International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Car to Car Communication Using Iot

Prof Ashwini K[1], DrSuganda P[2], Akash P[3], Netravati S[4], Sushmita M[5]

Electronics and Communication, Rural Engineering College Hulkoti

INTRODUCTION

The target of including learning is to make a canny well ordered space, which is quickly usable and encouraged into our homes, our workplaces, our paths, our vehicles, and all over the place. This new idea must be immaterial; it must mix in with our standard condition and should be available when we need it.One of the uses of this thought involves giving our vehicles and avenues with capacities to make the street logically secure (data about the traffic, difficulties, dangers, potential ad libbed courses, air, and so on.) and to make our break and about ceaselessly wonderful (Internet get the chance to, sort out beguilements, helping two individuals search after each other out on the town, visit, and so on.). These applications are ordinary instances of what we call an Intelligent Transportation System (ITS) whose objective is to improve security, suitability and satisfaction in street transport using ne Customary traffic the board structures depend upon joined foundations where cameras and sensors executed along the road assemble information on thickness and traffic state and transmit this data to a central unit to process it and settle on legitimate decisions. This kind of framework is all around expensive concerningaffiliation and is delineated by a long response time for managing and data move in a setting where data transmission suspension is fundamental and is essential in this sort of structure. Furthermore, these contraptions put on roads need accidental and over the top assistance. Thus, for enormous scale blueprint of this sort of structure, colossal undertaking is required in the correspondence and sensor foundation. Not w advances for data and correspondence (NTIC). withstanding, with the lively improvement of remote correspondence headways, region and sensors, another decentralized (or semi-concentrated) designing reliant on vehicle-to-vehicle trades (V2V) has made a verifiable interest these latest couple of years for vehicle makers, R&D social order and telecom executives. This sort of structure depends upon an appropriated and self-ruling framework and is contained the vehicles themselves without the help of a fixed foundation for information organizing.

1.1 IOT:

Internet of things (**IoT**) describes physical objects (or groups of such objects) with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.The Internet of things (IoT) falls under the Electronics & Communication and ComputerScience Engineering. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they The only need to be connected to a network,and be individually addressable.[7][8]

The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, as well as machine learning.Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things.In the consumer market, IoT technology is most synonymous with products

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems.
There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks.

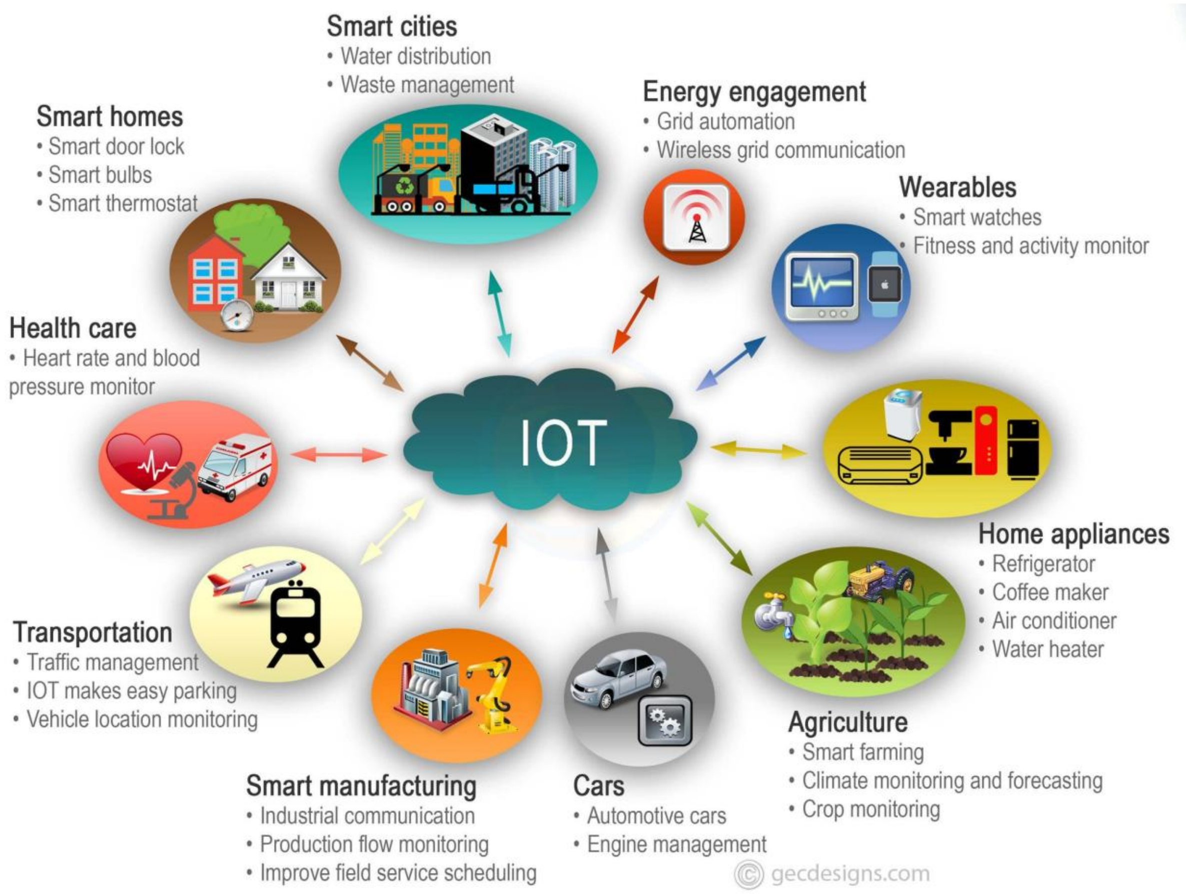


Fig1.1: A Modern Example of IOT

1.1.1 HISTORY:

Such as Electronic Route Guidance System (ERGS) and CACS in the United States and Japan respectively.While the term Inter-Vehicle Communications (IVC) began to circulate in the early 1980s. Various media were used before the standardization activities began, such as lasers, infrared, and The beginnings of vehicular communications go back to the 1970s. Work began on projects radio waves.
The PATH project in the United States between 1986 and 1997 was an important breakthrough in vehicular communications projects.Projects related to vehicular communications in Europe were launched with the PROMETHEUS project between 1986 and 1995.Numerous subsequent projects have been implemented all over the world such as the Advanced Safety Vehicle (ASV) program,CHAUFFEUR I and II,FleetNet,CarTALK 2000,etc.

In the early 2000s, the term Vehicular Ad Hoc Network (VANET) was introduced as an application of the principles of Mobile Ad-Hoc Networks (MANETs) to the vehicular field. The terms VANET and IVC do not differ and are used interchangeably to refer to communications between vehicles with or without reliance on roadside infrastructure, although some have argued that IVC refers to direct V2V connections Only.Many projects have appeared in EU, Japan, USA and other parts of the world for example, ETC, SAFESPOT, PREVENT, COMESafety, NOW, IVI.
Several terms have been used to refer to vehicular communications. These acronyms differ from each other either in historical context, technology used, standard, or country (vehicle telematics, DSRC, WAVE, VANET, IoV, 802.11p, ITS-G5, V2X). Currently, cellular based on 3GPP-Release 16and WiFi based on IEEE 802.11p have proven to be potential communication technologies enabling connected vehicles. However, this does not negate that other technologies for example, VLC, ZigBee, WiMAX, microwave, mmWave are still a vehicular communication research area.
Many organizations and governmental agencies are concerned with issuing standards and regulation for vehicular communication (ASTM, IEEE, ETSI, SAE, 3GPP, ARIB, TTC, TTA, CCSA, ITU, 5GAA, ITS America, ERTICO, ITS Asia-Pacific). 3GPP is working on standards and specifications for cellular-based V2X communications, while IEEE is working through the study group Next Generation V2X (NGV) on the issuance of the standard 802.11bd.

1.1.2 APPLICATIONS OF IOT:

The number of connected devices now dwarfs the number of humans on earth.
Researchers at Frost & Sullivan put the number of active IoT-connected devices at 41.76 billion in 2023. IoT Analytics researchers estimate the number at 16.7 billion active endpoints in 2023, while Statista estimates 15.14 billion
Despite variations in the actual figures and what's included in the count, one thing is clear: There's a mind-blowing number of IoT devices in the world.
That might not be surprising, though, considering the multiple areas where IoT is being used. Those IoT connections span the globe and permeate nearly all places: homes, offices, factories, farms, vehicles and even space.
Here's a detailed look at the top 12 use cases of IoT.

1. Self-driving and connected vehicles

Autonomous vehicles are one of the most notable examples of IoT in action, with longtime automotive companies such as BMW Group, Ford Motor Company and General Motors along with newer entries such as Tesla, all working on self-driving vehicles.
Self-driving cars and trucks use a slew of connected devices to safely navigate roadways in all sorts of traffic and weather conditions. The technologies in use include AI-enabled cameras, motion sensors and onboard computers.
Although regulatory, safety and technical concerns over the use of self-driving vehicles exist, the market for self-driving vehicles is expected to grow rapidly in upcoming years. An Allied Market Research report predicted that the global autonomous vehicle market will hit nearly $2.2 trillion by 2030, up from $76.13 billion in 2020.

Meanwhile, IoT connections also exist on conventional vehicles, with manufacturers installing connected devices to monitor performance and manage computerized systems.

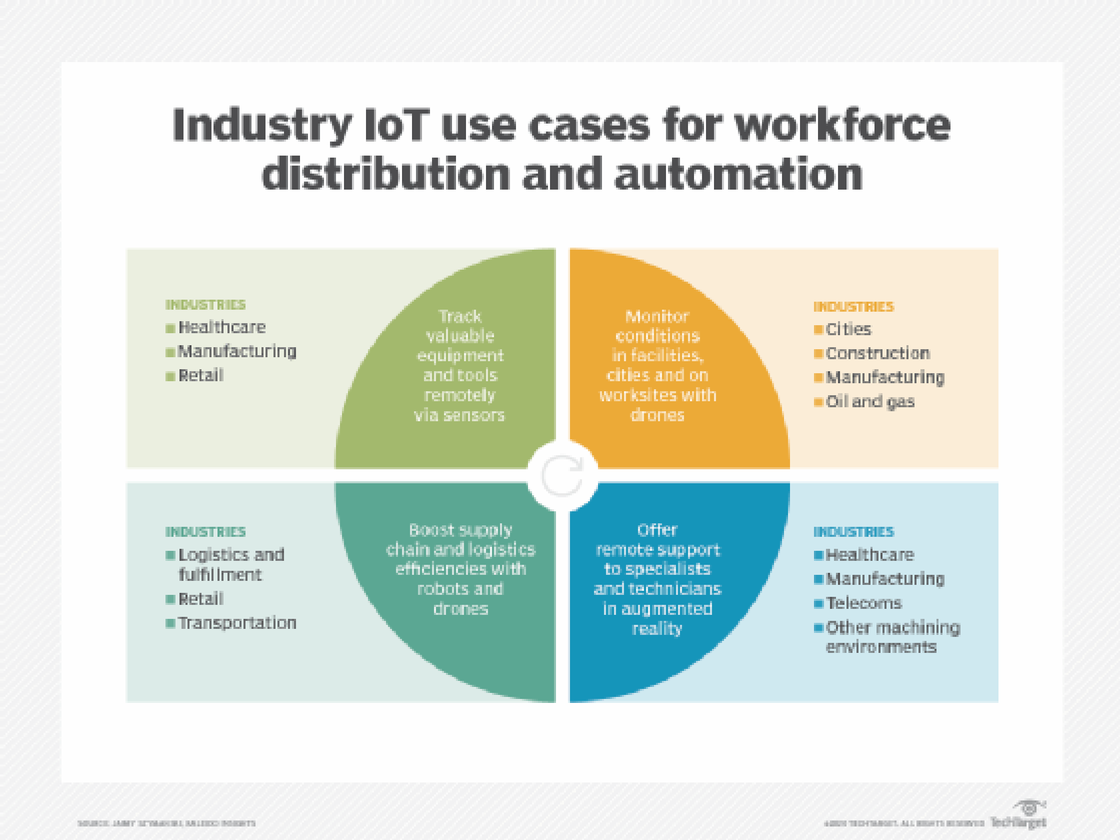
Commercial fleets such as municipal buses and corporate delivery trucks are often fitted with additional IoT technologies, such as connected systems to monitor for safety issues. Personal cars and trucks can be fitted with similar technology, which frequently comes from insurance companies, that collects and transmits telemetry data to verify good driving habits.

2. Logistics and fleet management

Companies are using sensors, telematics, GPS and analytics to see where their vehicles are at any given moment, estimate when they'll arrive at their destination and whether external conditions warrant updating routes or expected arrival times.

This technology ecosystem also enables companies to identify ways to improve operations through predictive maintenance, more driver training and route optimization.

Although logistics companies are among the primary users of such technologies, others also use IoT to track their fleets. For example, bike and scooter rental companies -- sometimes known as the micromobility industry -- use IoT to know where their wares are at any point in time.



IoT can add business value to several industries, including construction, manufacturing, retail and transportation.

3. Traffic management

Part of what enables self-driving cars is smart traffic management, which is also powered by IoT.Like the vehicles themselves, roadway infrastructure has become more connected during the past decade, with cameras, sensors, traffic light controls, parking meters and even smartphone traffic apps transmitting data that's used to help avert traffic jams, prevent accidents and ensure smooth travel.

For example, cameras detect and transmit data about traffic volume to central management groups that analyze the information to determine whether, what and when mitigation steps must be taken.
Sensors on traffic signals can detect varying levels of light in the sky and adjust the brightness of the signals, helping ensure they're always visible to drivers.
Connected devices can be used to detect open parking spaces and transmit that information to kiosks or apps to alert drivers.
Monitors on bridges collect and transmit data for analysis about their structural health, alerting authorities to maintenance needs before there's any sort of issue or failure.

4. Smart grids, including smart meters

Utilities are also using IoT to bring efficiency and resiliency to their energy grids.
Historically, energy flowed one way along the grid: from the generation site to the customer. However, connected devices now enable two-way communication along the entire energy supply chain: from generation through distribution to use, thereby improving the utilities' ability to move and manage it.
Utilities can analyze real-time data transmitted by connected devices to detect blackouts and redirect distribution and respond to changes in energy demand and load.
Meanwhile, smart meters installed at individual homes and businesses provide information about both real-time use and historical usage patterns that customers and the utilities can analyze to identify ways to improve efficiency.
Experts see the development of a fully smart grid as critical to key sustainability goals but note significant work is required to get to that point. A 2022 report from the International Energy Agency, for example, said that "investments in smart grids need to more than double through to 2030 to get on track with the Net Zero Emissions by 2050 Scenario, especially in emerging markets and developing economies."

5. Environmental monitoring

Connected devices can collect data that indicates the health and quality of air, water and soil, as well as fisheries, forests and other natural habitats. They can also collect weather and other environmental data.
As such, IoT delivers the ability to not only access more real-time data about the environment at any given time and place, but it also enables a range of organizations in various industries to use that data to glean actionable insights.
Such information can help government agencies better monitor and even predict natural disasters, such as tornados, as well as better manage and protect land and wildlife populations. Companies can use this data to better limit their carbon footprint, more effectively document compliance with environmental regulations and more efficiently plan around weather conditions that affect their business.
The market for environment-related IoT systems is seeing strong growth. A spring 2023 report from Global Industry Analysts valued the worldwide market for environmental sensing and monitoring tech at $14 billion in 2022 but expects it to grow to $19.3 billion by 2030.

6. Connected buildings and building security

Property owners are using the power of IoT to make buildings smarter, meaning they're more energy-efficient, comfortable, convenient, healthier and possibly safer.

An IoT ecosystem in a commercial building could include monitoring of the HVAC infrastructure that uses real-time data and automation technologies to constantly measure and adjust the temperature for optimum energy efficiency and comfort. Meanwhile, cameras using AI could aid in crowd management to ensure smooth flow of foot traffic or support public safety at large-scale events such as sold-out concerts.
On the home front, consumers can install smart technologies, such as door locks, appliances, thermostats and smoke detectors, which help them with their everyday needs by, for example, coordinating temperature controls to the owners' schedules.
Additionally, IoT capabilities power modern security systems both in commercial and residential buildings with connected cameras and sensors detecting and registering movement or activity -- i.e., a doorbell ringing. Those cameras and sensors then transmit that information to other systems, which can be programmed to analyze the data and automated to take specific actions based on that data, or to actual humans, such as homeowners who can determine what course of action to take.
This IoT market segment is also seeing remarkable growth: According to a 2023 report, the global smart building market stood at $72.8 billion in 2022 and is predicted to hit $304.3 billion by 2032, with a compound annual growth rate of 15.8% over the 10-year span.

7. Smart cities

Smart cities are consolidating IoT deployments across many facets to give them a holistic view of what's happening in their jurisdictions.
As such, smart cities incorporate connected traffic management systems and their own smart buildings. They might incorporate private smart buildings, too. Smart cities might also tie into smart grids and use environmental monitoring to create an even larger IoT ecosystem that provides real-time views of the various elements that affect life in their municipalities.
Similar to smaller, more confined IoT deployments, the objective with smart cities is to collect real-time data for analysis that provides insights that municipal officials can then use for better decision-making and automated controls to yield more efficient, effective, resilient and safer communities. Case in point: Copenhagen, the capital of Denmark, is using IoT technologies to reach its goal of being a carbon-neutral city by 2025.

8. Supply chain management

Supply chain management has been undergoing modernization, thanks to low-power sensors, GPS and other tracking technologies that pinpoint assets as they move along a supply chain. Such information lets managers both more effectively plan and more confidently reassure stakeholders about the location of items shipped or received.
That visibility is beneficial, but it's only the start of the evalue proposition that IoT brings tothis disciplin. IoT technologies can also monitor and manage delivery requirements, for example, measuring and maintaining a specified temperature throughout transport to ensure quality and safety controls. Additionally, back-end analytics capabilities can use IoT-generated data to determine supply chain improvements, such as more efficient routes or shipping times.

9. Digital payments

IoT also has a role to play in digital payments, which will continue to expand, according to researchers at Frost & Sullivan. The firm's post "The Top Growth Opportunities for IoT in 2023," states that as "cities become more digitally integrated, the use of payments will increase, and IoT will play a critical role."

10. Healthcare and consumer health and wellness

The healthcare industry as well as the consumer health and wellness market have numerous examples of IoT in action.

Medical institutions use connected devices throughout their care delivery processes, with many of those devices specifically designed to monitor patient vital signs and health conditions. Connected monitors, for example, can monitor, record and transmit a patient's heart rate, glucose levels or blood pressure; some also can determine if readings are within or outside a predetermined acceptable range and alert the patient or healthcare provider if that happens.

Consumers also have access to such devices and more, with smartwatches and other wearables capable of tracking, transmitting and analyzing different wellness markers, from daily steps taken to the amount of quality sleep received.

11. Predictive maintenance

Another prominent use of IoT, as well as one of the most widespread examples of it, is in understanding machine health and identifying in advance when they'll need service.

Sensors are placed in a plethora of different mechanical systems, from airplanes and mining equipment to manufacturing assembly lines and household appliances. These sensors collect, store and transmit data about performance, which when analyzed can pinpoint maintenance needs and potential problems before they're required -- enabling owners to take preventative action thereby avoiding degraded performance and equipment failures.

12. Agricultural, commercial, industrial and retail management and operations

IoT has numerous applications in nearly every sector, from agriculture to space exploration

* For example, manufacturing uses IoT to monitor factory production and provide predictive maintenance on equipment. A manufacturer might use machine-to-machine connected devices as part of an industrial IoT deployment to more accurately map workloads. A factory could track wear and tear on equipment to schedule preventive maintenance at an optimal time. Companies can use employee badges or wearable devices embedded with RFID chips to manage and control physical access to facilities.
* Farmers can opt for location technologies integrated with environmental monitors and their field equipment to both automate and maximize their seed allocations.
* Transportation and logistics companies, including international shipping companies, use IoT technologies to track their fleets and track and monitor goods as they're transported. Some also track the conditions in which goods are stored; a transportation company, for example, can track, monitor and adjust the temperature in a refrigerated truck to ensure it's kept within an optimal temperature range for the items being transported.
* Retailers are using IoT systems to support automation and robotics capabilities in their warehouses. They're also using IoT for inventory control and, increasingly, for in-store customer experience and personalized experience.

1.2 TOOLS:

Keeping track of all the IoT devices on a network, ensuring they are secure and operating correctly are critical -- and challenging -- activities. IoT monitoring tools simplify keeping devices updated with the latest software and firmware, automatically provide status information and identification, and are part of a larger cybersecurity strategy.

IT administrators can deploy and use IoT monitoring tools on their own or in conjunction with IoT analytics applications, security software, and identity and access management platforms.

Features to look for in IoT monitoring software

Remote monitoring

Remote monitoring of IoT devices is a must; devices are often deployed in inaccessible or hard-to-reach locations. It's not practical or economical to send IT teams on location to physically inspect or monitor the devices. It's also not a good use of time to have them switch between several monitoring applications to check in on the various devices. An IoT monitoring application with remote monitoring capabilities eliminates the extra work and centralizes the information IT teams need to do their work.

Automated discovery and alerting

Automation for IoT device monitoring doesn't need to be sophisticated. Setting baseline performance limits that trigger alarms when an IoT device crosses the threshold is enough. This level of automation reduces the workload for IT teams by engaging them only when there is a true issue that requires their attention. Most monitoring software comes with default performance thresholds that IT can use as they are or tweak them to meet their environment's needs.

Connection security

Whether they're connected through a private network or the internet, IoT devices can be vulnerable to cyber attacks. Hackers can take advantage of default device passwords to enter a network and then move about at will. Many companies fail to change these default passwords because there are so many devices to manage. They might not be aware of the devices at all, or their security processes might mistakenly ignore IoT devices. An IoT monitoring application reduces this risk because it finds and identifies every connected device, monitors performance 24/7 and has the ability to force regular password updates.

Free trial to assess suitability

Many IT teams rely on the free monitoring tools distributed by IoT device manufacturers. The market for IoT-specific monitoring tools isn't well developed, and many organizations hesitate to pay for one. However, free tools are sometimes poorly designed; they're often more of a marketing tool than a true tech tool. A free trial of a paid application can help IT teams understand whether an IoT monitoring tool is suitable for their tech stack, processes and infrastructure

Free trial to assess suitability

Many IT teams rely on the free monitoring tools distributed by IoT device manufacturers. The market for IoT-specific monitoring tools isn't well developed, and many organizations hesitate to pay for one. However, free tools are sometimes poorly designed; they're often more of a marketing tool than a true

tech tool. A free trial of a paid application can help IT teams understand whether an IoT monitoring tool is suitable for their tech stack, processes and infrastructure

IoT monitoring software options to consider

AWS IoT Device Management

AWS IoT Device Management is a specialized module in the Amazon family of cloud services. It's easy to set up, and it's accessible via a web browser. It can handle bulk and individual IoT device registration, and it groups devices to make monitoring and updating more efficient. It also offers superior intrusion prevention and detection. AWS prices by number of devices and usage.

Splunk for Industrial IoT

Splunk for Industrial IoT was designed for industrial markets but is useful for any industry. It uses built-in machine learning and AI features to predict performance and device health and enforce security. It offers a wide range of integrations, and it scales easily across enterprise-sized networks. It's available for a 60-day free trial as part of Splunk Enterprise.

DatadogIoT Monitoring

DatadogIoT Monitoring is a cloud-based infrastructure monitoring suite with specialized features for IoT device monitoring. It is SaaS-based and includes high-level security across all transmissions. It also maintains two-way encrypted communications with all devices. It supports device autodiscovery and updates network topology in real time as needed. Flexible pricing options let customers scale their IoT monitoring. Datadog offers a free 14-day trial for an unlimited number of devices.

Domotz

Domotz is another cloud-based IoT device monitoring application with a long list of features. It uses highly encrypted connections with all monitored devices and integrates with a wide range of other tools, including those that managed service providers use. It offers a simple but effective dashboard that is easy to use and includes extensive security scanning, such as VPN on demand and perimeter network security scanning. It's available without a contract, and organizations can test it out with a 21-day free trial.

1.3 ADVANTAGES:

1. Smart home devices

Smart home devices are the most common consumer IoT products. AI assistant speakers, smart locks and connected appliances are just some smart home device examples that help people live more efficient, connected lives.

An IoT-connected fridge can tell homeowners when they have low stock of specific groceries. Similarly, an IoT doorbell could sense when someone approaches the front door, even before they ring the bell.

Smart home devices have faced some controversy regarding security. Hackers have strategies to take advantage of these devices and gain access to homes. Smart locks are particularly infamous for smart home hacking. These devices can be easy to hack because anyone can access a smart home device with a simple passcode or the tap of a button in a smartphone app.

2. Industrial sensors

Industrial IoT (IIoT) supports use cases in automation, safety and data collection. With industrial sensors, companies can gain valuable insights and capabilities in the workplace, warehouses, manufacturing plants and construction sites.

Industrial vehicles and machinery can have IIoT sensors that track performance and output. Such sensors also enable predictive maintenance, which prevents expensive breakdowns, costly repairs and downtime.

IoT sensors also monitor various processes, such as inventory management in warehouses. Organizations of any size can implement digitized inventory management with IoT sensors that monitor item status and condition. The most tech-literate companies have been known to double their earnings per employee compared to companies without much tech buy-in.

3. Industrial robots

Industrial organizations also use IoT technology in robots. For example, many warehouses use autonomous courier robots. These robots have IoT sensors that report their location and performance on the warehouse floor. IoT sensors help with navigation and can scan QR codes throughout the warehouse to guide their route.

In industrial fields, IoT robots can fill in labor gaps and improve operational efficiency. Managers must train employees on how to safely work alongside these robots. Inadequate training may lead to accidents or misuse of the robots, even though they can largely function autonomously.

Managers must ensure they choose the right connectivity option. Large facilities like warehouses may require a specialized approach to achieve reliable connectivity throughout. Because Ethernet, Wi-Fi, low-power WAN, satellites and cellular networks can all support IoT devices, organizations must research what option is best for their robotic deployment.

4. Healthcare devices

In healthcare facilities, IoT devices are crucial to patient care. IoT medical devices help nurses remotely monitor patient vitals so that nursing teams can provide better care to their departments.

Healthcare IoT technology can also monitor patient health outside the doctor's office. Remote at-home monitoring helps protect patients and avoid repetitive, costly hospital visits. Even smartwatches have become valuable medical IoT tools that provide electrocardiograms and accurately track vital signs, such as heart rate and blood glucose.

Reliability and security are the main challenges of IoT healthcare devices. Like any IoT device, the convenience of connectivity leaves these medical devices vulnerable to hacking. Additionally, if the device loses its connection or malfunctions, it must have strong safeguards in place to prevent patient harm.

Healthcare providers must have strong cybersecurity protocols to protect the sensitive patient data that IoT medical devices collect.

5. Connected cars

IoT has continuously evolved in the automotive industry. The technology is in vehicle entertainment, navigation and maintenance monitoring systems.

Adoption of IoT for automotive use cases will only grow as autonomous vehicles become more advanced and commonplace; many consumers now have advanced driver assistance systems that sense nearby cars or can even help parallel park the car.
IoT makes life easier for both drivers and vehicle manufacturers. With IoT, a car's entertainment or driver assistance system can receive remote updates without any need for the driver to take the car to the dealership.
Over-the-air software updates also enable car manufacturers to stay ahead of potential security risks. Unfortunately, IoT-enabled vehicles have been hacked, including rare instances while on the road.

LITERATURE SURVEY

As part of its objective to help create a European ecosystem for “sustainable, sovereign and safe mobility”, Software République has unveiled H1st vision (Human First vision), a collaborative vehicle concept that incorporates functional and innovative technologies and illustrates human-centred visions of the mobility of the future.
Software République is an open innovation ecosystem, created two years ago by Atos, DassaultSystèmes, Orange, Renault Group, STMicroelectronics and Thales. It also includes seven startups and a partner: Arkamys, Compredict, Epicnpoc, Eyelights, Kardome, Stern Tech and JCDecaux. It has set itself a roadmap to launch 10 new services and products, incubate 50 or more startups and offer services in more than 50 geographies around the world – all by 2025.
H1st vision is described by the consortium as representing a “revolution” in mobility, bringing together 20 concrete operational innovations to offer a “true experience” beyond automotive mobility. It stresses that H1st vision is not just a concept car, but instead a tangible vision of tomorrow’s mobility experience.
The physical vehicle doubles up as a virtual twin in a digital universe, where systems that are currently independent – such as infrastructure, energy, public services and users of different categories – can communicate. This is designed to make it possible to model, visualise and simulate use cases that could arise in the real world. Thanks to its interconnection with its environment, H1st vision is said to be in “in constant dialogue” with digital and physical ecosystems.
Connected vehicle technologies include data on available charging stations from Mobilize; the Plug Inn app, a French peer-to-peer EV charging community, delivered by Renault; Orange smart parking space sensors; and a module combining Narrowband IoT (NB-IoT) cellular communication and localisation by satellite (GNSS) from STMicroelectronics.
The operational features are said to be centred on people – taking care of the driver, their passengers and other road users. These features include all-new and secure biometric access control, a one-of-a-kind sound experience inside, optimised vehicle range and charging, a driver and vehicle health monitor and assistant, and predictive alerts to protect occupants and other people on the road.
When drivers approach their vehicle, postural then facial recognition software kicks in to allow access to the passenger compartment and then enable ignition. Users create a profile on a tablet, and their personal data (name, height, a photo of their face and a video of them walking) is entered, encoded and stored in what is said to be an ultra-secure virtual container integrated into the car, the Digital ID Wallet.
Using cameras on the wing mirrors, the vehicle recognises the registered driver and passengers approaching on either side, based on their posture, when they are up to six metres away. An avatar

welcomes the user and assists them with all the available features. H1st vision also has an in-car payment interface and a smart parking assistant.

To make mobility sustainable, silicon carbide parts are used in the critical systems powering electric vehicles and in the charging stations, said to increase efficiency significantly. Powerbox charging stations work both ways, with vehicle-to-grid functionality so the car can also support the grid or help power your home during consumption spikes.

An array of physical and virtual sensors watch over the health of the occupants and, in the event of an incident, H1st vision can place a mobile or satellite call to emergency services. The state of wear of key vehicle components, such as the battery or tyres, is also monitored, and H1st vision can generate its own health certificate.

The virtual twin in a digital city shows the car's extended connection to its environment – city 4.0, other people using the road, other infrastructure, and so on – to spot potential hazards, keep people safe if they are vulnerable and optimise rescue vehicle traffic flows.

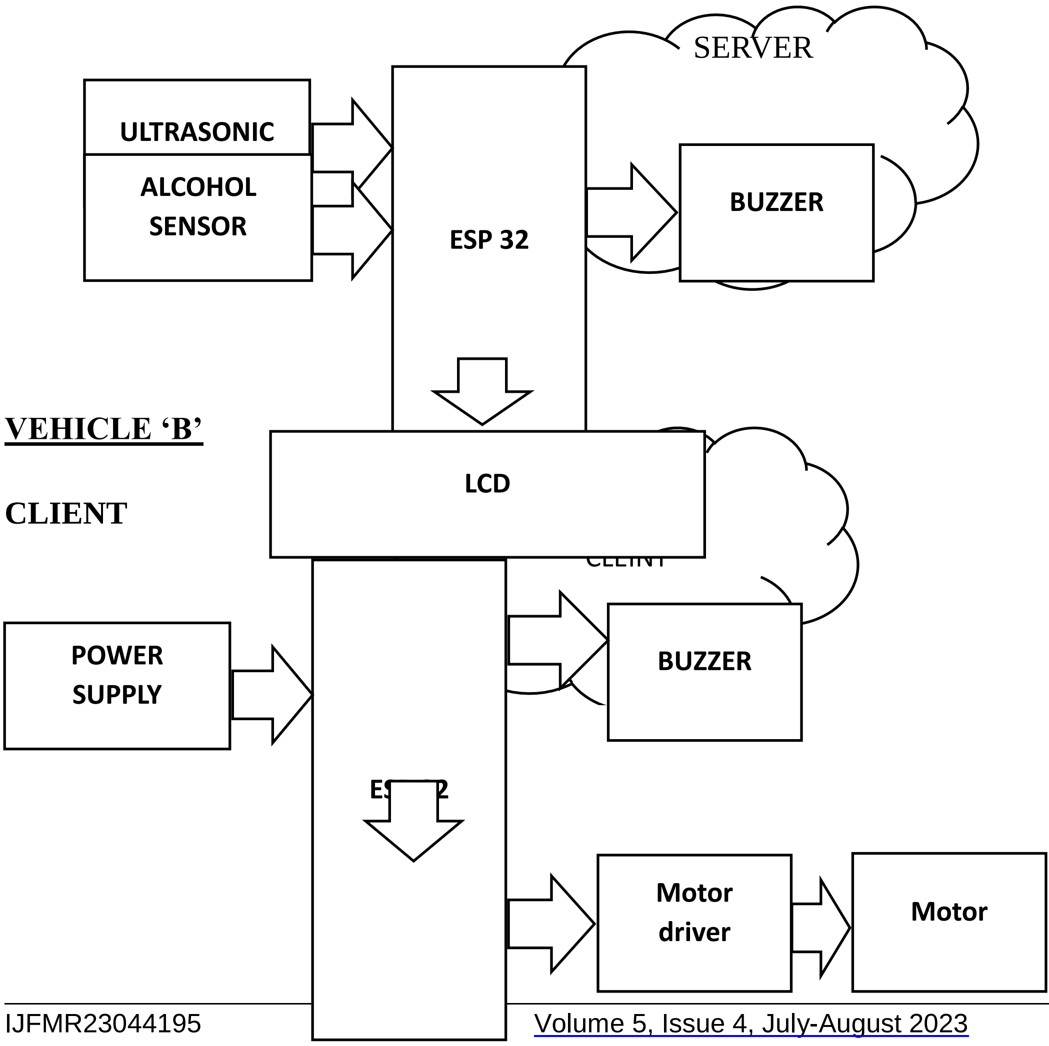
The H1st vision concept car is a connected, physical and virtual object that demonstrates both a robust method for collaboration between the partners and how the technologies of DassaultSystèmes, Orange, ST Microelectronics, Renault Group and Thales complement each other to build the mobility of the future," said Software République's chief operating officer, Eric Feunteun.

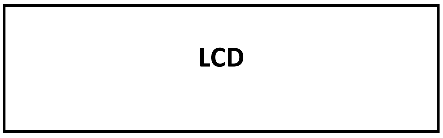
BLOCK DIAGRAM

<u>VEHICLE 'A'</u>

<u>VEHICLE 'B'</u>

CLIENT

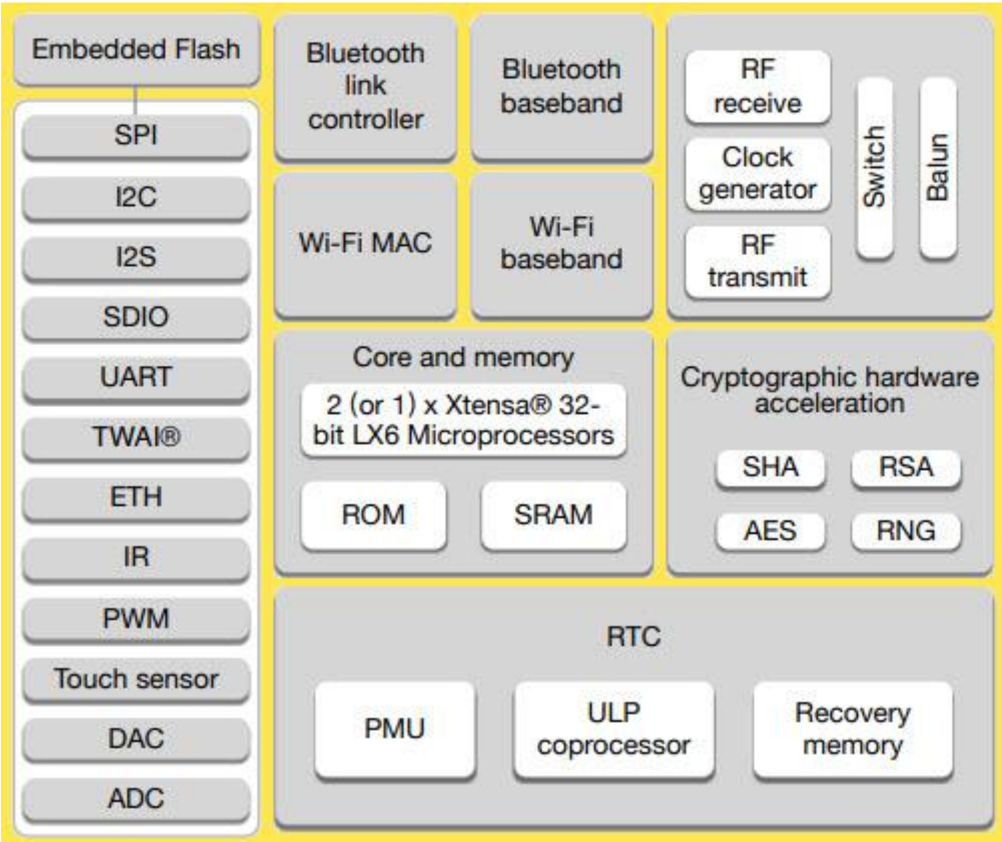




3.1 ESP32

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.



Another important thing to know about ESP32 is that it is manufactured using TSMC's ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

Specifications of ESP32

ESP32 has a lot more features than ESP8266 and it is difficult to include all the specifications in this Getting Started with ESP32 guide. So, I made a list of some of the important Specifications of ESP32 here. But for complete set of specifications, I strongly suggest you to refer to the Datasheet.

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.

- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I^2C, 2 x I^2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

Fig 3.1: ESP32

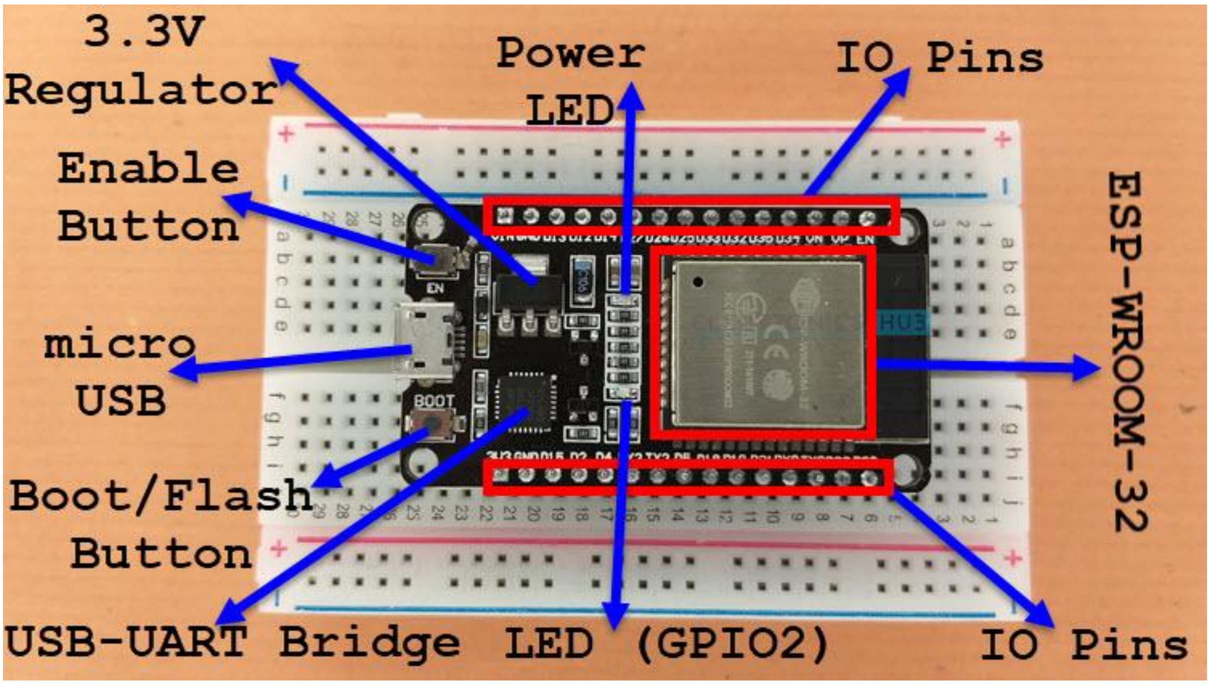
Layout

We will see what a typical ESP32 Development Board consists of by taking a look at the layout of one of the popular low-cost ESP Boards available in the market called the ESP32 DevKit Board.

The following image shows the layout of an ESP32 Development Board which I have.

IMPORTANT NOTE: There are many ESP32 Boards based on ESP-WROOM-32 Module available in the market. The layout, pinout and features vary from board to board.

The board which I have has 30 Pins (15 pins on each side). There are some board with 36 Pins and some with slightly less Pins. So, double check the pins before making connections or even powering up the board.

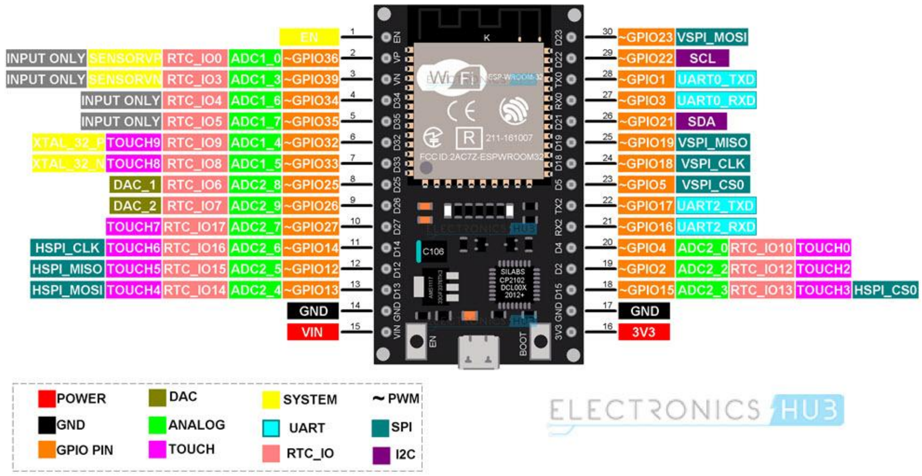


As we can see from the image, the ESP32 Board consists of the following:

- ESP-WROOM-32 Module
- Two rows of IO Pins (with 15 pins on each side)
- CP2012 USB – UART Bridge IC
- micro–USB Connector (for power and programming)
- AMS1117 3.3V Regulator IC
- Enable Button (for Reset)
- Boot Button (for flashing)
- Power LED (Red)
- User LED (Blue – connected to GPIO2)
- Some passive components

An interesting point about the USB-to-UART IC is that its DTR and RTS pins are used to automatically set the ESP32 in to programming mode (whenever required) and also rest the **board after programming.**

Pinout of ESP32 Board



This pinout is for the 30 – pin version of the ESP Board. In the pinout tutorial, I will explain the pin out of both the 30 – pin as well as the 36 – pin version of the ESP Boards.

3.2 LED'S

A **light-emitting diode** (**LED**) is a semiconductor light source. LEDs are used as indicator lamps in many devices and are increasingly used for other lighting.

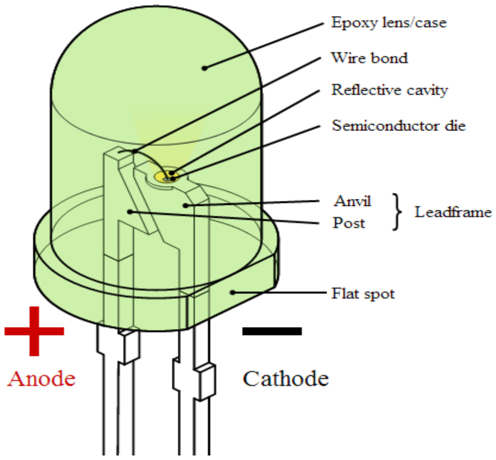
The LED consists of a chip of semiconducting material doped with impurities to create a *p-n junction*. As in other diodes, current flows easily from the p-side, or anode, to the n-side, or cathode, but not in the reverse direction. Charge-carriers—electrons and holes—flow into the junction from electrodes with different voltages. whena electron meet a hole, it falls into lower level, and releases the energy in

The wavelength of the light emitted, and thus its color depends on the band gap energy of the materials forming the *p-n junction*. In silicon or germanium diodes, the electrons and holes recombine by a *non-radiative transition*, which produces no optical emission, because these are indirect band gap materials. The materials used for the LED have a direct band gap with energies corresponding to near-infrared, visible, or near-ultraviolet light.

LED development began with infrared and red devices made with gallium arsenide. Advances in materials science have enabled making devices with ever-shorter wavelengths, emitting light in a variety of colors.

LEDs are usually built on an n-type substrate, with an electrode attached to the p-type layer deposited on its surface. P-type substrates, while less common, occur as well. Many commercial LEDs, especially GaN/InGaN, also use sapphire substrate.

Most materials used for LED production have very high refractive indices. This means that much light will be reflected back into the material at the material/air surface interface. Thus, light extraction in LEDs is an important aspect of LED production, subject to much research and development.



Parts of an LED. Although not directly labeled, the flat bottom surfaces of the anvil and post embedded inside the epoxy act as anchors, to prevent the conductors from being forcefully pulled out from mechanical strain or vibration

3.3 LIQUID CRYSTAL DISPLAY (LCD)

The LCD panel used in this block interfaced with micro-controller through output port. This is a 16 character × 2Line LCD module, capable of display numbers, characters, and graphics. The display contains two internal byte-wide registers, one for commands (RS=0) and the second for character to be displayed (RS=1). It also contains a user programmed Ram area (the character RAM) character that can be formed using dot matrix that can be programmed to generate any desired. Two distinguished between these areas, the hex command byte will be signify that the display RAM address 00h is chosen.

LCD can add a lot to our application in terms of providing a useful interface for theuser, debugging an application or just giving it a "professional" look. The most common type of LCD controller is the Hitachi 44780 which provides a relatively simple interface between a processor and an LCD. Using this inter is often not attempted by inexperienced designers and programmers because it is difficult to find good documentation on the interface, initializing the interface can be problem and the displays themselves are expensive.

Connection to a PC parallel port is mostly simple. These displays can handle eight bit input directly. They also need two extra lines to control which kind of data has just arrived and when the data is meant to be stable. Those signals are also called RS (Register Select, instruction or data register) and EN (enable).

So it has to control ten data lines (8 bits + RS + EN) and one common ground (GND) line, which make eleven lines to the parallel port. Data read back is not supported by the driver and so it does not require extra line for this. The following table shows the needed connection.

BLOCK DIAGRAM:

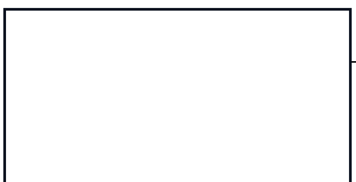


Fig 3.3: 16x2 LCD connection to microcontroller

The pin outs are as follows:

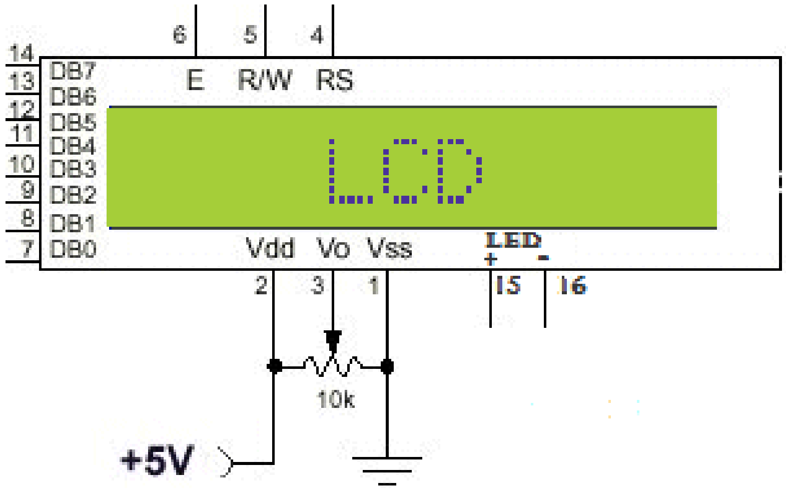


Fig 3.3: LCD Pin out.

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)
Pin no. 15	**LED+**	Anode of LED for Backlit
Pin no. 16	**LED-**	Cathode of LED for Backlit

3.4 ULTRASONIC SENSOR

HC-SR04 Hardware Overview

At its core, the HC-SR04 Ultrasonic distance sensor consists of two ultrasonic transducer. The one acts as a transmitter which converts electrical signal into 40 KHz ultrasonic sound pulses. The receiver listens for the transmitted pulses. If it receives them it produces an output pulse whose width can be used to determine the distance the pulse travelled. As simple as pie!

The sensor is small, easy to use in any robotics project and offers excellent non-contact range detection between 2 cm to 400 cm (that's about an inch to 13 feet) with an accuracy of 3mm. Since it operates on 5 volts, it can be hooked directly to an Arduino or any other 5V logic microcontrollers.

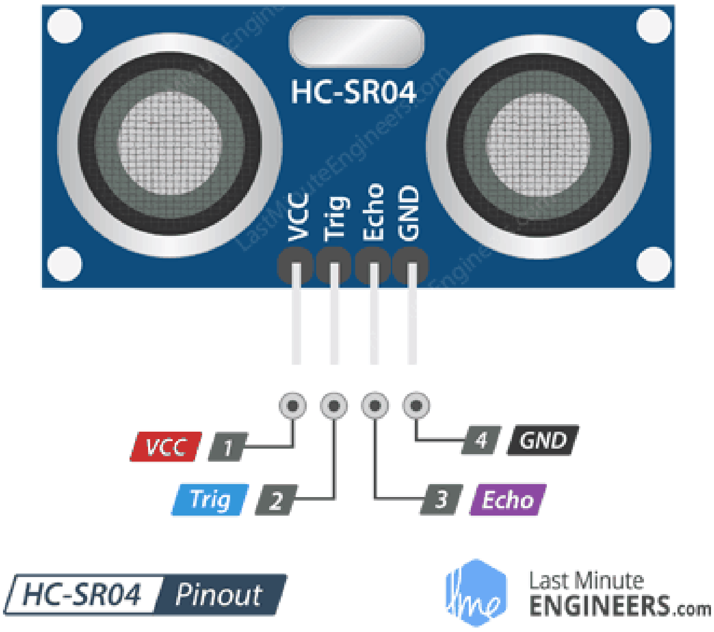
Here are complete specifications:

Operating Voltage	DC 5V
Operating Current	15mA
Operating Frequency	40KHz
Max Range	4m
Min Range	2cm
Ranging Accuracy	3mm
Measuring Angle	15 degree
Trigger Input Signal	10µS TTL pulse
Dimension	45 x 20 x 15mm

HC-SR04 ULTRASONIC SENSOR

Pinout of Ultrasonic Sensor

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com



VCC is the power supply for HC-SR04 Ultrasonic distance sensor which we connect the 5V pin on the Arduino.

Trig (Trigger) pin is used to trigger the ultrasonic sound pulses.

Echo pin produces a pulse when the reflected signal is received. The length of the pulse is proportional to the time it took for the transmitted signal to be detected.

GND should be connected to the ground of Arduino.

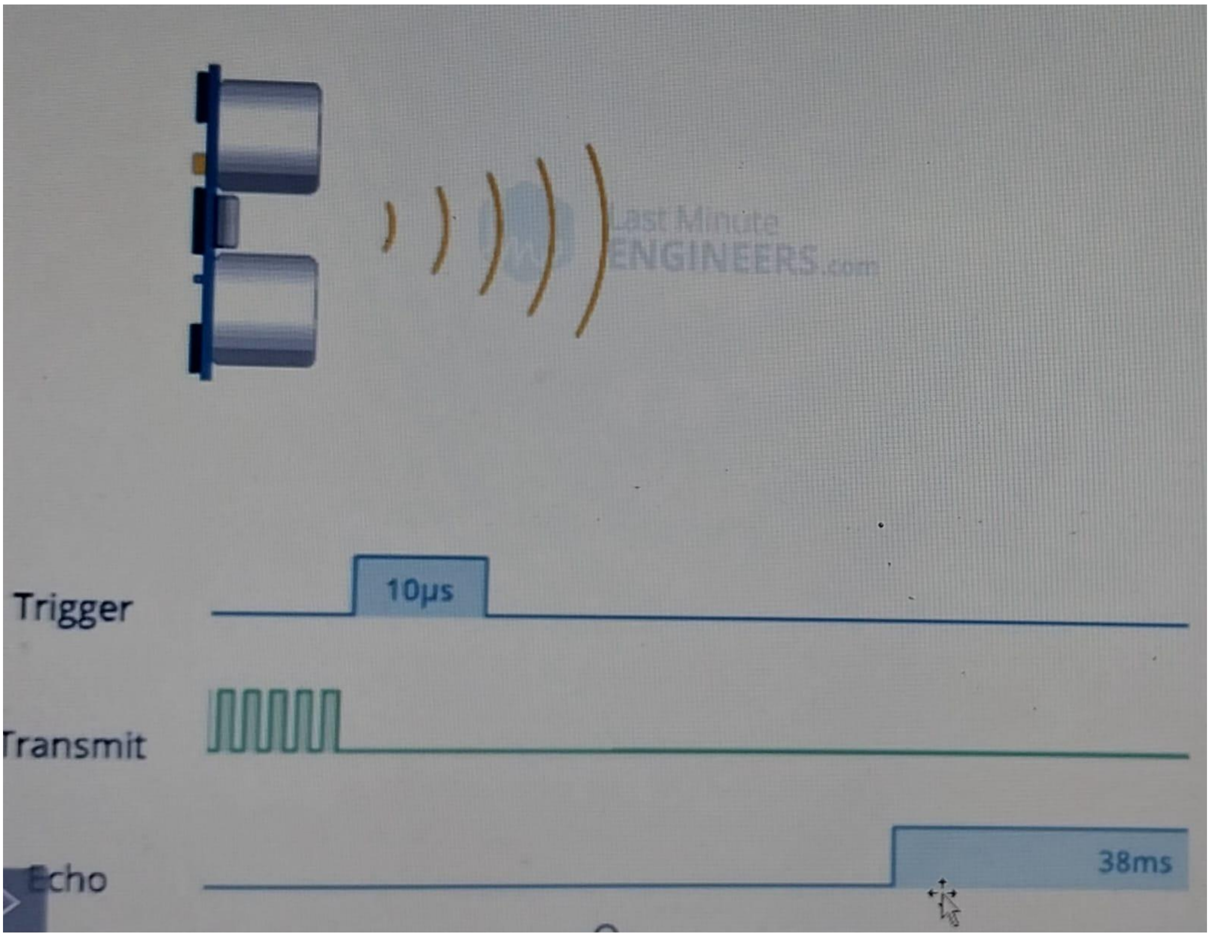
How Does HC-SR04 Ultrasonic Distance Sensor Work?

It all starts, when a pulse of at least 10 µS (10 microseconds) in duration is applied to the Trigger pin. In response to that the sensor transmits a sonic burst of eight pulses at 40 KHz. This 8-pulse pattern makes the "ultrasonic signature" from the

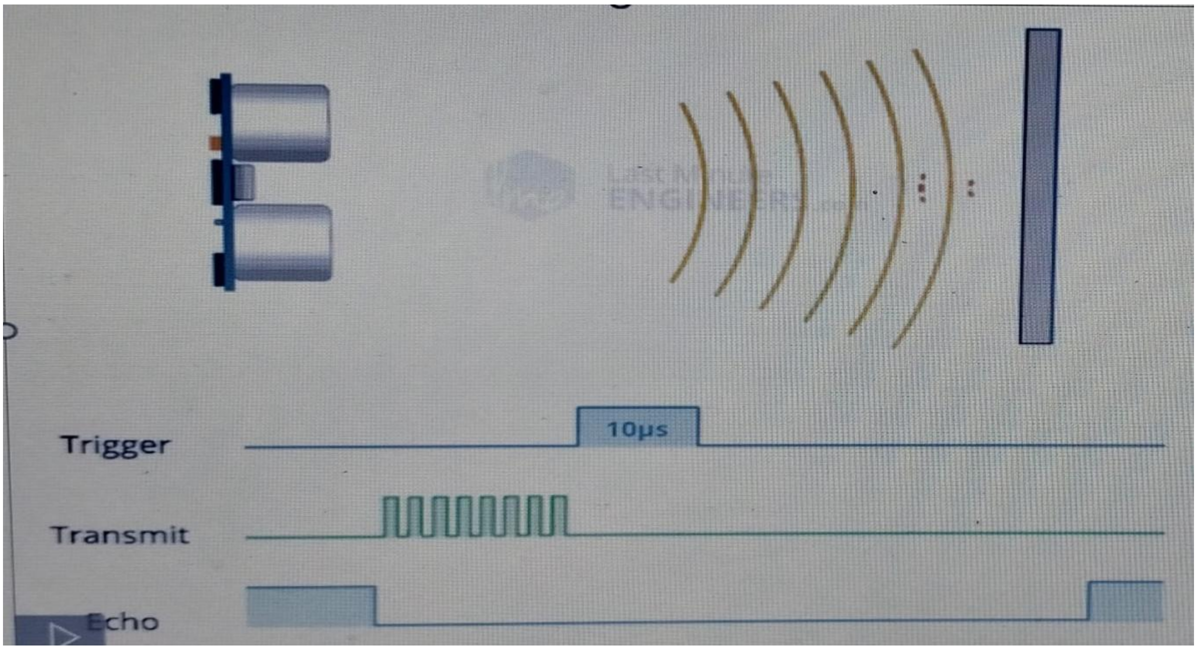
device unique, allowing the receiver to differentiate the transmitted pattern from the ambient ultrasonic noise.

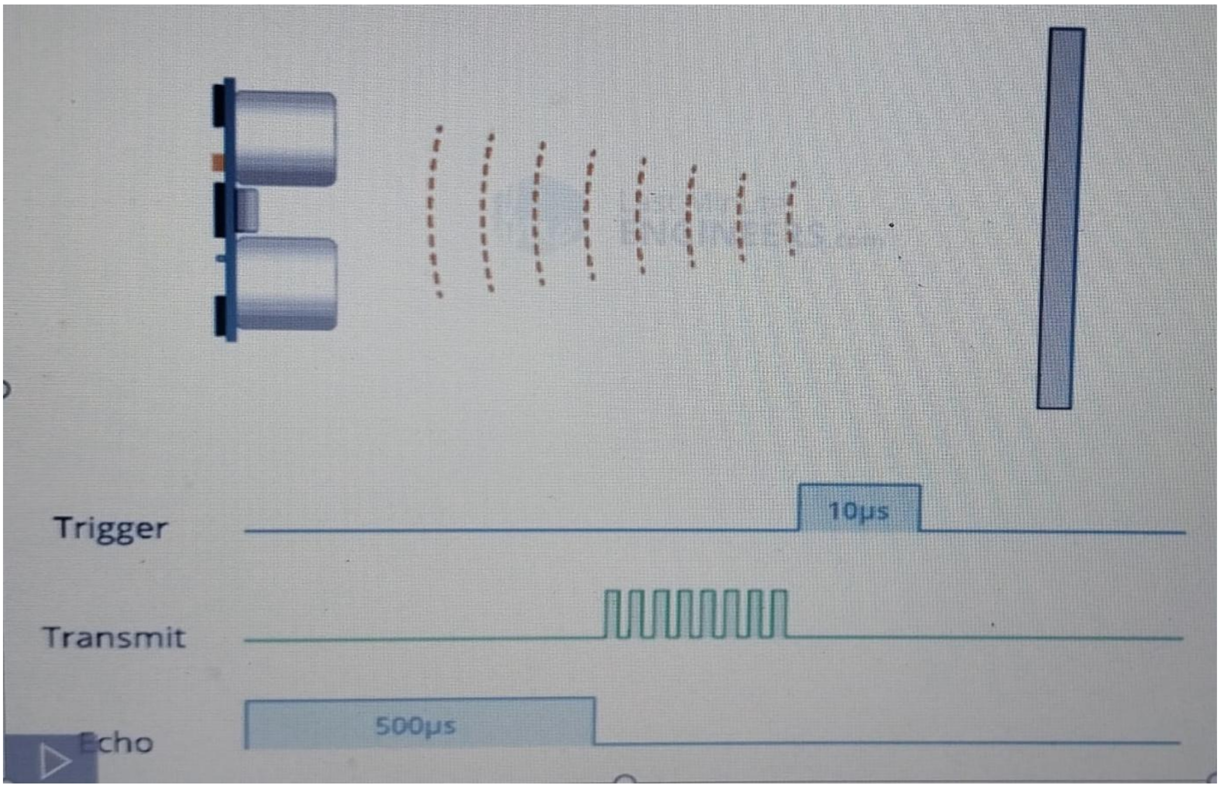
The eight ultrasonic pulses travel through the air away from the transmitter. Meanwhile the Echo pin goes HIGH to start forming the beginning of the echo-back signal.

In case, If those pulses are not reflected back then the Echo signal will timeout after 38 mS (38 milliseconds) and return low. Thus a 38 mS pulse indicates no obstruction within the range of the sensor.

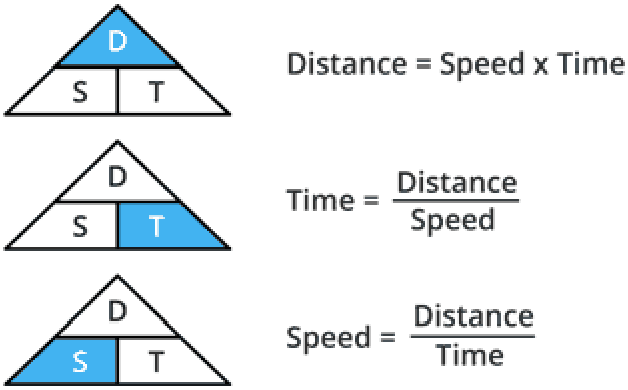


If those pulses are reflected back the Echo pin goes low as soon as the signal is received. This produces a pulse whose width varies between 150 µS to 25 mS, depending upon the time it took for the signal to be received.





The width of the received pulse is then used to calculate the distance to the reflected object. This can be worked out using simple distance-speed-time equation, we learned in High school. In case you forgot, an easy way to remember the distance, speed and time equations is to put the letters into a triangle.



Let's take an example to make it more clear. Suppose we have an object in front of the sensor at an unknown distance and we received a pulse of width 500 µS on
the Echo pin. Now let's calculate how far the object from the sensor is. We will use the below equation.
Distance = Speed x Time
Here, we have the value of Time i.e. 500 µs and we know the speed. What speed do we have? The speed of sound, of course! Its 340 m/s. We have to convert the speed of sound into cm/µs in order to calculate the distance. A quick Google search for "speed of sound in centimeters per microsecond" will say that it is 0.034 cm/µs. You could do the math, but searching it is easier. Anyway, with that information, we can calculate the distance!
Distance = 0.034 cm/µs x 500 µs

But this is not done! Remember that the pulse indicates the time it took for the signal to be sent out and reflected back so to get the distance so, you'll need to divide your result in half.
Distance = (0.034 cm/µs x 500 µs) / 2
Distance = 8.5 cm
So, now we know that the object is 8.5 centimeters away from the sensor.

3.5 ALCOHOL /GAS SENSOR
This is a robust Gas sensor suitable for sensing **LPG**, **Smoke**, **Alcohol**, **Propane**, **Hydrogen**, **Methane** and **Carbon Monoxide** concentrations in the air. If you are planning on creating an indoor air quality monitoring system; breath checker or early fire detection system, MQ2 Gas Sensor Module is a great choice.

What is MQ2 Gas Sensor?MQ2 is one of the commonly used gas sensors in MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type Gas Sensor also known as **Chemiresistors** as the detection is based upon change of resistance of the sensing material when the Gas comes in contact with the material. Using a simple voltage divider network, concentrations of gas can be detected.

MQ2 Gas sensor works on 5V DC and draws around 800mW. It can detect **LPG**, **Smoke**, **Alcohol**, **Propane**, **Hydrogen**, **Methane** and **Carbon Monoxide** concentrations anywhere from 200 to 10000ppm.

Here are the complete specifications

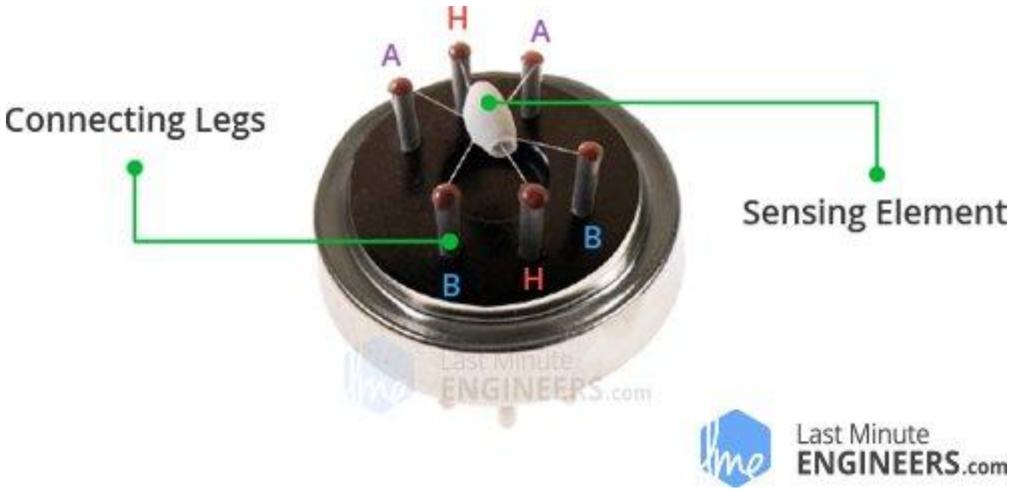
Operating voltage	5V
Load resistance	20 KΩ
Heater resistance	33Ω ± 5%
Heating consumption	<800mw
Sensing Resistance	10 KΩ – 60 KΩ

Concentration Scope	200 – 10000ppm
Preheat Time	Over 24 hour

Internal structure of MQ2 Gas Sensor

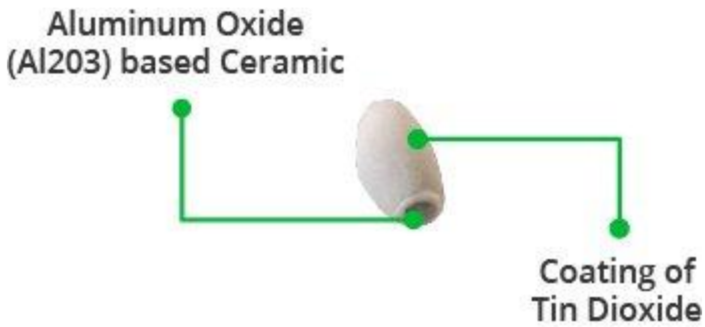
The sensor is actually enclosed in two layers of fine stainless steel mesh called **Anti-explosion network**. It ensures that heater element inside the sensor will not cause an explosion, as we are sensing flammable gases.

It also provides protection for the sensor and filters out suspended particles so that only gaseous elements are able to pass inside the chamber. The mesh is bound to rest of the body via a copper plated clamping ring.

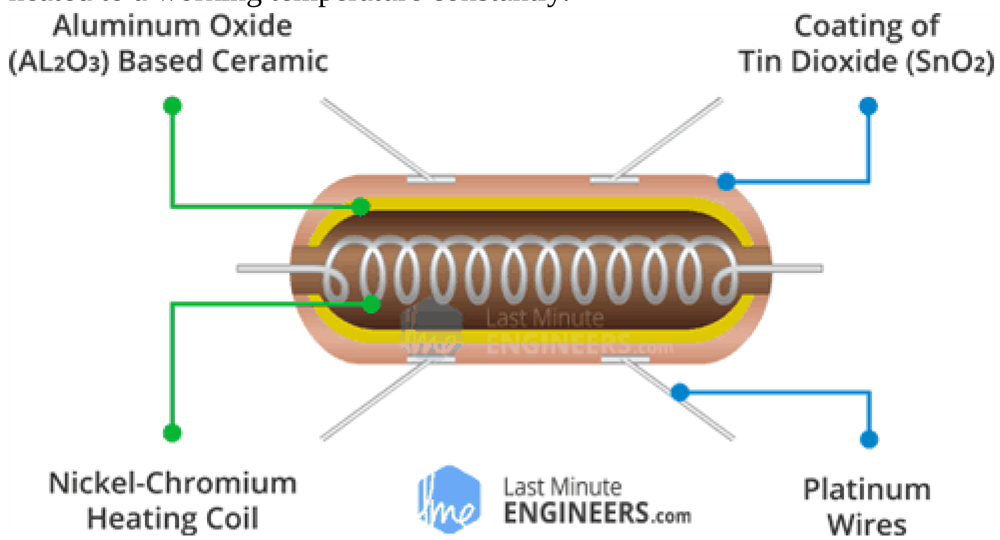


This is how the sensor looks like when outer mesh is removed. The star-shaped structure is formed by the sensing element and six connecting legs that extend beyond the Bakelite base. Out of six, two leads (**H**) are responsible for heating the sensing element and are connected through **Nickel-Chromium coil**, well known conductive alloy.

The remaining four leads (**A** & **B**) responsible for output signals are connected using **Platinum Wires**. These wires are connected to the body of the sensing element and convey small changes in the current that passes through the sensing element.



The tubular sensing element is made up of **Aluminum Oxide** (AL_2O_3) based ceramic and has a coating of **Tin Dioxide** (SnO_2). The Tin Dioxide is the most important material being sensitive towards combustible gases. However, the ceramic substrate merely increases heating efficiency and ensures the sensor area is heated to a working temperature constantly.



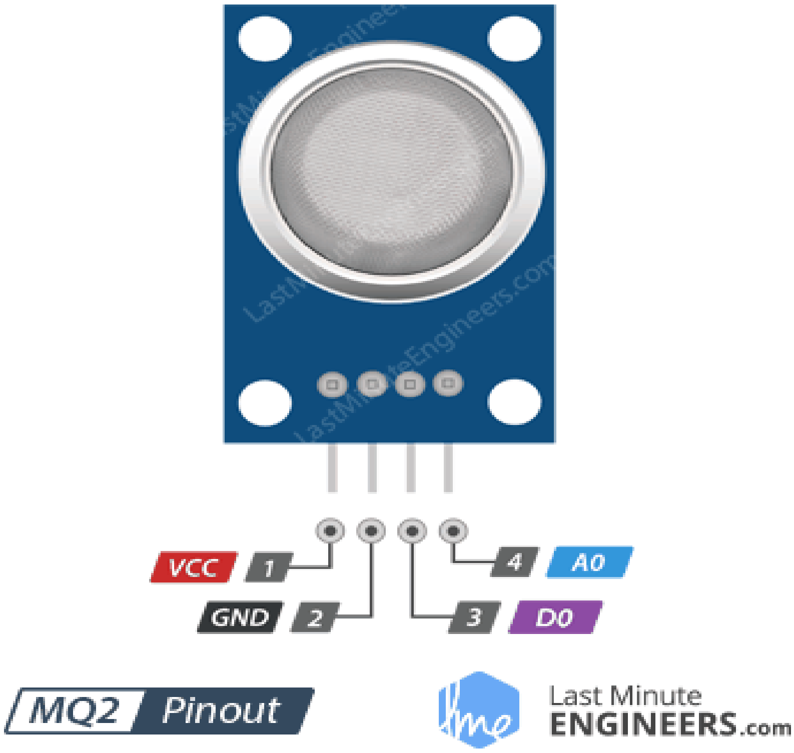
So, the Nickel-Chromium coil and Aluminum Oxide based ceramic forms a **Heating System**; while Platinum wires and coating of Tin Dioxide forms a **Sensing System**.

How does a gas sensor work?

When tin dioxide (semiconductor particles) is heated in air at high temperature, oxygen is adsorbed on the surface. In clean air, donor electrons in tin dioxide are attracted toward oxygen which is adsorbed on the surface of the sensing material. This prevents electric current flow.

In the presence of reducing gases, the surface density of adsorbed oxygen decreases as it reacts with the reducing gases. Electrons are then released into the tin dioxide, allowing current to flow freely through the sensor

MQ2 Gas Sensor Module Pinout



VCC supplies power for the module. You can connect it to 5V output from your Arduino.
GND is the Ground Pin and needs to be connected to GND pin on the Arduino.
D0 provides a digital representation of the presence of combustible gases.
A0 provides analog output voltage in proportional to the concentration of smoke/gas.

3.6 Electro-mechanical buzzer

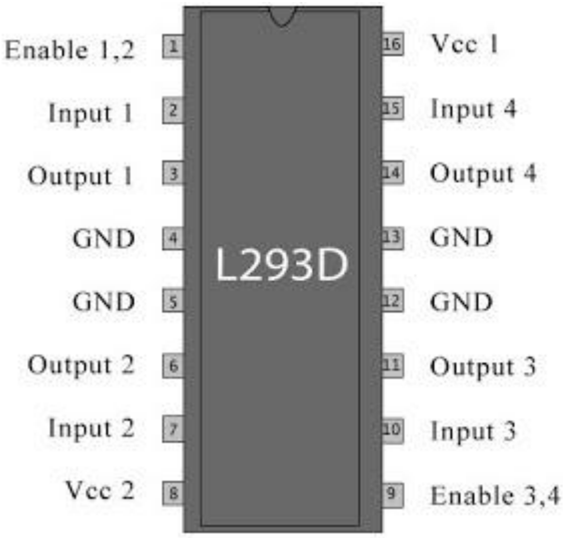
A buzzer is in the mechanical form of a small rectangular or cylindrical housing, with electrical connection for direct mounting on rigid printed circuit, or with electrical connection consisting of flexible electrical son. In the latter case, the buzzer has two small brackets. The loudness of such a component is about 85 dB / cm (note that it does not specify the sound level meter - as for HP, as a business perspective, it would seem probably too little power. As for sweets which are given the price per 100g and not for one kilogram).

It requires a DC voltage to operate, it should generally be between 3 V and 28 V, depending on the model. A buzzer designed to operate at 6 V generally works very well for any supply voltage between 4 V and 8 V, and a buzzer designed to operate at 12 V can work perfectly at a voltage between 6 V and 28 V (see characteristics given by the manufacturer for not making stupidity). There are also buzzers that work directly on the AC mains 230 V. This type of buzzer is convenient to use, because unlike piezoelectric buzzers simple (simple piezoelectric transducers without associated electronics), it has no work, except of course the eventual control stage which will enable it. He provides a simple DC voltage and presto, it sounds.

Simple Piezo-electric:- A buzzer (transducer) piezoelectric requires an AC voltage to operate, a few volts to several tens of volts (3V to 30V for example). It presents an optimal resonance frequency a few kHz (between 1 kHz and 5 kHz in general, eg, 2 kHz, 2.8 kHz or 3 kHz). It is this type of transducer that can be found on the back of the watch with an alarm function.

3.7 L293 MOTOR DRIVER

Generally, L293D motor driver can control two motor at one time or called is a dual H-Bridge motor driver. By using this IC, it can interface DC motor which can be controlled in both clockwise and counter clockwise direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Below shown the pin diagram of L293D motor driver

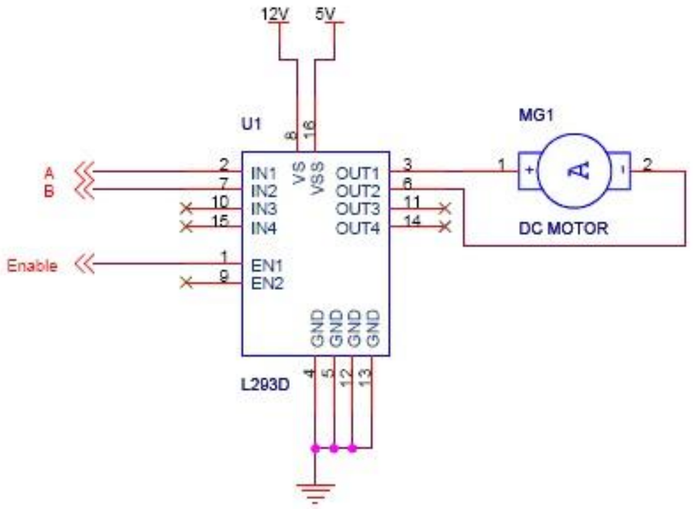


and the pin description of L293D motor driver

Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc 2
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc 1

Besides that, with this L293D driver motor it will control four DC motors at one time but with fix direction of motion. L293D has output current of 600mA and peak output current of 1.2A per channel.

Moreover for protection of circuit from back EMF output diode are included within the L293D. The output supply hich is external supply has a wide range from 4.5V to 36V which has made L293D a best choice for DC motor driver. A simple schematic for interfacing a DC gear motor using L293D driver motor is shown below:



and below the truth table for L293D driver motor

Truth Table

A	B	Description
0	0	Motor stops
0	1	Motor runs clockwise
1	0	Motor runs anti-clockwise
1	1	Motor stops

For truth table above, the Enable has to be set to 1 and motor power used is 12V but it is depends on motor power that used (range 4.5V to 36V). The rotation of the DC motor can be control by combinations of A and B in programming assembling and from the truth table it is clear to explain the rotations of the motor. Picture below shown the connection of DC gear motor to L293D driver motor.

SOFTWARE

4.1 Software Introduction:

Software introduction:

Arduino IDE is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules.

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

It is available for all operating systems i.e. (MAC, Windows, Linux and runs on the Java) Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code.

A range of Arduino modules available including Arduino Uno, Arduino Mega, ArduinoLeonnardo, Arduino Micro, and many more .Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code

Fig 4.1: ARDUINO

4.1.1: Definition of Arduino IDE

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.

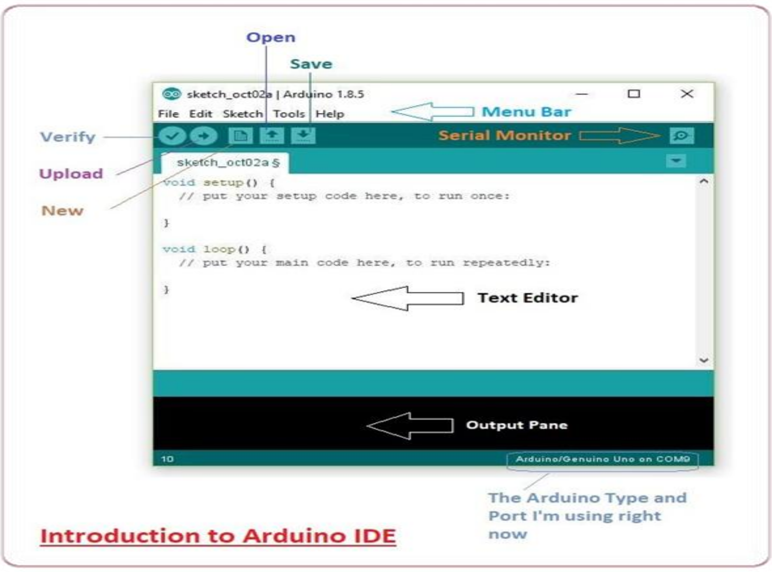
You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino software (IDE), based on processing.

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.

4.1.2. ABOUT IDE

The IDE environment is mainly distributed into three sections :

***Menu Bar *Text Editor *Output Pane**

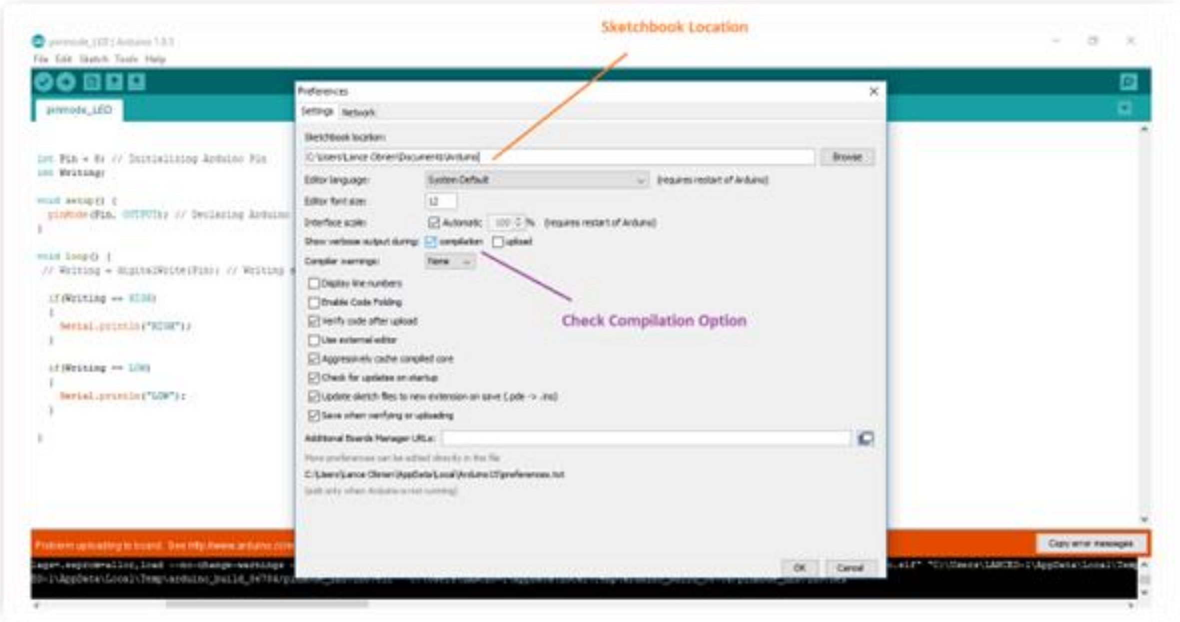


The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

File - You can open a new window for writing the code or open an existing one. The following table shows the number of further subdivisions the file option is categorized into

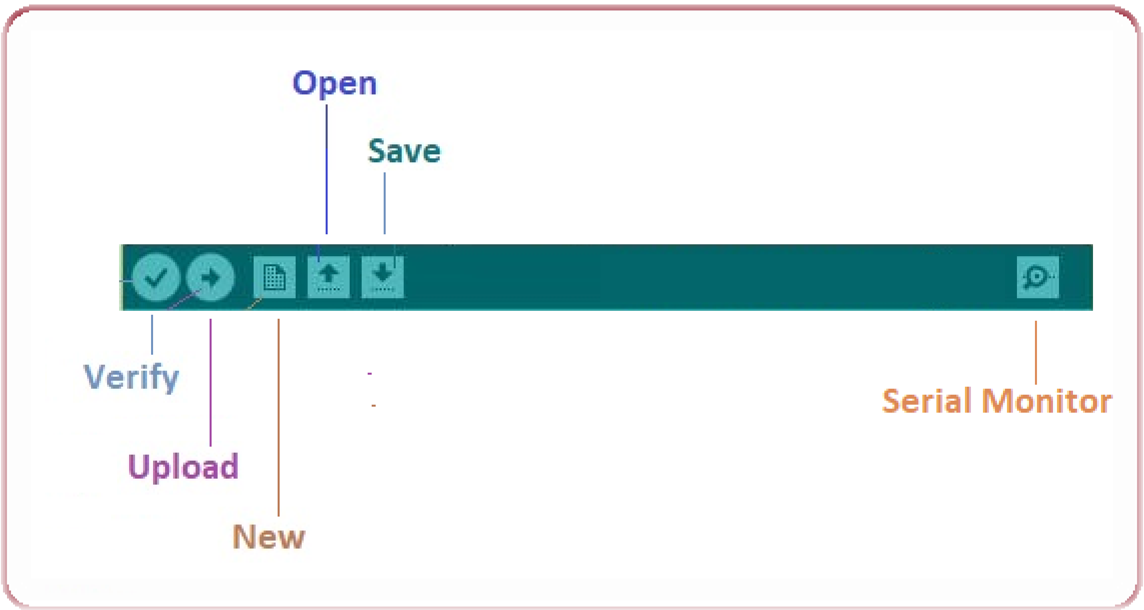
File	
New	This is used to open new text editor window to write your code
Open	Used for opening the existing written code
Open Recent	The option reserved for opening recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Default examples already stored in the IDE software
Close	Used for closing the main screen window of recent tab. If two tabs are open, it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program
Save as	It will allow you to save the recent program in your desired folder
Page setup	Page setup is used for modifying the page with portrait and landscape options. Some default page options are already given from which you can select the page you intend to work on
Print	It is used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editor page
Quit	It will quit the whole software all at once

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.



And at the end of the compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

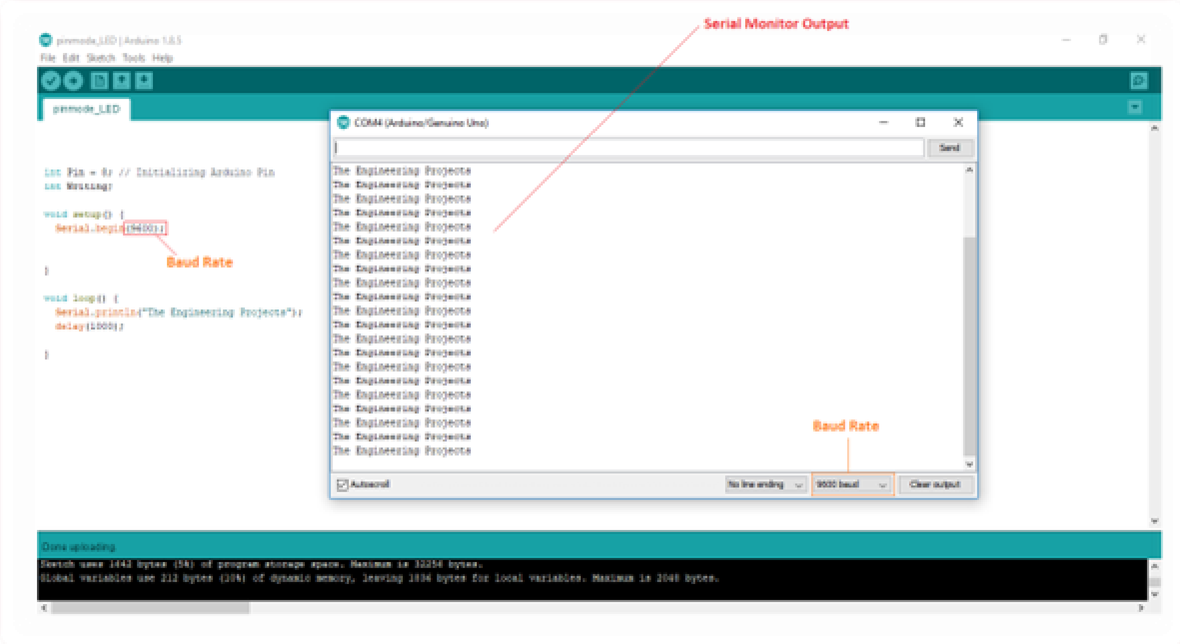
Edit - Used for copying and pasting the code with further modification for font
Sketch - For compiling and programming
Tools - Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
Help - In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.
The **Six Buttons** appearing under the Menu tab are connected with the running program as follows.



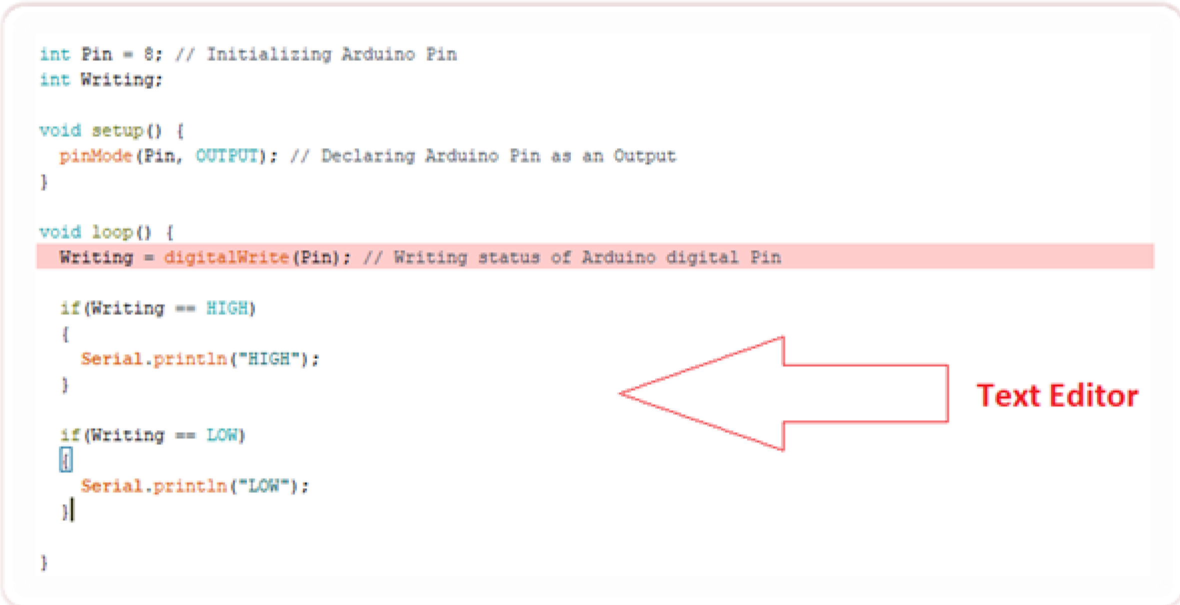
The checkmark appearing in the circular button is used to verify the code. Click this once you have written your code.
The arrow key will upload and transfer the required code to the Arduino board.
The dotted paper is used for creating a new file.
The upward arrow is reserved for opening an existing Arduino project.
The downward arrow is used to save the current running code.

The button appearing on the top right corner is a **Serial Monitor** - A separate pop-up window that acts as an independent terminal and plays a vital role in sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly.
The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.

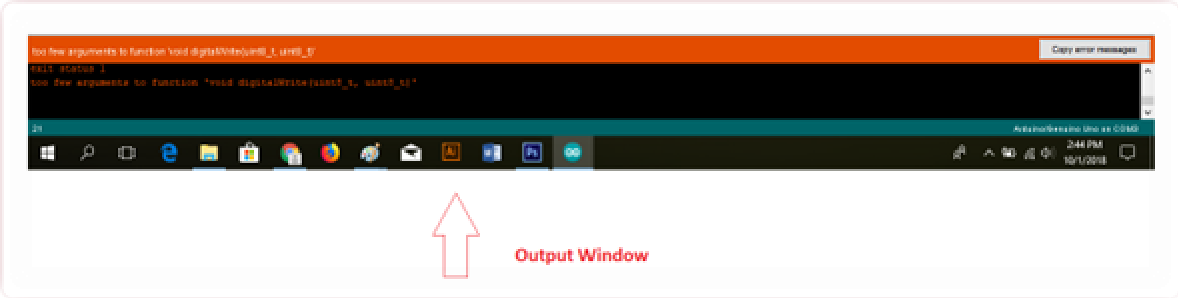
You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.



The main screen below the Menu bard is known as a simple text editor used for writing the required code.



The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors that occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.



More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

4.2 PROGRAM:

4.2.1SERVER:

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);


// Set your access point network credentials
const char* password = "123456789";


bool Flag1 = 0, Flag2 = 0;
String USSTAT = "", AlcoholSTAT = "";
floatBat_Vg = 0;

#define TRIG    15
#define ECHO    2
#define Alcohol 35
#define buzz    23
#define VBat    34

void LCD_DISP(String Row1, String Row2, int D)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(Row1);
lcd.setCursor(0, 1);
lcd.print(Row2);
```

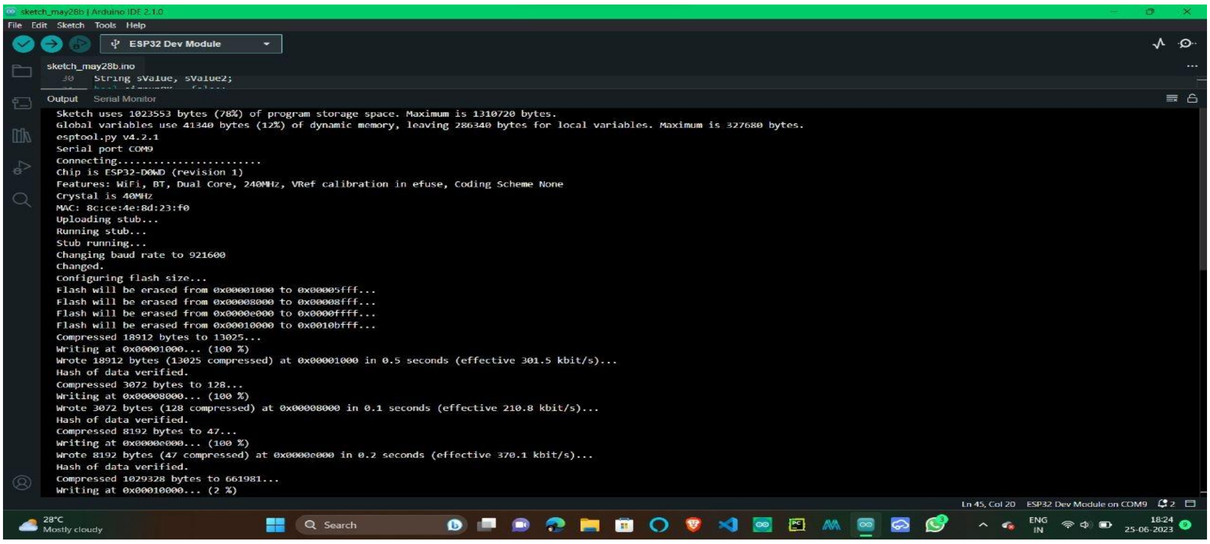
```
delay(D);
}


void Distance()
{
 USSTAT = "";
digitalWrite(TRIG, LOW);
delayMicroseconds(2);
digitalWrite(TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG, LOW);
int duration = pulseIn(ECHO, HIGH);
int distance = (duration / 58.138) ;
Serial.print("Distance : ");
Serial.println(distance);
if (distance < 30)
  {
   USSTAT = "Obstacle Detect";
digitalWrite(buzz, HIGH);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Obstacle Detect");
delay(1000);
digitalWrite(buzz, LOW);
  }
else
  {
   USSTAT = "Normal";
lcd.clear();
  }

}

voidReadAlcoholSensor(void)
{
AlcoholSTAT = "";
Serial.print("Alcohol:");
Serial.println(digitalRead(Alcohol));
if (digitalRead(Alcohol) == 1)
  {
AlcoholSTAT = "Normal";
lcd.clear();
```

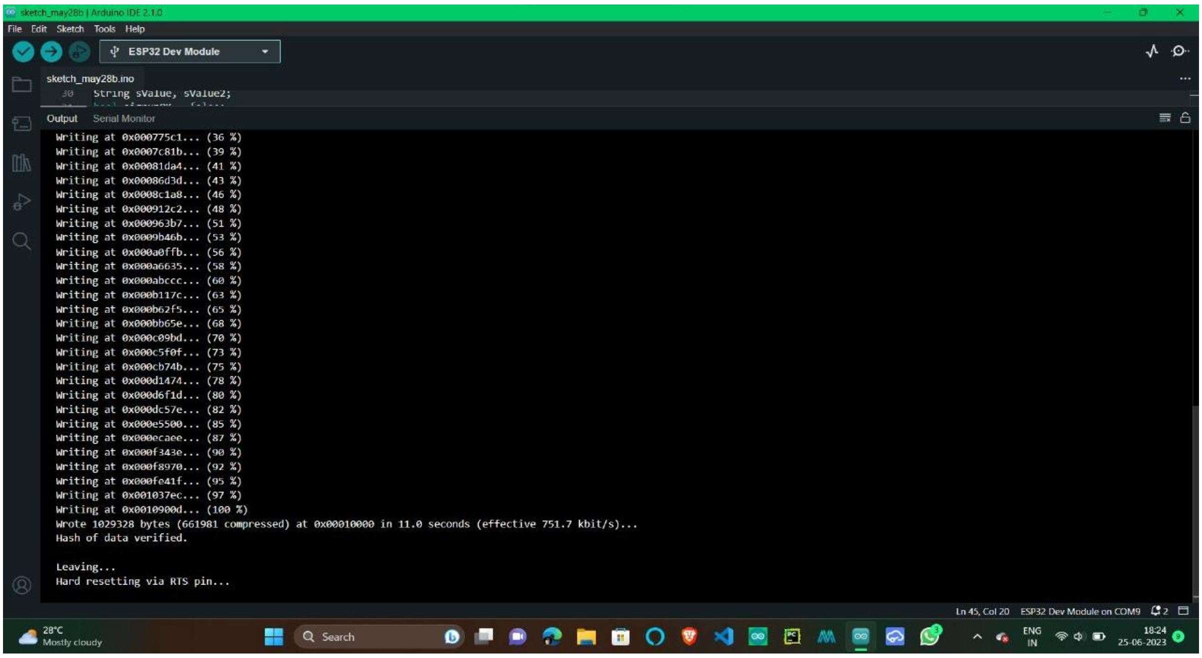
```
  }
else
  {
AlcoholSTAT = "Alcohol Detect";
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Alcohol Detected");
digitalWrite(buzz, HIGH);
delay(1000);
digitalWrite(buzz, LOW);
  }
}

voidBatteryVg()
{
intval = analogRead(VBat);
Bat_Vg = (val * 10.0) / 4096.0;
}

String US()
}

String AlcoholCheck()
{
returnAlcoholSTAT;
}

String VBAT()
{
return String(Bat_Vg);
}


void LCDDISP(String R1, String R2, int D)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(R1);
lcd.setCursor(0, 1);
lcd.print(R2);
delay(D);
}
```

```
void setup() {
  // Serial port for debugging purposes
Serial.begin(9600);
Serial2.begin(9600);
Serial.println();
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("    WEL-COME    ");
delay(1000);
pinMode(TRIG, OUTPUT);
pinMode(buzz, OUTPUT);
pinMode(ECHO, INPUT);
pinMode(Alcohol, INPUT);
digitalWrite(buzz, LOW);
  // Setting the ESP as an access point
Serial.print("Setting AP (Access Point)…");
  // Remove the password parameter, if you want the AP (Access Point) to be open
Serial.print("AP IP address: ");
Serial.println(IP);

server.on("/US", HTTP_GET, [](AsyncWebServerRequest * request) {
request->send_P(200, "text/plain", US().c_str());
  });
server.on("/Alcohol", HTTP_GET, [](AsyncWebServerRequest * request) {

  });

server.on("/Bat", HTTP_GET, [](AsyncWebServerRequest * request) {
request->send_P(200, "text/plain", VBAT().c_str());
  });

bool status;

  // default settings
  // (you can also pass in a Wire library object like &Wire2)

  // Start server
server.begin();
}

void loop()
{
```

```
ReadAlcoholSensor();
delay(500);
Distance();
delay(500);
}
```

OUTPUT:





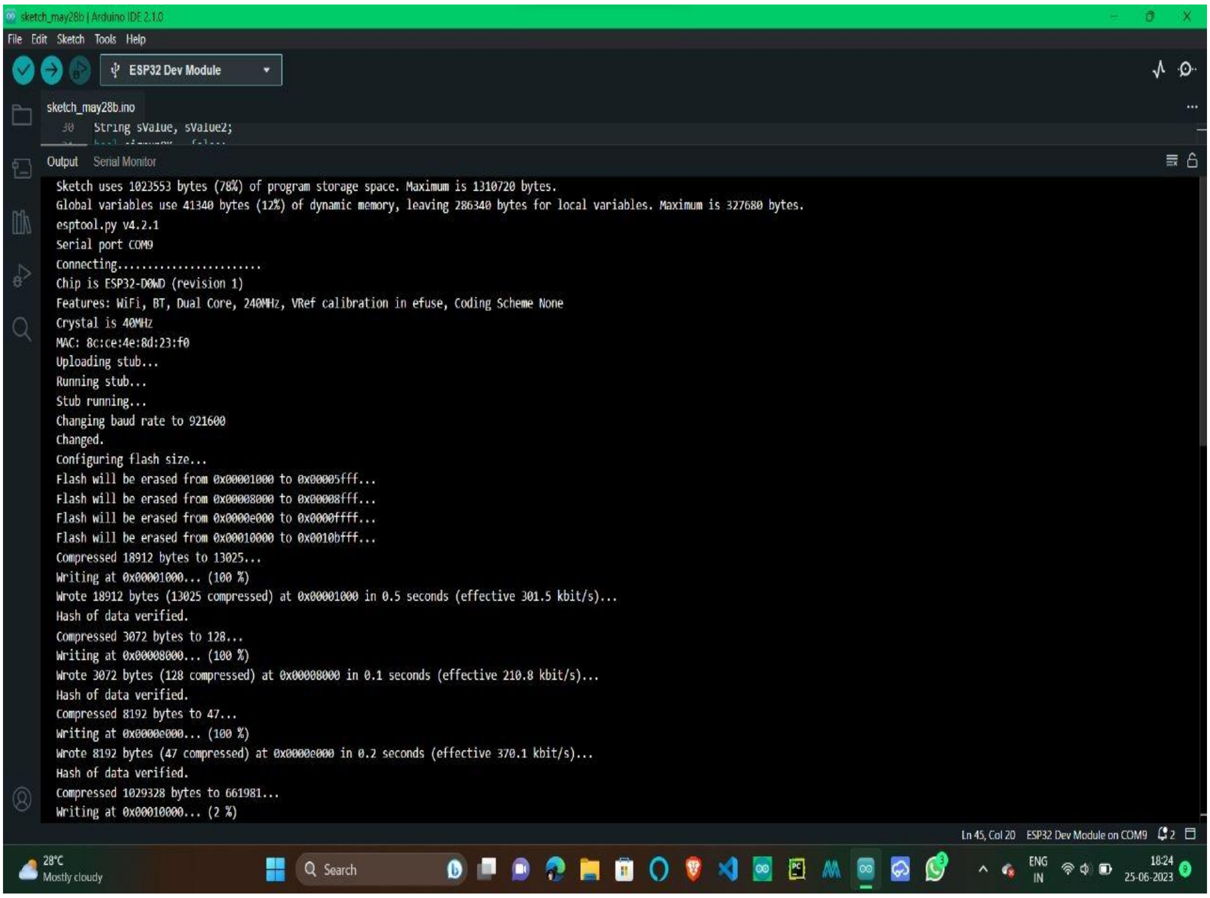
4.2.2 CLIENT:

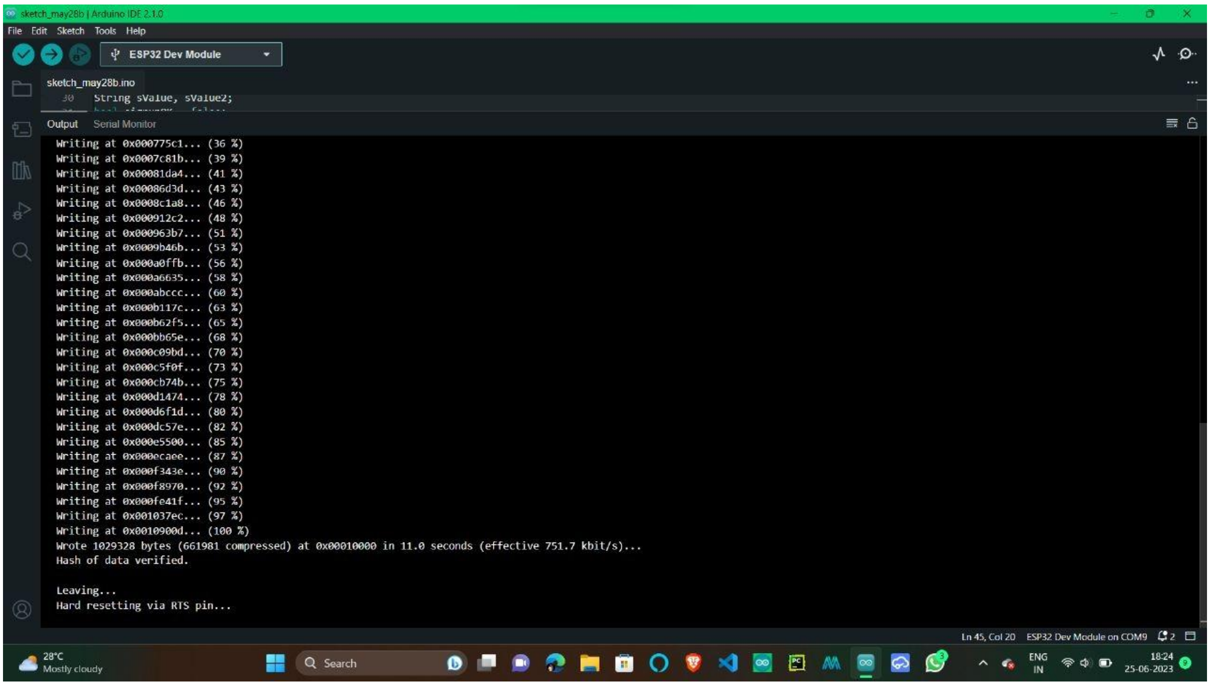
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);


const char* ssid = "ESP32-Access-Point";
const char* password = "123456789";

//Your IP address or domain name with URL path
const char serverNameUS = "http://192.168.4.1/US";
const char serverNameDoor = "http://192.168.4.1/Alcohol";


String USSTAT;
String AlcoholSTAT;

unsigned long previousMillis = 0;
const long interval = 1000;

int M1 = 27, M2 = 26;


voidLCD_Disp(String Row1, String Row2, int Delay)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(Row1);
lcd.setCursor(0, 1);
lcd.print(Row2);
delay(1000);
}

void setup()
{
Serial.begin(9600);
pinMode(M1, OUTPUT);
digitalWrite(M1, HIGH);
pinMode(M2, OUTPUT);
digitalWrite(M2, HIGH);

lcd.init();
```

```
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("   WEL - COME   ");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connecting....!");
WiFi.begin(ssid, password);
Serial.println("Connecting");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
  }
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
lcd.clear();
lcd.setCursor(0, 1);
lcd.print("Connected  ");
delay(1000);
digitalWrite(M1, LOW);
digitalWrite(M2, LOW);

}

void loop() {
unsigned long currentMillis = millis();

if (currentMillis - previousMillis>= interval)
  {
   // Check WiFi connection status
if (WiFi.status() == WL_CONNECTED )
   {
Serial.println(AlcoholSTAT);
if ((-1 != USSTAT.indexOf("Obstacle Detect")) || (-1 != AlcoholSTAT.indexOf("Alcohol Detect")))
     {
digitalWrite(M1, HIGH);
digitalWrite(M2, HIGH);
     }
else
     {
digitalWrite(M1, LOW);
digitalWrite(M2, LOW);
```

```
      }
LCD_Disp("OBS:" + USSTAT, "DS:" + AlcoholSTAT, 1000);
previousMillis = currentMillis;
lcd.clear();
    }
else
    {
Serial.println("WiFi Disconnected");
LCD_Disp("   Server Not   ", "   Available    ", 500);
    }
  }
}

String httpGETRequest(const char* serverName) {
WiFiClient client;
HTTPClient http;

  // Your Domain name with URL path or IP address with path
http.begin(client, serverName);

  // Send HTTP POST request
inthttpResponseCode = http.GET();

  String payload = "--";

if (httpResponseCode> 0) {
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
payload = http.getString();
  }
else {
Serial.print("Error code: ");
Serial.println(httpResponseCode);
  }
  // Free resources
http.end();

return payload;
}
```

OUT PUT:


sketch_may28b | Arduino IDE 2.1.0
File Edit Sketch Tools Help
ESP32 Dev Module
sketch_may28b.ino
String sValue, sValue2;
Output Serial Monitor
Sketch uses 1023553 bytes (78%) of program storage space. Maximum is 1310720 bytes.
Global variables use 41340 bytes (12%) of dynamic memory, leaving 286340 bytes for local variables. Maximum is 327680 bytes.
esptool.py v4.2.1
Serial port COM9
Connecting.......................
Chip is ESP32-D0WD (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 8c:ce:4e:8d:23:f0
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Flash will be erased from 0x00001000 to 0x00005fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Flash will be erased from 0x0000e000 to 0x0000ffff...
Flash will be erased from 0x00010000 to 0x0010bfff...
Compressed 18912 bytes to 13025...
Writing at 0x00001000... (100 %)
Wrote 18912 bytes (13025 compressed) at 0x00001000 in 0.5 seconds (effective 301.5 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.1 seconds (effective 210.8 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 47...
Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.2 seconds (effective 370.1 kbit/s)...
Hash of data verified.
Compressed 1029328 bytes to 661981...
Writing at 0x00010000... (2 %)
Ln 45, Col 20 ESP32 Dev Module on COM9
28°C Mostly cloudy
Search
ENG IN
18:24 25-06-2023


sketch_may28b | Arduino IDE 2.1.0
File Edit Sketch Tools Help
ESP32 Dev Module
sketch_may28b.ino
String sValue, sValue2;
Output Serial Monitor
Writing at 0x000775c1... (36 %)
Writing at 0x0007c81b... (39 %)
Writing at 0x00081da4... (41 %)
Writing at 0x00086d3d... (43 %)
Writing at 0x0008c1a8... (46 %)
Writing at 0x000912c2... (48 %)
Writing at 0x000963b7... (51 %)
Writing at 0x0009b46b... (53 %)
Writing at 0x000a0ffb... (56 %)
Writing at 0x000a6635... (58 %)
Writing at 0x000abccc... (60 %)
Writing at 0x000b117c... (63 %)
Writing at 0x000b62f5... (65 %)
Writing at 0x000bb65e... (68 %)
Writing at 0x000c09bd... (70 %)
Writing at 0x000c5f0f... (73 %)
Writing at 0x000cb74b... (75 %)
Writing at 0x000d1474... (78 %)
Writing at 0x000d6f1d... (80 %)
Writing at 0x000dc57e... (82 %)
Writing at 0x000e5500... (85 %)
Writing at 0x000ecaee... (87 %)
Writing at 0x000f343e... (90 %)
Writing at 0x000f8970... (92 %)
Writing at 0x000fe41f... (95 %)
Writing at 0x001037ec... (97 %)
Writing at 0x0010900d... (100 %)
Wrote 1029328 bytes (661981 compressed) at 0x00010000 in 11.0 seconds (effective 751.7 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Ln 45, Col 20 ESP32 Dev Module on COM9
28°C Mostly cloudy
Search
ENG IN
18:24 25-06-2023

RESULTS

5.1 Hardware Implemenation

The intelligent sensors on machine for smart interaction. Internet of things (IoT) gives a wide scope in many application domains where number of smart gadgets per person is increasing exponentially with time. The automobile sector is also one of the application Domain where vehicle can be madeintelligent by using IoT. This paper presents the interface of ESP 32 development board, sensor shield and smart phone. The proposed schematic is for car security which gives the access of car through WI FI , car is becoming a form idable sensor platform, absorbing information from the environment, from other cars (and from the driver) and feeding it to other cars and infrastructure to assist in safe navigation, pollution control and traffic management.

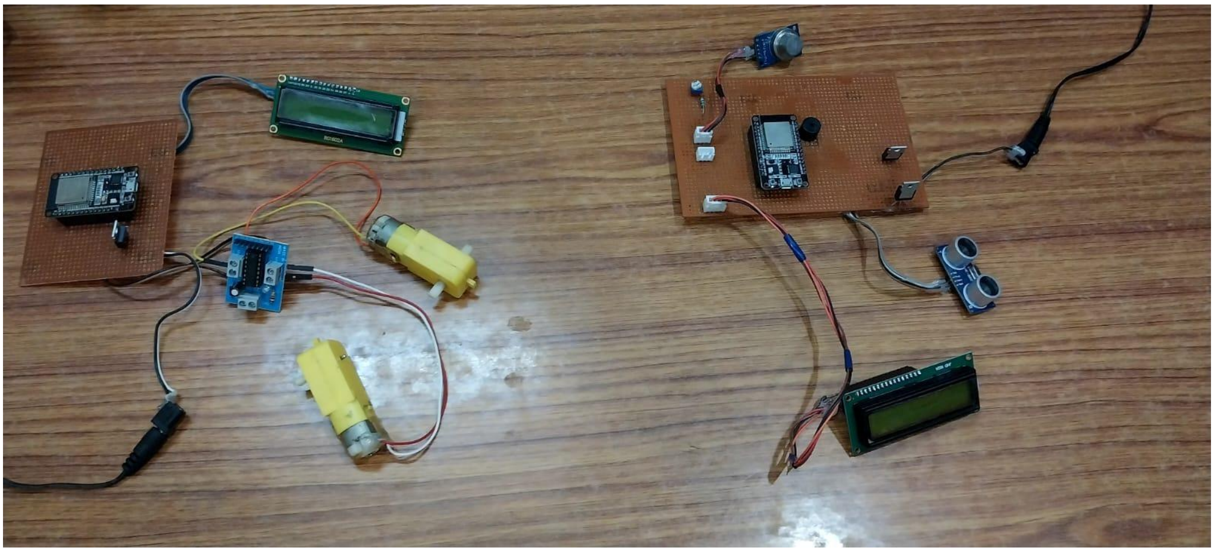


Fig5.1.1: Initial Step of Hardware Implementation

The Vehicle Grid essentially becomes an Internet of Things (IOT), which we call Internet of Vehicles (IOV), capable to make its own decisions about driving customers to their destinations. Like other important IOT examples (e.g., smart buildings), the Internet of Vehicles will not merely upload data to the Internet using V2I. It will also use V2V communications between peers to complement on board sensor inputs and provide safe and efficient navigation. In this paper, we first describe several vehicular applications that leverage V2V and V2I. Communications with infrastructure and with other vehicles, however, can create privacy and security violations. In the second part of the paper we address these issues and more specifically focus on the need to guarantee location privacy to mobile users. We argue on the importance of creating public, open "smart city" data repositories for the research community and propose privacy preserving techniques for the anonymous uploading of urban Sensor data from Vehicles.

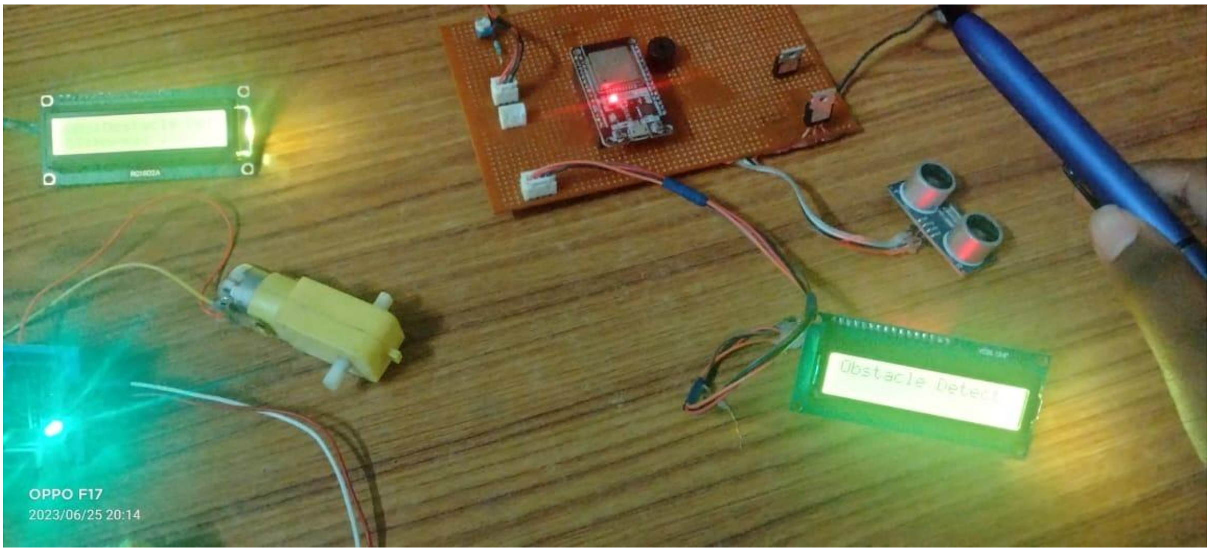


Fig5.1.2:Obstacle Detected

The Fig5.1.2 shows the Obstacle Detected on the Car to Car Communication System
The Ultrasonic sensor or HC-SRO4 is used to measure the distance of the object using SONAR.
It emits the Ultrasound at a frequency of 40KHZ or 40000 Hz. The frequency travels through the air and strikes the object on its path. The rays bounce back from the object and reach back to the module.
The four terminals of HC-SRO4 are VCC, TRIG, ECHO, and GND. The voltage supply or VCC is +5V. We can connect the ECHO and TRIG terminal to any of the digital I/O pin on the specific Arduino board.
The Ultrasonic sensors work best for medium ranges.
The resolution is 0.3cm.
The medium ranges of the sensor are 10cm to 3m. It works best at this duration.
The maximum range the sensor may detect is 4.5m.

Fig5.1.3:The Obstacle Detected On the Car to Car Communicaion

Fig5.1.4:Alcohol Detected

Fig5.1.4 Shows the Alcohol detection of Car to Car Communication SystemWe have provided a very effective solution to develop an intelligent system for vehicles for alcohol detection whose core is Arduino. Since sensor has fine sensitivity range around 10meters, it can suit to any vehicle and easily be hidden from the subjects. The whole system has also an advantage of small volume and more reliability. As the growing public perception is that is that vehicle safety is more important, advances in public safety is gaining acceptance that in than in the past. Future scope of this system is to control the accidents causes due to alcohol consumption. This system improves the safety of the human being. And hence providing the effective development in the automobile industry regrading to reduce the accidents due to alcohol.

5.2 CONLUSION

Our system efficiently checks the accidents occurs or not and drunken driving. By implementing this system in vehicle, a safe journey is possible which would decrease the injuries during accidents and also reduce the accident rate due to drunken driving. This system has also accident prevention technology called as ultrasonic sensor, which would reduce the accident of the vehicle in crowd areas. We can describe that this is a safety features for vehicle because if alcohol detected or accident happens then it will stop automatically. In case of alcohol detected or any accident happens it would communicate with the other vehicle.

REFERENCES

1. ^a b"Dedicated Short Range Communications (DSRC) Home". Home". leearmstrong.com. Archived from the original on 2012-11-19. Retrieved 2008-02-29.
2. ^Hartenstein, H.; Laberteaux, K.P. (2008). "A tutorial survey on vehicular ad hoc networks". IEEE Communications Magazine. 46 (6): 164–171. doi:10.1109/MCOM.2008.4539481. ISSN 0163-6804. S2CID 3160950.
3. ^Tsugawa, S. (2003). "Inter-vehicle communications and their applications to intelligent vehicles: an overview". Intelligent Vehicle Symposium, 2002. IEEE. Versailles, France: IEEE. 2: 564–569. doi:10.1109/IVS.2002.1188011. ISBN 978-0-7803-7346-4. S2CID 62061334.

4. ^a b"California Partners for Advanced Transportation". University of California, Berkeley. Retrieved 29 April 2022.
5. ^Williams, M. (1988). "PROMETHEUS-The European research programme for optimising the road transport system in Europe". IEE Colloquium on Driver Information: 1/1–1/9.
6. ^"Background of the development of ASV (Advanced Safety Vehicle)". www.mlit.go.jp. Retrieved 2021-08-13.
7. "Promote Chauffeur II - TRIMIS - European Commission". TRIMIS. 2009-10-12. Retrieved 2021-08-13.
8. ^Franz, W. J.; Eberhardt, R.; Luckenbach, T. (2001). "FLEETNET - INTERNET ON THE ROAD".
9. ^Reichardt, D.; Miglietta, M.; Moretti, L.; Morsink, P.; Schulz, W. (2003). "CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication". Intelligent Vehicle Symposium, 2002. IEEE. Versailles, France: IEEE. 2: 545–550. doi:10.1109/IVS.2002.1188007. ISBN 978-0-7803-7346-4. S2CID 60703429.
10. ^Sichitiu, Mihail; Kihl, Maria (2008). "Inter-vehicle communication systems: a survey". IEEE Communications Surveys & Tutorials. 10 (2): 88–105. doi:10.1109/COMST.2008.4564481. ISSN 1553-877X. S2CID 18052278.
11. ^"ETC(Electronic Toll Collection System) - Global standard ETC started". www.mlit.go.jp. Retrieved 2021-08-13.
12. ^"Safespot". www.safespot-eu.org. Retrieved 2021-08-13.
13. ^"PReVENT :: Home". www.prevent-ip.org. Retrieved 2021-08-13.
14. ^"Communications for eSafety - TRIMIS - European Commission". TRIMIS. 2013-06-24. Retrieved 2021-08-13.
15. ^"´Network-on-Wheels´ at the University of Mannheim". pi4.informatik.uni-mannheim.de. Retrieved 2021-08-13.
16. ^"Public Roads - The Intelligent Vehicle Initiative: Advancing 'Human-Centered' Smart Vehicles , Sept/Oct 1997 -". www.fhwa.dot.gov. Retrieved 2021-08-13.
17. ^"ITS Standards Program | Fact Sheets | ITS Standards Fact Sheets". www.standards.its.dot.gov. Retrieved 2021-08-13.
18. ^"Work Programme - Work Item Detailed Report". portal.etsi.org. Retrieved 2021-08-14.
19. ^"Release 16". www.3gpp.org. Retrieved 2021-08-14.
20. ^Alalewi, Ahmad; Dayoub, Iyad; Cherkaoui, Soumaya (2021). "On 5G-V2X Use Cases and Enabling Technologies: A Comprehensive Survey". IEEE Access. 9: 107710–107737. doi:10.1109/ACCESS.2021.3100472. hdl:20.500.12210/55004. ISSN 2169-3536. S2CID 236939427. Text was copied from this source, which is available under a Creative Commons Attribution 4.0 International License.
21. ^"Welcome to TTA - Telecommunications Technology Association of Korea". www.tta.or.kr. Retrieved 2021-08-14.
22. ^"ITSasia-pacific". itsasia-pacific.com. Retrieved 2021-08-14.
23. ^"V2X". www.3gpp.org. Retrieved 2021-08-14.
24. ^"IEEE P802.11 - TASK GROUP BD (NGV) - GROUP INFORMATION UPDATE". www.ieee802.org. Retrieved 2021-08-14.
25. ^M. Peden; Richard Scurfield; D. Sleet; D. Mohan; et al. "World report on road traffic injury prevention" (PDF). World Health Organization. Retrieved 2008-02-29.

26. ^"Crashes Vs. Congestion -- What's the Cost to Society?" (PDF). American Automobile Association. Archived from the original (PDF) on 2012-02-01. Retrieved 2011-11-30.
27. ^"Vehicle Infrastructure Integration (VII)". its.dot.gov. Retrieved 2008-02-29.
28. ^Boehmlaender, Dennis; Hasirlioglu, Sinan; Yano, Vitor; Lauerer, Christian; Brandmeier, Thomas; Zimmer, Alessandro (2015). "Advantages in Crash Severity Prediction Using Vehicle to Vehicle Communication". 2015 IEEE International Conference on Dependable Systems and Networks Workshops. IEEE: 112–117. doi:10.1109/dsn-w.2015.23. ISBN 978-1-4673-8044-7. S2CID 13183260.
29. ^"Frequency of Target Crashes for IntelliDrive Safety Systems" (PDF). NHTSA. October 2010. Archived from the original (PDF) on 5 April 2021. Retrieved 27 April 2022.
30. ^"The world health report 2002 - Reducing Risks, Promoting Healthy Life". World Health Organization. Archived from the original on December 2, 2002. Retrieved 2008-02-29.
31. ^Bigelow, Pete. "Feds Want V2V Communication in New Cars Starting in 2021". Car and Driver. Retrieved 2017-01-29.
32. ^Harding, J (2014). "Vehicle-to-vehicle communications: Readiness of V2V technology for application" (PDF). nhtsa.gov. Archived from the original (PDF) on 2018-11-15. Retrieved 2016-04-28.
33. ^"Cars are ready to talk to one another – unless we use their airwaves for Wi-Fi". Los Angeles Times. Retrieved 2021-11-28.
34. ^Austroads. "Austroads' Submission to the '2014 Review of the Motor Vehicle Standards Act 1989" (PDF). Department of Infrastructure and Development (Australia). Retrieved 2017-01-29.
35. ^"Radiocommunications (Intelligent Transport Systems) Class Licence 2017". Federal Register of Legislation. Retrieved 2018-10-09.
36. ^"No lights, no signs, no accidents – future intersections for driverless cars (video)". Reuters.com. 22 March 2012. Retrieved 28 April 2012.
37. ^Andert, Edward; Khayatian, Mohammad; Shrivastava, Aviral (18 June 2017). "Crossroads". Crossroads: Time-Sensitive Autonomous Intersection Management Technique. Institute of Electrical and Electronics Engineers Inc. pp. 1–6. doi:10.1145/3061639.3062221. ISBN 9781450349277. S2CID 6173238.
38. ^Khayatian, Mohammad; Mehrabian, Mohammadreza; Shrivastava, Aviral (2018). "RIM: Robust Intersection Management for Connected Autonomous Vehicles". 2018 IEEE Real-Time Systems Symposium (RTSS). Institute of Electrical and Electronics Engineers Inc. pp. 35–44. doi:10.1109/RTSS.2018.00014. ISBN 978-1-5386-7908-1. S2CID 52093557.
39. ^"5GAA live demos show C-V2X as a market reality". 5GAA. 14 November 2019. Retrieved 29 April 2022.
40. ^"Towards 5G Mobility: The role of efficient discrete semiconductors". Wevolver. 29 January 2020. Retrieved 29 April 2022