RAILWAY TRACK CRACK AND FAULT DETECTION SYSTEM USING GPS GSM

A Report submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Technology

In

Computer Science and Engineering (Internet of Things)

By

Paidi Jagadeesh 2111CS050023

Ailuri Raju Kumar 2111CS050036

Mudedla Vinay 2111CS050050

Under the esteemed guidance of

Mrs. Affrose

M.Tech(Phd)

Assistant Professor

Department of Computer Science and Engineering (Internet of Things)

School of Engineering

MALLA REDDY UNIVERSITY

Maisammaguda, Dulapally, Hyderabad, Telangana 500100

2025

RAILWAY TRACK CRACK AND FAULT DETECTION SYSTEM USING GPS GSM

A Report Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology
In

Computer Science and Engineering (Internet of Things)

By

Paidi Jagadeesh 2111CS050023
Ailuri Raju Kumar 2111CS050036
Mudedla Vinay 2111CS050050

Under the esteemed guidance of

Mrs. Affrose
M.Tech(Phd)
Assistant Professor

Department of Computer Science and Engineering (Internet of Things)

School of Engineering
MALLA REDDY UNIVERSITY
Maisammaguda, Dulapally, Hyderabad, Telangana 500100

2025

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

Department of Computer Science and Engineering (Cyber Security)

CERTIFICATE

This is to certify that the project report entitled **"Railway Track Crack and Fault Detection System Using GPS GSM"**, submitted by **Paidi Jagadeesh(2111CS050023), Ailuri Raju Kumar(2111CS050036), Mudedla Vinay(2111CS050050),** towards the partial fulfillment for the award of Bachelor's Degree in Computer Science and Engineering - Cybersecurity from the Department of Cybersecurity, Malla Reddy University, Hyderabad, is a record of bonafide work done by him/ her. The results embodied in the work are not submitted to any other University or Institute for award of any degree or diploma.

Internal Guide	***Head of the Department***
Mrs. Affrose	**Dr. G.Anand Kumar**
Assistant Professor	**CSE(Cyber Security & IoT)**

External Examiner

DECLARATION

We hereby declare that the project report entitled **"Railway Track Crack and Fault Detection System Using GPS GSM"** has been carried out by us and this work has been submitted to the **Department of Computer Science and Engineering (Internet of Things), Malla Reddy University,** Hyderabad in partial fulfillment of the requirements for the award of degree of Bachelor of Technology. We further declare that this project work has not been submitted in full or part for the award of any other degree in any other educational institutions.

Place:

Date:

Paidi Jagadeesh	2111CS050023
Ailuri Raju Kumar	2111CS050036
Mudedla Vinay	2111CS050050

ACKNOWLEDGEMENT

We extend our sincere gratitude to all those who have contributed to the completion of this project report. Firstly, We would like to extend our gratitude to **Dr. V. S. K Reddy, Vice-Chancellor**, for his visionary leadership and unwavering commitment to academic excellence.

We would also like to express my deepest appreciation to our project guide **Mrs. Affrose, Assistant Professor**, whose invaluable guidance, insightful feedback, and unwavering support have been instrumental throughout the course of this project for successful outcomes.

We extend our gratitude to **Dr. G. Latha, PRC-convenor,** for giving valuable inputs and timely guidelines to improve the quality of our project through a critical review process. We thank our project coordinator **Dr. B. Nageshwar Rao,** for his timely support.

We are also grateful to **Dr. G. Anand Kumar, Head of the Department of Internet of Things,** for providing us with the necessary resources and facilities to carry out this project

We are deeply indebted to all of them for their support, encouragement, and guidance, without which this project would not have been possible.

ABSTRACT

Railways are the most preferable transport system because of their reliability, passenger safety, and ease to travel. If any misalignment or crack occurs, it creates a loss of lives. If these cracks and misalignments are not taken care of early, they may result in derailments and eventually result in a significant loss of life. To overcome this issue, a railway track crack detection system is proposed. The components like Global Position System (GPS), BUZZER, IR SENSOR, ULTRASONIC SENSOR, ARDUINO, Global System for Mobile (GSM), BATTERY, and D.C. MOTOR will be used. The cracks are detected through sensor rays. GPS can send the position of the explicit area and GSM sends the message to the station premises. The sensor detects the crack and sends a message to the maintenance room, and all connections are made with ARDUINO. Here in the place of the train, a wheeled robot will be used with the help of the dc motor. This concept will give high accuracy and no problem will be occurred during detection and saves so many lives. It can be used for railway stations and used in metro trains, gaming systems, and any crack detection system.

INDEX

Contents	Page No.

1. INTRODUCTION

Depending on the fast developments in railway systems, high-speed trains are used for speed transportation, and rai1 transportation is increased day by day. The most of the people uses railway for transportation because it is essentia1 for transferring the goods and passengers from one place to another p1ace and low cost. And also, the railway system is providing faci1ity such as high speed, with economica1, environment friend1y, safety, and better characteristics of rai1ay systems. These can be done by time to time maintenance and control measurements. But depending on different factors, deformations and derailment may occur on the superstructure of railways, because of improper maintenance and the currently irregular and manual track line monitoring mistake from workers. Such deformation is determining on time and taking precautions is very important for the safety of railway systems. Therefore, solution for this problem is introducing in this project. To provide the protection to the railway accident because of cracks occurs in the railway track. This system is used in-between two stations which will detect the cracks present on the track using IR sensors which transmit sine waves for an ideal track. If a crack is detected then this sensor will send a signal to the Arduino Uno board which will activate the GPS receiver. The GPS receiver will send the exact location which will then be messaged to the main authorities. Once the sensor sends a signal to the controller, the controller will activate the webcam. The webcam will provide the live video of the track. The live video and the data from the GPS will be updated in the designed application of the wireless camera. Using this smart technology will be a part of the brave new digitalized world which will be able to prevent the loss of precious life and property.

1.1 Introduction of Embedded System:

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner.

This is in direct contrast to the personal computer in the family room. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function rather; it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do wish it. One customer may use it for a network file server another may use it exclusively for playing games, and a third may use it to write the next great American novel.

Frequently, an embedded system is a component within some larger system. For example, modern cars and trucks contain many embedded systems. One embedded system controls the anti-lock brakes, other monitors and controls the vehicle's emissions, and a third displays information on the dashboard. In some cases, these embedded systems are connected by some sort of a communication network, but that is certainly not a requirement.

At the possible risk of confusing you, it is important to point out that a general-purpose computer is itself made up of numerous embedded systems. For example, my computer consists of a keyboard, mouse, video card, modem, hard drive, floppy drive, and sound card-each of Which is an embedded system? Each of these devices contains a processor and software and is designed to perform a specific function. For example, the modem is designed to send and receive digital data over analog telephone line. That's it and all of the other devices can be summarized in a single sentence as well.

If an embedded system is designed well, the existence of the processor and software could be completely unnoticed by the user of the device. Such is the case for a microwave oven, VCR, or alarm clock. In some cases, it would even be possible to build an equivalent device that does not contain the processor and software. This could be done by replacing the combination with a custom integrated circuit that performs the same functions in hardware. However, a lot of flexibility is lost when a design is hard-cooled in this way. It is mush easier, and cheaper, to change a few lines of software than to redesign a piece of custom hardware.

1.2 History and Future:

Given the definition of embedded systems earlier is this chapter; the first such systems could not possibly have appeared before 1971. That was the year Intel introduced the world's first microprocessor. This chip, the 4004, was designed for use in a line of business calculators produced by the Japanese Company Busicom. In 1969, Busicom asked Intel to design a set of custom integrated circuits-one for each of their new calculator models. The 4004 was Intel's response rather than design custom hardware for each calculator, Intel proposed a general-purpose circuit that could be used throughout the entire line of calculators. Intel's idea was that the software would give each calculator its unique set of features.

The microcontroller was an overnight success, and its use increased steadily over the next decade. Early embedded applications included unmanned space probes, computerized traffic lights, and aircraft flight control systems. In the 1980s, embedded systems quietly rode the waves of the microcomputer age and brought microprocessors into every part of our kitchens (bread machines, food processors, and microwave ovens), living rooms (televisions, stereos, and remote controls), and workplaces (fax machines, pagers, laser printers, cash registers, and credit card readers).

It seems inevitable hat the number of embedded systems will continue to increase rapidly. Already there are promising new embedded devices that have enormous market potential; light switches and thermostats that can be central computer, intelligent air-bag systems that don't inflate when children or small adults are present, pal-sized electronic organizers and personal digital assistants (PDAs), digital cameras, and dashboard navigation systems. Clearly, individuals who possess the skills and desire to design the next generation of embedded systems will be in demand for quite some time.

1.3 Real Time Systems:

One subclass of embedded is worthy of an introduction at this point. As commonly defined, a real-time system is a computer system that has timing constraints. In other words, a real-time system is partly specified in terms of its ability to make certain calculations or decisions in a timely manner. These important calculations are said to have deadlines for completion. And, for all practical purposes, a missed deadline is just as bad as a wrong answer.

The issue of what if a deadline is missed is a crucial one. For example, if the real-time system is part of an airplane's flight control system, it is possible for the lives of the passengers and crew to be endangered by a single missed deadline. However, if instead the system is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the system is a hard real-time system. Real-time systems at the other end of this discussion are said to have "soft" deadlines.

All of the topics and examples presented in this book are applicable to the designers of real-time system who is more delight in his work. He must guarantee reliable operation of the software and hardware under all the possible conditions and to the degree that human lives depend upon three system's proper execution, engineering calculations and descriptive paperwork.

Application Areas : Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication,Data communication, telecommunications, transportation, military and so on.

Consumer appliances: At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today’s high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing.

Office automation: The office automation products using em embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

Industrial automation: Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

Medical electronics: Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

Computer networking: Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems.

Telecommunications: In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), sate11ite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

Wireless technologies: Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20’h century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

Insemination: Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easy testing and measurement in the field by field-personnel.

Security: Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in. every industrial segment- consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines.Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

Finance: Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society.

1.4 Overview of Embedded System Architecture

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the 'firmware'. The embedded system architecture can be represented as a layered architecture as shown in Fig.

The operating system runs above the hardware, and the application software runs above the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system.

For small appliances such as remote control units, air conditioners, toys etc., there is no need for an operating system and you can write only the software specific to that application. For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run fora long time you don't need to reload new software.

Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are:

- Central Processing Unit (CPU)
- Memory (Read-only Memory and Random Access Memory)
- Input Devices
- Output devices
- Communication interfaces
- Application-specific circuitry

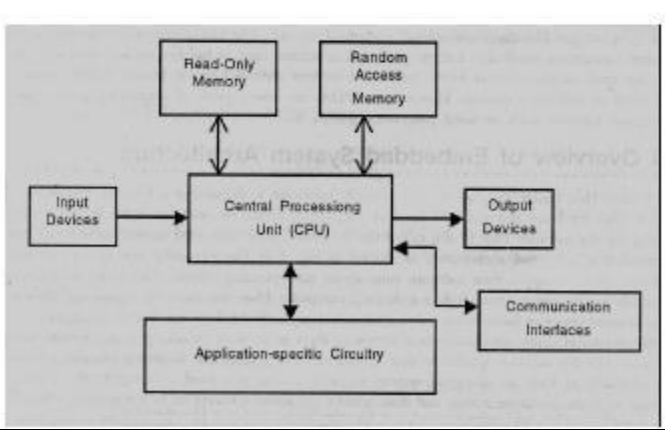


Fig 1.1 Blocks of hardware embedded system

Central Processing Unit (CPU):

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

Memory:

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is program is executed.

Input devices:

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device foruser interaction; they take inputs fromsensors or transducers 1'fnd produce electrical signals that are in turn fed to other systems.

Output devices:

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few Light Emitting Diodes (LEDs) to indicate the health status of the system modules, or for visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display some important parameters.

Communication interfaces:

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

Application-specific circuitry:

Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a way that the power consumption is minimized.

LITERATURE REVIEW

1. Railway track crack detection using robot

K. R. Jyothisree; C. A. Amalraj; A. T. Jose; A. Rajeev; T. A. Alhana

To ensure public safety, railroad transportation needs to be constantly inspected and promptly maintained. Traditional manual inspections are expensive and time-consuming, but the accuracy of flaw identification depends on the inspector's skill and productivity. Railway track inspection using robots is an innovative technology that has revolutionized the way railway tracks are inspected. It involves the use of autonomous robots that are designed to inspect and monitor railway tracks. The robots are equipped with ultrasonic sensors that enable them to detect any kind of defect or damage to the tracks. They are programmed to move along the track and detect potential defects or damages. This technology has improved the safety standards of railway tracks by helping the operators to detect and repair any defects in a timely manner. It has also helped reduce the need for manual track inspection, thus reducing the costs associated with it. Furthermore, the use of robots for track inspection has also helped in reducing the risk of human error during the inspection process.

2. Railway Track Crack Detection Using IoT Model

Sai Sahana R K; Anush P Upadya; Ashish P Upadya; Md Mushtaq; Chandrashekhar Pomu Chavan

Railway transportation is one of the most used means of transportation in India and many other countries, as large number of people use rail for transportation everyday, safety of the passengers is one of the important goals to be met. The railway department is still in its growing phase so; there are few issues to be focused on. Some of them are, is to reduce the number of accidents that occur due to derail and crack in the track. In this paper, we are building a IoT based model to detect crack using Arduino and IR sensor, once the crack is found, the location of the crack will be fetched with the help of GPS using GSM module, a message will be sent to the railway authorities’ registered mobile number about the crack being found, along with the location of the crack in the form of a Google maps link. This model will be fixed on to railway track maintenance vehicle which run independent of train. This model is built to be simple as it operates outdoors.

3. IoT Based Railway Track Faults Detection and Localization Using Acoustic Analysis

Hafeez Ur Rehman Siddiqui; Adil Ali Saleem; Muhammad Amjad Raza; Kainat Zafar

Rail is one of the most energy efficient and economical modes of transportation. Regular railway track health inspection is an essential part of a robust and secure train operation. Delayed investigations and problem discoveries pose a serious risk to the safe functioning of rail transportation. The traditional method of manually examining the rail track using a railway cart is both inefficient and susceptible to mistakes and biasness. It is imperative to automate inspection in order to avert catastrophes and save countless lives, particularly in zones where train accidents are numerous. This research develops an Internet of Things (IoT)-based autonomous railway track fault detection scheme to enhance the existing railway cart system to address the aforementioned issues. In addition to data collection on Pakistani railway lines, this work contributes significantly to railway track fault identification and classification based on acoustic analysis, as well as fault localization. Based on their frequency of occurrences, six types of track faults were first targeted: wheel burnt, loose nuts and bolts, crash sleeper, creep, low joint, and point and crossing. Support vector machines, logistic regression, random forest, extra tree classifier, decision tree classifier, multilayer perceptron and ensemble with hard and soft voting were among the machine learning methods used. The results indicate that acoustic data can successfully assist in discriminating track defects and localizing these defects in real time. The results show that MLP achieved the best results, with an accuracy of 98.4 percent.

EXISTING SYSTEM:

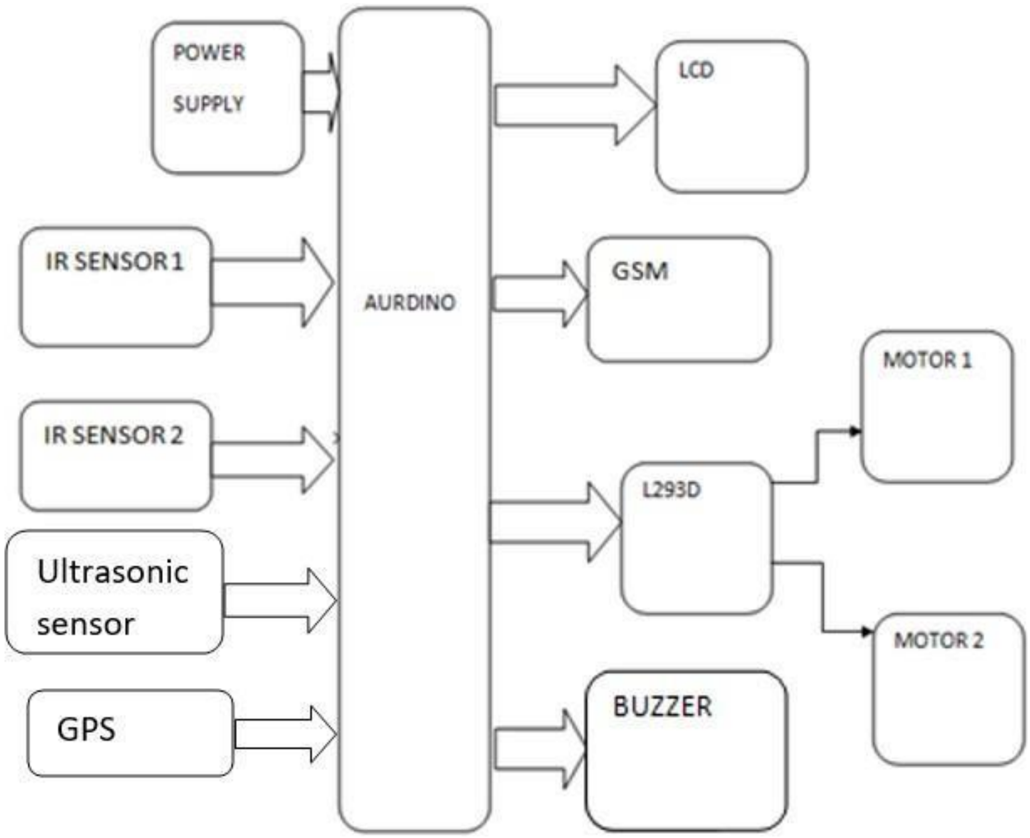
In the existing system cracks on the track can be identified by techniques like visual inspection, video streaming, , eddy current, magnetic field methods. Visual inspection is the oldest of all these methods in which the components are scanned visually. In India this method is used widely although it produces poorest result. In video streaming a web camera is used to continuously monitor the track. In this technique minute hairline cracks cannot be seen and high cost method. In eddy current method, current is made to pass through the track for flaw detection and the results produced are not accurate. All these techniques take a lot of processing power and an extreme amount of time causing the robot speed slow and thereby inconvenient.

PROPOSED SYSTEM:

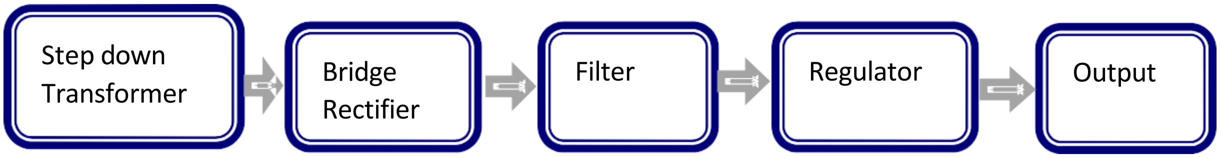
The proposed system overcomes the limitations of the existing system that are used for the detection of faulty tracks. In this proposed system we are using Arduino UNO board. The Arduino is an open source integrated development environment which simplified the coding greatly. The proposed system consists of Ultrasonic sensor for crack detection and . Motor driver L293D is used to drive the DC motors.

2. BLOCK DIAGRAM OF PROJECT AND FUNCTIONING

BLOCK DIAGRAM:



POWER SUPPLY BLOCKDIAGRAM:

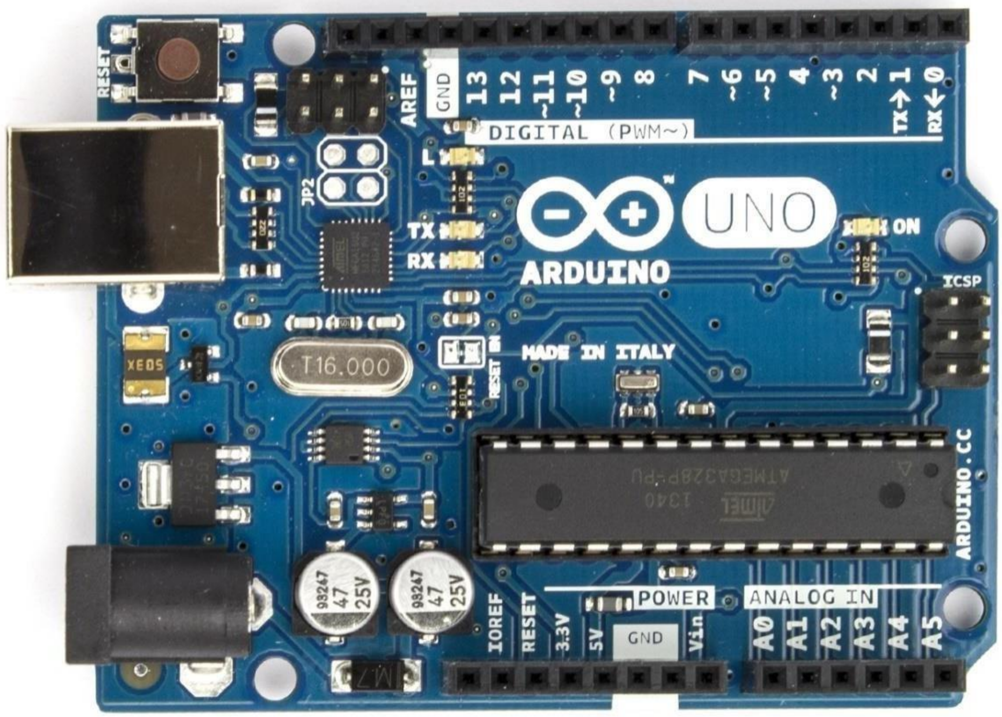


3. DESCRPTION OF COMPONENTS

3.1 ARDUINO

Overview The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). 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 USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2. "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 the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.



Summary:

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors

Power

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 (wall-wart) 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 recommended range is 7 to 12 volts.

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

Memory:

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of 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 pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 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. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:
- AREF.Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

3.2 POWER SUPPLY

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

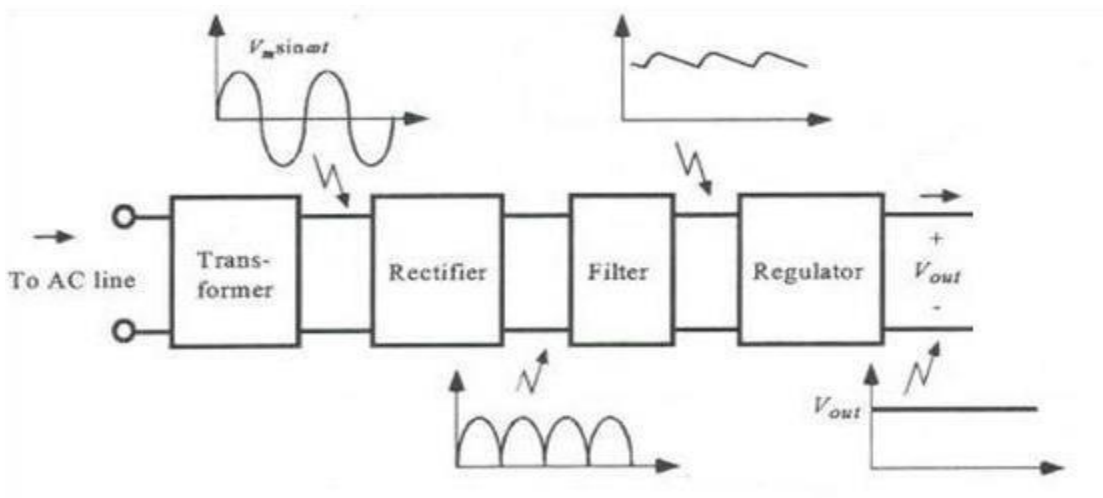


Fig 3.5 Block Diagram of Power supply

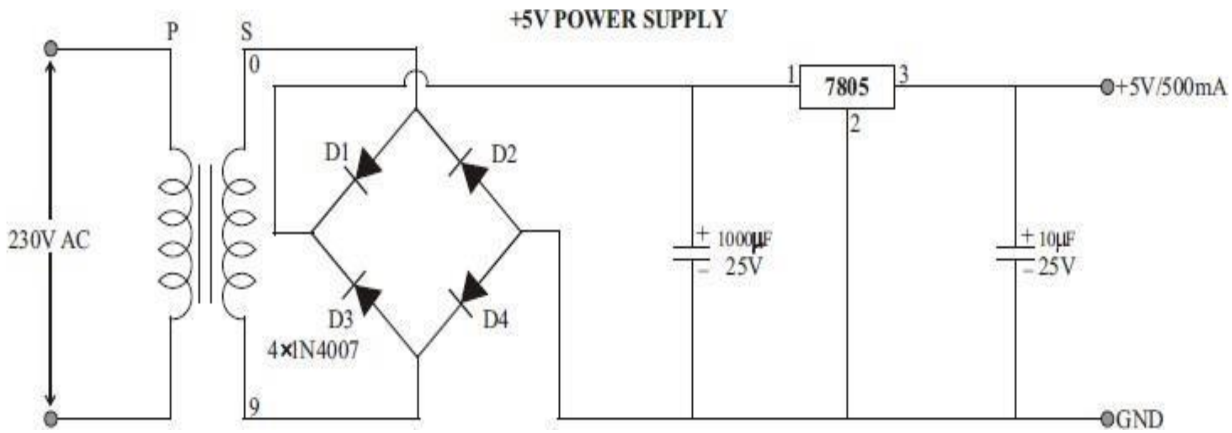


Fig 3.6 Circuit Diagram of Power supply

3.2.1 Step down Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

3.2.2 Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

3.2.3 Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

3.2.4 Voltage Regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

Features:

• Output Current up to 1A.

• Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V.

• Thermal Overload Protection. • Short Circuit Protection.

• Output Transistor Safe Operating Area Protection.

3.3 LCD/DISPLAY

Liquid Crystal Display also called as LCD is very helpful in providing user interface as well as for debugging purpose. The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers

Pin Description:

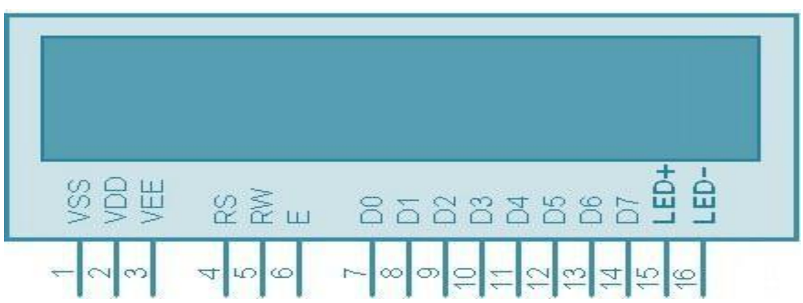


Fig 3.13 Pin Diagram of LCD

Pin No.	Name	Description
1	**VSS**	Power supply (GND)

2	**VCC**	Power supply (+5V)
3	**VEE**	Contrast adjust
4	**RS**	0 = Instruction input 1 = Data input
5	**R/W**	0 = Write to LCD module 1 = Read from LCD module
6	**EN**	Enable signal
7	**D0**	Data bus line 0 (LSB)
8	**D1**	Data bus line 1
9	**D2**	Data bus line 2
10	**D3**	Data bus line 3
11	**D4**	Data bus line 4
12	**D5**	Data bus line 5
13	**D6**	Data bus line 6
14	**D7**	Data bus line 7 (MSB)
15	**LED+**	Back Light VCC
16	**LED-**	Back Light GND

Table 3.5 Pin Description of LCD

Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	F	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5x10 dots F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Table3.6 Command Operation of LCD

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48

2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add	128+add
17	Set CGRAM address or set pointer to CGRAM location	0x40+add	64+add

Table 3.7 Command List of LCD

Sending Commands to LCD

To send commands we simply need to select the command register. Everything is same as we have done in the initialization routine. But we will summarize the common steps and put them in a single subroutine. Following are the steps:

- move data to LCD port
- select command register
- select write operation
- send enable signal
- wait for LCD to process the command
- Sending Data to LCD
- To send data move data to LCD port
- select data register
- select write operation.

DISPLAY

- often called a **notebook**, is a small, portable personal computer (PC) with a "clamshell" form factor, typically having a thin LCD or LED computer screen mounted on the inside of the upper lid of the clamshell and an alphanumeric keyboard on the inside of the lower lid. The clamshell is opened up to use the computer. Laptops are folded shut for transportation, and thus are suitable for mobile use.[1] Its name comes from lap, as it was deemed to be placed on a person's lap when being used. Although originally there was a distinction between laptops and notebooks (the former being bigger and heavier than the latter), as of 2014, there is often no longer any difference.[2] Today, laptops are commonly used in a variety of settings, such as at work, in education, for playing games, Internet surfing, for personal multimedia, and general home computer use.

- Laptops combine all the input/output components and capabilities of a desktop computer, including the display screen, small speakers, a keyboard, data storage device, optical disc drive, pointing devices (such as a touchpad or trackpad), a processor, and memory into a single unit. Most modern laptops feature integrated webcams and built-in microphones, while many also have touchscreens. Laptops can be powered either from an internal battery or by an external power supply from an AC adapter. Hardware specifications, such as the processor speed and memory capacity, significantly vary between different types, makes, models and price points.

- Design elements, form factor and construction can also vary significantly between models depending on intended use. Examples of specialized models of laptops include rugged notebooks for use in construction or military applications, as well as low production cost laptops such as those from the One Laptop per Child (OLPC) organization, which incorporate features like solar charging and semi-flexible components not found on most laptop computers. Portable computers, which later developed into modern laptops, were originally considered to be a small niche market, mostly for specialized field applications, such as in the military, for accountants, or for traveling sales representatives. As the portable computers evolved into the modern laptop, they became widely used for a variety of purposes.[3]

3.4 ULTRASONIC SENSOR

The working principle, applications and limitations of ultrasonic sensors

AUGUST 6, 2019 BY LYNNETTE REESE 4 COMMENTS

FacebookTwitterLinkedInEmail

Figure 1: HC SR04 ultrasonic sensor. (Source: Digikey

The ultrasonic sensor (or transducer) works on the same principles as a radar system. An ultrasonic sensor can convert electrical energy into acoustic waves and vice versa. The acoustic wave signal is an ultrasonic wave traveling at a frequency above 18kHz. The famous HC SR04 ultrasonic sensor generates ultrasonic waves at 40kHz frequency.

Typically, a microcontroller is used for communication with an ultrasonic sensor. To begin measuring the distance, the microcontroller sends a trigger signal to the ultrasonic sensor. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor. When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter. As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.

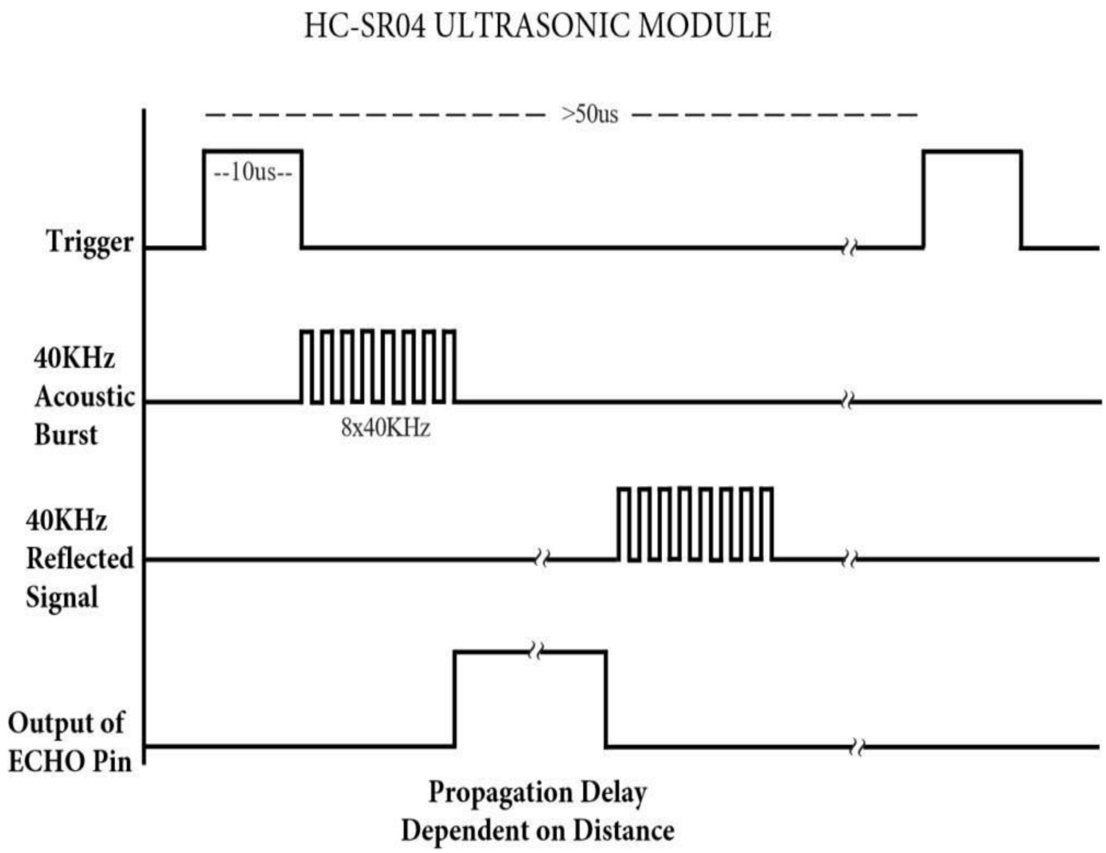


Figure 2: Representation of trigger signal, acoustic bursts, reflected signal and output of echo pin. (Source: HC-SR04 User Guide)

The microcontroller interprets the time signal into distance using the following functions:

$$Distance\ (cm) = \frac{echo\ pulse\ width\ (uS)}{58}$$

$$Distance\ (inch) = \frac{echo\ pulse\ width\ (uS)}{148}$$

Theoretically, the distance can be calculated using the TRD (time/rate/distance) measurement formula. Since the calculated distance is the distance traveled from the ultrasonic transducer to the object—and back to the transducer—it is a two-

way trip. By dividing this distance by 2, you can determine the actual distance from the transducer to the object. Ultrasonic waves travel at the speed of sound (343 m/s at 20°C). The distance between the object and the sensor is half of the distance traveled by the sound wave.[iv] The following equation calculates the distance to an object placed in front of an ultrasonic sensor:

$$distance = \frac{time\ taken\ x\ speed\ of\ sound}{2}$$

Applications

Multiple areas of engineering use ultrasonic sensors. “No-contact” distance measuring is very useful in automation, robotics, and instrumentation. Below, we investigate the applications of ultrasonic sensor

3.5 GSM

SM 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 services 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.

GSM Modem

There are various cell sizes in a GSM system such as macro, micro, pico and umbrella cells. Each cell varies as per the implementation domain. There are five different cell sizes in a GSM network macro, micro, pico and umbrella cells. The coverage area of each cell varies according to the implementation environment.

Time Division Multiple Access

TDMA technique relies on assigning different time slots to each user on the same frequency. It can easily adapt to data transmission and voice communication and can carry 64kbps to 120Mbps of data rate.

GSM Architecture

A GSM network consists of the following components:

- **A Mobile Station:** It is the mobile phone which consists of the transceiver, the display and the processor and is controlled by a SIM card operating over the network.
- **Base Station Subsystem:** It acts as an interface between the mobile station and the network subsystem. It consists of the Base Transceiver Station which contains the radio transceivers and handles the protocols for communication with mobiles. It also consists of the Base Station Controller which controls the Base Transceiver station and acts as a interface between the mobile station and mobile switching centre.

- **Network Subsystem:** It provides the basic network connection to the mobile stations. The basic part of the Network Subsystem is the Mobile Service Switching Centre which provides access to different networks like ISDN, PSTN etc. It also consists of the Home Location Register and the Visitor Location Register which provides the call routing and roaming capabilities of GSM. It also contains the Equipment Identity Register which maintains an account of all the mobile equipments wherein each mobile is identified by its own IMEI number. IMEI stands for International Mobile Equipment Identity.

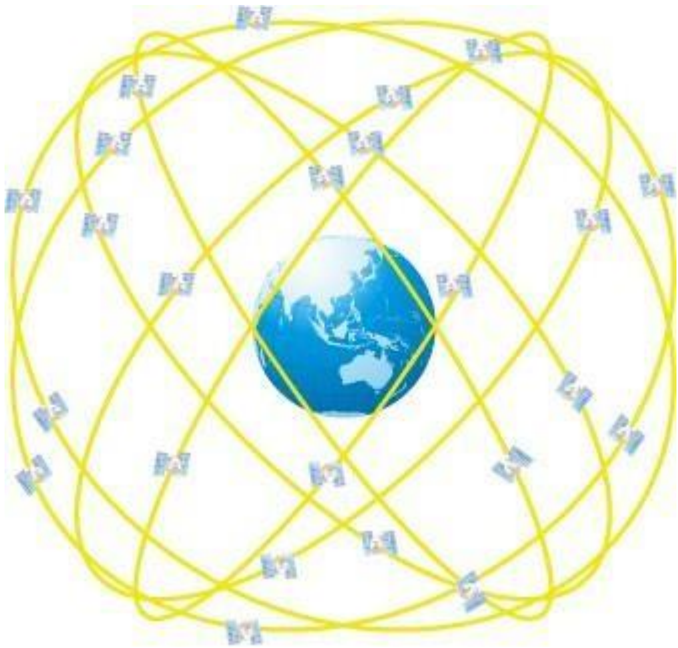
Features of GSM Module:

- Improved spectrum efficiency
- International roaming
- Compatibility with integrated services digital network (ISDN)
- Support for new services.
- SIM phonebook management
- Fixed dialing number (FDN)
- Real time clock with alarm management
- High-quality speech
- Uses encryption to make phone calls more secure
- Short message service (SMS)

The security strategies standardized for the GSM system make it the most secure telecommunications standard currently accessible. Although the confidentiality of a call and secrecy of the GSM subscriber is just ensured on the radio channel, this is a major step in achieving end-to- end security.

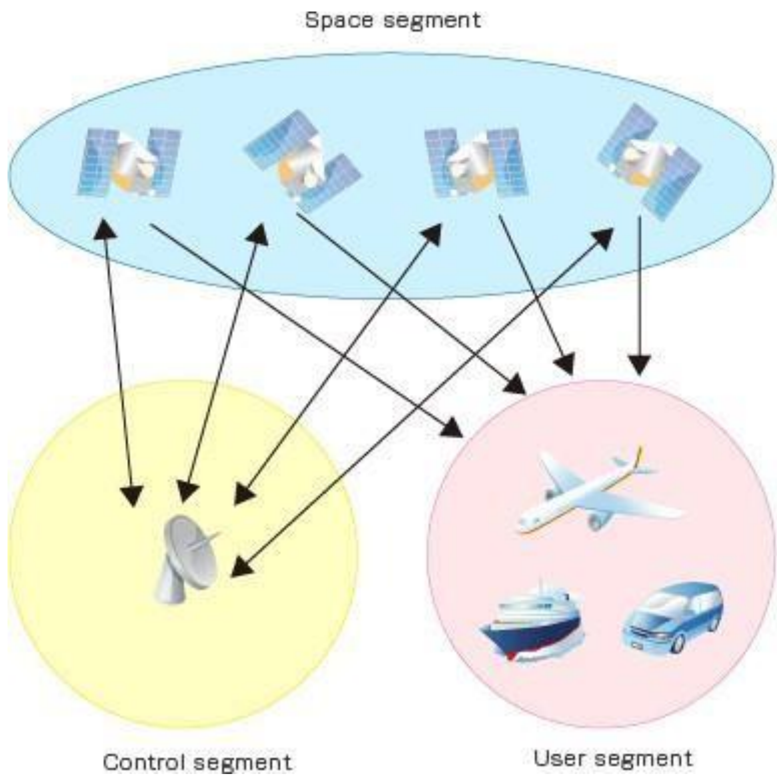
GPS(Global Positioning System)

GPS stands for Global Positioning System by which anyone can always obtain the position information anywhere in the world.



Basic structure of gps

Three-block configuration



GPS consists of the following three segments.

Space segment (GPS satellites)

A number of GPS satellites are deployed on six orbits around the earth at the altitude of approximately 20,000 km (four GPS satellites per one orbit), and move around the earth at 12-hour-intervals.

Control segment (Ground control stations)

Ground control stations play roles of monitoring, controlling and maintaining satellite orbit to make sure that the deviation of the satellites from the orbit as well as GPS timing are within the tolerance level.

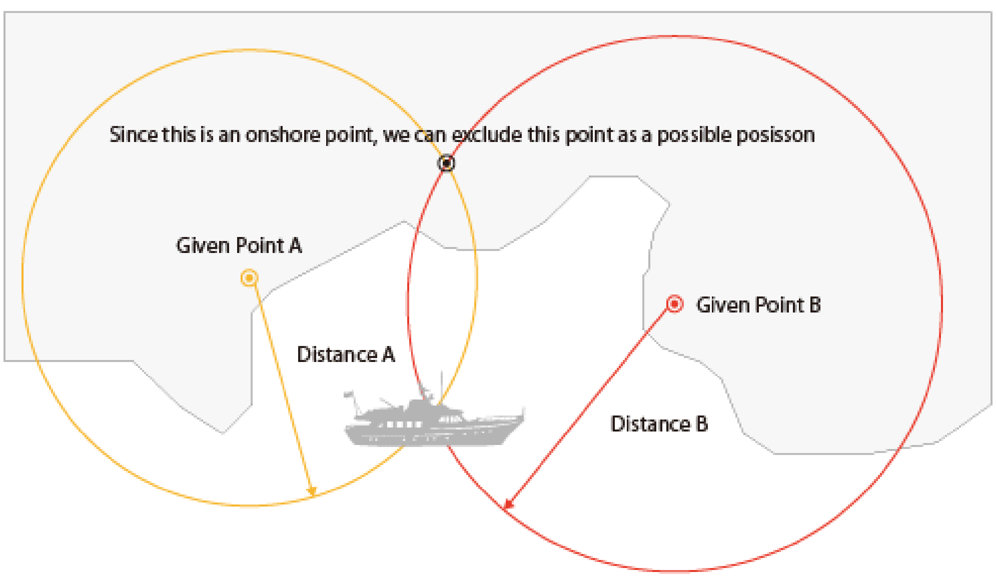
User segment (GPS receivers)

User segment (GPS receivers)

GPS positioning

Firstly, the signal of time is sent from a GPS satellite at a given point. Subsequently, the time difference between GPS time and the point of time clock which GPS receiver receives the time signal will be calculated to generate the distance from the receiver to the satellite. The same process will be done with three other available satellites. It is possible to calculate the position of the GPS receiver from distance from the GPS receiver to three satellites. However, the position generated by means of this method is not accurate, for there is an error in calculated distance between satellites and a GPS receiver, which arises from a time error on the clock incorporated into a GPS receiver. For a satellite, an atomic clock is incorporated to generate on-the-spot time information, but the time generated by clocks incorporated into GPS receivers is not as precise as the time generated by atomic clocks on satellites. Here, the fourth satellite comes to play its role: the distance from the fourth satellite to the receiver can be used to compute the position in relations to the position data generated by distance between three satellites and the receiver, hence reducing the margin of error in position accuracy.

The Fig below illustrates an example of positioning by two dimensions (position acquisition by using two given points). We can compute where we are at by calculating distance from two given points, and the GPS is the system that can be illustrated by multiplying given points and replacing them with GPS satellites on this figure.



4. ARDUINO SOFTWARE

Arduino IDE (Integrated Development Environment) is required to program the Arduino Uno board. Download it here.

PROGRAMMING ARDUINO

Once arduino IDE is installed on the computer, connect the board with computer using USB cable. Now open the arduino IDE and choose the correct board by selecting Tools>Boards>Arduino/Genuino Uno, and choose the correct Port by selecting Tools>Port. Arduino Uno is programmed using Arduino programming language based on Wiring. To get it started with Arduino Uno board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink. Once the example code (also shown below) is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the Arduino's built-in LED blinking. Below is the example code for blinking:

ARDUINO – INSTALLATION

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

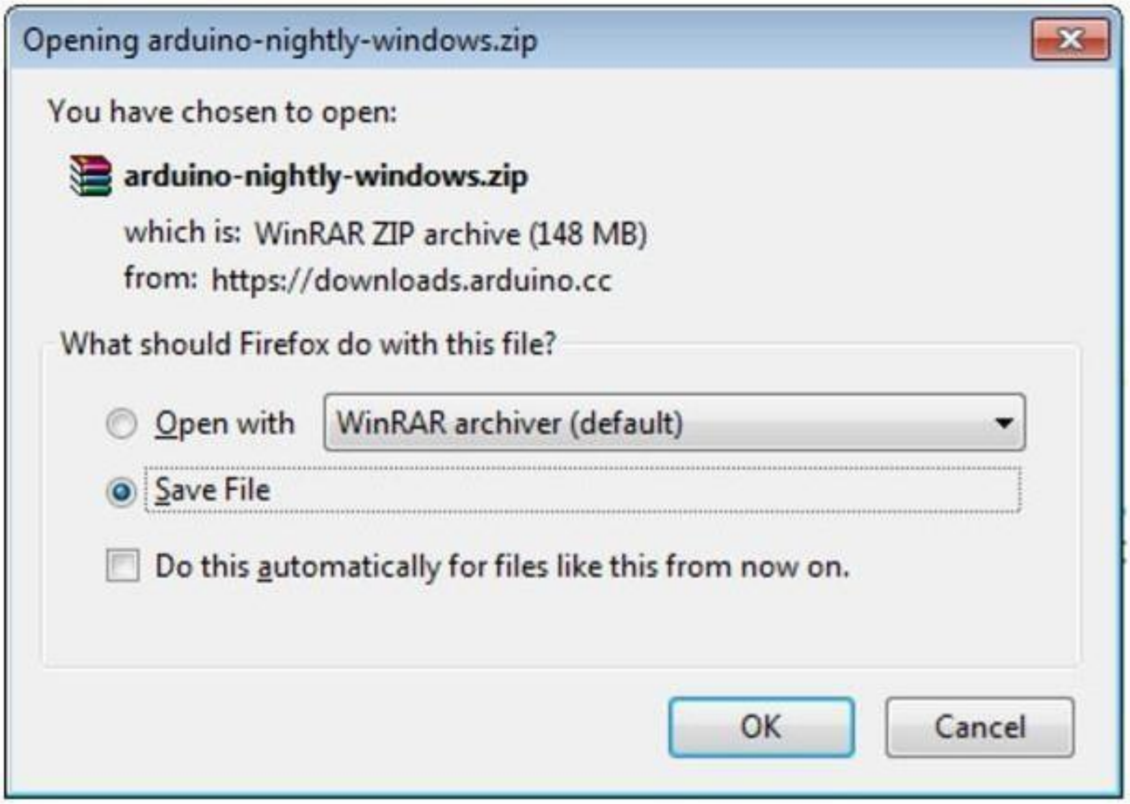
In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



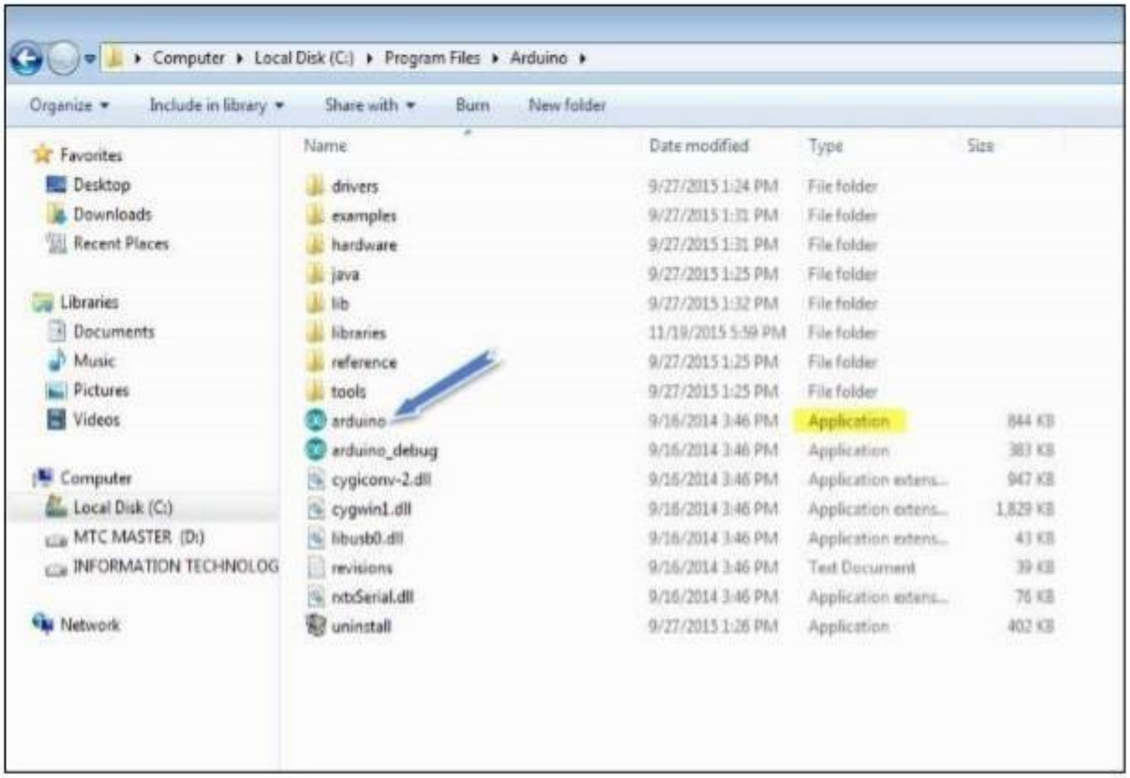
Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4: Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Doubleclick the icon to start the IDE



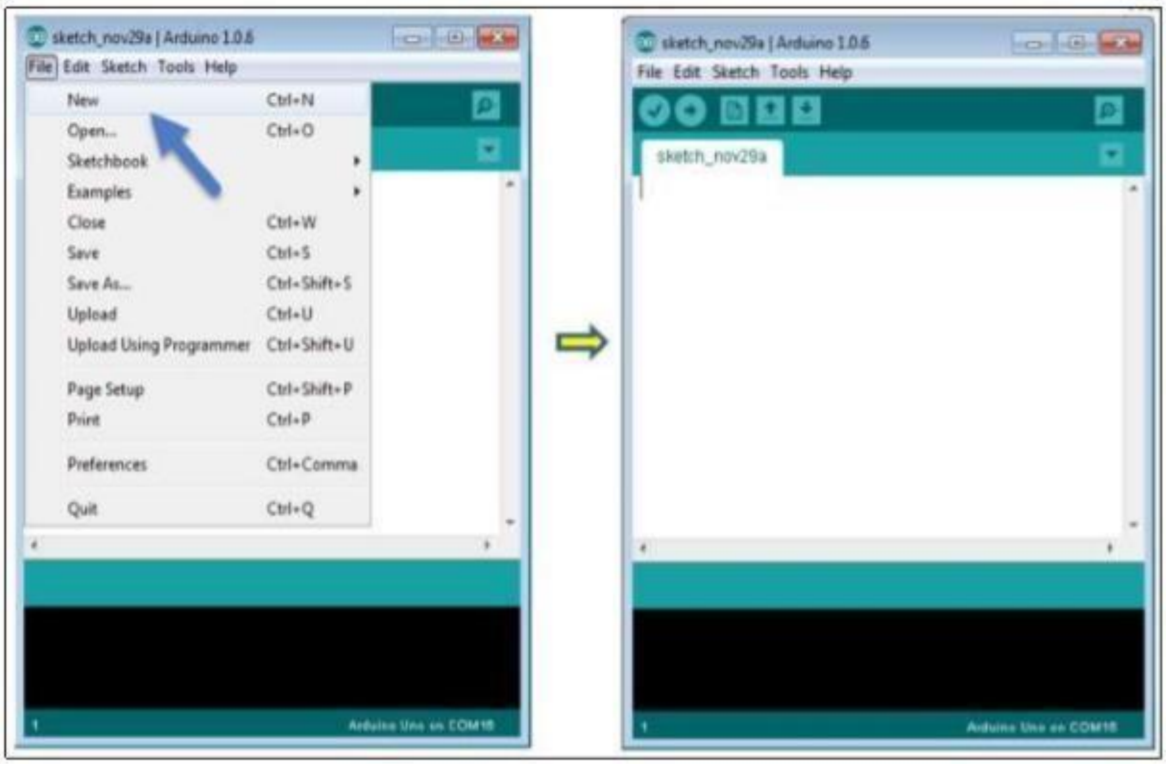
Step 5: Open your first project.

Once the software starts, you have two options:

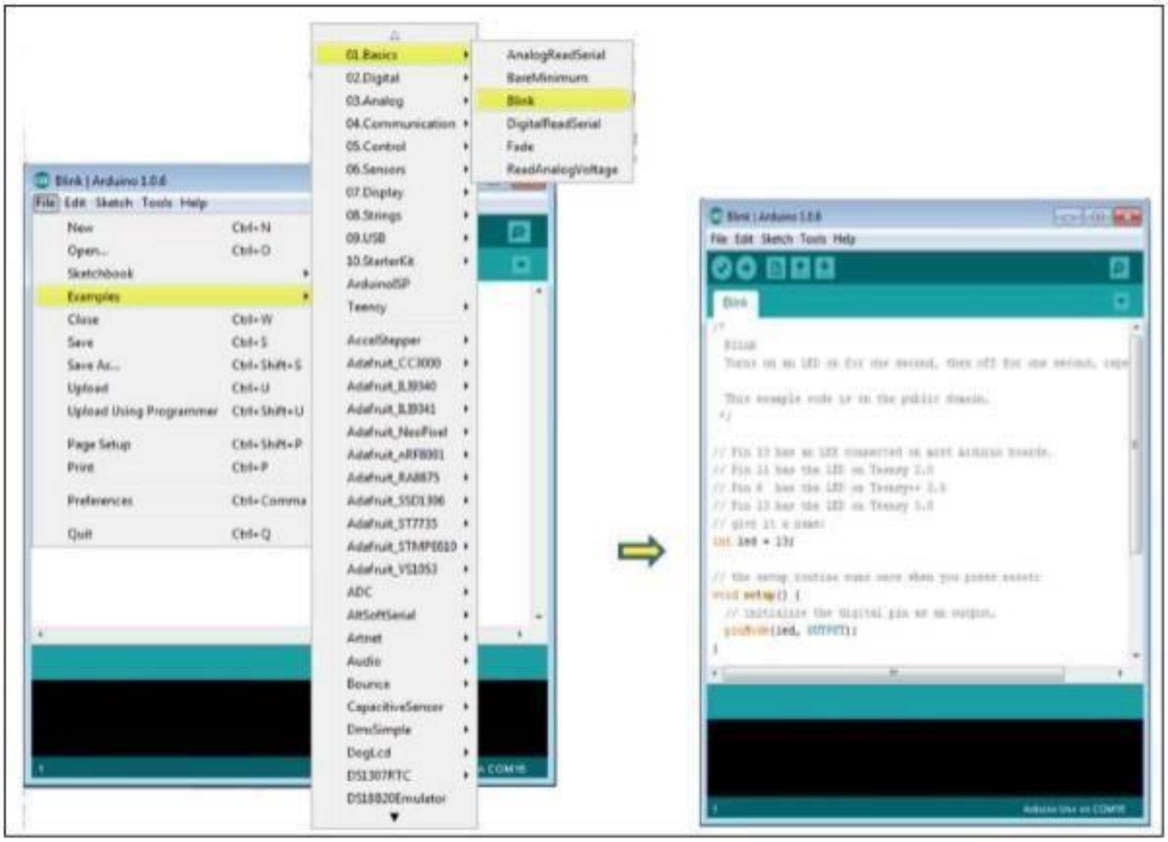
 Create a new project.

 Open an existing project example.

To create a new project, select File --> New



To open an existing project example, select File -> Example -> Basics -> Blink.

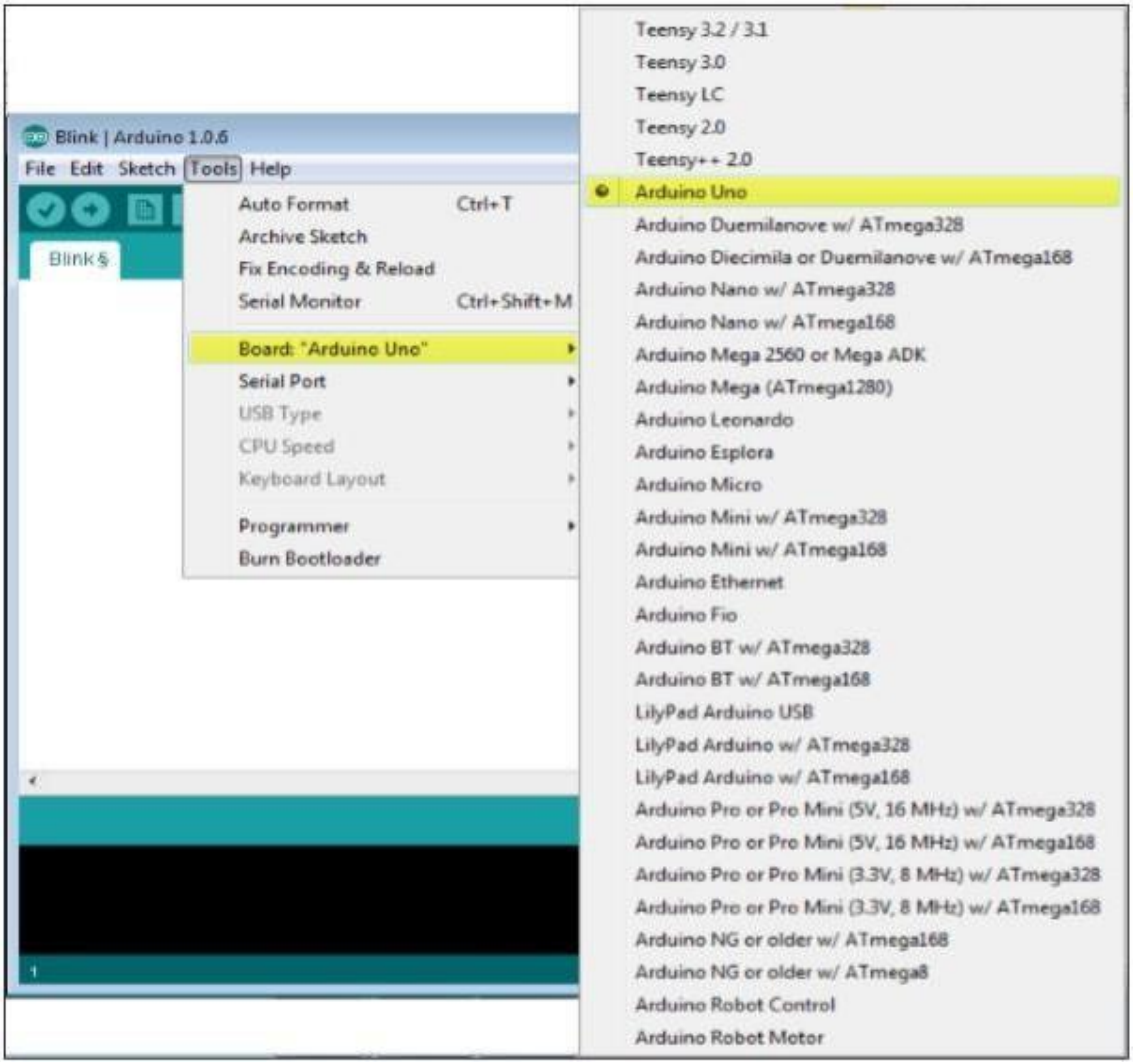


Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

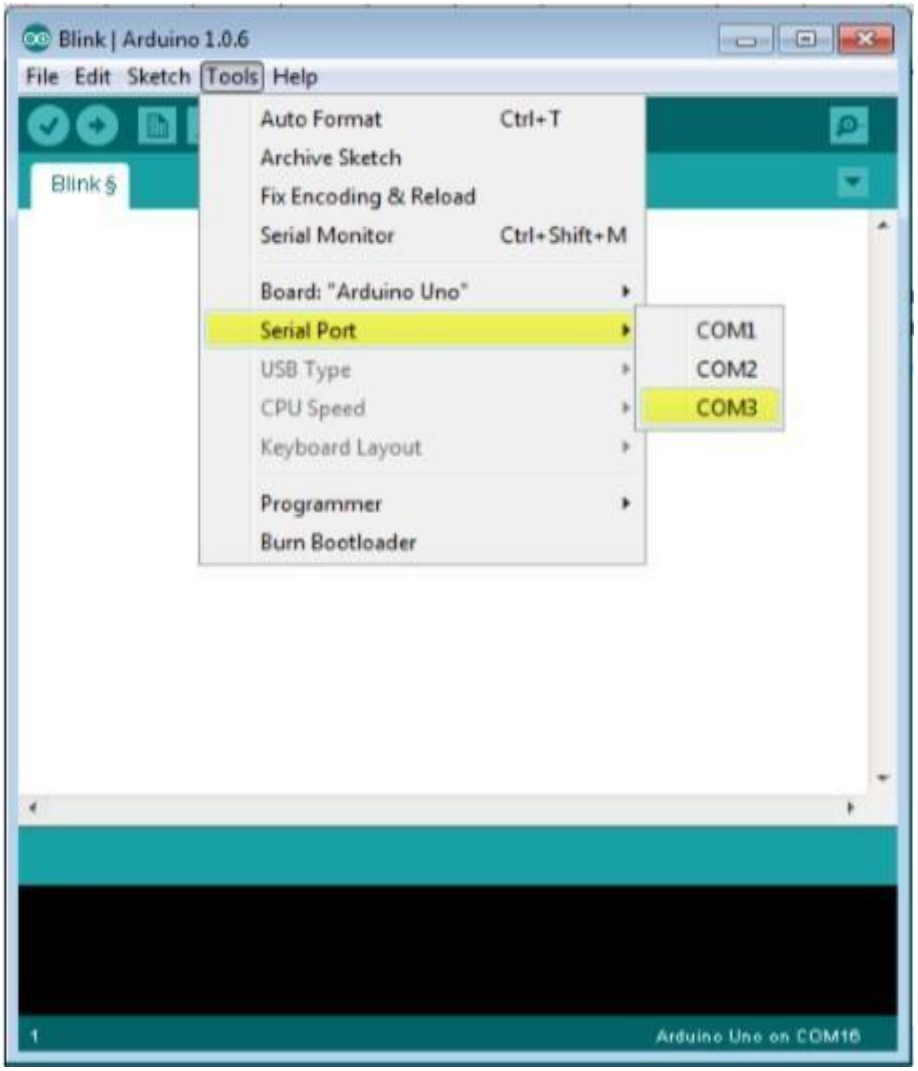
Go to Tools -> Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

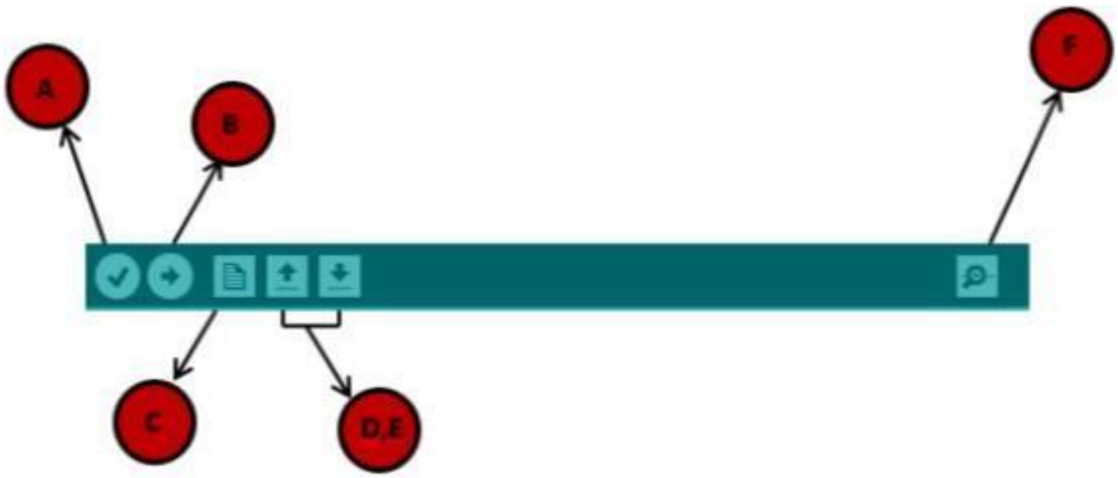
Step 7: Select your serial port.

Select the serial device of the Arduino board. Go to Tools -> Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8: Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar



A- Used to check if there is any compilation error.

B- Used to upload a program to the Arduino board.

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketch.

E- Used to save your sketch.

F- Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

ARDUINO – PROGRAM STRUCTURE

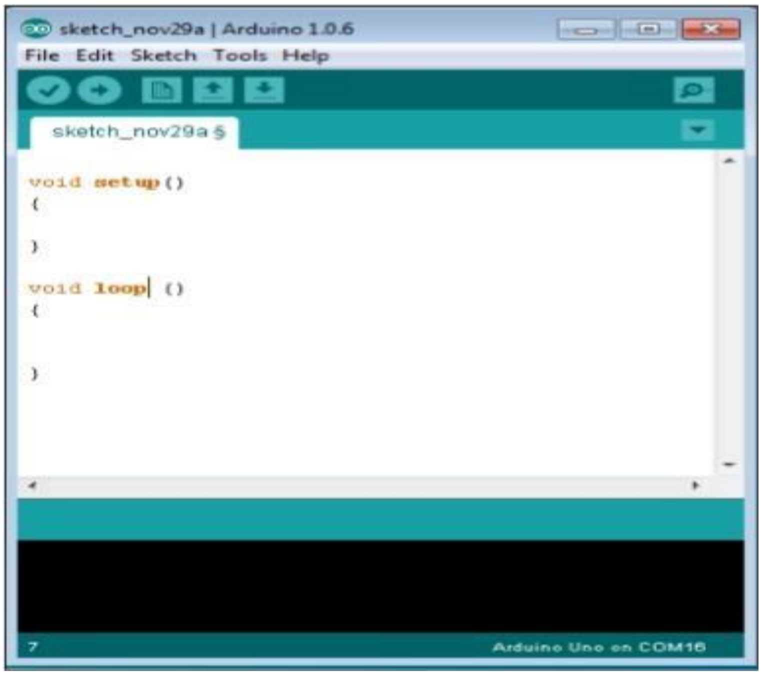
we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch: The first new terminology is the Arduino program called “sketch”.

Structure Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions:

- Setup( ) function
- Loop( ) function



```
Void setup ( )
{

}
```

PURPOSE: The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

INPUT: -

OUTPUT: -

RETURN:

```
Void Loop ( )
{

}
```

PURPOSE: After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

INPUT: -

OUTPUT: -

RETURN:

5. Code

```
#include <LiquidCrystal.h>

#include <Servo.h>

#include <SoftwareSerial.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

SoftwareSerial mySerial(2, 3);

String textMessage;

#define m11 4

#define m12 5

#define m21 6

#define pump1 7

const int trigPin = A0;

const int echoPin = A1;

long duration;

int distance;


int servoPin = A4;

Servo Servo1;


int memsx=0,memsy=0;
```

```
int hbtc=0,hbtc1=0,rtrl=0;

unsigned char rcv,count,gchr='x',gchr1='x',robos='s';


char rcvmsg[10],pastnumber[11];

 //char pastnumber1[11],pastnumber2[11];//pastnumber3[11];

  int ii=0,rchkr=0;


float tempc=0,weight=0;

float vout=0;



int sti=0;

String inputString = "";         // a string to hold incoming data

boolean stringComplete = false;  // whether the string is complete


{

  unsigned char rcr;

  do{

      rcr = mySerial.read();

    }while(rcr != 'K');

}
```

```
{
  lcd.begin(16,2);
  Serial.begin(115200);
  mySerial.begin(9600);
  Servo1.attach(servoPin);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  lcd.setCursor(0,0);
  lcd.print("  WELCOME  ");
  pinMode(m11,OUTPUT);
  pinMode(m12,OUTPUT);
  pinMode(m21,OUTPUT);
  pinMode(pump1,OUTPUT);
  digitalWrite(m11,LOW);
  digitalWrite(m12,LOW);
  digitalWrite(m21,LOW);
  digitalWrite(pump1,LOW);
  Serial.println("Initializing...");
  Servo1.write(0);
  delay(1000);
```

```
  gsminit();
serialEvent();
lcd.clear();



char memss='x';



   {
     int IR1 = digitalRead(A2);
    int IR2 = digitalRead(A3);
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    int B= duration*0.034/2;

  if(IR1 == LOW)
    {
```

```
 lcd.clear();
 lcd.print("Train Entry");
 lcd.setCursor(0, 1);
 lcd.print("Gate Closed");
digitalWrite(pump1,HIGH);
delay(2000);
digitalWrite(m11,HIGH);
digitalWrite(m12,LOW);
digitalWrite(m21,LOW);
digitalWrite(pump1,LOW);
Servo1.write(160);
     delay(1000);
mySerial.write("AT+CMGS=\"");
mySerial.write(pastnumber);
mySerial.write("\"\r\n");  delay(3000);
mySerial.write(" Train Entry ");
mySerial.write("\"\r\n");  delay(300);
mySerial.write(0x1A);delay(4000);delay(4000);
}

if(IR2 == LOW)
```

```
{
 lcd.clear();
 lcd.print("Train Exit");
 lcd.setCursor(0, 1);
 lcd.print("Gate Open");
 digitalWrite(pump1,HIGH);
 delay(2000);
 digitalWrite(m11,LOW);
 digitalWrite(m12,HIGH);
 digitalWrite(m21,LOW);
 digitalWrite(pump1,LOW);
 Servo1.write(0);
 delay(1000);
 mySerial.write("AT+CMGS=\"");
 mySerial.write(pastnumber);
 mySerial.write("\"\r\n");  delay(3000);
 mySerial.write(" Train Exit ");
 mySerial.write("\"\r\n");  delay(300);
 mySerial.write(0x1A);delay(4000);delay(4000);
 }
```

```
if(B > 40)
 {
   lcd.clear();
  lcd.print("Railway Track");
  lcd.setCursor(0, 1);
  lcd.print("Crack Detected");
  digitalWrite(pump1,HIGH);
  delay(2000);
  digitalWrite(m11,HIGH);
  digitalWrite(m12,LOW);
  digitalWrite(m21,LOW);
  digitalWrite(pump1,LOW);
      delay(1000);
  mySerial.write("AT+CMGS=\"");
  mySerial.write(pastnumber);
  mySerial.write("\"\r\n");  delay(3000);
  mySerial.write("Railway Track Crack Detected");
  mySerial.write("\"\r\n");  delay(300);
  mySerial.write("https://maps.app.goo.gl/dFq48QEYG78ELhis8");delay(300);
  mySerial.write("\"\r\n");
  mySerial.write(0x1A);delay(4000);delay(4000);
```

```
      }


  if(IR1 == HIGH && IR2 == HIGH && B<40)
    {
      lcd.clear();
    lcd.print("No Alerts");
      digitalWrite(m11,LOW);
      digitalWrite(m12,HIGH);
      digitalWrite(m21,LOW);
      digitalWrite(pump1,LOW);
      delay(1000);
      }


  if(mySerial.available()>0)
        {
      textMessage = mySerial.readString();
      Serial.print(textMessage);
      delay(10);
    }
}
```

```
void serialEvent()
{
  while (mySerial.available())
      {

         char inChar = (char)mySerial.read();
           //sti++;
           //inputString += inChar;
          if(inChar == '*')
            {sti=1;
              inputString += inChar;
             //  stringComplete = true;
             // gchr = inputString[sti-1]
            }
          if(sti == 1)
            {
                inputString += inChar;
            }
          if(inChar == '#')
            {sti=0;
              stringComplete = true;
```

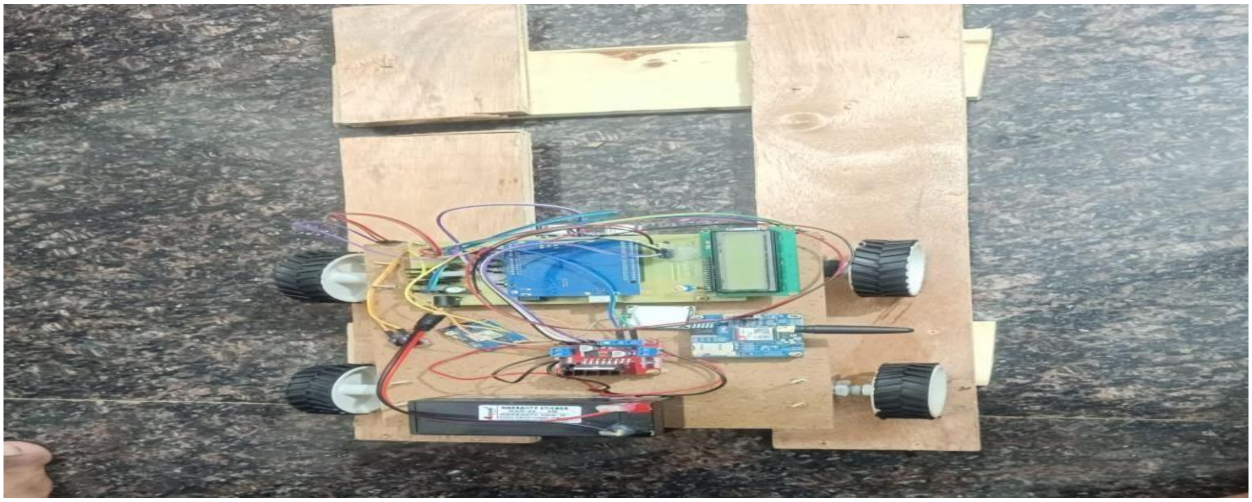
```
            }
        }
    }
    int readSerial(char result[])
    {
      int i = 0;
      while (1)
      {
        while (mySerial.available() > 0)
        {
          char inChar = mySerial.read();
          if (inChar == '\n')
            {
            result[i] = '\0';
            mySerial.flush();
            return 0;
            }
          if (inChar != '\r')
            {
            result[i] = inChar;
            i++;
```

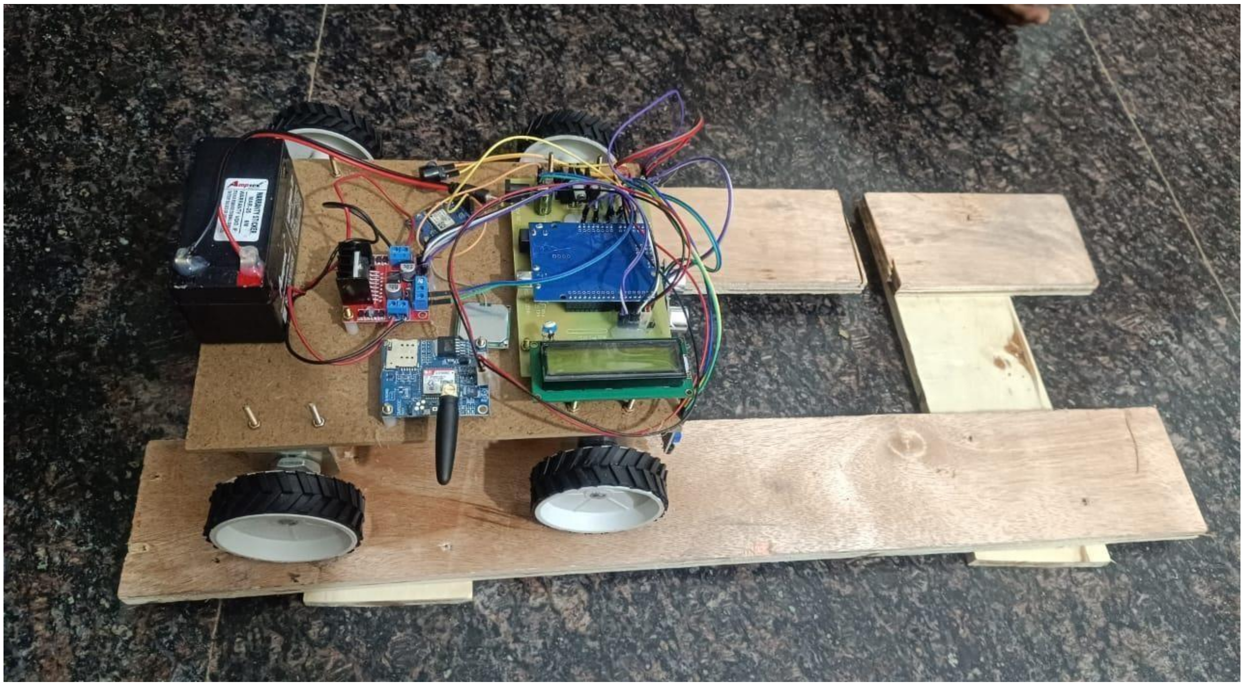
```
        }
    }
  }
}


int readSerial1(char result[])
{
  int i = 0;
  while (1)
  {
    while (mySerial.available() > 0)
    {
      char inChar = mySerial.read();
      if (inChar == '*')
        {
          result[i] = '\0';
          mySerial.flush();
          return 0;
        }
      if (inChar != '*')
        {
```

```
          result[i] = inChar;
          i++;
         }
    }
  }
}
void gsminit()
{
  mySerial.write("AT\r\n");                    okcheck();
  mySerial.write("ATE0\r\n");                  okcheck();
  mySerial.write("AT+CMGF=1\r\n");             okcheck();
  mySerial.write("AT+CNMI=2,2,0,0,0\r");       okcheck();
  mySerial.write("AT+CSMP=17,167,0,0\r\n");  okcheck();
  Serial.print("SEND MSG STORE");
  Serial.print("MOBILE NUMBER");
      lcd.clear();
      lcd.print("SEND MSG STORE");
      lcd.setCursor(0, 1);
      lcd.print("MOBILE NUMBER");
  do{
     rcv = mySerial.read();
```

```
    }while(rcv != '*');
     readSerial(pastnumber);
     pastnumber[10]='\0';
  Serial.print(pastnumber);
    lcd.clear();
    lcd.print(pastnumber);
    delay(1000);
   mySerial.write("AT+CMGS=\"");
   mySerial.write(pastnumber);
   mySerial.write("\"\r\n"); delay(3000);
   mySerial.write("registered\r\n");
   mySerial.write(0x1A);
   delay(4000);delay(5000);
  //delay(1000);
}
```

6. RESULT





7. CONCLUSION

The proposed system has the ability to detect the cracks and obstacles if any on the track. There are many advantages with the proposed system as compared with the traditional detection techniques which include low cost, low power consumption, fast detecting system without human intervention and less analysis time. By this model we can easily avoid train accidents and derailments so that many human lives can be saved.

8. REFRENCES

[1] Vikas Desai, "Design and Implementation of GSM and GPS Based Vehicle Accident Detection System", IJIT, Vol 01, Issue 03,pp. 1- 4,2013.

[2] C.Prabha, R.Sunitha, R.Anitha, "Automatic Vehicle Accident Detection and Messaging System Using GSM And GPS Modem", IJAREEIE, Vol. 3, Issue 7, pp. 1-5, 2014.

[3] Vikram Singh Kushwaha, Deepa Yadav, "Car Accident Detection System using Gps, Gsm and Bluetooth" in IJERGS May-June 2015.

[4] AboliRavindraWakure,ApurvaRajendraPatkar, "Vehicle Accident Detection and Reporting System using Gps And Gsm.", IJERGS, Vol 10,Issue 4,pp.-25- 28, April 2014.

[5] N. Watthanawisuth, "Wireless Black Box using MEMS Accelerometer and GPS Tracking for Accidental Monitoring of Vehicles", IEEE conference in Jan, 2012.

[6] Hoang Dat Pham, "Development of vehicle tracking system using GPS and GSM modem" IEEE conference in Dec, 2013.

[7] Rashida Nazir, Ayesha Tariq, Sadia Murawwat, Sajjad Rabbani,"Accident Prevention and Reporting System using GSM (SIM 900D) and GPS (NMEA 0183)", Int. J. Communications, Network and System Sciences, 2014, 7, 286-293 Published Online August 2014.

[8] Ravi Shankar Shekhar, Purushottam Shekhar, Ganesan P, "Automatic Detection of Squats in Railway Track", Innovations in Information Embedded and Communication Systems, 2015

[9] Sanket Salvi, Shashank Shetty , "AI based Solar Powered Railway Track Crack Detection and Notification System with Chatbot Support", Third International Conference on I-SMAC, 2019

[10] B. S. Sathish, Ganesan P, A. Ranganayakulu, Dola Sanjay S, Jagan Mohan Rao, "Advanced Automatic Detection of Cracks in Railway Tracks", International Conference on Advanced Computing & Communication Systems (ICACCS), 2019

[11] Nilisha Patil , Dipakkumar Shahare , Shreya Hanwate , Pranali Bagde , Karuna Kamble, Prof. Manoj Titre, "Designing of Improved Monitoring System for Crack Detection on Railway Tracks", d International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, 2021.

[12]