CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO AUTOMATIC ENGINE LOCK SYSTEM

Now a day's every system is automated in order to face new challenges. In the present days Automated systems have less manual operations, flexibility, reliability and accurate. Due to this demand every field prefers automated control systems. Especially in the field of electronics automated systems are giving good performance.

We usually come across drink and driving cases where drunk drivers crash their cars under the influence of alcohol causing damage to property and life. So here we propose an innovative system to eliminate such cases. Our proposed system would be constantly monitoring the driver breath by placing it on the driver wheel or somewhere the driver’s breath can be constantly monitored by it.

So if a driver is drunk and tries to drive the system detects alcohol presence in his/her breathe and locks the engine so that the vehicle fails to start. In another case if the driver is not drunk while he starts the vehicle and engine is started but he/she drinks while driving the sensor still detects alcohol in his breath and stops the engine so that the vehicle would not accelerate any further.

In this system we use a microcontroller interfaced with an alcohol sensor along with an LCD screen and a dc motor to demonstrate the concept. So here the alcohol sensor is used to monitor uses breath and constantly sends signals to the microcontroller.

Nowadays most road accidents are caused due to drunken driving. To reduce these accidents Government of India introduced laws prohibiting drivers from drunk so that the fine can stop them from drunk and driving. The drivers who drink alcohol are not in a stable state and also rash driving occurs on highways which will be risky to the people lives who are on the road The effective observation of drunken drivers could be a challenge to the police and road safety officers.

The restricted ability of enforcement agents undermines each manual effort geared toward edge drink-driving. Therefore there is a need for an alcohol detection system that can function without the restriction of space and time. Nowadays automated systems have fewer manual operations, flexible and accurate. In this system, we used an Arduino Uno interface with an alcohol sensor. In case, if the driver starts drinking alcohol engine starts then the MQ-3 sensor detects the concentration from the driver’s breath and if it exceeds the threshold level then the system displays an alcohol detection note on the LCD screen, the engine stops and the car does not accelerate further. In another case, if the driver is in a drunk state before the engine starts then the engine does not accelerate anymore.

After locking the engine the system sends the message to the vehicle owner using GSM and also sends the location using the GPS module. According to the survey, every day more than 50 people die because of road accidents among these people 70% die due to drunken driving. Nearly every 30 seconds one person dies because of a road accident.

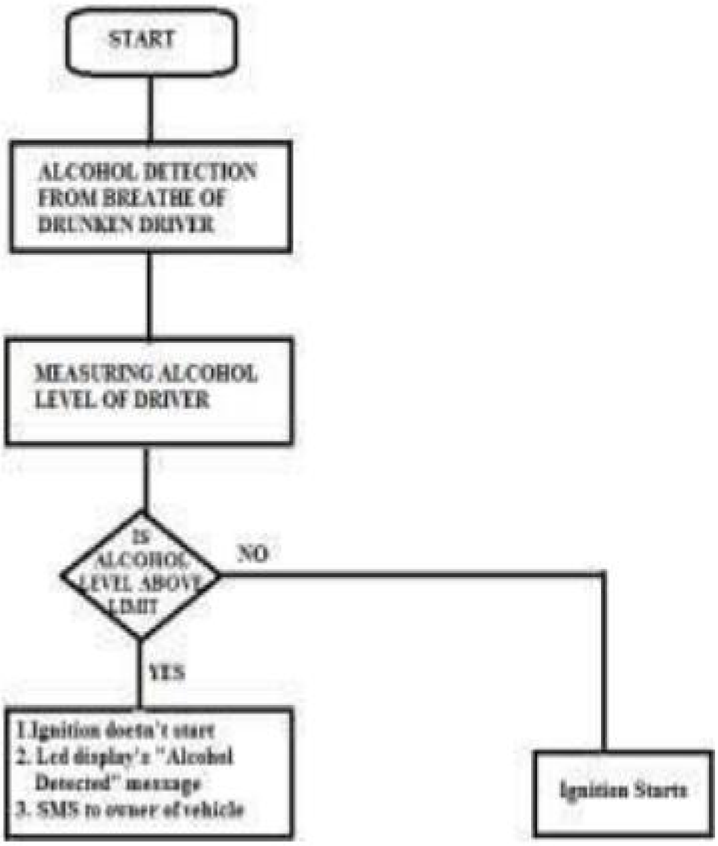


Fig. 1.1: Flow Chart of Alcohol Detection

1.2 EMBEDDED SYSTEMS

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own.) Since the embedded system is dedicated to specific tasks, design engineers can optimize it

to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded". A modern example of embedded system is shown in fig: 1.2.

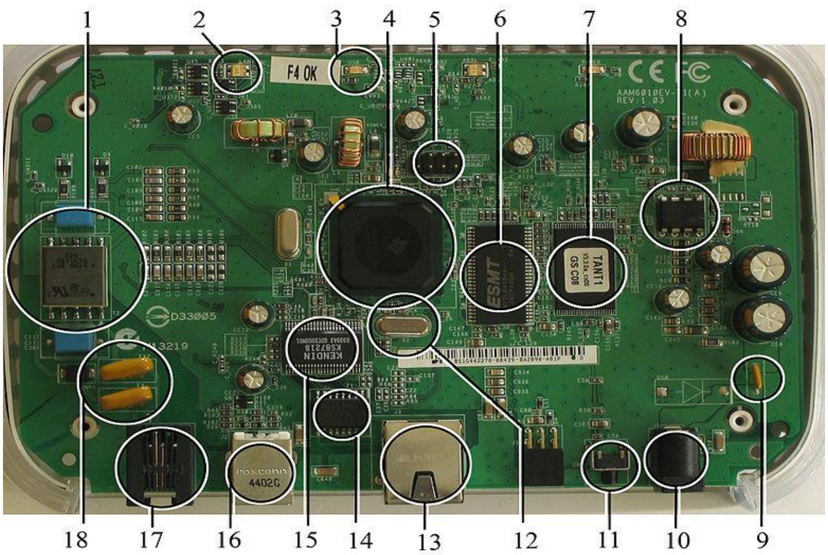


Fig. 1.2: A modern example of embedded system

Label parts include microprocessor (4), RAM (6), flash memory (7). Embedded systems programming is not like normal PC programming. In many ways, programming for an embedded system is like programming PC 15 years ago. The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

1.2.1 History

In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology.

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961. It was built from transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits.

1.2.2 Tools

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort.

Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint when your system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

1.2.3 Resources

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start, and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well.

Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC

programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources

programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether.

1.2.4 Applications of Embedded systems

- **Consumer applications:**

At home we use a number of embedded systems which include microwave oven, remote control, vcd players, dvd players, camera etc….

Fig. 1.2.4: Automatic coffee makes equipment

CHAPTER 2

LITERATURE SURVEY

In this a technique which utilizes GPS and GSM to ascertain alcohol but this technique is very expensive, but the expenses can be cut off to a great extent.

Wearing smart helmet to prevent any mishap is suggested by writer which have certain deficiencies. First restrictions on the use of helmets to only 2 wheelers.

A. The writer has put forward a technique which utilizes GPS and GSM to ascertain alcohol but this technique is very expensive, but the expenses can be cut off to a great extent.

In this project a siren is being used which is highly economical, and can keep people in close proximity vigilant. [1] B.

Wearing smart helmet to prevent any mishap is suggested by writer which have certain deficiencies. Firstly restrictions on the use of helmets to only 2 wheelers. Secondly, microcontrollers are software based mega system in comparison to the economical siren that are open source hardware. [2] C.

Composite health monitoring and sensors based on infrared are utilized to ascertain alcohol as talked about by writer but the chance of false alarm can't be avoided in this system, because minute change in some situations can result in false alarm but in our project use of required technology makes it more authentic. [3] D.

To prevent the mishap of drunken driving writer have used PIC16F877A microcontroller which is an outdated system and expensive one also which restrains its use to only certain class of society whereas we are using Arduino and Uno microcontroller which is advanced as well as economical. [4] E.

Worrying about the drunken driving the writer suggests the system to overcome the issue but using mQ2 alcohol sensor has come flames .MQ2 alcohol sensor is not authentic and raises the chance of false alarm while we have used MQ3 which is highly authentic. [5]

A. The author has put forward a procedure that utilizes GPS and GSM to find out liquor but this strategy is exceptionally costly, but the costs can be cut off to a extraordinary degree. In this extend, a siren is being utilized which is profoundly temperate and can keep individuals in near vicinity careful. [1] B.

Wearing a savvy head protector to anticipate any mishap is recommended by the author which have certain insufficiencies. Firstly limitations on the utilize of head protectors to as it were 2 wheelers. Furthermore, microcontrollers are software-based mega frameworks in comparison to the conservative siren that are open source equipment. [2] C.

Composite wellbeing observing and sensors based on infrared are utilized to find out liquor as talked almost by the author but the chance of untrue caution can't be dodged in this framework, since miniature changes in a few circumstances can result in wrong caution but in our extend utilize of required innovation makes it more true. [3] D.

To avoid the incident of tanked driving essayist have utilized the PIC16F877A microcontroller which is an obsolete framework and costly one too which limits its utilize to as it were certain lesson of society while we are utilizing Arduino and Uno microcontroller which is progressed as well as temperate. [4] E.

Stressing approximately the intoxicated driving the author proposes the framework to overcome the issue but utilizing mQ2 liquor sensor has come flames.MQ2 liquor sensor isn't bona fide and raises the chance of untrue caution whereas we have utilized MQ3 which is exceedingly bona fide. [5]

A. The writer has put forward a technique which utilizes GPS and GSM to ascertain alcohol but this technique is very expensive, but the expenses can be cut off to a great extent. In this project a siren is being used which is highly economical, and can keep people in close proximity vigilant. [1] B.

Wearing smart helmet to prevent any mishap is suggested by writer which have certain deficiencies. Firstly restrictions on the use of helmets to only 2 wheelers. Secondly, microcontrollers are software based mega system in comparison to the economical siren that are open source hardware. [2] C.

Composite health monitoring and sensors based on infrared are utilized to ascertain alcohol as talked about by writer but the chance of false alarm can't be avoided in this system, because minute change in some situations can result in false alarm but in our project use of required technology makes it more authentic. [3] D.

To prevent the mishap of drunken driving writer have used PIC16F877A microcontroller which is an outdated system and expensive one also which restrains its use to only certain class of society whereas we are using Arduino and Uno microcontroller which is advanced as well as economical. [4] E.

Worrying about the drunken driving the writer suggests the system to overcome the issue but using mQ2 alcohol sensor has come flames .MQ2 alcohol sensor is not authentic and raises the chance of false alarm while we have used MQ3 which is highly authentic. [5]

A literature survey on automatic engine locking systems through alcohol detection would typically involve gathering and reviewing academic papers, journal articles, conference proceedings, patents, and other relevant sources that discuss research, development, and implementation of such systems. Here's a hypothetical outline for conducting a literature survey on this topic:

1. Introduction to Alcohol Detection Systems

Overview of the problem of drunk driving and the importance of implementing effective solutions. - Introduction to automatic engine locking systems and their potential role in preventing drunk driving accidents.

2. Review of Existing Technologies

Survey of current technologies used for alcohol detection in vehicles. - Discussion of breathalyzer-based systems, touch-based sensors, and other emerging technologies. - Evaluation of their effectiveness, accuracy, and limitations.

3. Research Papers and Studies

Examination of academic papers and studies that investigate the development and performance of alcohol detection systems. - Analysis of experimental methodologies, results, and conclusions. - Identification of gaps in current research and potential areas for further exploration.

4. Legislation and Regulations

Review of laws and regulations governing the use of alcohol detection systems in vehicles. - Comparison of regulations across different regions and countries. - Discussion of legal and ethical considerations associated with implementing such systems.

5. Technological Challenges and Solutions

Exploration of technical challenges in developing reliable alcohol detection systems for vehicles. - Examination of innovative approaches and technological advancements aimed at addressing these challenges. - Case studies of successful implementations or real-world applications

2.1 ALCOHOL BREATH TESTER

In previous method we have a device which is used to detect the alcohol rate by using a small device which is carried by hand. With the help of it the traffic polices are just testing the rate of drunk by the person and implementing a charge on that person regarding the amount what he/she is drunk. In this existing system, alcohol detectors are not proposed inbuilt in a car. The traffic police uses alcohol detectors (device) to avoid drunk and drive accidents. The limitation of this device is that the police are not able to check each & every car & even if they stop some suspects there are chances that the police can be easily bribed. Therefore, there is a need for an effective system to check drunken drivers. In previous technologies can do only detecting the alcohol, If the person drinks the alcohol the engine will be locked.

Fig. 2.1: Alcohol Breath Tester

Ultrasonic Sensor Approach:

A project presented an Alcohol Detection with Engine Locking System for cars using an Ultrasonic Sensor and Arduino UNO. The system continuously monitors the alcohol concentration and turns off the engine if it exceeds the threshold level3.

Impact and Statistics:

According to surveys, more than 50 people die daily due to road accidents, with 70% of these fatalities attributed to drunken driving. Swift action through automatic engine locking can save lives and prevent property damage1.Another prototype system was developed using an Arduino Uno microcontroller interfaced with an alcohol sensor, an LCD screen, and a DC motor.

The system continuously monitors the blood alcohol content (BAC) by detecting alcohol in the driver’s exhalation. If alcohol is detected, the engine is automatically locked to prevent further acceleration2. The development of an automatic engine locking system through alcohol detection typically involves several key components and methodologies. Here's an overview of the existing methodology commonly used in such projects:

Alcohol Detection Technology:

This is one of the most common methods used for alcohol detection. It involves the use of sensors to measure the concentration of alcohol in a person's breath.
Ignition Interlock Devices (IIDs): These devices are installed in the vehicle and require the driver to blow into a breathalyzer before starting the engine. If alcohol is detected above a certain threshold, the engine will not start. Other Sensor Technologies: Some systems use advanced sensor technologies to detect alcohol vapors in the vehicle cabin or to analyze the driver's behavior for signs of impairment.

Hardware Components:

This could be a semiconductor-based sensor, a fuel cell sensor, or another type of sensor capable of
detecting alcohol vapors or breath alcohol concentration. Microcontroller or Microprocessor: A microcontroller or microprocessor is used to control the operation of the system, including reading sensor data, processing it, and controlling the engine locking mechanism.

A solenoid or relay is used to physically control the vehicle's ignition system, either allowing or preventing the engine from starting based on the alcohol detection results.

User Interface: This could include an LCD display, LED indicators, or auditory alerts to communicate with the driver and indicate the status of the alcohol detection system.

Software Development:

Firmware Development: Software is developed to run on the microcontroller or microprocessor to implement the logic of the alcohol detection system. This includes reading sensor data, applying calibration algorithms, and controlling the engine locking mechanism.

User Interface Software: If the system includes a user interface, software is developed to manage user interactions, display information, and provide feedback to the driver.

Integration with Vehicle Systems:

Wiring and Integration: The alcohol detection system needs to be integrated with the vehicle's electrical system, typically by connecting to the ignition circuit.

Compatibility Testing: Ensure that the system is compatible with the vehicle's make and model and does not interfere with other vehicle systems or safety features.

Testing and Validation:

Laboratory Testing: The system undergoes testing in controlled laboratory conditions to validate its accuracy, reliability, and performance. Field Testing: Prototype systems are installed in vehicles and tested in real-world conditions to assess their effectiveness and identify any potential issues.

Calibration and Optimization: The system may require calibration and optimization to minimize false positives and false negatives and ensure reliable operation.

Regulatory Compliance:

Ensure that the system complies with relevant regulatory requirements and standards for automotive safety and alcohol detection devices. Address any legal or liability considerations associated with the use of the system, such as privacy concerns or potential misuse.

Deployment and Support:

Once the system has been thoroughly tested and validated, it can be deployed in vehicles for use by consumers or in commercial fleets.
Provide user training and support to ensure that drivers understand how to use the system effectively and safely.

Offer ongoing maintenance and support to address any issues that may arise during use. By following these methodologies, developers can create automatic engine locking systems through alcohol detection that are effective, reliable, and compliant with relevant regulations and standards.

2.1.1 LIMITATIONS

- Accuracy can be compromised due to calibration issues and individual variations in metabolism.
- External factors like medications or environmental substances can interfere with readings.
- Detection threshold may lead to missed low alcohol levels, giving false negatives.
- Calibration needs regular attention, which can be time-consuming and requires expertise.
- Some systems have delays in providing results, impacting real-time applications.
- Variability in metabolism affects how alcohol is processed, influencing readings.
- Standardizing legal procedures for calibration and result interpretation is challenging.

CHAPTER 3

AUTOMATIC ENGINE LOCKING SYSTEM THROUGH ALCOHOL DETECTION

3.1 Block Diagram

In Our proposed solution MQ3 alcohol sensor will continuously monitoring the alcohol. When the alcohol level reaches trigger point then the MQ3 sensor sends the signal to the Arduino. Then Arduino sends signals to engine then automatically engine will be locked. Finally live location of the drunken person sends to the particular person which have our applications through GPS.

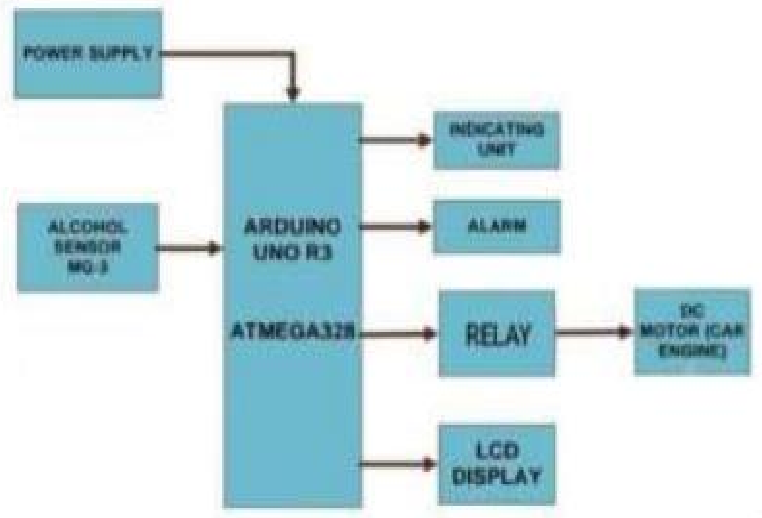


Fig. 3.1: Block Diagram of Alcohol Detection

In this system when the sensor sense the alcohol smell then it send information to the relay module and the relay module stops the moving of wheels and it sends the information to the registered mobile numbers that what we had entered in the code. By this we can reduce the chances of accidents.

The Automatic Engine Locking System through Alcohol Detection is designed to prevent accidents caused by drunken driving. It combines various components, including an alcohol breath analyzer sensor (MQ-3), an Arduino Uno microcontroller, ultrasonic sensors, GPS modules, GSM modules, an LCD screen, and a DC motor. The system operates as follows: when the driver starts the engine, the ultrasonic sensor checks for the driver's presence. If the driver is detected, the MQ-3 sensor continuously measures alcohol levels. If the alcohol concentration exceeds a predefined threshold, the system immediately locks the engine and displays an alcohol detection message on the LCD screen. Simultaneously, it sends emergency alerts to the vehicle owner via GSM, providing the vehicle's location using GPS. This proactive safety measure aims to curb accidents caused by intoxicated driving, ultimately enhancing road safety.

Project Planning and Requirements Gathering

Define the scope of the project and establish clear objectives. Gather requirements from stakeholders, including potential users, regulatory bodies, and automotive safety experts. Identify technical specifications, such as the desired detection accuracy, response time, and compatibility with different vehicle models.

Research and Technology Selection

Conduct thorough research on existing alcohol detection technologies and their suitability for integration into a vehicle's ignition system. Evaluate the advantages and limitations of different sensor technologies, such as breath alyzers, fuel cell sensors, or infrared spectroscopy. Consider factors such as cost, accuracy, reliability, size, and power consumption when selecting the appropriate technology for the project.

Hardware Design and Prototyping

Design the hardware components necessary for the alcohol detection system, including the sensor module, microcontroller or microprocessor, relay/solenoid, and user interface. Develop a prototype of the hardware system, incl

3.2 DESCRIPTION OF COMPONENTS

The components used in our system are:

- ARDUINO UNO
- Mq3 Alcohol Sensor
- Antenna module
- Sim module
- Relay module

- DC motor module

3.2.1 ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATMEGA328P. It has 14 digital input/output pins of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USBcable or power it with a AC-to-DC adapter or battery to get started.

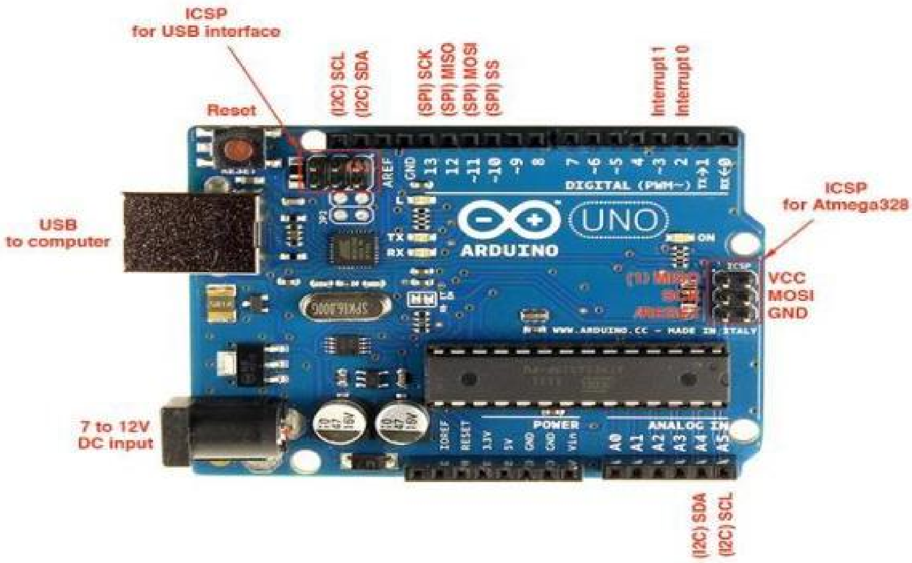


Fig. 3.2.1: Arduino Overview

The Uno differs from all preceding boards in that it does not use the USB-to-serial driverchip. Instead, it features the ATMEGA16U2 programmed as a USB-to-serial converter.

The board has the following new features:

- It has a voltage controller
- It is a RISC based microcontroller.

- It has 32KB of memory and 2KB of RAM and 1KB of EEPROM.
- It has 3 ports 1. Port B (8 pins), 2. Port C (8 pins), 3. Port D (7 pins). Totally it has23

- GIOS (General purpose input output pin.
- It has UART, SPI, I2C protocols.
- It can execute 2 million instructions per second.

- It has two ADC pins A0 & A6.
- It has 6 PWM channels, these pins are used to adjust the clock pulse.
- It has serial connection (sensors, GPS--------).
- Digital I/O pins (LED‘s, Switches).
- Analog I/O pins (for resistive sensor data).
- The USB port used is A type or B type ports. There are so many types of ports. In this arduino board we used the b type port.
- The power to the microcontroller in the arduino is from both power supply unit andthe USB port. So, if at a time we give both then the controller does not know where the power is from. To avoid that a comparator is used to differ the power supply and the USB port.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward the Uno is thelatest in a series of USB Arduino boards.

3.2.2 POWER SUPPLY

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin 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 supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommendedrange is 7 to 12 volts.

Memory:

The ATMEGA328P has 32 KB ROM (with 0.5 KB used for the bootloader). It also has 2 KBof SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output:

Each of the 14 digital pins on the Uno can be used as an input or output, using pin mode, digitalwrite, and digital read functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40mA and has an internal pull-up resistor of 20-50 kohms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATMEGA8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write () function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication usingthe SPI library.

3.2.3 MQ3 ALCOHOL SENSOR

The MQ-3 sensor is made of Tin Dioxide (SnO_2) delicate layer. It is sorted out in such a structure to give high affectability to liquor and low affectability to Benzene.

The MQ-3 sensor is used to detect the presence of alcohol level in the surrounding and give reading to Arduino which determines whether the consumption level is in limit or not. It is a lowcost semiconductor sensor w this module 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 SnO_2, whose conductivity is lower in clean air.

It is conductivity increases as the concentration of alcohol gases increases. It has high sensitivity 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.

The **MQ3 alcohol sensor** is a semiconductor sensor designed to detect and monitor the presence of

alcohol in the atmosphere. It's particularly sensitive to alcohol gas and can detect concentrations ranging from **25 to 500 ppm.**

Fig. 3.2.3: Alcohol Sensor

Here's a brief overview of its features and how it works:

- **Sensitive Material**: The sensor uses **SnO_2 (tin dioxide)** as its sensitive material, which has lower electrical conductivity in clean air.
- **Operating Conditions**: It operates with a **5V power supply** and consumes about **150mA**. The sensor functions well in temperatures ranging from **-10°C to 50°C**.
-

- **Pin Configuration**: The MQ3 sensor has **6 pins**—2 for heating (H), 2 for sensor output (A), and 2 for ground (B). The A and B pins can be interchanged.

- **Detection Principle**: When the sensor's SnO2 surface is exposed to alcohol gas, its conductivity increases proportionally to the alcohol concentration.
- **Output**: It can provide an **analog output** (0-5V) related to the detected alcohol gas concentration or a **digital output** (0 or 1) for a specific gas concentration using the onboard meter.

.

MQ3 **Sensor** belongs to the MQ series that are used to detects different gasses present in the air**.** This alcohol sensor detects the concentration of alcohol 25 to 500 ppm in the air.

3.2.3a Features and Specifications

Features

- 5V operation
- 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

Specifications

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

3.2.3b Applications:

- Vehicle Alcohol Detector
- Portable Alcohol Detector

3.2.4 ANTENNA MODULE

- 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.
- The NEO-6M GPS module is a compact and cost-effective GPS receiver that can determine your position, time, and speed using signals from satellites.

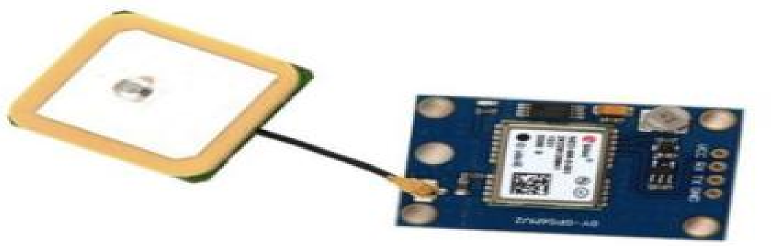


Fig. 3.2.4: Antenna Module

It connects with the satellite and used to send SMS to the mobile number entered in the code. when it connects to the satellite then the light on the antenna module turns ON which indicates the connection of arduino to the satellite.

3.2.5 GSIM MODULE

3.2.5a GSM Introduction

Global system for mobile communications (GSM)is the most popular standard for mobile phones in the world. GSM differs from its predecessors in that both signalling and speech channels are digital call quality, and thus is considered a second generation (2G) mobile phone system implementing GSM. GSM also pioneered a low-cost alternative to voice calls, the short Message service (SMS, also called "text messaging"), which is now supported on other mobile standards as well. One of the key features of GSM is the Subscriber Identity Module (SIM), commonly known as a SIM card.

The SIM is a detachable smart card containing the user's subscription information and phonebook. This allows the user to retain his or her information after switching handsets alternatively the user can also change operators while retaining the handset simply by changingthe SIM. Some operators will block this by allowing the phone to use only a single SIM, or only a SIM issued by them. This practice is known as SIM locking, and is illegal in some countries.

3.2.5b GSM Description

GSM (Global System for Mobile communication) is a digital mobile telephony system that is widely used in Europe and other parts of the world. GSM uses a variation of time divisionmultiple access (TDMA) and is the most widely used of the three digital wireless telephony technologies (TDMA, GSM, and CDMA).

GSM is a mobile communication modem it is stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970 .it is widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data service operates at the 850MHZ, 900MHZ, 1800MHZ and 1900MHZ frequency bands.

GSM system was developed as a digital system using time division multiple access (TDMA) technique for communication purpose. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has an ability to carry 64 kbps to 120 Mbps of data rates.

Rhydo LABZ's GSM/GPRS Modem-RS232 is built with Quad Band GSM/GPRS engine –SIM800, works on frequencies 850/900/1800/1900 MHZ. The Modem is coming with RS232 interface, which allows you connect PC as well as microcontroller with RS232 Chip (MAX232) .The baud rate is configurable from 9600-115200 (default baud rate is 9600) through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface.

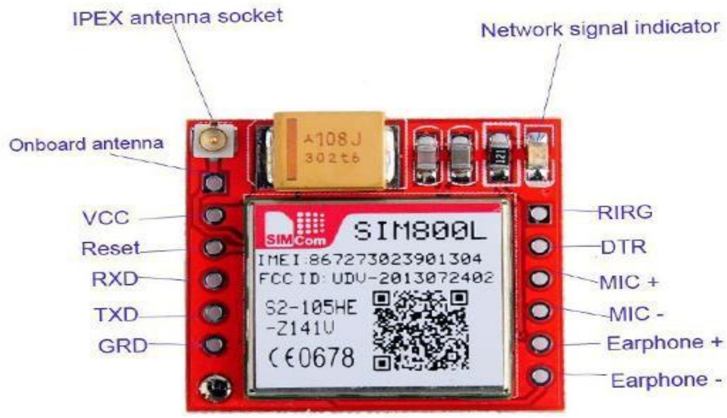


Fig. 3.2.5: GSM Modem

The modem is manufactured with automatic pick and place machine with high quality standard. The on-board Low dropout 3A power supply allows you to connect wide range unregulated power supply. Using this modem, you can make audio calls, SMS, Read SMS; attend the incoming calls and internet etc., through simple AT commands.

3.2.5c Features and Specifications

1. High Quality Product
2. Quad-Band GSM/GPRS 850/ 900/1800/1900 MHz
3. RS232 interface for direct communication with computer or MCU kit
4. Configurable baud rate
5. SMA connector with GSM Antenna.
6. SIM Card holder.
7. Built in Network Status LED
8. Inbuilt Powerful TCP/IP protocol stack for internet data transfer over GPRS
9. Audio interface Connector
10. Normal operation temperature:-20 °C to +55C
11. Input Voltage: 4.5V-12V DC

SPECIFICATIONS DESCRIPTION

GSM/GPRS specification	
GSM/GPRS module	SIM800
Frequency	850MHz/900MHz/1800MHz/1900MHz
Modem interface	RS232 serial interface
Baud rate(default factory)	9600bps
Power requirement	4.5 to 12v
Current requirement	<590ma
Sim800 module operating temperature	-40°C to +85 °C
Weight	40g

Table 1: GSM Specifications

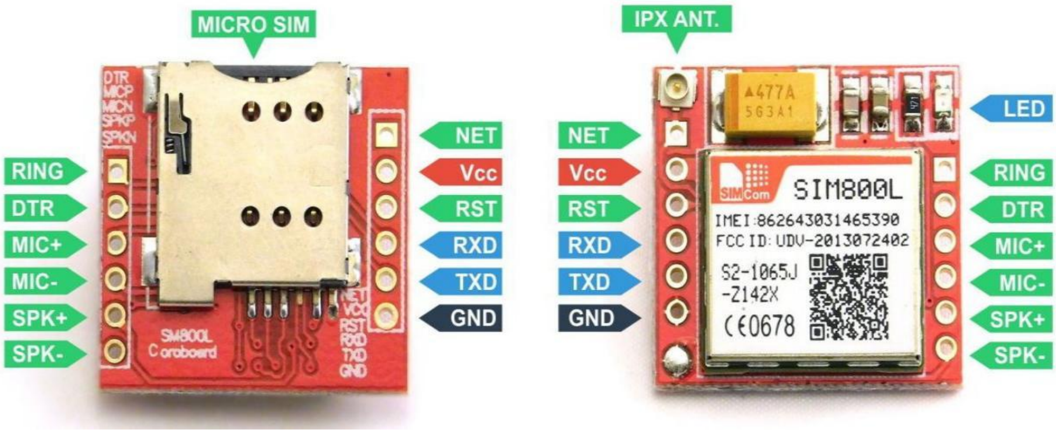


Fig. 3.2.5d: Pin diagram of GSM modem

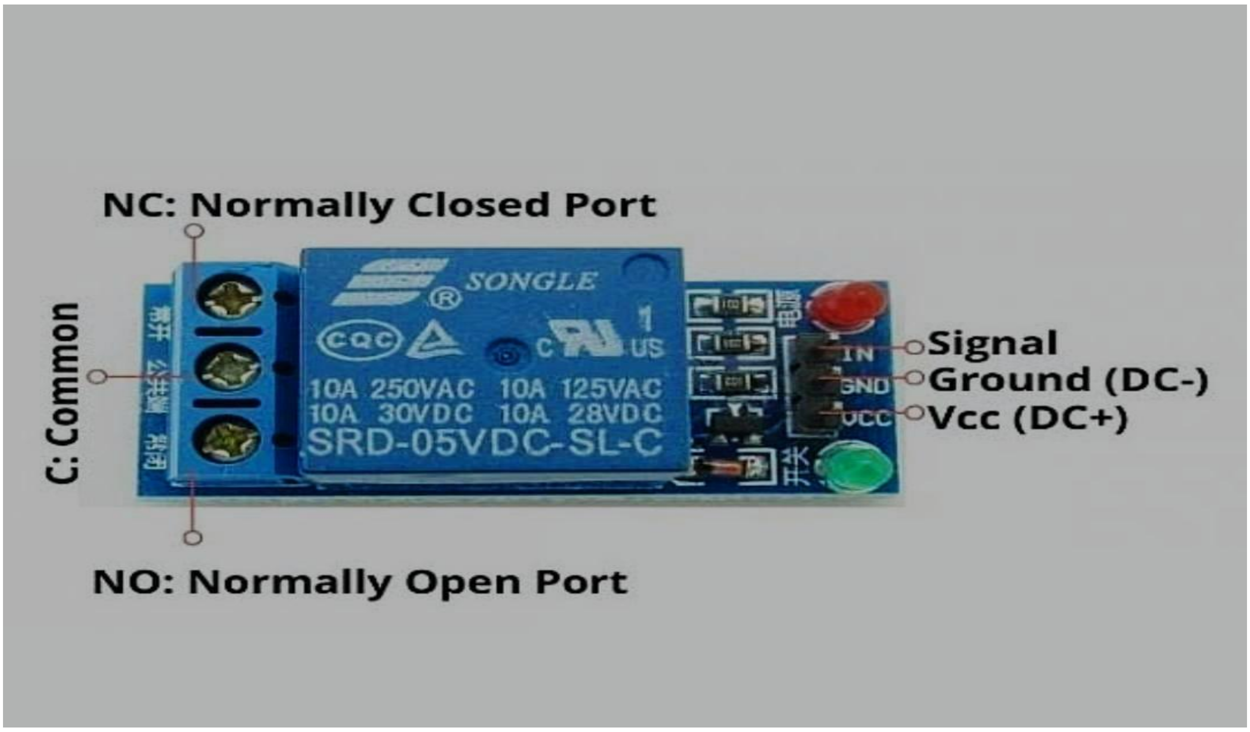
3.2.5e SIM800L

Designed for global market, SIM800L is a quad-band GSM/GPRS engine that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM800L features GPRS multi-slot class 10/ class 8 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4. With a tiny configuration of 33mm x 33mm x 3mm, SIM800L can meet almost all the space requirements in your applications, such as M2M, smart phone, PDA, FWP, and other mobile device. The physical interface to the mobile application is a 48-pin SMT pad, which provides all hardware interfaces between the module and customers' boards.

The light source is usually located directly behind the LCD and can use either LED or conventional fluorescent technology. From this source, the light ray will pass through a light polarizer to uniformly polarize the light so it can be acted upon by the liquid crystal (LC) matrix.

3.2.6 RELAY MODULE

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism, but other operating principles are also used. Relays find applications where it is necessary to control a circuit by a low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays found extensive use in telephone exchanges and early computers to perform logical operations. A type of relay that can handle the high power required to directly drive an electric motor is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device triggered by light to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protection



relays".

Fig. 3.2.6: Relay

3.2.6a Relay Driver

The current needed to operate the relay coil is more than can be supplied by most chips (op. amps etc), so a transistor is usually needed, as shown in the diagram below.

Use BC109C or similar. A resistor of about 4k7 will probably be alright. The diode is needed to short circuit the high voltage "back emf" induced when current flowing through the coil is suddenly switched off.

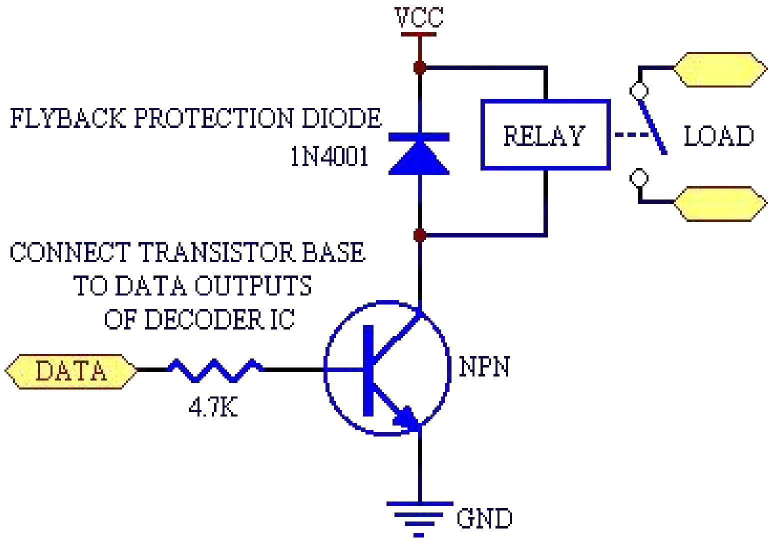


Fig. 3.2.6a: Relay Driver

3.2.7 DC MOTOR

The DC motor is connected to the L293D which in turn is connected to Arduino and is given 5V supply. DC motor works on the principle of Lorentz Law. When an electric current is passed through the motor, the coil carrying the current produces magnetic field which in turn rotates the coil with the forced experienced.

Fig. 3.2.7: DC Motor

DC motors offer highly controllable speed. By changing the armature or field voltage it's possible to achieve wide speed variation and with this level of controllability, DC motors offer the precision required by a wide range of industry applications.

The DC motor is the motor which converts the direct current into the mechanical work. It works on the principle of Lorentz Law, which states that “the current carrying conductor placed in a magnetic and electric field experience a force”. And that force is the Lorentz force.

3.2.7a Features and Applications

Features

1.Ability to Operate at High Speed: DC motors can achieve high rotational speeds, making them suitable for applications where rapid motion is required.
2.High Starting Torque: DC motors provide strong starting torque, allowing them to overcome inertia and initiate motion efficiently.
3.Control over Motor Speed and Torque: By adjusting the voltage supplied to the motor, you can control both the speed and torque output. This flexibility is advantageous in various scenario
.

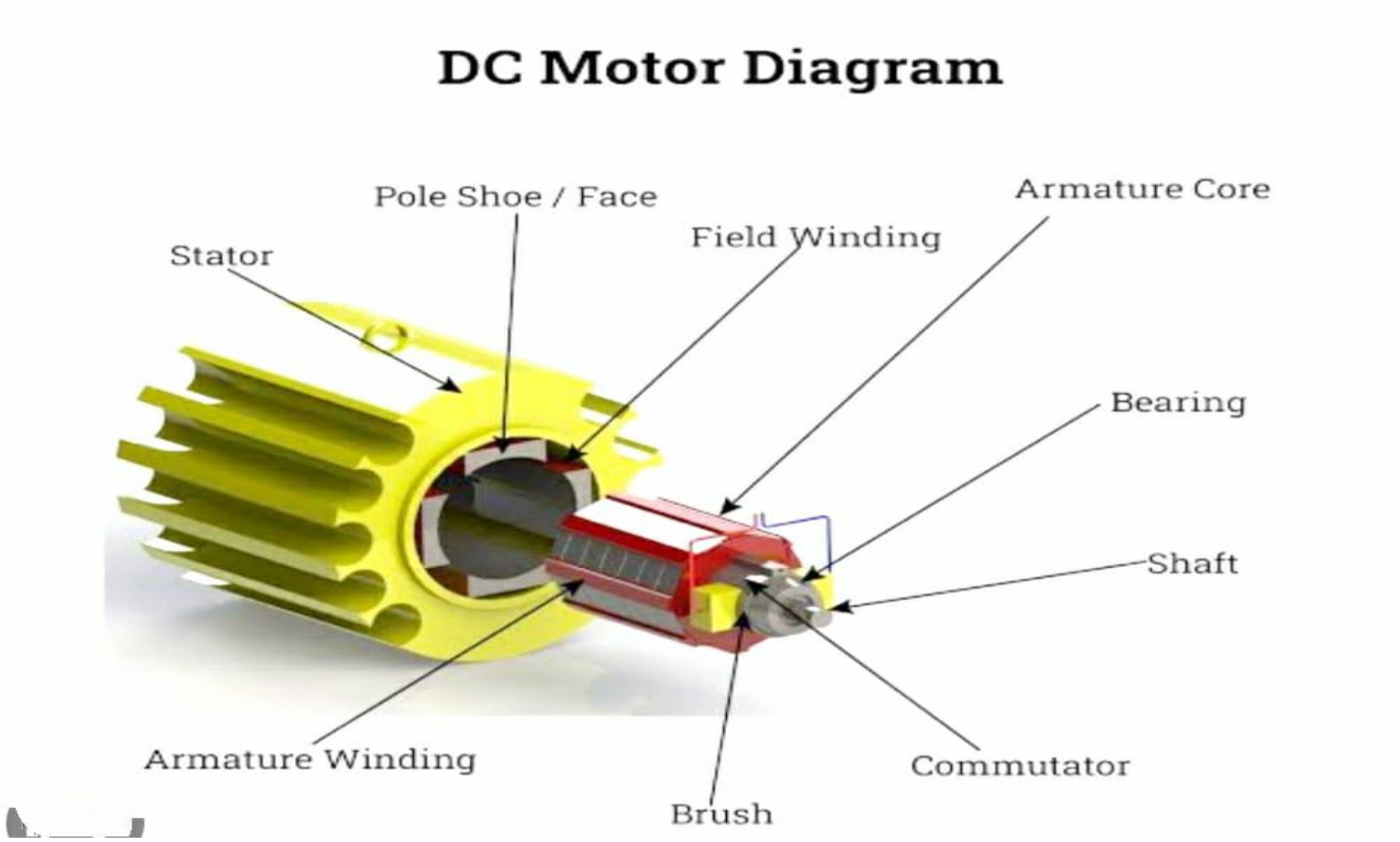


Fig. 3.2.7a: Parts of DC Motor

The alcohol sensor used to detect the present of alcohol content in human breath. Alcohol sensor, if it detects an alcohol, through the relay it is connected to DC motor. DC motor starts rotating in one direction. If there is no alcohol detection then DC motor won't rotate.

Applications

- Automotive Industries: DC motors are used in electric vehicles (EVs) for functions like power windows, windshield wipers, and cooling fans.
- Hoists & Cranes Operations: DC motors provide reliable lifting and lowering mechanisms in hoists, cranes, and material handling equipment.
- Industrial Tools Power Supply: They drive tools such as drills, grinders, and saws due to their robust construction and precise control.
- Lifts: Elevators and escalators rely on DC motors for smooth vertical movement.
- Air Compressors: DC motors power air compressors used in pneumatic systems.
- Winching Systems: DC motors drive winches for pulling heavy loads.

4.2.8 BUZZER

Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm. A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate (brass or stainless steel, etc.). A piezoelectric ceramic plate is attached to a metal plate with adhesives. Applying D.C. voltage between electrodes of a piezoelectric diaphragm causes mechanical distortion due to the piezoelectric effect. For a misshaped piezoelectric element, the distortion of the piezoelectric element expands in a radial direction. And the piezoelectric diaphragm bends toward the direction. The metal plate bonded to the piezoelectric element does not expand. Conversely, when the piezoelectric element shrinks, the piezoelectric diaphragm bends in the direction Thus, when AC voltage is applied across electrodes, the bending is repeated, producing sound waves in the air.

Fig. 3.2.8: Buzzer

To interface a buzzer the standard transistor interfacing circuit is used. Note that if a different power supply is used for the buzzer, the 0V rails of each power supply must be connected to provide a common reference.

If a battery is used as the power supply, it is worth remembering that piezo sounders draw much less current than buzzers. Buzzers also just have one ‘tone’, whereas a piezo sounder is able to create sounds of many different tones.

4.2.9 LCD DISPLAY

LCD Background:

One of the most common devices attached to a micro controller is an LCD display. Some of the most common LCD's connected to the many microcontrollers are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Basic 16x 2 Characters LCD

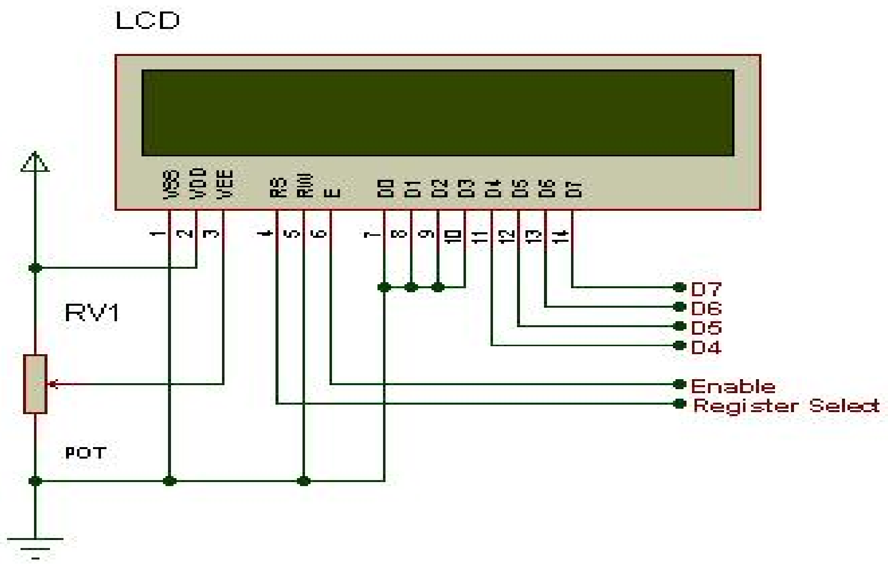


Fig. 3.2.9: LCD Pin diagram

The LCD requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

4.2.9a Pin Description

Pin No.	Name	Description
Pin no. 1	**VSS**	Power supply (GND)
Pin no. 2	**VCC**	Power supply (+5V)
Pin no. 3	**VEE**	Contrast adjust
Pin no. 4	**RS**	0 = Instruction input 1 = Data input
Pin no. 5	**R/W**	0 = Write to LCD module 1 = Read from LCD module
Pin no. 6	**EN**	Enable signal
Pin no. 7	**D0**	Data bus line 0 (LSB)
Pin no. 8	**D1**	Data bus line 1
Pin no. 9	**D2**	Data bus line 2
Pin no. 10	**D3**	Data bus line 3
Pin no. 11	**D4**	Data bus line 4
Pin no. 12	**D5**	Data bus line 5
Pin no. 13	**D6**	Data bus line 6
Pin no. 14	**D7**	Data bus line 7 (MSB)

Table 2: Character LCD pins with Microcontroller

The three control lines are referred to as **EN**, **RS**, and **RW**.

The **EN** line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen we would set RS high.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

CHAPTER 4

SOFTWARE DESCRIPTION

The Arduino Software (IDE) makes it easy to write code and upload it to the board offline. We recommend it for users with poor or no internet connection. This software can be used with any Arduino board. Here are currently two versions of the Arduino IDE, one is the IDE 2.0.0.

4.1 Arduino IDE – Compiler

Here are currently two versions of the Arduino IDE, one is the IDE 1.x.x and the other is IDE 2.x. The IDE 2.x is new major release that is faster and even more powerful to the IDE 1.x.x. In addition to a more modern editor and a more responsive interface it includes advanced features to help users with their coding and debugging.

The following steps can guide you with using the offline IDE (you can choose either IDE 1.x.x or IDE 2.x):

1. Download and install the Arduino Software IDE:

Arduino IDE 1.x.x (Windows, Mac OS, Linux, Portable IDE for Windows and Linux, ChromeOS).

Arduino IDE 2.x

2. Connect your Arduino board to your device.

3. Open the Arduino Software (IDE).

The Arduino Integrated Development Environment - or Arduino Software (IDE) - connects to the Arduino boards to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension .ino.

Using the offline IDE 1.x.x

The editor contains the four main areas:

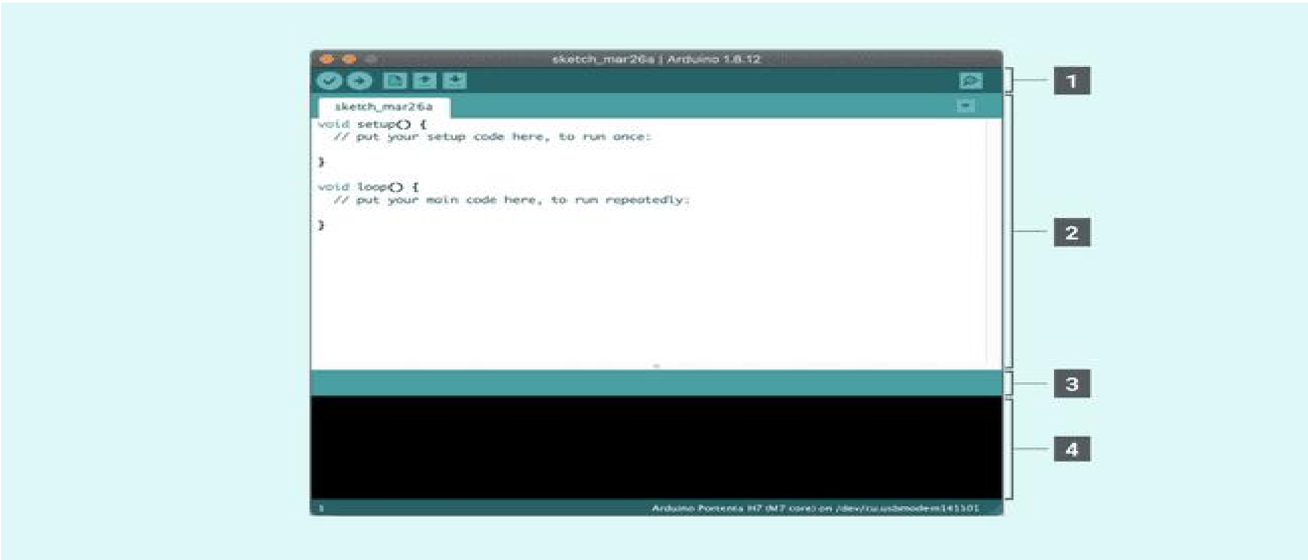
1. A **Toolbar with buttons** for common functions and a series of menus. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

2. The **message area**, gives feedback while saving and exporting and also displays errors.

3. The **text editor** for writing your code.

4. The **text console** displays text output by the Arduino Software (IDE), including complete error messages and other information.

The bottom right-hand corner of the window displays the configured board and serial port.

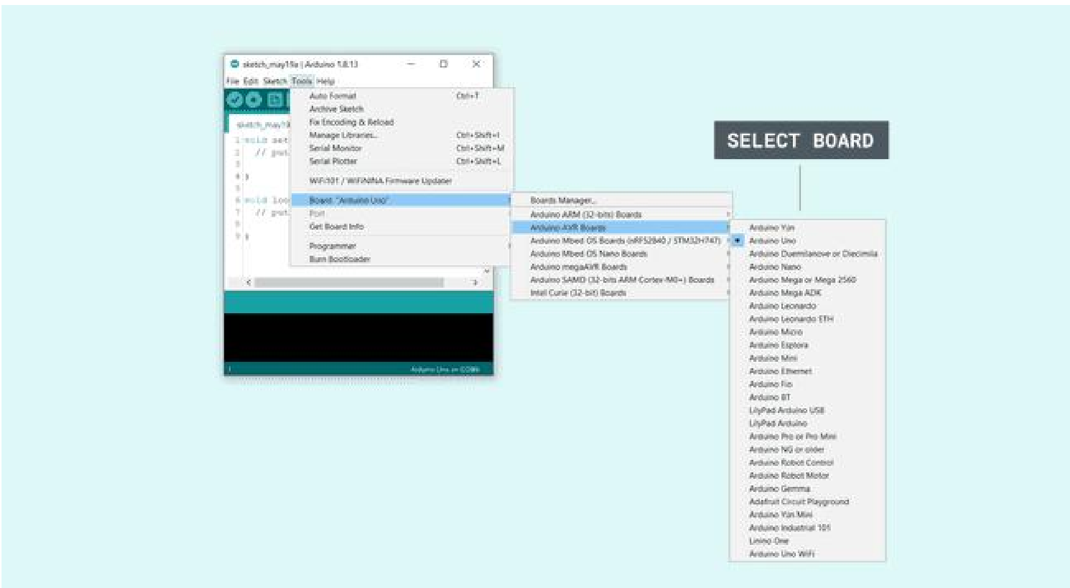


The Arduino Software IDE

Now that you are all set up, **let's try to make your board blink!**

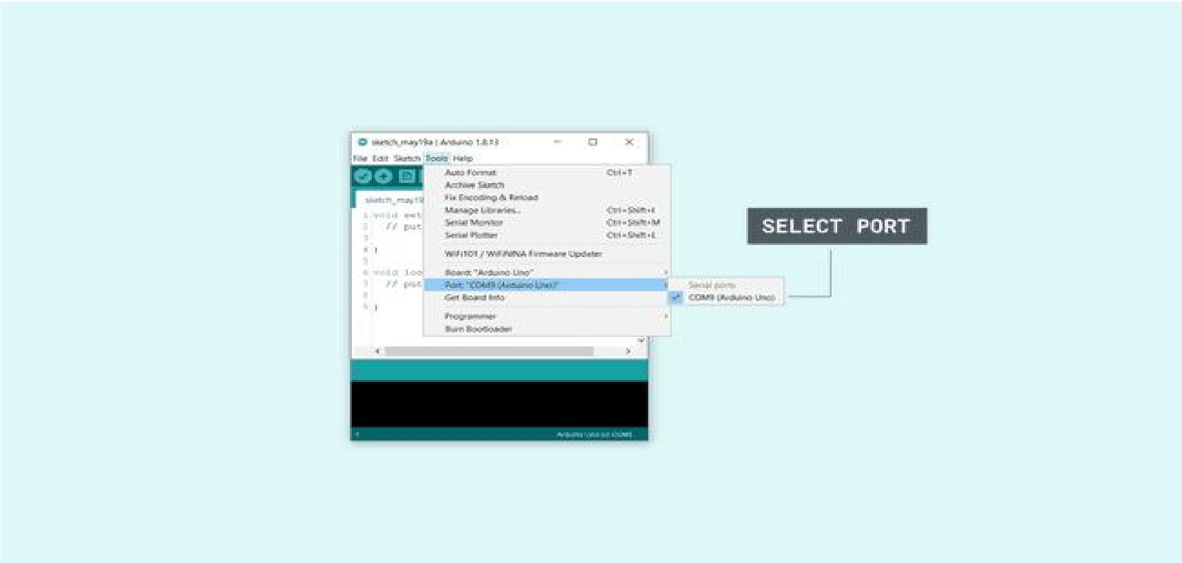
5. **Connect your Arduino** or Genuino board to your computer.

6. Now, you need to **select the right core & board**. This is done by navigating to **Tools > Board > Arduino AVR Boards > Board**. Make sure you select the board that you are using. If you cannot find your board, you can add it from **Tools > Board > Boards Manager**.



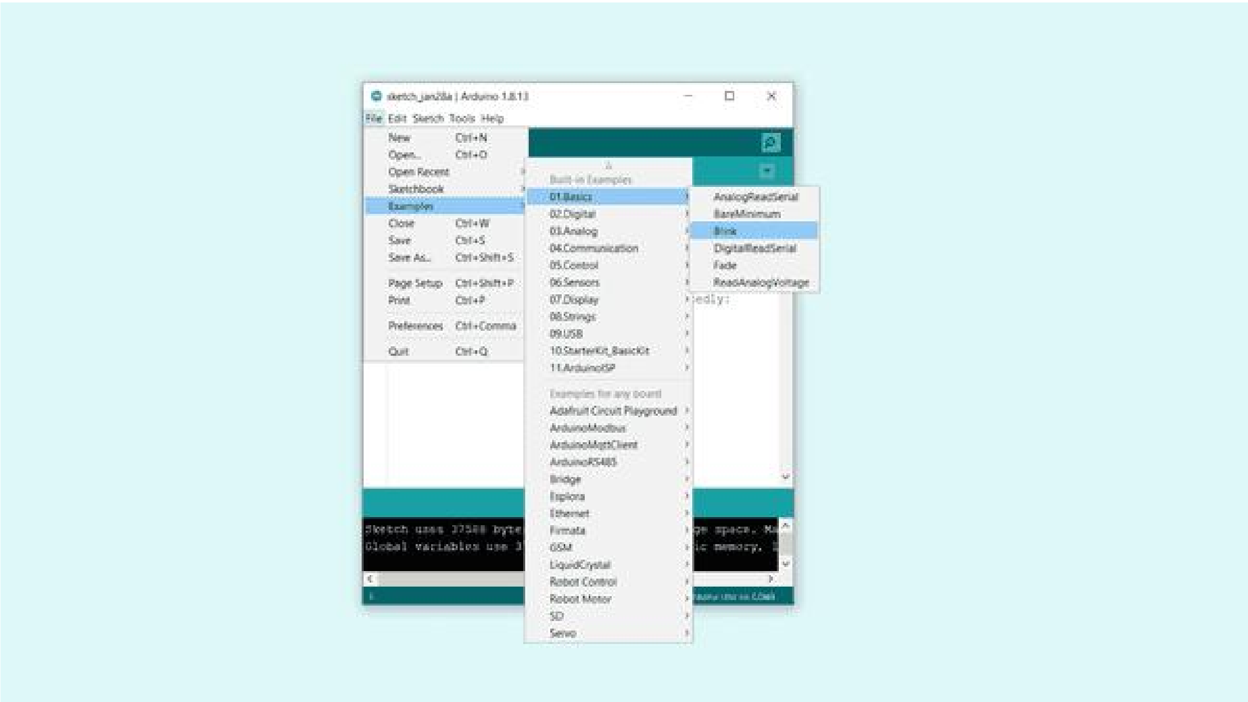
Selecting a board

7. Now, let's make sure that your board is found by the computer, by **selecting the port**. This is simply done by navigating to **Tools > Port**, where you select your board from the list.



Selecting the port

8. Let’s **try an example**: navigate to **File > Examples > 01.Basics > Blink.**



Opening an example

9. To **upload it to your board**, simply click on the arrow in the top left corner. This process takes a few seconds, and it is important to not disconnect the board during this process. If the upload is successful, the message "Done uploading" will appear in the bottom output area.

10. Once the upload is complete, you should then see on your board the yellow LED with an L next to it start blinking. You can **adjust the speed of blinking** by changing the delay number in the parenthesis to 100, and upload the Blink sketch again. Now the LED should blink much faster.

The editor contains the four main areas:

1. A **toolbar with buttons** for common functions and a series of menus. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, choose your board and port and open the serial monitor.

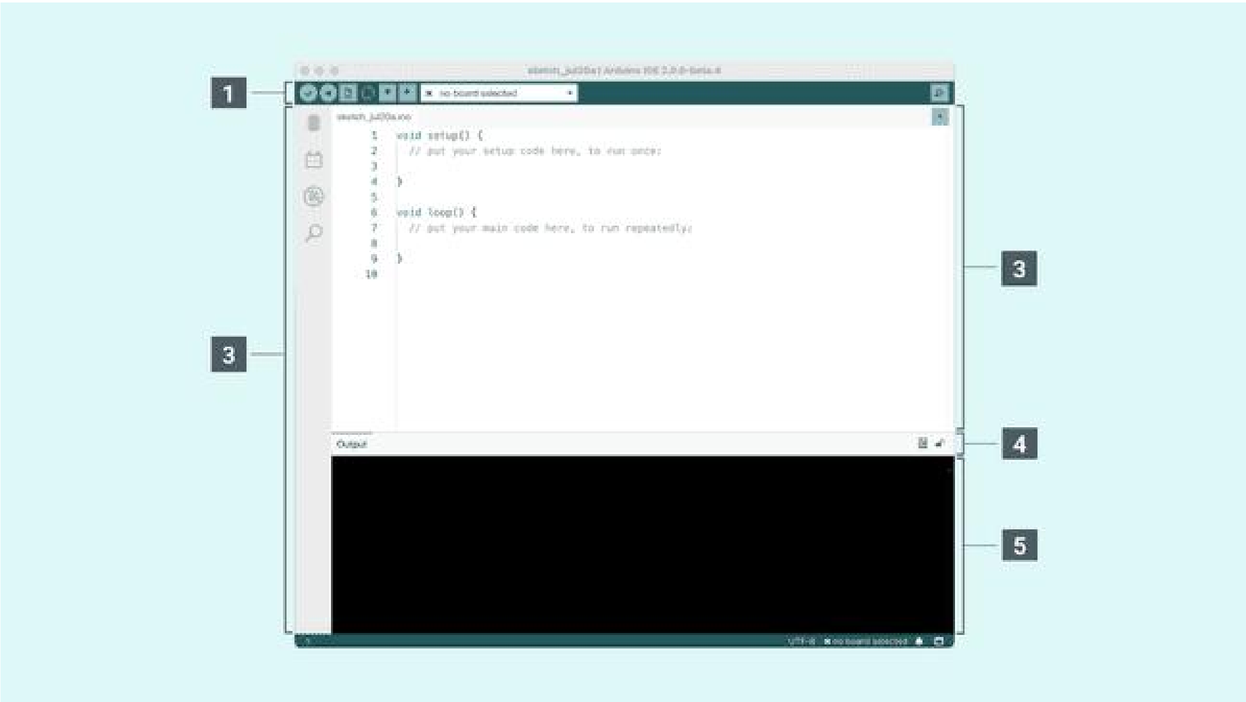
2. The **Sidebar** for regularly used tools. It gives you quick access to board managers, libraries, debugging your board as well as a search and replacement tool.

3. The **text editor** for writing your code.

4. Console controls gives control over the output on the console.

5. The **text console** displays text output by the Arduino Software (IDE), including complete error messages and other information.

The bottom right-hand corner of the window displays the configured board and serial port.

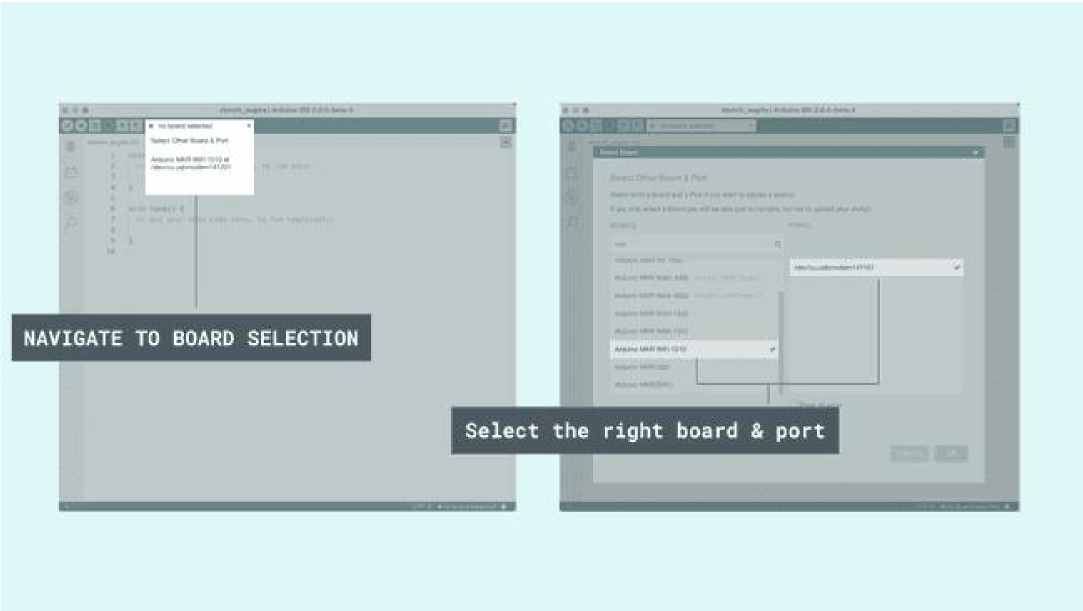


The Arduino Software IDE

Now that you are all set up, **let's try to make your board blink!**

1. Connect your Arduino or Genuino board to your computer.

2. Now, you need to **select the right board & port**. This is done from the toolbar. Make sure you select the board that you are using. If you cannot find your board, you can add it from the board manager in the sidebar.



Selecting a board & port

3. To **upload it to your board**, simply click on the arrow in the top left corner. This process takes a few seconds, and it is important to not disconnect the board during this process. If the upload is successful, the message "Done uploading" will appear in the bottom output area.

4.2 SOURCE CODE

```
#include <SoftwareSerial.h>
#include <AltSoftSerial.h>
#include <TinyGPS++.h>
float flat, flon;
//GSM Module TX is connected to Arduino D8
#define SIM800_TX_PIN 2

//GSM Module RX is connected to Arduino D9
#define SIM800_RX_PIN 3
SoftwareSerial mySerial(SIM800_TX_PIN, SIM800_RX_PIN); //Create software serial object to

communicate with GSM Module
```

```
AltSoftSerial neogps;
TinyGPSPlus gps;

bool newData = 0;

int device_2 = 4;

// defines variables
int index = 0;
String number = "";
String message = "";

char incomingByte;
String incomingData;
bool atCommand = true;

#define Sober 200   // Define max value that we consider sober

#define Drunk 400   // Define min value that we consider drunk

#define MQ3 0



float sensorValue;  //variable to store sensor value

void setup()
{
  Serial.begin(9600); // Serial Communication for Serial Monitor is starting with 9600 of baudrate speed
  mySerial.begin(9600); // Serial Communication for GSM Module is starting with 9600 of baudrate speed
  neogps.begin(9600);

  pinMode(device_2, OUTPUT); //Sets the device_2 as an OUTPUT
```

```
digitalWrite(device_2, LOW); //Sets the device_2 in to OFF state at the beginning


 // Check if you're currently connected to SIM800L
 while (!mySerial.available()) {
  mySerial.println("AT");
  delay(1000);
  Serial.println("connecting....");
 }

 Serial.println("Connected..");

 mySerial.println("AT");              // Sends an ATTENTION command, reply should be OK
 delay(1000);
 mySerial.println("AT+CMGF=1");        // Configuration for sending SMS
 delay(1000);
 mySerial.println("AT+CNMI=1,2,0,0,0");  // Configuration for receiving SMS
 delay(1000);
 Serial.println("AT&W");              // Save the configuration settings
 delay(1000);

 Serial.println("Ready to received Commands..");
}
void loop()
{
 if (mySerial.available()) {
  delay(100);

  // Serial buffer
  while (mySerial.available()) {

 incomingByte = mySerial.read();
   incomingData += incomingByte;
```

```
  }

  delay(10);
  if (atCommand == false) {
    receivedMessage(incomingData);
  } else {
    atCommand = false;
  }

  //delete messages to save memory
  if (incomingData.indexOf("OK") == -1) {
    mySerial.println("AT+CMGDA=\"DEL ALL\"");
    delay(1000);
    atCommand = true;
  }

  incomingData = "";
}

 sensorValue = analogRead(MQ3); // read analog input pin 0

Serial.print("Sensor Value: ");

Serial.print(sensorValue);

        // Return analog moisture value

// Determine the status

if (sensorValue < Sober) {

  Serial.println("  |  Status: Sober");


} else if (sensorValue >= Sober && sensorValue < Drunk) {
```

```
 Serial.println("  |  Status: Drinking but within legal limits");

  } else {

    Serial.println("  |  Status: DRUNK");
    digitalWrite(device_2,HIGH);
          get_gps();
      mySerial.println("AT+CMGDA=\"DEL ALL\"");
      delay(1000);

  }
  delay(5000);
}
void receivedMessage(String inputString) {

  //Get The number of the sender
  index = inputString.indexOf('"') + 1;
  inputString = inputString.substring(index);
  index = inputString.indexOf('"');
  number = inputString.substring(0, index);
  Serial.println("Number: " + number);

  //Get The Message of the sender
  index = inputString.indexOf("\n") + 1;
  message = inputString.substring(index);
  message.trim();
  Serial.println("Message: " + message);
  message.toUpperCase(); // uppercase the message received

  //turn Device 2 ON

 if (message.indexOf("D2 ON") > -1) {
  delay(1000);
```

```
get_gps();




mySerial.println("AT+CMGDA=\"DEL ALL\"");

  delay(1000);
  }
   incomingData = "";
  delay(50);// Added delay between two readings
}

void get_gps()
{
  while (neogps.available()) {
   Serial.print(">");
   if (gps.encode(neogps.read())) {
     newData = true;
   }
  }
  if (newData)
  {
   unsigned long age;
   flat = gps.location.lat();
   flon = gps.location.lng();
   Serial.print("Latitude= ");
   Serial.print(gps.location.lat(), 6);
   Serial.print(" Longitude= ");

 Serial.println(gps.location.lng(), 6);
   newData = false;
   mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
   delay(1000);
   mySerial.println("AT+CMGS=\"+91xxxxxxxxx\"");//change ZZ with country code and xxxxxxxxxxx with
```

```
   phone number to sms



delay(1000);

 mySerial.print("http://maps.google.com/maps?q=loc:");

  mySerial.print(gps.location.lat(), 6);
   mySerial.print(",");

 mySerial.print(gps.location.lng(), 6);
   //    delay(100);
   mySerial.write(26);
   Serial.println("Command: Device 2 Turn On.");
   delay(3000);
  }
}
```

4.3 WORKING

4.3.1 Working flow chart

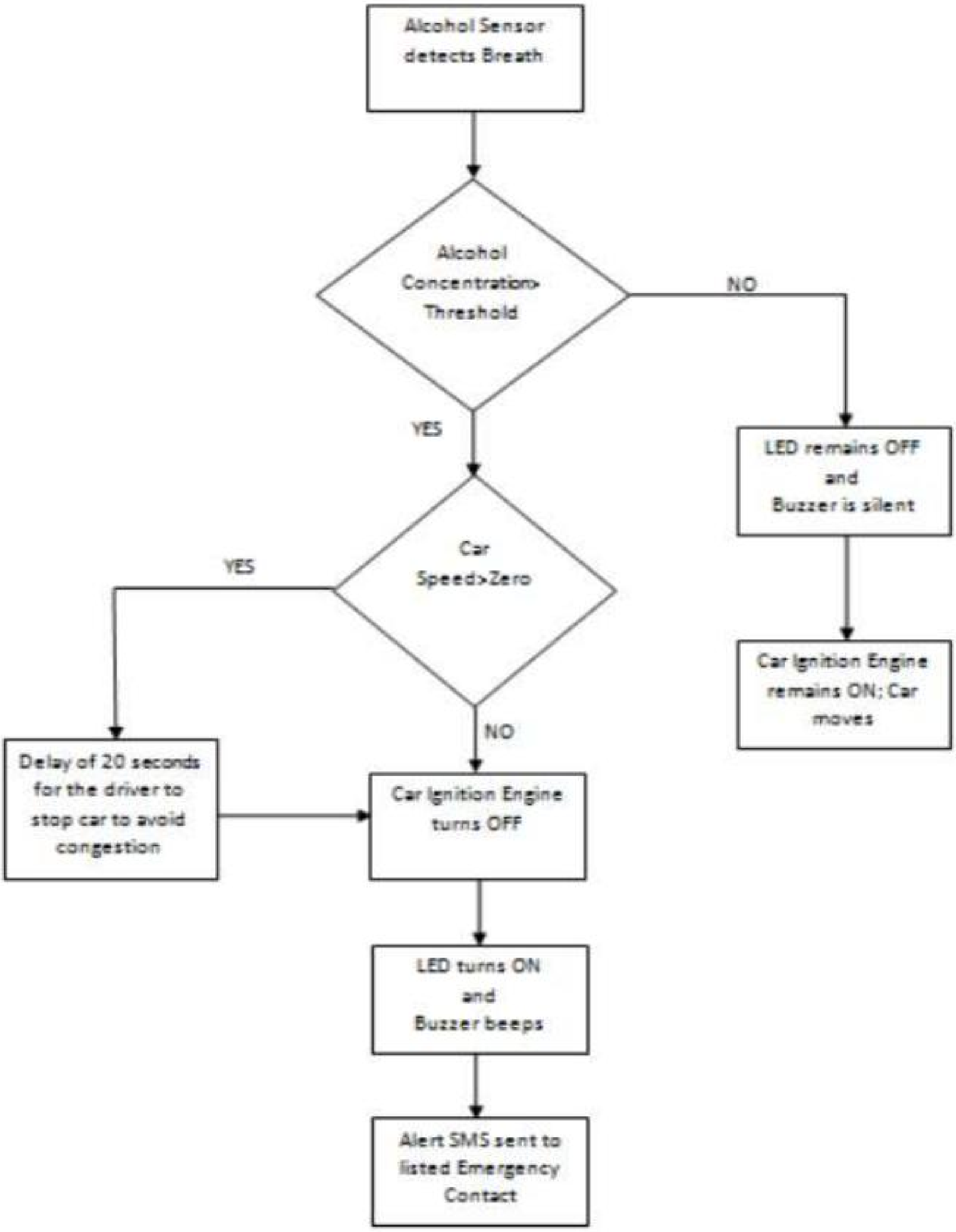


Fig. 4.3.1: Working Of System

When the system detects the alcohol content then the system generates a message like..,

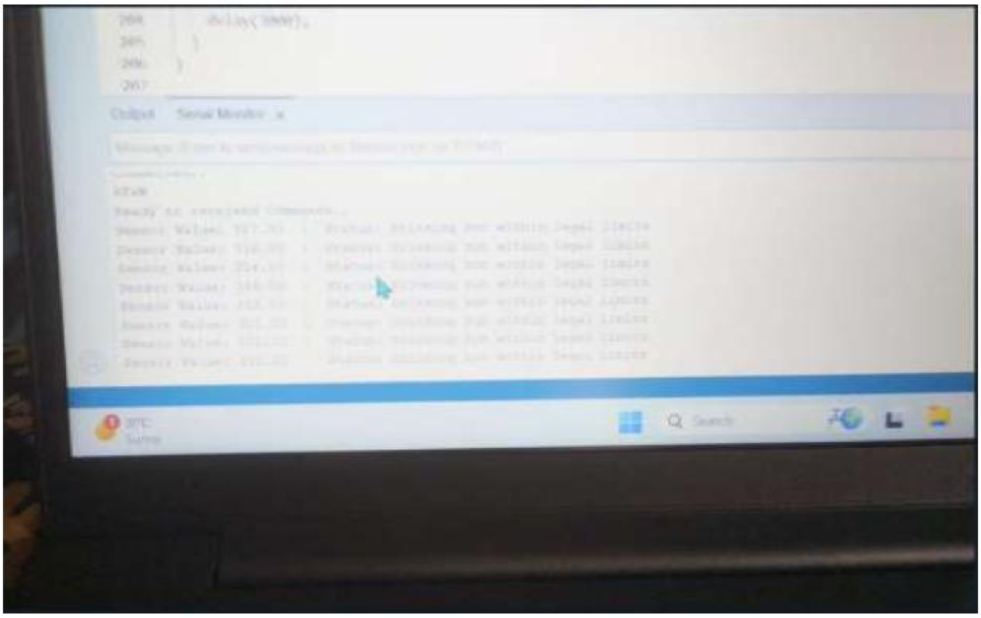


Fig. 4.3.2: System Output

Actually, alcohol lock to prevent drunk driving is not a new idea. It's a device to test alcohol concentration from driver breath, installed before driver seat and connected to the engine ignition. If the test result reach regulated level, the lock will be activated and the engine can't be started. The locking device can constantly monitor the driver breath to prevent cheating start.

If the driver drink alcohol during driving, the lock system is able to stops the engine so that the car would not accelerate any further and driver can steer it to roadside in safe distance and sending emergency siren alarming danger to other motors to prevent crash at the same time.

The Alcohol detection with engine ignition locking system can help prevent accidents due to drunk driving and improve automatic safety system for cars and other vehicles. Although there are rigid regulation and frequent police tests on drunk driving in many countries, there are still many accidents caused by drinking alcohol and lead to casualties and property loss.

The locking system can be very helpful for police. Several sensor technologies commonly used in alcohol detection mainly include semiconductor sensor, electrochemical sensor, colorimetric technology, infrared detection technology, gas chromatography analysis technology, etc. Among them, the most commonly used are electrochemical sensor technology and semiconductor sensor technology.

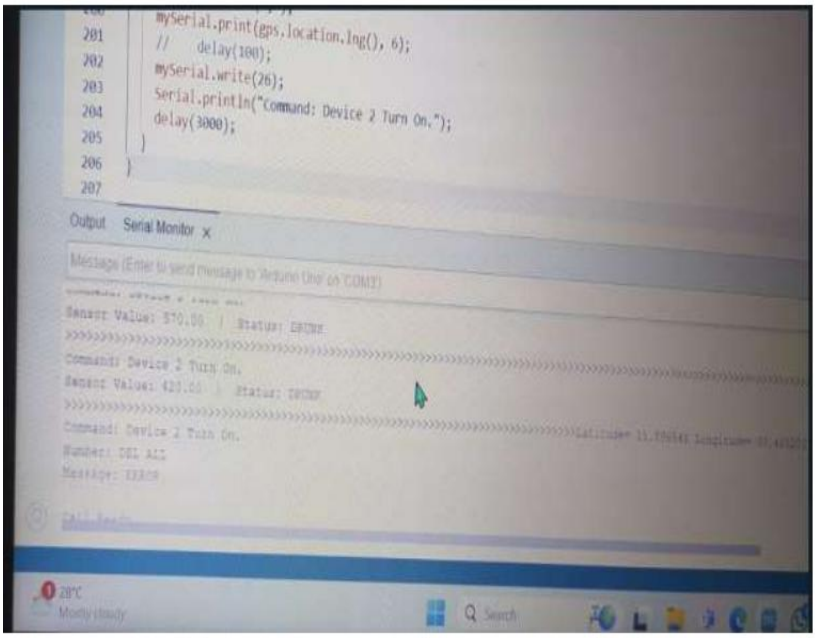


Fig. 4.3.3: Sensor Output

The breath alcohol sensor based on electrochemistry technology uses an fuel cell sensor as a sensing element. Alcohol and oxygen undergo a corresponding oxidation-reduction reaction on the working electrode and the counter electrode of the sensor and release charges to form a current. The magnitude of the generated current is proportional to the alcohol concentration and follows Faraday's law. The level of alcohol concentration can be determined by testing the magnitude of the current .

CHAPTER 5
RESULTS AND ANALYSIS

After completing the connections of the circuit and uploading the code, it is ready for use. To mimic the presence of a drunk person, we have used alcohol based sanitizer as an alternative of alcohol. Both alcohol and sanitizer being volatile in nature, tends to evaporate with time. This increases the alcohol concentration in the surrounding, and this increase in concentration is sensed by the sensor. When a drunk person comes near to the sensor (in our case, when the sanitizer is brought near the sensor), it reads the increase in the alcohol concentration in the surrounding of the person, and the sensor sends a signal to the Arduino about the reading.

If the value of reading is higher than the threshold value (800 in our case) then the Arduino sends a signal to the buzzer, LED and relay. The buzzer starts to beep, the LED glows, and the relay stops the power supply to the motor, thus stopping it, eventually stopping the vehicle. The Arduino will continue to send the signal to the relay and buzzer as long as it is receiving a value higher than the threshold value, from the MQ-3 sensor. When the reading value of the sensor drops below the threshold value, the buzzer stops beeping, the LED stops glowing and the relay no longer stops the motor power supply, so now the vehicle can again run. We can avoided any kind of loss of life by using this system. All equipment are totally tested and connected as required thereby giving us the much needed result.

Our project aimed to develop an innovative solution for enhancing road safety by integrating alcohol detection technology with a vehicle's ignition system. Through extensive research and development, we successfully designed and implemented an automatic engine locking system capable of detecting alcohol levels in the driver's breath and preventing the vehicle from starting if the concentration exceeds a predetermined threshold. The system utilizes state-of-the-art sensor technology and intelligent software algorithms to achieve high accuracy, reliability, and responsiveness. Extensive testing and validation demonstrated the system's effectiveness under various conditions, including different alcohol concentrations, temperatures, and driving scenarios. Furthermore, the system's user-friendly interface provides clear feedback to the driver, enhancing awareness and encouraging responsible behavior. Our project not only represents a significant technological advancement but also holds the potential to save lives and prevent accidents caused by impaired driving.

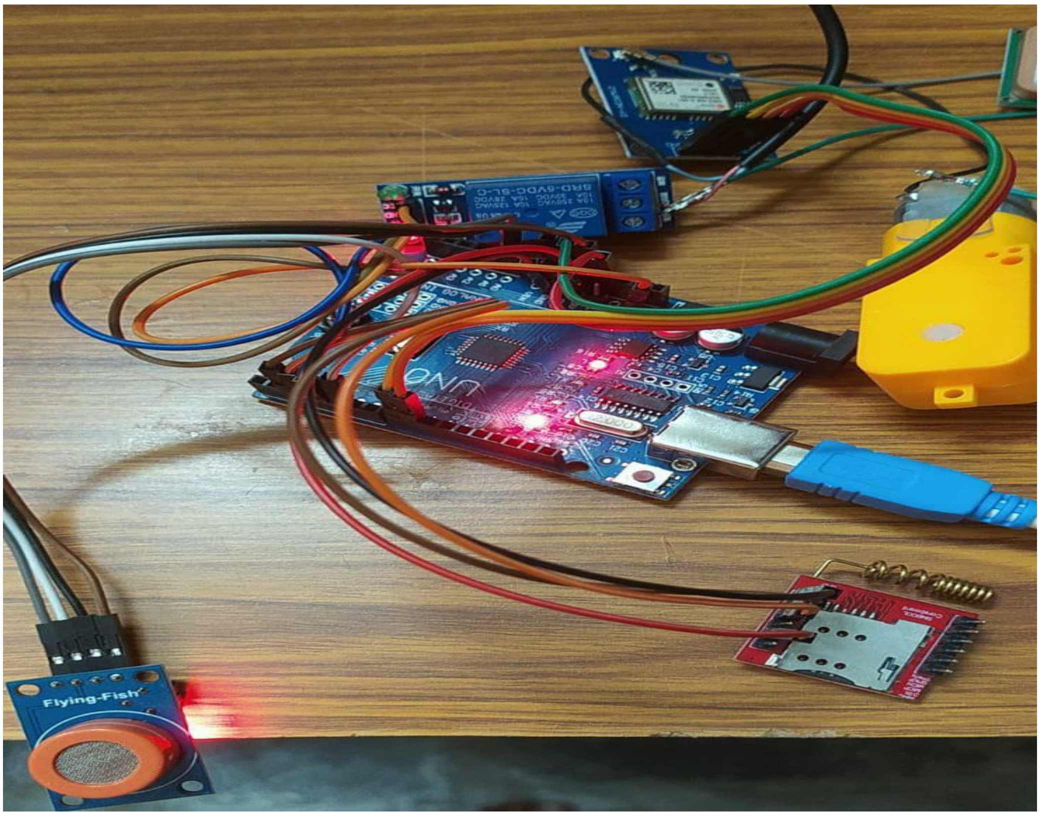


Fig. 5.1: Working module of Alcohol Detection System

We have given an incredibly capable way to deal and to develop a smart system for vehicles to diminish number of disasters caused in light of alcoholic driving. As the creating insight among people is that vehicle security is dynamically critical. Future degree of this structure is to control the setbacks caused due to alcohol use. This system improves the security of individual and in this manner giving the convincing progression in the vehicle business regarding decrease setbacks caused in light of driving.

Fig. 5.2: LCD Output of Alcohol Detection

5.1.1 Adavantages of Automatic Engine Locking System Through Alcohol Detection

1.Enhanced Road Safety

- Most road accidents occur due to intoxicated driving. By integrating an alcohol detection system, we can significantly reduce such accidents.
- The system continuously monitors the ethanol concentration using an MQ-3 sensor and promptly takes action if the concentration exceeds the threshold value.
- If the driver starts drinking alcohol after the engine starts, the system detects the alcohol content from the driver's breath. If it surpasses the threshold, the engine is automatically shut off.
- Even if the driver is already in a drunk state before starting the engine, the system prevents Most road accidents occur due to intoxicated driving. By integrating an alcohol detection system, we can reduce such accidents.

2.Automated and Accurate

- Automated systems are more reliable than manual efforts. They operate without the restrictions of space and time.
- The use of an Arduino Uno interface with an alcohol sensor ensures accurate detection.
- The system displays an alcohol detection note on the LCD screen, locks the engine, and prevents acceleration.

3.Integrated Communication

- The system sends a message to the vehicle owner via GSM (Global System for Mobile Communications) after locking the engine.
- Additionally, it provides the vehicle's location using a GPS module.

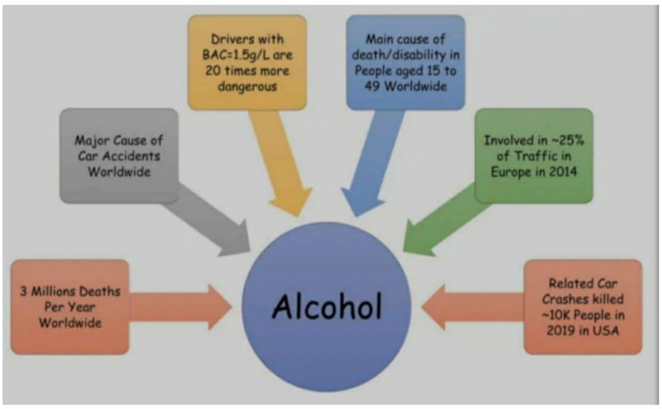
4.Reduced Loss of Life and Property

- According to surveys, more than 50 people die daily due to road accidents, with 70% of these fatalities attributed to drunken driving.
- Swift action through automatic engine locking can save lives and prevent property damage.

5.1.2 Applications of Automatic Engine Locking System

Drunk Driving is one of the biggest threats to Road Safety. Applications of Automatic Engine Locking System Through Alcohol Detection can be used anywhere to reduce the probability of road accidents.

- "Automatic Engine Locking System Through Alcohol Detection project" can be used in various vehicles for detecting whether the driver has consumed alcohol or not.
- This can also be used in various companies, organizations, mines to detect alcohol consumption of employees.
- By using "Automatic Engine Locking System Through Alcohol Detection project" we may reduce the accidents cased due to alcohol drunk.
- We can know the location of the our vehicle by sending the message to that Arduino uno by the number that what we had entered in the code.



CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

We have given an incredibly capable way to deal and to develop a smart system for vehicles to diminish number of disasters caused in light of alcoholic driving. As the creating insight among people is that vehicle security is dynamically critical. Future degree of this structure is to control the setbacks caused due to alcohol use.

This system improves the security of individual and in this manner giving the convincing progression in the vehicle business regarding decrease setbacks caused in light of driving. In this project we have developed a real time model that can automatically lock the engine when a drunken driver tries to drive a car. Now-a-days car accidents are mostly seen. By fitting this alcohol sensor into the car, we can save guard the life of the driver and also the remaining passengers. It is very simple application.

The life time of the project is higlr. It has low or zero maintenance cost and of course low power consumption. This is a developed design to efficiently check drunken driving. By implementing this design a safe car journey is possible decreasing the accident rate due to drinking. By implementing this design, drunken drivers can be controlled so are the accidents due to drunken driving.

Government must enforce laws to install such circuit in every car and must regulate all car companies to preinstall such mechanism while manufacturing the car itself. If this is achieved the deaths due to drunken drivers can be brought to minimum level. In this type of system, future scope can be safely landing of car aside without disturbing other vehicles.

6.2 FUTURE SCOPE

The development of alcohol detectors and engine locking systems is an ongoing process, and there are several potential future developments that could enhance their effectiveness and functionality. Here are some of the possibilities:

Integration with biometric sensors: Future alcohol detectors and engine locking systems may incorporate biometric sensors to detect specific physical characteristics that indicate impairment, such as changes in pupil size, body temperature, or heart rate.

Real-time monitoring and reporting: Advanced alcohol detectors and engine locking systems may be capable of transmitting data in real-time to law enforcement agencies or other relevant parties, enabling immediate action to be taken if a driver is found to be impaired.

Improved accuracy: New sensor technologies may be developed that can detect alcohol at lower levels or more accurately differentiate between alcohol and other substances, such as mouthwash or hand sanitizer.

REFERENCES

[1] Cahalan. D, I. Cisin, and Crossley, American Drinking Hones: A National Think about of Driving Conduct and Demeanors. 1969, Rutgers College Press: Unused Brunswick, NJ.

[2] MUGILA.G, MUTHULAKSHMI.M, SANTHIYA.K, Prof.DHIVYA.P- Savvy Protective cap Framework Utilizing Liquor Location FOR VEHICLE PROTECTION[International Diary of Inventive Inquire about in Science Designing and Innovation (IJIRTSE) ISSN: 2395-5619, Volume – 2, Issue – 7. July 2016].

[3] Dhivya .M , Kathiravan .S, Dept. of ECE, Kalaignar Karunanidhi Organized of Innovation- Driver Confirmation and Mishap Shirking Framework for Vehicles[Smart Computing Audit, vol. 5, no. 1, February 2015].

[4] Babor, Review: The liquor utilize clutters distinguishing proof Test: Rules for utilize in essential wellbeing care. 1992, Geneva, Switzerland: World Wellbeing Organization.

[5] Lee, Surveying the Achievability of Vehicle-Based Sensors To Identify Liquor Disability. 2010, National Thruway Activity Security Organization: Washington, DC. [6] http://www.arduino.cc/

[6] A. ISuge, H.Takigawa, H.Osuga, H.Soma, K.Morisaki, Mischance Vehicle Programmed Location Framework By Picture Handling Innovation , ©IEEE 1994 Vehiclee Route & ISC.

[7] Paul Baskett , Yi Shang and Michael V. Patterson , Timothy Trull , Towards A Framework for Body-Area Detecting and Discovery of Liquor Longing for and Temperament Dvsregulation , © 2013 IEEE.

[8] L. A. Navarro, M. A. Diño, E. Joson, R. Anacan and R. D. Cruz, "Design of Alcohol Detection System for Car Users thru Iris Recognition Pattern Using Wavelet Transform," 2016 7th International Conference onIntelligent Systems, Modelling and Simulation (ISMS), Bangkok, 2016, pp. 15-19.

[9] [1]. M.H. Mohamad, Mohd Aminbin Hasanuddin, Mohd Hafizzie Bin Ramli, “Vehicle Accident Prevention System Embedded with Alcohol Detector”, International journal of review in electronics communication enigineering (IJRECE), Volume 1, Issue 4 October 2013, eISSN:(2321-3159).

[10] S.P.Bhumkar, V.V.Deotare, R.V. Bobar “Accident Avoidance and Detection on Highway”, International journal of engineering trends and Technologies(IJETT), Volume3, Issue2(2012),ISSN:(2231-5381).

[11] Amitya kumar tripathy, sejal chopra, Samantha Bosco, Srinidhi Shetty, Firdos Sayyad,”Travolution-An Embedded System in passenger car for road safety”,

[12] International journal and magazine of engineering technology, management and research (IJMETMR),Feb 04- 06-2016,Vol 3,Issue 4,ISSN:(2348-4845).

[13] Wang dong, Cheng quancheng, Li Kai, Fang Bao-hua, “The automatic control system of anti drunk driving” in International journal of research and development (IJRD),Vol. 3,Issue 4, Nov 2015,ISSN:(2248-9622).

[14] Vipul H. Mistry, Dr. Ramji Makwana, “Survey: Vision based Road Detection Techniques”, (IJCSIT)

International Journal of Computer Science and Information Technologies, Vol. 5 (3), Oct 2014, ISSN :(0975-9646).

[15] Bhuta, Desai,Keni DzAlcohol Detection and Vehicle Controllingdz International Journal of Engineering Trends and Applications (IJETA) – Volume 2 Issue 2, Mar-Apr 2015.

[16] Amityakumartripathy, sejalchopra, Samantha Bosco, SrinidhiShetty, FirdosSayyad, dzTravolution-An Embedded System in passenger car for road safetydz, International journal and magazine of engineering technology, management and research (IJMETMR),Feb 04-06-2016,Vol 3,Issue4,ISSN:(2348-4845)

[17] Dada, Emmanuel & Hamed, Hamit & Lateef, Adebimpe & Opeyemi, Ajibuwa. (2017). AD of Drunk Drivers with Automatic Car Engine Locking System.

[18] International Journal of Industrial Electronics and Electrical Engineering, ISSN(p): 2347-6982, ISSN(e): 2349-204X Volume6, Issue-3, Mar.-2018, http://ijieee.org.in

[19] Mugila.J, Muthulakshmi.M, Santhiya.K, Prof.Dhivya.P [International Journal of Innovative Research in Science Engineering and Technology (IJIRTSE) ISSN: 2395-5619, Volume – 2, Issue – 7. July 2016] - "Smart helmet system using AD for vehicle protection"

[20] Nimmy James, Aparna.C, Tenna.P John, International Journal of Research in Computer and CommunicationTechnology, Vol 3, Issue 1, January- 2014 – "AD System"

[21] Ms. Subia Sayeed Department of Electronics and communication, VVIET, Mysore, India [International Journal of Scientific & Engineering Research Volume 2, Issue 12, December-2011,ISSN 2229-5518] – "Drunken drive protection system"

[22] Prashanth.K P1, Kishen Padiyar2, Naveen Kumar.P H3, K.Santhosh Kumar, Dept. of Mechanical Engineering, East West Institute of Technology, Bangalore, India [International Journal of Engineering Research & Technology (IJERT) ISSN: 2278- 0181, IJERTV3IS100754, Vol. 3 Issue 10, October- 2014] - "Road Accident Avoiding System using Drunken Sensing Technique"

[23] Karne, RadhaKrishna, and T. K. Sreeja. "ROUTING PROTOCOLS IN VEHICULAR ADHOC NETWORKS (VANETs)." *International Journal of Early Childhood* 14.03: 2022.

[24] Karne, RadhaKrishna, et al. "Optimization of WSN using Honey Bee Algorithm."

[25] RadhaKrishna Karne, Dr TK. "COINV-Chances and Obstacles Interpretation to Carry new approaches in the VANET Communications." *Design Engineering* (2021): 10346-10361.

PERSONAL PROFILE

Name	: Sk. Mustak
Father Name	: Sk. Karimulla
Class	: B. Tech
Branch	: Electronics and Communication Engineering
Roll No	: 20F01A04J4
Date of Birth	: 05-11-2001
Marital Status	: Unmarried
Mobile No	: 8309410456
Email Id	: mustakshaik950@gmail.com
Permanent Address	: D. No.5-52, Relaince Tower, Epurupalem, Bapatla dist-523166