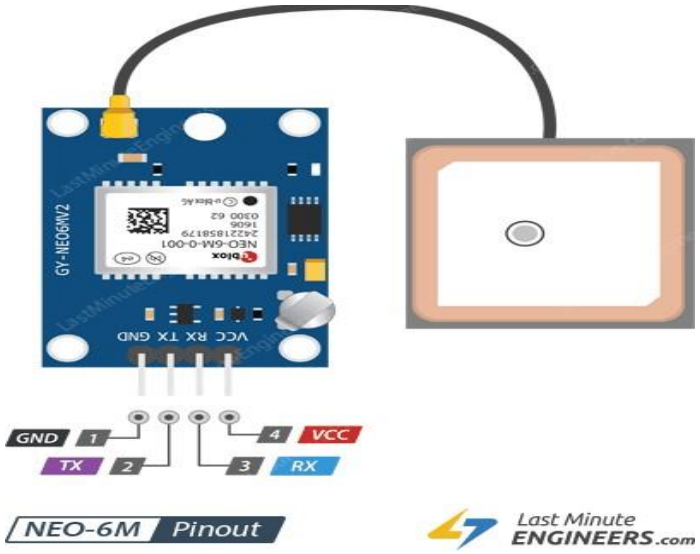

Fig.3.14 NEO-6M GPS Module Pinout

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

TxD (Transmitter) pin is used for serial communication. RxD (Receiver) pin is used for serial communication.

VCC supplies power for the module. We can directly connect it to the 5V pin on the Arduino.

3.6 GSM MODULE

In this project we are using sim800L.The SIM800L module is a complete Quad-band GSM/GPRS solution in a LGA type which can be embedded in the customer, it has a set of TTL level serial interface, a set of power supply interface. Besides, there are a set of antenna interface on this module. The function of GSM module is to send the SMS to the respected authorities given in the number when the speed is more than threshold value.

3.6.1 Features:

- Supply voltage range 3.4 ~ 4.4V and
- Current of 1A or more (the current is very important)
- It features Bluetooth, FM and Embedded AT (AT commands)
- Quad-band 850/900/1800/1900MHz
- Operation temperature: -40 ~85 degree Celsius

3.6.2 Pinout Description:

There are 7 total pins on the SIM800L, which we are going to use to interface with Arduino.

VCC: External Supply Voltage input for SIM800L

GND: External Ground for SIM800L

VDD: Microcontroller Supply voltage input for SIM800L

RST: Reset pin for SIM800L

RXD: Serial communication (Receiver Pin)

TXD: Serial communication (Transfer Pin)

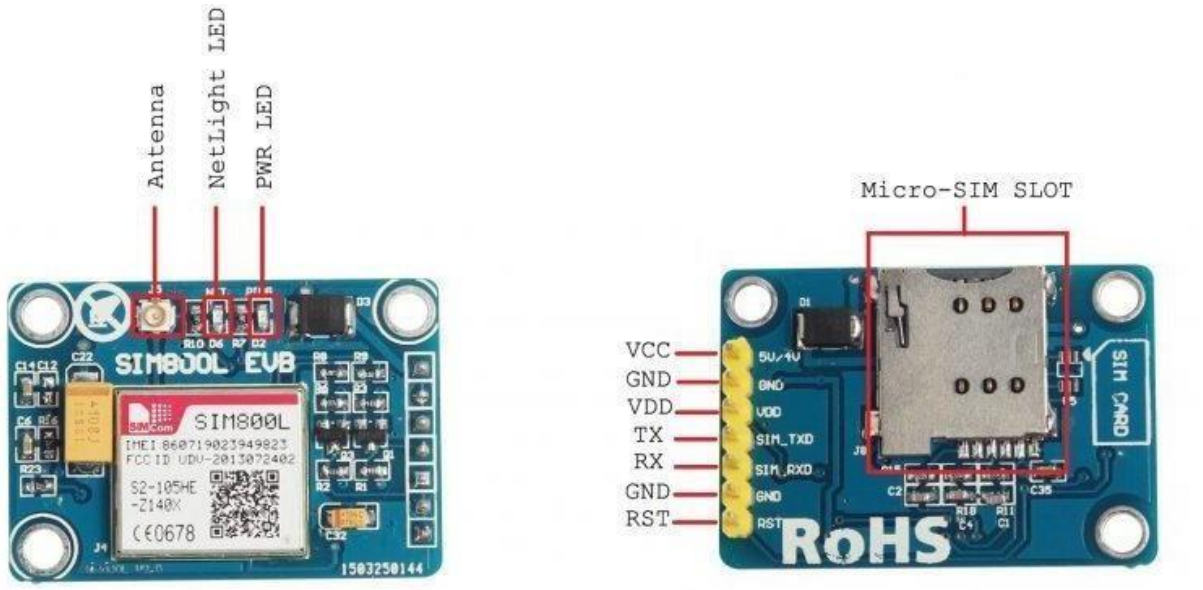

Fig.3.15 Pinout description of SIM800L

3.7 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 slight characteristic 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.

Fig.3.16 Ethanol

5.4 LIBRARIES

The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g.working with hardware or manipulating data.

Libraries are a collection of code that makes it easy for you to connect sensor, display, module, etc. There are several library functions like standard library functions, Ethernet library functions, user library functions which make the Arduino to connect to internet.

CHAPTER-4

CIRCUIT OPERATION

4.1 BLOCK DIAGRAM

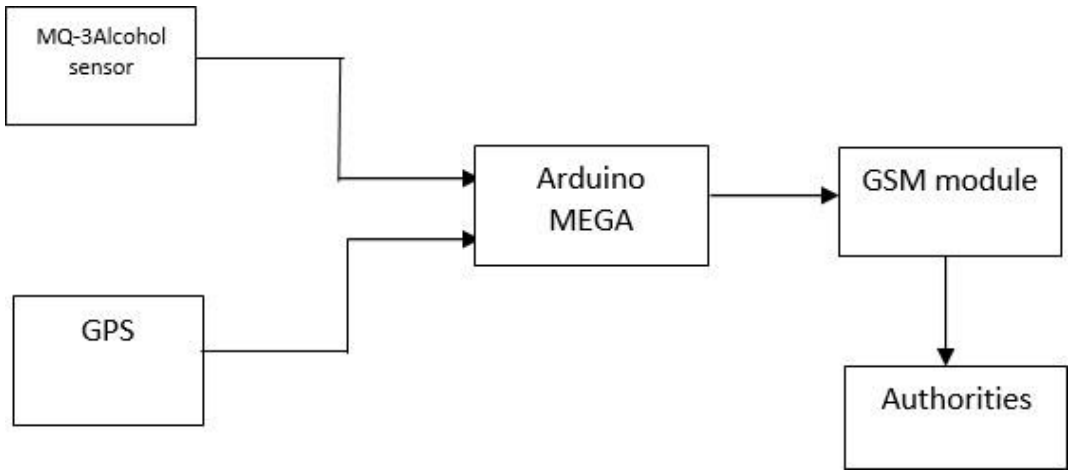

Fig.4.1 Block diagram of entire system

4.2 WORKING

Firstly, the alcohol sensor senses the alcohol level of the person, this information is sent through Arduino MEGA. Now it compares with the value in the program if it is more than the threshold value the bike doesn't start else the bike starts. Now while travelling the GPS continuously calculates the location and speed and sends it to Arduino MEGA and Arduino MEGA compares with threshold value and if it crosses the speed then the Arduino MEGA sends an instruction to GSM and now the GSM module sends the message to the respective authorities which includes the latitude, longitude and the speed.

4.3 FLOW CHART

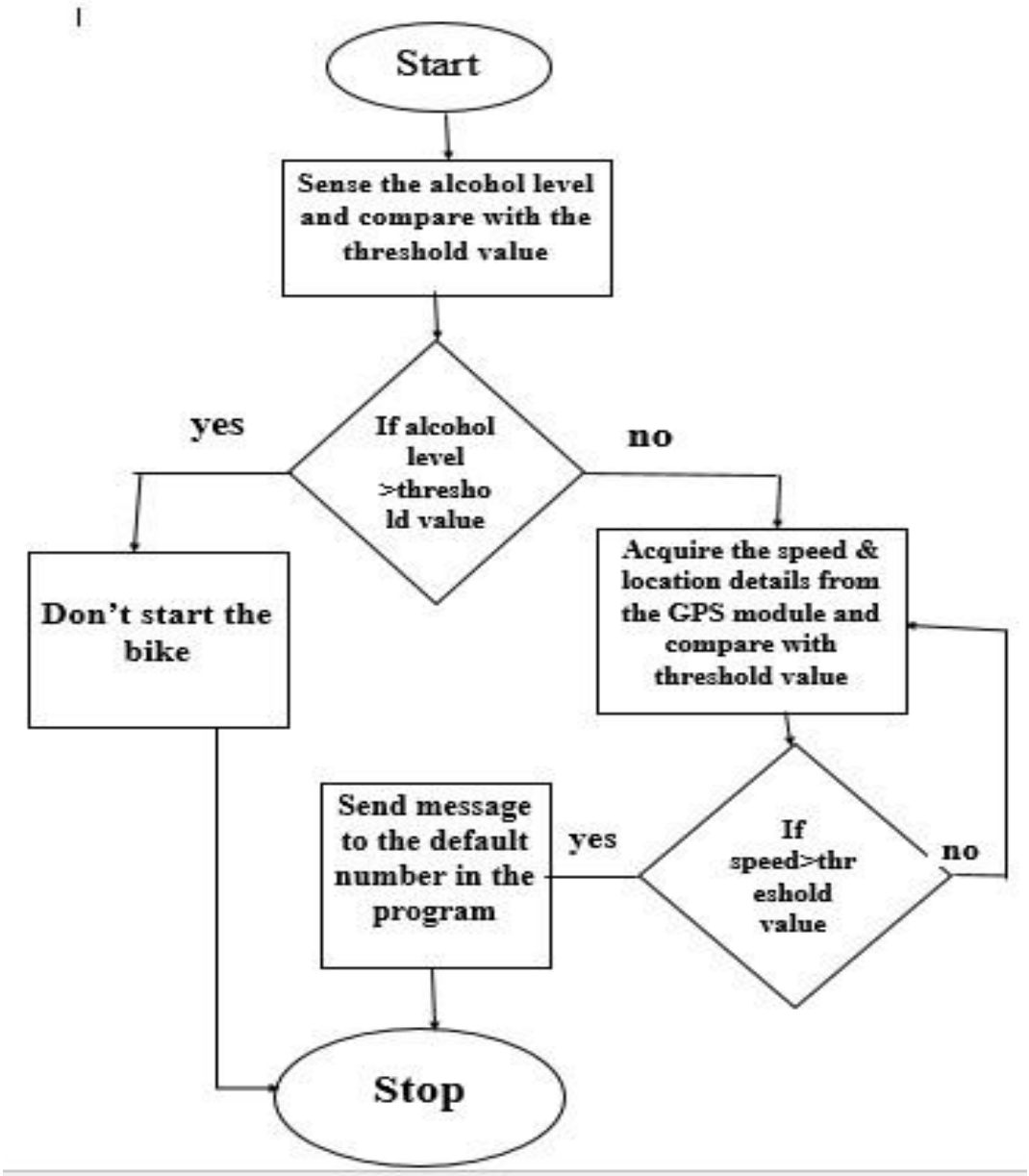

Fig.4.3 Flow chart of the device

4.4CIRCUIT DIAGRAM

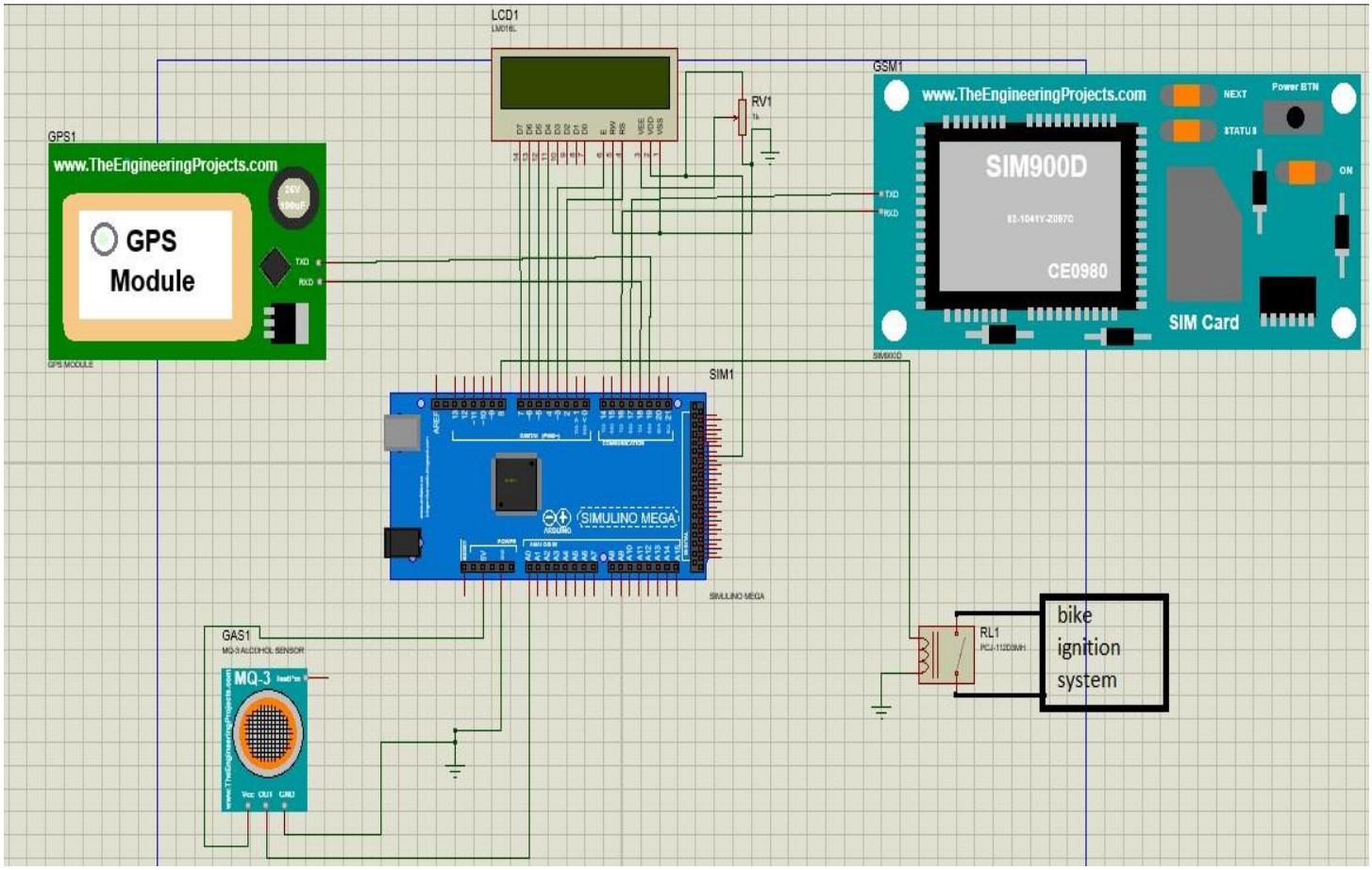

Fig.4.4 Circuit diagram

CONCLUSION & FUTURE SCOPE

The work helps in reducing the road fatalities due to alcohol consumption and rash driving. By installing such device, the ignition of the vehicle is also controlled i.e. the vehicle doesn't start if the driver consumes alcohol and the speed of the driver is continuously supervised and by sending alerts to authorities while crossing the limit promotes the road safety not only to the people driving but also the pedestrians. Thus the work reduces the road fatalities caused by two major reasons which are alcohol consumption and rash driving.

The proposed system deals with the detection of the accidents. But this can be extended by providing medication to the victims at the accident spot. By increasing the technology, we can also avoid accidents by providing alerts systems that can stop the vehicle to overcome the accidents.