CHAPTER - 1

INTRODUCTION

1.1 GENERAL

Deforestation and illegal activities like tree smuggling contribute significantly to environmental imbalance and are major concerns when addressing global warming. To tackle these challenges, a system has been developed that focuses on restricting such illegal activities and protecting valuable tree species. By doing so, the system aims to preserve the environment, maintain ecological balance, and combat one of the critical factors contributing to global warming. The system's primary objective is to prevent tree smuggling and protect high-value tree species such as sandalwood, red sandalwood, teak (sag wan), and other medicinally significant plants, which are often targeted due to their high economic value. Additionally, the system detects and responds to fire-related damages, ensuring prompt action to minimize losses and protect the surrounding ecosystem. Equipped with a cut detection sensor, a fire detection sensor, and a location tracking sensor, the tree unit monitors physical tampering, temperature changes, and geographic coordinates. In case of illegal activities or fire incidents, the system transmits real-time location information to higher authorities, enabling quick response and intervention.

By preventing deforestation and illegal activities, the system contributes to environmental conservation, biodiversity preservation, and ecological sustainability. This innovative solution not only safeguards valuable and endangered tree species but also plays a crucial role in mitigating global warming and preserving natural resources for future generations.

1.2 EXISTING SYSTEM

Current forest monitoring systems mainly depend on manual patrolling and a few remote sensors, which have several limitations. These methods often result in delayed responses, as they cannot provide real-time alerts or quick action when illegal logging or forest fires occur. The coverage of these systems is usually limited, leaving many areas unmonitored, and the high operational costs make them less practical for large-scale use

1.3 DISADVANTAGES OF EXISTING SYSTEM

- Delayed detection of forest fires and illegal logging.
- Limited monitoring coverage in vast forest areas.
- Inability to provide real-time alerts or responses.
- Over-reliance on manual patrolling, prone to inefficiency.

1.4 PROPOSED SYSTEM

Our proposed system is designed to provide advanced security and protection for forests by combining various modern technologies. It includes ultrasonic sensors, vibration sensors, fire sensors, an Arduino microcontroller, an LCD display, buzzers, relay switches, a DC water pump, and GSM modules. The ultrasonic and vibration sensors play a key role in detecting unauthorized movements, such as people entering restricted areas or illegal logging activities. Fire sensors are included to quickly spot any signs of a fire starting in the forest. All the information collected by these sensors is processed by the Arduino microcontroller, which acts as the brain of the system. When the sensors detect a problem, the system responds immediately by activating buzzers to sound alarms, sending alert messages through GSM modules to inform authorities, and turning on a DC pump to control fires before they spread. This setup allows for real-time monitoring and quick actions, making it an effective solution to protect forests and respond to threats promptly.

1.5 ADVANTAGES OF PROPOSED SYSTEM

- Detects problems in real-time and sends alerts immediately.
- Monitors a large area using different types of sensors.
- Automatically acts to control fires and stop illegal activities.
- Reduces the need for expensive manual patrols.
- Responds quickly to prevent damage from spreading.

CHAPTER -2

PROJECT DESCRIPTION

2.1 GENERAL INTRODUCTION TO EMBEDDED SYSTEM

Embedded systems are designed to perform specific tasks rather than serve as general-purpose computers capable of handling a variety of functions. Some embedded systems have real-time performance constraints, which are critical for reasons such as safety and usability. In contrast, other systems may have minimal performance requirements, allowing the hardware to be simplified and costs to be reduced.

To run applications in an embedded system, a microprocessor or microcontroller is necessary. In the case of a microprocessor, external memory is used, which increases its size and enables it to handle multiple operations. However, in a microcontroller, the memory is built-in, and the system is tailored for specific applications, offering faster processing speeds. This makes microcontrollers more commonly used in embedded systems as compared to microprocessors.An embedded system can be defined as a computing device designed to perform a focused, specific job. Examples of embedded systems include household appliances like air conditioners, VCD players, DVD players, printers, fax machines, and mobile phones. Each of these devices contains a processor and specialized hardware designed to meet the specific needs of the application, along with embedded software (also known as firmware) that the processor uses to complete these tasks. Unlike general-purpose computers like desktops or laptops, which can perform a variety of functions such as gaming, word processing, and software development, embedded systems are designed to run fixed software to perform particular tasks.Embedded systems have several defining characteristics. They are designed for specific tasks and cannot be reprogrammed for different functions. These systems typically have limited resources, especially in terms of memory, and do not include secondary storage devices like CD-ROMs or floppy disks. Many embedded systems are also required to meet strict time constraints. Some, known as real-time systems, must complete tasks within precise time limits, and failing to meet these deadlines can lead to severe consequences, such as the loss of life or property damage. Power efficiency is another important factor, as many embedded systems are battery-powered and must consume minimal energy.

Finally, some embedded systems are built to function in extreme environmental conditions, such as high temperatures or humidity

2.2 OVERVIEW OF EMBEDDED SYSTEM ARCHITECTURE

Embedded systems are designed with custom-built hardware centered around a Central Processing Unit (CPU). This hardware includes memory chips that store the software, commonly referred to as "firmware." The architecture of an embedded system is typically represented as a layered structure: the hardware forms the base, the operating system (if present) runs above it, and the application software operates on top of the operating system. This layered design is similar to that of general-purpose computers, such as desktop systems. However, a key distinction lies in the fact that an operating system is not always a mandatory component in embedded systems.

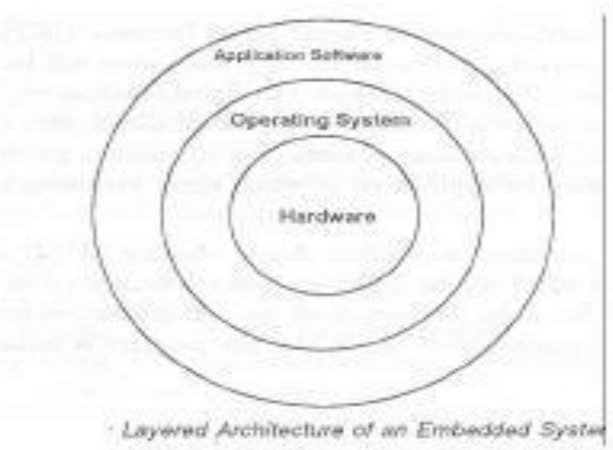


Fig. 2.2 (a) Overview of Embedded System

For small devices like remote controls, air conditioners, and toys, an operating system is unnecessary, and application-specific software can be directly written. However, for applications requiring complex processing, using an operating system is recommended. In such cases, the application software is integrated with the operating system, and the complete software package is transferred onto the memory chip. Once loaded, the software operates reliably for extended periods without the need for frequent updates or reloading.

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

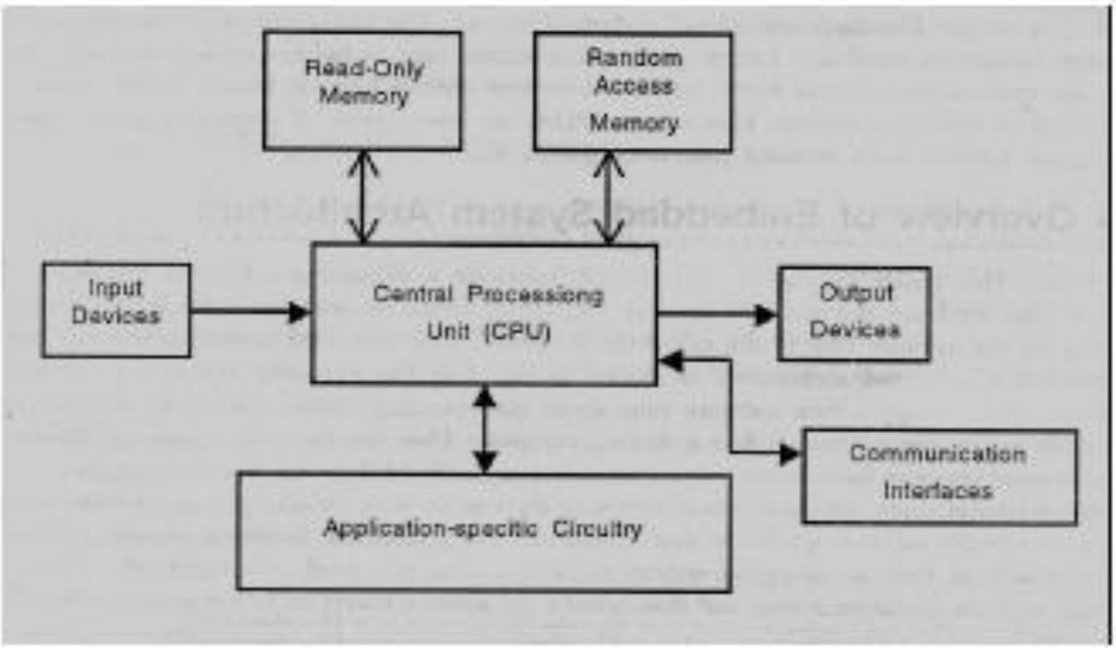


Fig.2.2 (b) Hardware blocks of Embedded Systems

Central Processing Unit (CPU):

The Central Processing Unit (CPU) in an embedded system can be a microcontroller, microprocessor, or Digital Signal Processor (DSP). A microcontroller is a cost-effective option that integrates various components, such as memory, serial communication interfaces, and analog-to-digital converters, directly onto the chip. This makes it ideal for small-scale applications, as it minimizes the need for external components. In contrast, microprocessors are more powerful but require additional external components for operation. DSPs, on the other hand, are specifically designed for signal processing tasks, making them suitable for applications like audio and video processing.

Memory:

Memory in an embedded system is classified into Random Access Memory (RAM) and Read-Only Memory (ROM). RAM is temporary storage that loses its data when the power is switched off, while ROM retains data permanently. Firmware, the software controlling the system, is stored in ROM for reliable operation. When the system is powered on, the processor reads the instructions from ROM and executes them. ROM ensures the firmware remains intact and consistent, while RAM is used for temporary data processing during program execution. Both types of memory work together to ensure the system functions efficiently.

Input Devices:

Unlike desktop computers, the input devices in embedded systems are often limited in functionality. They typically do not include a keyboard or mouse, making interaction more challenging. Many embedded systems feature a simple keypad, where pressing a specific key sends a corresponding command. In some cases, the keypad may only be used to input numeric values. However, many embedded systems, particularly those used in process control, lack direct user input devices entirely. Instead, they rely on sensors or transducers to gather data and produce electrical signals, which are then transmitted to other systems for further processing.

Output Devices:

The output devices in embedded systems are also quite limited in terms of functionality. Many embedded systems feature a few Light Emitting Diodes (LEDs) that indicate the health status of system modules or provide visual alerts for alarms. Additionally, a small Liquid Crystal Display (LCD) may be used to display important parameters or system information. These simple output devices are designed to provide essential feedback in a straightforward and efficient manner, without the complexity found in more advanced systems.

2.3 BLOCK DIAGRAM

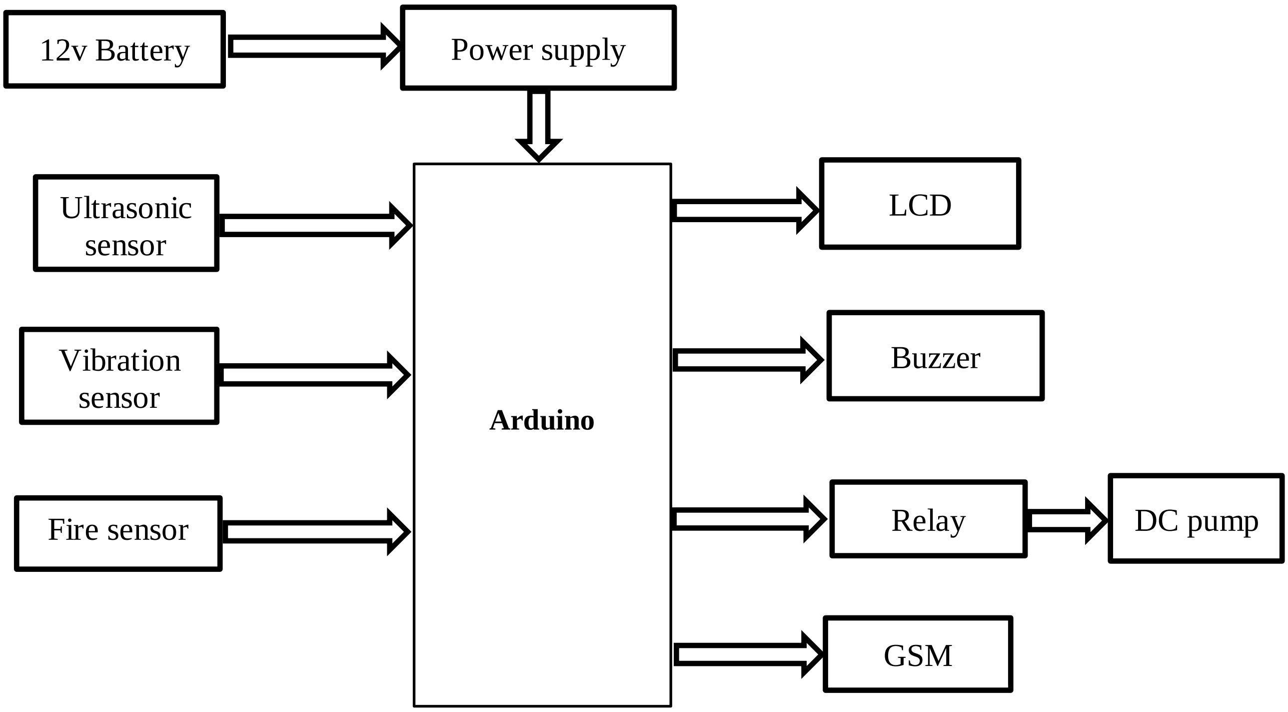


Fig:2.3 Block diagram

2.4 MODULES

2.4.1 POWER SUPPLY

The power supply section is the section which provide +5V for the components to work. IC LM7805 is used for providing a constant power of +5V.The ac voltage, typically 220V, is connected to a transformer, which steps down that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

A regulator circuit removes the ripples and also retains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

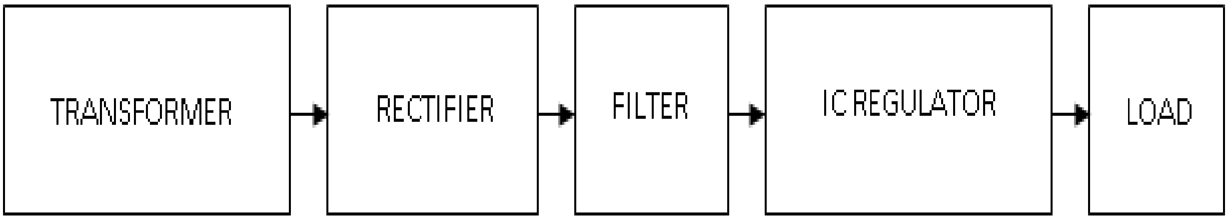


Fig 2.4.1. Block Diagram of Power Supply

2.4.2 Transformer

Transformers convert AC electricity from one voltage to another with little loss of power.Transformers work only with AC and this is one of the reasons why mains electricity is AC.Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in India) to a safer low voltage.The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead, they are linked by an alternating magnetic field created in the soft-iron core of the transformer. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up. The transformer will step down the power supply voltage (0-230V) to (0- 6V) level.

Then the secondary of the potential transformer will be connected to the bridge rectifier, which is constructed with the help of PN junction diodes. The advantages of using bridge rectifier are it will give peak voltage output as DC.

2.4.3 Rectifier

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a center-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC

2.4.4 Bridge Rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners. Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

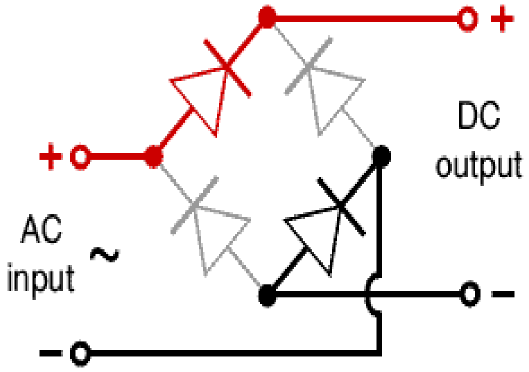


Fig2.4.4 Bridge Rectifier

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

i. The main advantage of this bridge circuit is that it does not require a special centre tapped transformer, thereby reducing its size and cost.
ii. The single secondary winding is connected to one side of the diode bridge network and the load to the other side as shown below.
iii. The result is still a pulsating direct current but with double the frequency.

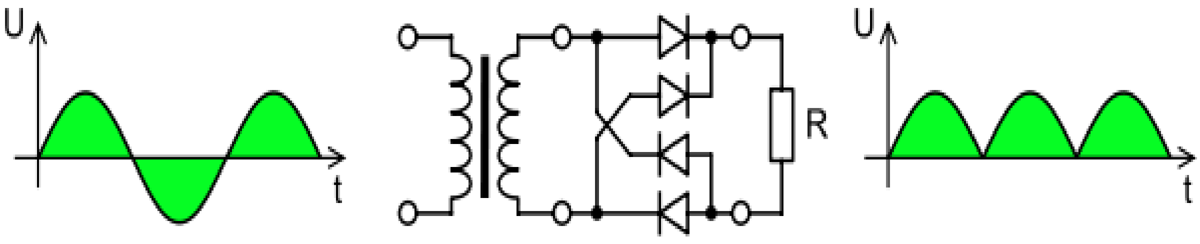


Fig2.4.4 (a): Output Waveform of DC

2.4.5 Smoothing

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

2.4.6 Voltage Regulators

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to Tens of watts.

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts. Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

Many of the fixed voltage regulator ICs has 3 leads and look like power transistors, such as the 7805 +5V 1Amp regulator. They include a hole for attaching a heat sink if necessary.

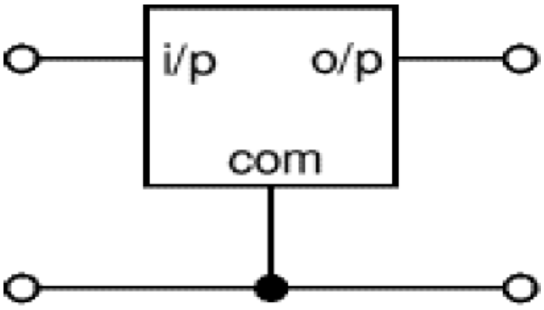


Fig2.4.6 (a) Regulator

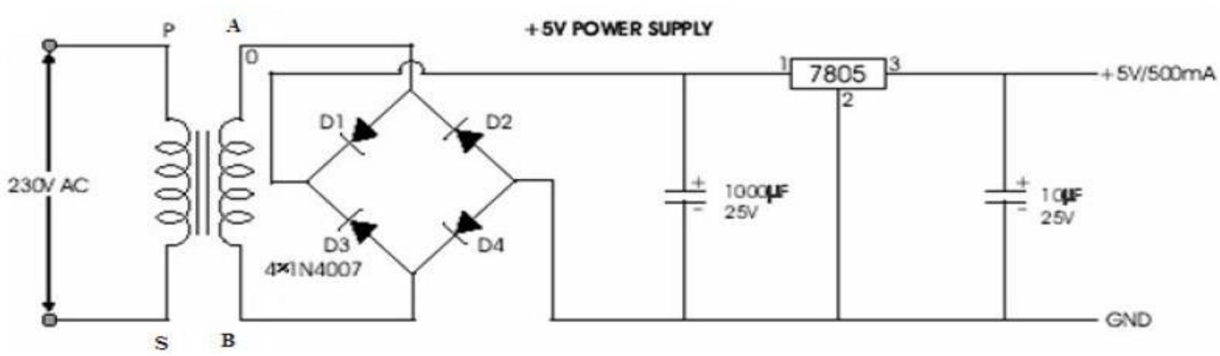


Fig 2.4.6 (b) Circuit Diagram of Power Supply

2.5 FIRE SENSOR/ FLAME SENSOR:

A flame sensor is a device designed to detect and respond to fire or flames. Its response depends on its setup and may include activating an alarm, controlling gas lines, or triggering a fire suppression system. Commonly used in industrial boilers, its main purpose is to ensure the boiler is functioning correctly. Compared to heat or smoke detectors, flame sensors are faster and more accurate due to their advanced detection mechanism.

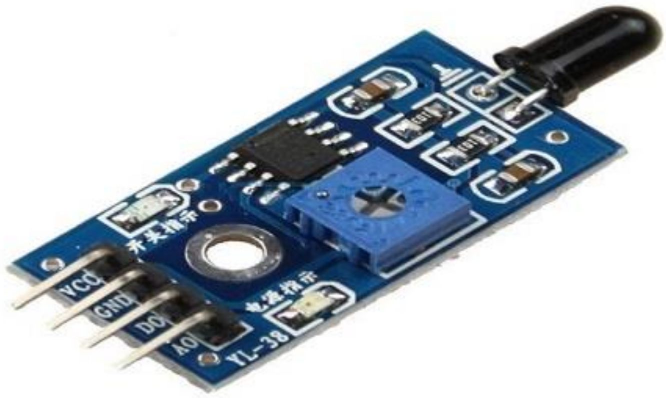


Fig:2.5 Fire sensor/ Flame sensor

Working principle: A flame sensor or detector is built using an electronic circuit combined with a receiver that detects electromagnetic radiation emitted by flames. This sensor operates using the infrared flame flash method, which enables it to detect flames even in challenging conditions. It can function effectively through layers of oil, dust, water vapor, or even ice. This makes it highly reliable in harsh environments where other sensors might struggle to work. Its ability to detect infrared radiation ensures accurate and efficient performance, making it suitable for industrial and safety applications.

Pins of fire sensor

- Pin1 (VCC pin): Voltage supply rages from 3.3V to 5.3V
- Pin2 (GND): This is a ground pin
- Pin3 (AOUT): This is an analog output pin (MCU.IO)
- Pin4 (DOUT): This is a digital output pin (MCU.IO)

2.6 LIQUID CRYSTAL DISPLAY (LCD)

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

- ❖ The declining prices of LCDs.
- ❖ The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
- ❖ Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.
- ❖ Ease of programming for characters and graphics.
- ❖ These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

A model described here is for its low price and great possibilities most frequently used in practice. It is based on the HD44780 microcontroller (Hitachi) and can display messages in two lines with 16 characters each. It displays all the alphabets, Greek letters, punctuation marks, mathematical symbols etc. In addition, it is possible to display symbols that user makes up on its own. Automatic message on display (shift left and right), appearance of the pointer, backlight etc. are considered as useful characteristics.

2.6.1 LCD CONNECTIONS

Depending on how many lines are used for connection to the microcontroller, there are 8-bit and 4-bit LCD modes. The appropriate mode is determined at the beginning of the process in a phase called “initialization”. In the first case, the data are transferred through outputs D0-D7 as it has been already explained. In case of 4-bit LED mode, for the sake of saving valuable I/O pins of the microcontroller, there are only 4 higher bits (D4-D7) used for communication, while other may be left unconnected. Consequently, each data is sent to LCD in two steps: four higher bits are sent first (that normally would be sent through lines D4-D7), four lower bits are sent afterwards. With the help of initialization, LCD will correctly connect and interpret each data received. Besides, with regards to the fact that data are rarely read from LCD (data mainly are transferred from microcontroller to LCD) one more I/O pin may be saved by simple connecting R/W pin to the Ground. Such saving has its price. Even though message displaying will be normally performed, it will not be possible to read from busy flag since it is not possible to read from display.

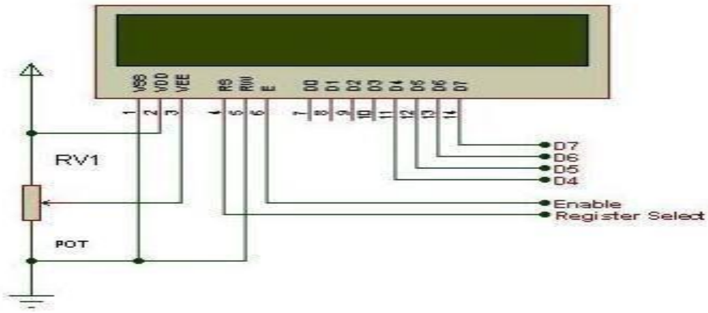


Fig 2.6.1 Pin Connections of LCD

2.7 ULTRASONOC SENSOR

Ultrasonic sensors are electronic devices that calculate the target's distance by emission of ultrasonic sound waves and convert those waves into electrical signals. The speed of emitted ultrasonic waves traveling speed is faster than the audible sound. There are mainly two essential elements which are the transmitter and receiver. Using the piezoelectric crystals, the transmitter generates sound, and from there it travels to the target and gets back to the receiver component. To know the distance between the target and the sensor, the sensor calculates the amount of time required for sound emission to travel from transmitter to receiver

Fig: 2.7 Ultrasonic sensor

2.7.1 WORKING PRINCIPLE:

It is either similar to sonar or radar which evaluates the target/object attributes by understanding the received echoes from sound/radio waves correspondingly. These sensors produce high-frequency sound waves and analyze the echo which is received from the sensor. The sensors measure the time interval between transmitted and received echoes so that the distance to the target is known.

Pins of Ultrasonic sensor

VCC – This pin has to be connected to a power supply +5V.

TRIG – This pin is used to receive controlling signals from the Arduino board. This is the triggering input pin of the sensor

ECHO – This pin is used for sending signals to the Arduino board where the Arduino calculates the pulse duration to know the distance. This pin is the ECHO output of the sensor.

GND – This pin has to be connected to the ground.

The below picture shows the **ultrasonic sensor block diagram** for distance measurement.

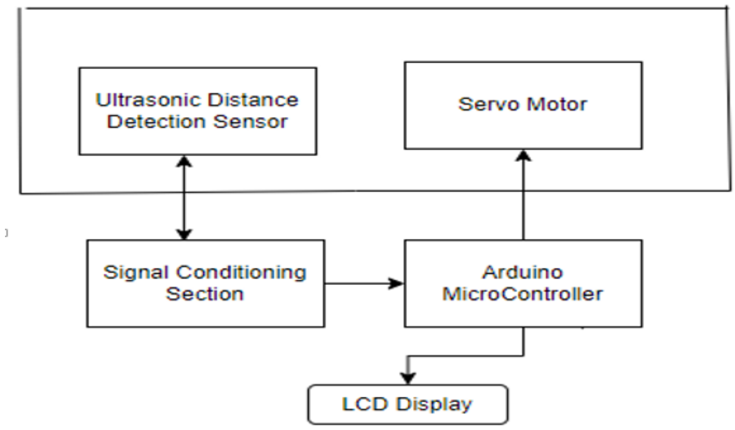


Fig: 2.7(a) Ultrasonic Sensor Block Diagram

2.8 VIBRATION SENSOR

The vibration sensor is also called a piezoelectric sensor. These sensors are flexible devices which are used for measuring various processes. This sensor uses the piezoelectric effects while measuring the changes within acceleration, pressure, temperature, force otherwise strain by changing to an electrical charge. This sensor is also used for deciding fragrances within the air by immediately measuring capacitance as well as quality.

2.8.1 WORKING PRINCIPLE

The core working principle of a vibration sensor revolves around converting mechanical vibrations (movement, oscillations, or displacement) into a measurable electrical signal. Different types of vibration sensors achieve this using various underlying mechanisms.

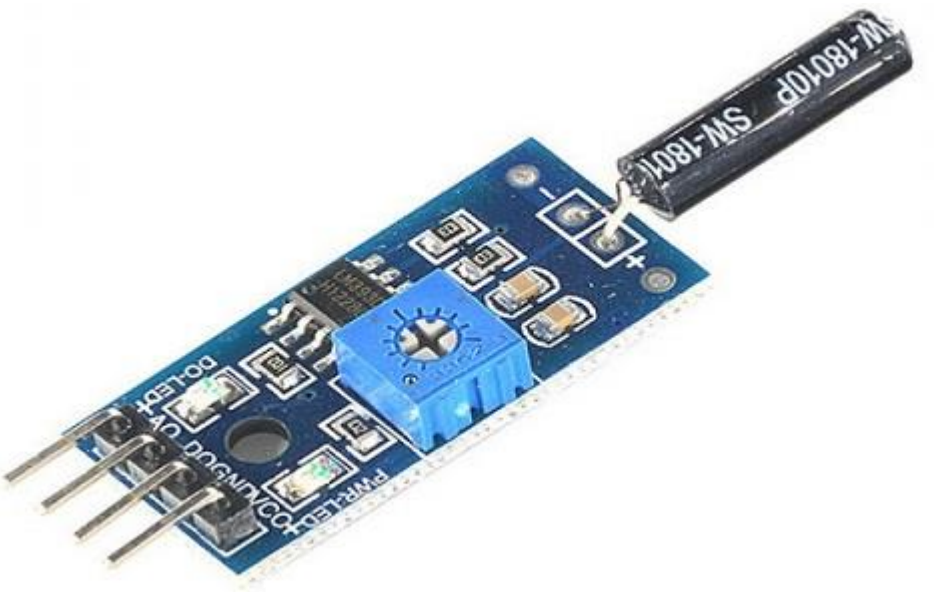


Fig 2.8.1: Vibrator sensor

The sensitivity of these sensors normally ranges from 10 mV/g to 100 mV/g, and there are lower and higher sensitivities are also accessible. The sensitivity of the sensor can be selected based on the application. So, it is essential to know the levels of vibration amplitude range to which the sensor will be exposed throughout measurements.

Thus, this is all about vibration sensor. From the above information, finally, we can conclude that vibration is a difficult measurement which includes different parameters. Based on the goals of vibration measurement, the measurement technologies have benefits and drawbacks. These sensors are mainly used for measuring, analyzing, displaying, proximity, acceleration, displacement, etc.

2.9 BUZZER

Buzzer is a kind of voice device that converts audio model into sound signal. It is mainly used to prompt or alarm. According to different design and application, it can produce music sound, flute sound, buzzer, alarm sound, electric bell and other different sounds.

2.9.1 WORKING PRINCIPLE

The working principle of a buzzer is based on the conversion of electrical energy into sound energy. Buzzers typically operate using electromagnetic, piezoelectric, or mechanical vibration mechanisms.

Fig 2.9.1: Buzzer

Typical applications include siren, alarm device, fire alarm, air defense alarm, burglar alarm, timer, etc. It is widely used in household appliances, alarm system, automatic production line, low-voltage electrical equipment, electronic toys, game machines and other products and industries.

2.10 RELAY

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

A relay is an electromagnetic switching device used to control the flow of electricity in a circuit. It acts as an automatic switch that opens or closes circuits in response to electrical signals. Relays are widely used in applications where it is necessary to control a high-power or high-voltage circuit with a low-power signal, ensuring electrical isolation and safety.

2.10.1 WORKING PRINCIPLE

A relay operates on the principle of electromagnetism. When an electrical current flows through the electromagnetic coil, it generates a magnetic field. This magnetic field pulls the armature, a movable metal component, towards the coil. The movement of the armature either closes or opens the contacts in the circuit, depending on the relay's configuration (normally open or normally closed).

When the current to the coil is turned off, the magnetic field collapses, and a spring mechanism returns the armature to its original position, resetting the contacts to their default state. This allows the relay to control the flow of current in one circuit using a separate low-power signal, ensuring electrical isolation between the controlling and controlled circuits.

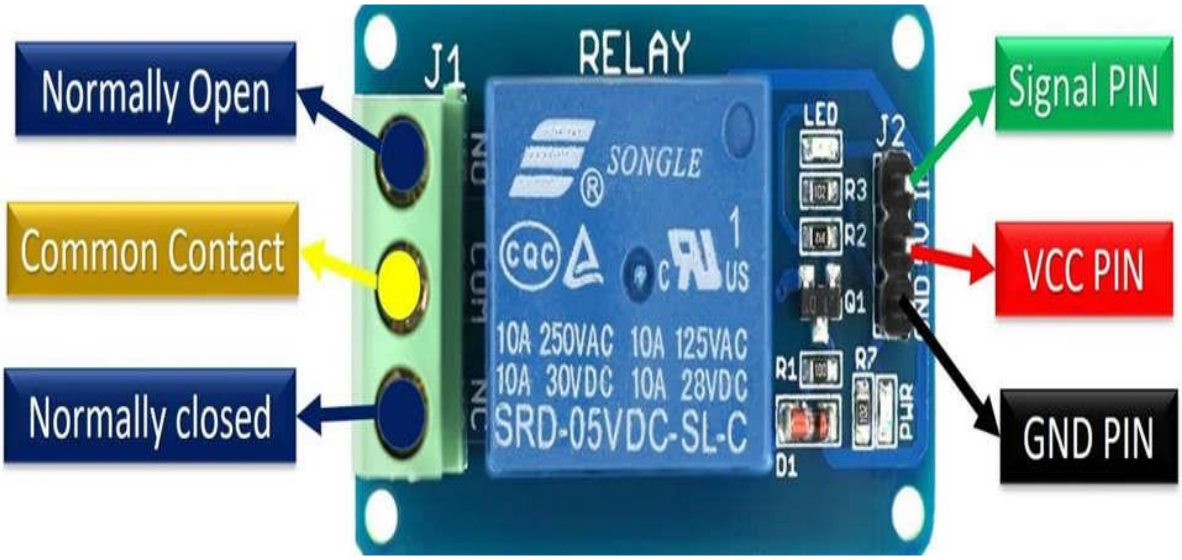


Fig 2.10.1: Relay

2.10.2 Pin configuration

SIGNAL PIN – It is a control pin. This signal will energize the relay coil to make contact with the common terminal with the normally open terminal. This pin can be active low or active high.

VCC PIN – That means it requires 5V DC to operate. Hence, connect the 5v DC power supply to this pin.

GND PIN – connected to ground.

COMMON PIN- This terminal is connected with the load that we want to switch with the relay module.

NC PIN – It is normally connected with the COM pin and forms a closed circuit. But this normally

closed connection breaks when the relay is activated by applying an active high or active low signal.

NO PIN – This pin is normally open unless we apply an activation signal to the signal pin of the 5V single channel relay. when the relay gets activated common pin disconnects the NC pin and connects the NO pin.

2.10.3 Internal circuit

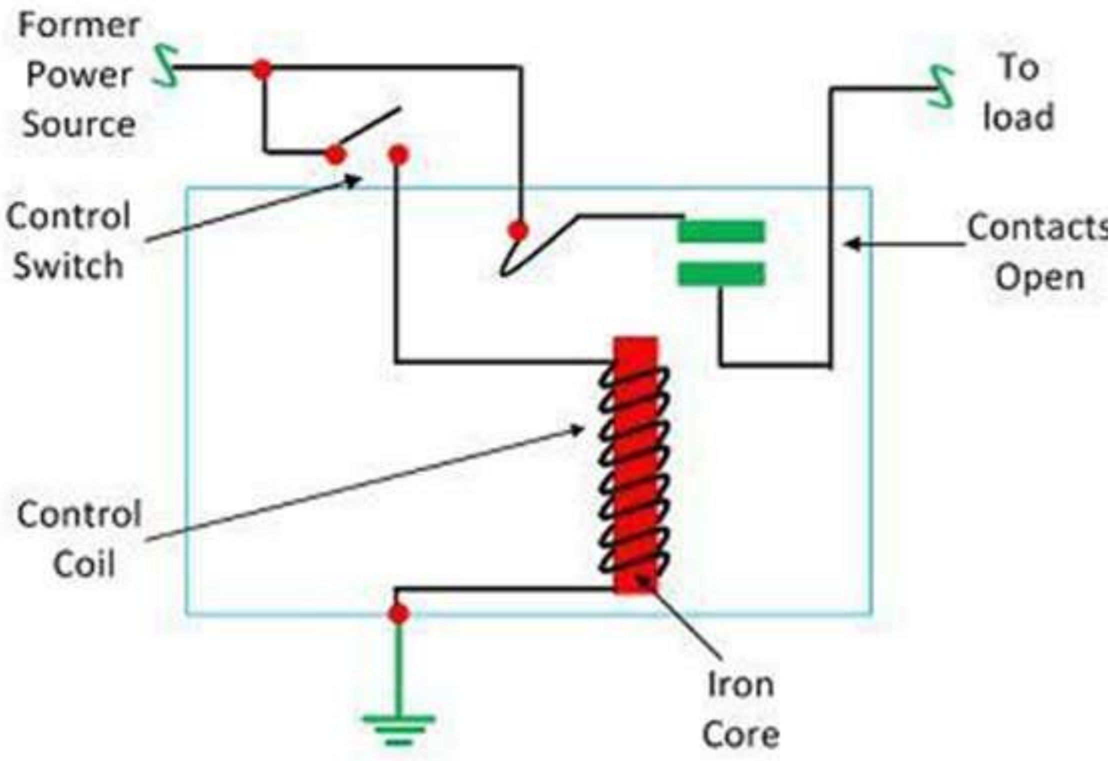


Fig 2.10.3. Internal circuit

- It works on the principle of electromagnetic attraction. When the circuit of the relay senses the fault current, it energizes the electromagnetic field which produces the temporary magnetic field.
- The inner section of the relay has an iron core which is wound by a control coil. The power supply is given to the coil through the contacts of the load and the control switch. The current flows through the coil and produces the magnetic field around it.
- Due to this magnetic field, the upper arm of the magnet attracts the lower arm. Hence close the circuit, which makes the current flow through the load. If the contact is already closed, then it moves oppositely and hence opens the contacts.

2.11 DC PUMP

A DC water pump is an electric pump with low voltage. It is quiet and uses little power. You can use it for many things, like cars, homes, and water wells. There are many DC water pumps in the market. The DC Fountain Pump VP30 is a small five-volt pump for solar fountains. You can also find DC Submersible Pumps VP30A and VP40A for aquariums, tabletop crafts, and fish tanks.

Fig 2.11: DC Pump

When selecting a small water pump, it’s important to check its power consumption. Most pumps draw up to 9.5 amps, so choosing one that matches your power needs is important. Check that the motor is corrosion-resistant and can withstand multiple freeze-thaw cycles. In addition, the housing should be intact and have no cracks. Finally, choose a pump that comes with a warranty. If a mini water pump fails after just a few years, it may be good to look for a brand that offers a warranty. After-sales support can also help you replace any parts if needed. Discover everything you need to know about DC pumps in our comprehensive guide, where we delve into their efficiency, applications, and advantages.

A DC water pump runs on a battery, power supply, or solar panel. These pumps are small and easy to use. Their primary use is to circulate, pressurize, and emulsify liquids. They are particularly useful in environments where water is in short supply. Listed below are some other uses for DC submersible pumps. This article will explain the most common applications for DC water pumps.

2.11.1 Working Principle

Here’s the ultimate guide if you’re considering buying a DC water pump for your RV. First, consider the size of your RV. A 12-volt DC water pump will run on its built-in 12-volt power system. A large commercial pump requires a 230-volt power supply, while a 12-volt model requires only a 120-volt power source. Both options will require a **mini water pump**.

You’ll need to ensure the pump can handle the required amount of water to get the proper flow. Ensure the pipe uses PVC that is safe for drinking water. A pump designed for 24 Volts may have a 3/4-inch outlet. For 12 Volts, you’ll need a pump compatible with the smaller pipe size. The difference in size will affect how much water a pump can draw from the tank. Fortunately, there are many reasons to consider a DC water pump. One major advantage of a **DC submersible pump** is its installation flexibility. You can easily install a DC submersible pump anywhere if you have an existing power source. Manufacturers can even install them without needing a dedicated electrical grid. This will increase the flexibility of installation.

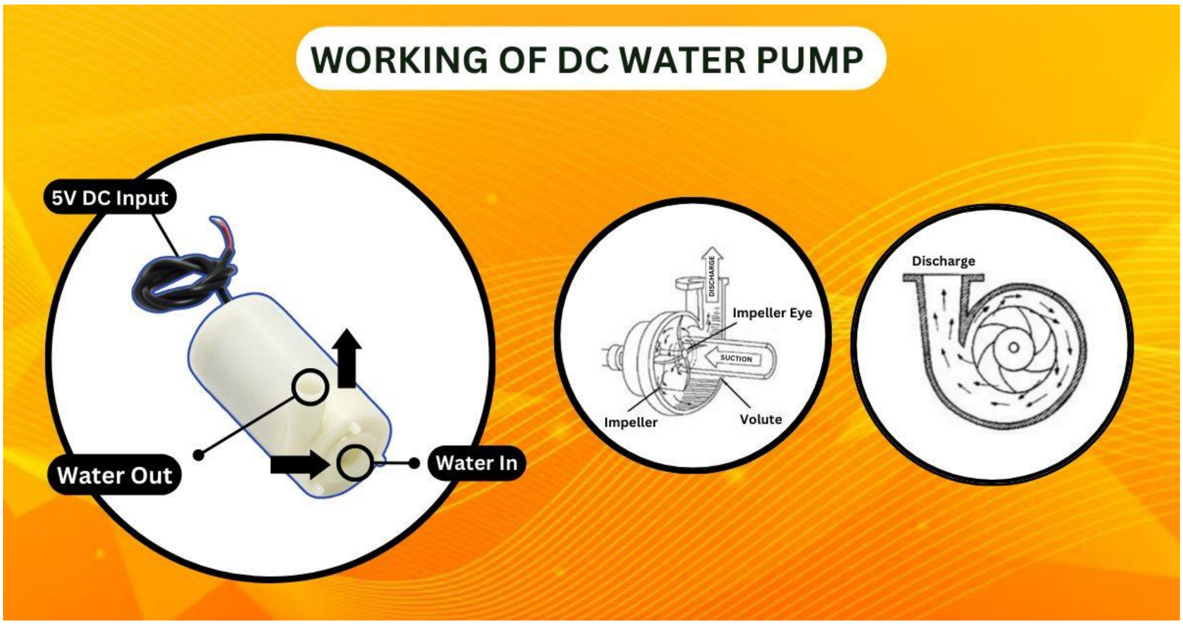


Fig 2.11.1 Working Principle

2.12 BATTERY

Solar panels allow you to generate your own electricity and can reduce your utility bills, but they won't do so at night or during a power outage. But a solution to both issues is getting more common and more affordable.

Solar batteries can be a valuable part of a residential solar system. They provide reliability, safety, economic benefits, and comfort for your family. Batteries allow you to use solar power 24/7, maximize

savings from your system, and have reliable power during bad weather and grid outages.

If you're looking for the answer to "How do solar batteries work?" this article will explain what a solar battery is, solar battery science, how solar batteries work with a residential solar power system, and the overall benefits of energy storage within your own system.

Fig 2.12 Battery

2.12.1 Working principle

A battery operates on the principle of electrochemical reactions, converting chemical energy into electrical energy. It consists of two electrodes: the anode (negative) and the cathode (positive), separated by an electrolyte. During discharge, oxidation occurs at the anode, releasing electrons and generating positive ions. The electrons flow through an external circuit, producing an electric current, while the ions travel through the electrolyte to the cathode, where a reduction reaction takes place. This flow of electrons powers devices connected to the battery.

In rechargeable batteries, the process is reversible during charging, with an external power source driving electrons back to the anode to restore chemical energy. The battery's voltage is determined by the difference in electrochemical potential between the electrodes, while its capacity depends on the active material and the number of electrons involved in the reactions. Various types of batteries, such as primary (non-rechargeable) and secondary (rechargeable), use different chemistries, influencing their energy density, lifespan, and efficiency. Rechargeable batteries like lithium-ion are widely used in applications ranging from portable electronics to electric vehicles due to their high performance and reusability.

2.13 ARDUINO UNO

The Arduino Uno is a widely used microcontroller development board that simplifies working with electronics and programming. It is built around the ATmega328P microcontroller, an 8-bit device that serves as the brain of the board. A microcontroller is essentially a small computer on a single integrated circuit, containing a processor, memory, and input/output peripherals. The Arduino Uno makes it easy for hobbyists, students, and professionals to use this microcontroller by providing a user-friendly platform.

The board includes features such as digital and analog input/output pins, a USB connection for programming and communication with a computer, and a power supply interface. It is programmed using the Arduino IDE, a straightforward software environment, which supports simplified coding in C/C++.

2.13.1 Arduino Uno Technical Specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table 2.13.1: Arduino Uno Technical Specifications

Fig 2.13.1: Arduino Uno

2.13.2 How to use Arduino Board?

The 14 digital input/output pins can be used as input or output pins by using pin Mode (), digital Read () and digital Write () functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 K Ohms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

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

PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analog Write () function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

In-built LED Pin 13: This pin is connected with a built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference () function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below.

- **AREF:** Used to provide reference voltage for analog inputs with analog Reference () function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

2.13.3 ATmega328P Microcontroller

The ATmega328P is an 8-bit microcontroller by Microchip Technology, widely used for its simplicity and versatility. It features 32 KB of flash memory, 2 KB of SRAM, 1 KB of EEPROM, and operates at up to 20 MHz with a voltage range of 1.8V to 5.5V. With 23 I/O pins, a 10-bit ADC, three timers, and support for UART, I2C, and SPI communication, it is well-suited for various tasks.

Its power-saving modes and robust interrupt handling make it efficient, while its compatibility with platforms like **Arduino** enhances accessibility for developers.

2.13.4 ATmega328 Pin Configuration

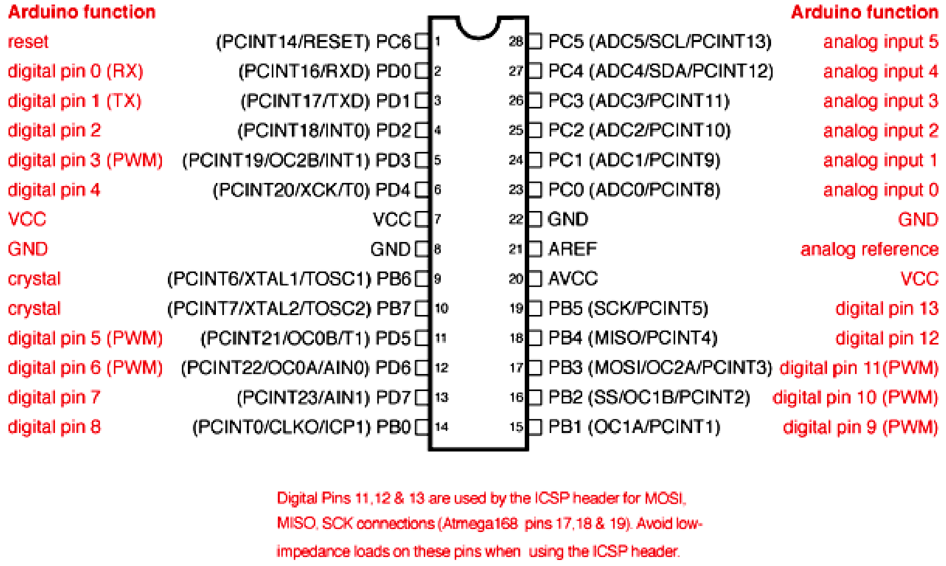


Fig 2.13.4: ATmega328 Pin Configuration diagram

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

Table 2.13.4: Each pin description

2.13.5 Communication

Arduino can communicate with a computer, other Arduino boards, or microcontrollers using its AT-mega328P microcontroller, which supports UART TTL serial communication via pins 0 (Rx) and 1 (Tx). An ATmega16U2 chip channels this over USB, appearing as a virtual COM port on a computer without needing external drivers (except a .inf file for Windows). The Arduino software has a Serial Monitor for exchanging text data, with RX and TX LEDs flashing during USB communication. It also supports I2C (using the Wire library) and SPI communication, while the Software Serial library enables serial communication on any digital pin.

2.13.6 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/Genuine 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 blink in.

2.14 GLOBAL SYSTEM FOR MOBILE (GSM)

A GSM modem is a specialized modem that uses wireless networks, like a mobile phone, to send and receive data. It requires a SIM card and can connect to a computer via USB or directly as a plug-in device. Unlike traditional modems that use telephone lines, GSM modems communicate through radio waves. GSM is a widely used mobile communication system operating at 850MHz, 900MHz, 1800MHz, and 1900MHz for voice and data transmission.

2.14.1 Features of The SIM900A GSM module

- GSM Communication
- SMA Antenna
- SIM Card Slot
- Serial Communication
- Voltage Input

- Digital and Analog I/O Pins
- SMS and Call Functions
- Data Transmission

Fig 2.14.1: GSM Module

2.14.2 GSM MODEM APPLICATIONS

Fig 2.14.2: GSM Modem Applications

2.14.3 Technical Details

- Single supply voltage: 3.4V – 4.5V
- Power saving mode: Typical power consumption in SLEEP mode is 1.5mA
- Frequency bands: SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can search the two frequency bands automatically. The frequency bands also can be set by AT command.
- GSM class: Small MS
- GPRS connectivity: GPRS multi-slot class 10 (default), GPRS multi-slot class 8 (option)

CHAPTER-3

SOFTWARE SPECIFICATIONS

3.1 INTRODUCTION TO ARDUINO IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

3.2 KEY FEATURES OF ARDUINO IDE

Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hard-ware (called a programmer) in order to load a new code onto the board. You can simply use a USBcable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

Finally, Arduino provides a standard form factor that breaks the functions of the micro- controller into a more accessible package.

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.

3.3 STEPS TO UPLOAD THE PROGRAM IN 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.

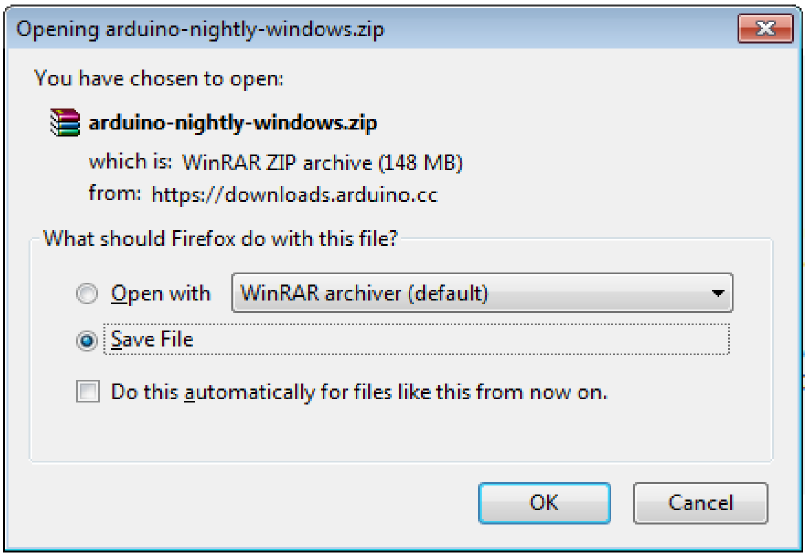
Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable.

In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega2560, 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.

Fig:3.3(a) USB Cable

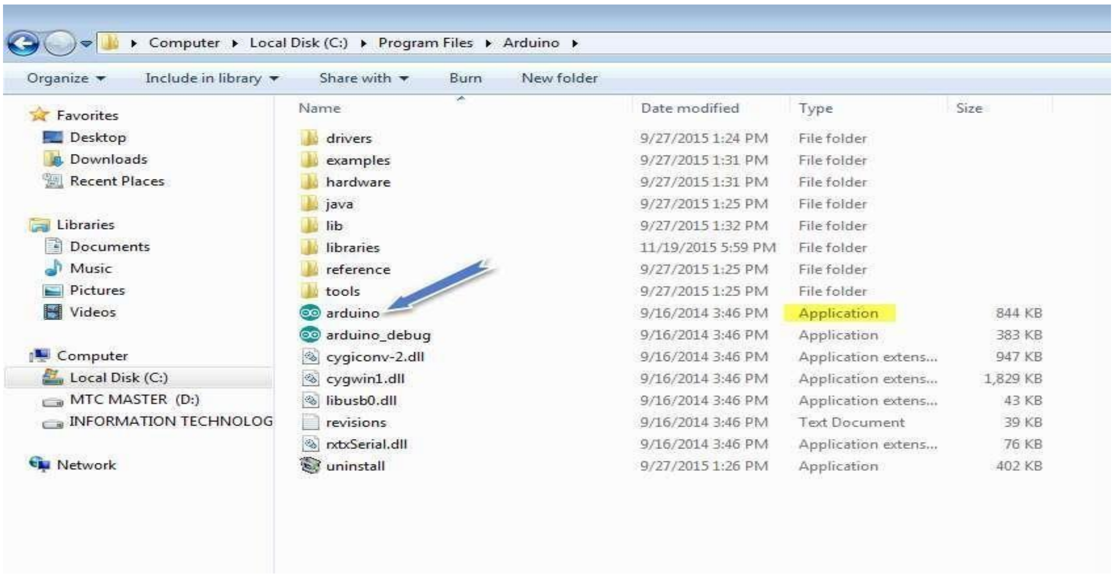
Step 2: Download Arduino IDE Software.

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.

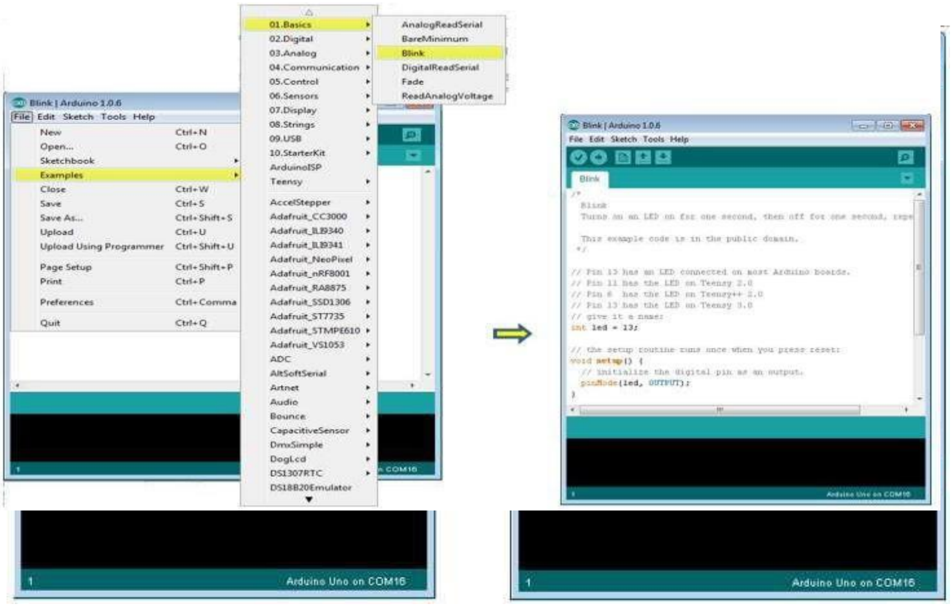
After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double click the icon to start the IDE.

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



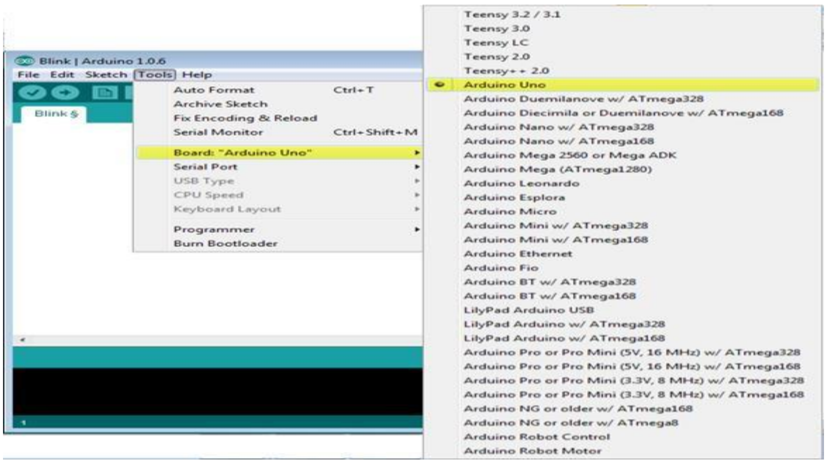
To open an existing project example, select File -> Example -> Basics -> Blink.

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.

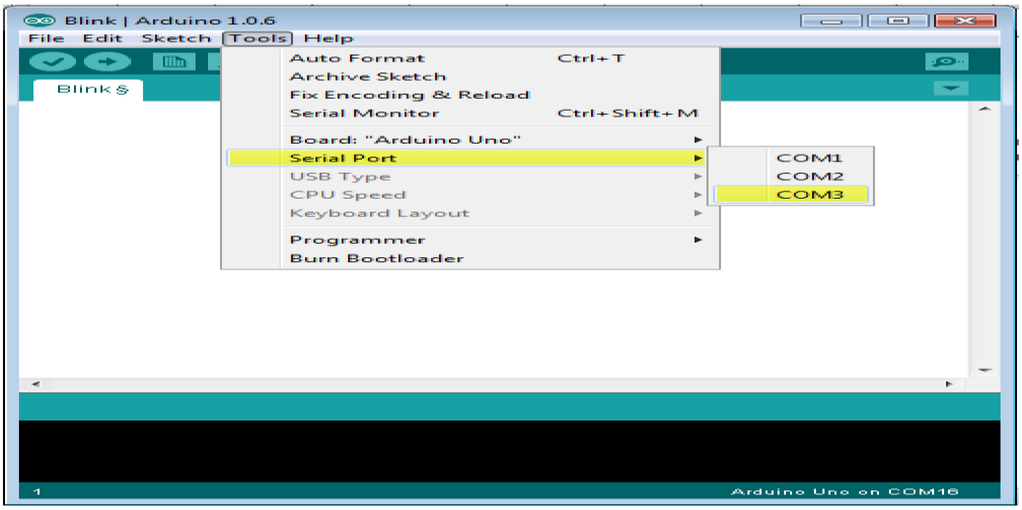
Go to Tools -> Board and select your board



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 serialport



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 sketches. 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.

3.4 ARDUINO PROGRAMMING STRUCTURE

In this 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.

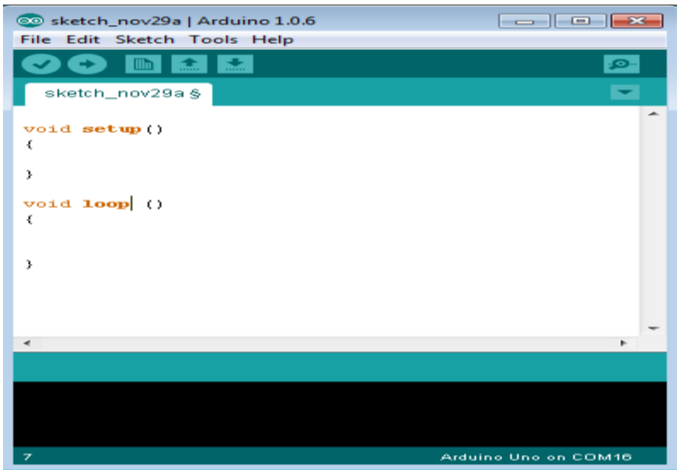
Sketch: The first new terminology is the Arduino program called “sketch”.

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 RE-
TURN
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 secutively, allowing your program to change and respond. Use it to actively control the Arduino board.

INPUT OUTPUT
RETURN

CHAPTER-4

IMPLEMENTATION

4.1 SCHEMATIC DIAGRAM

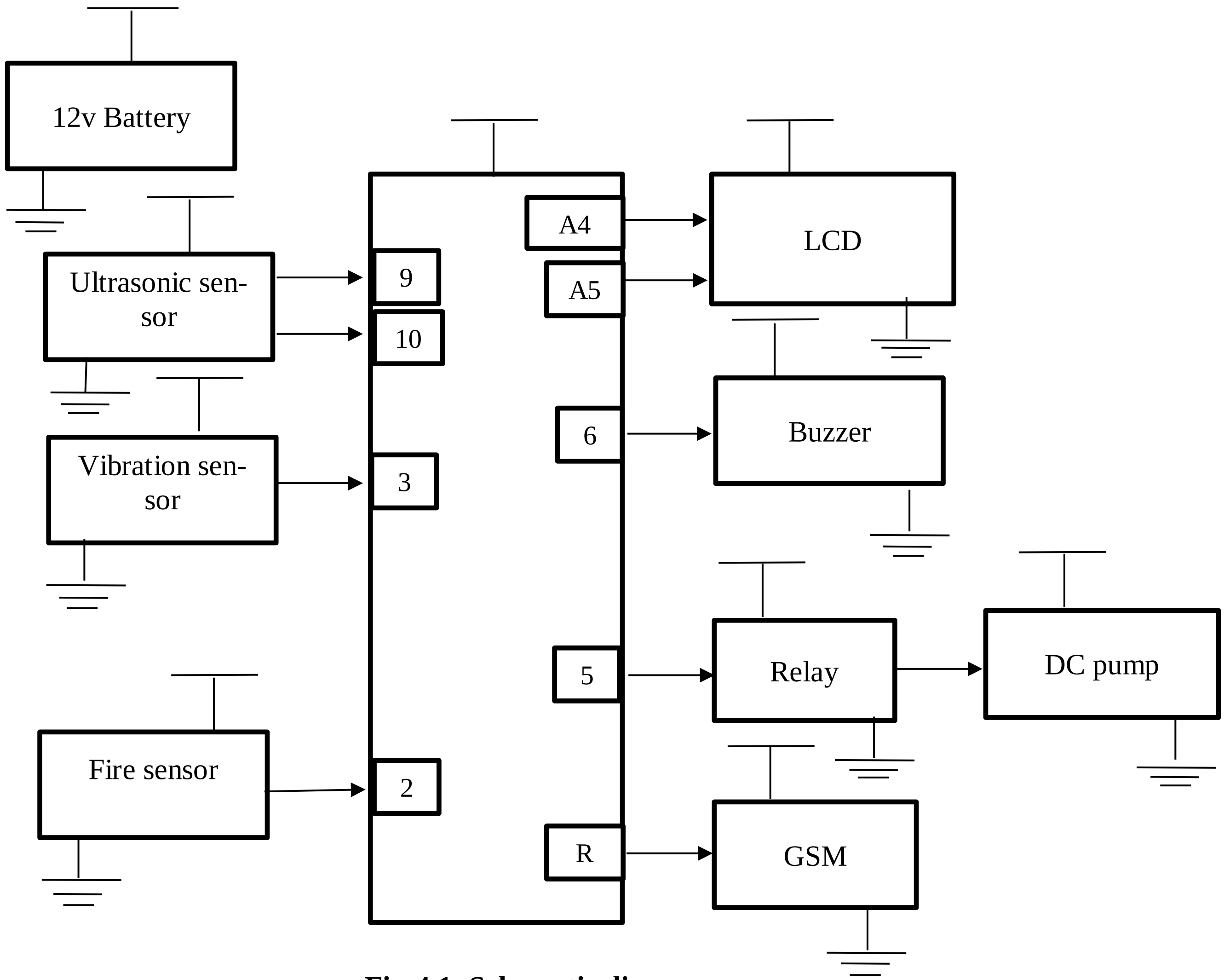


Fig:4.1: Schematic diagram

4.2 CODE IMPLEMENTATION

```
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0X27,16,2);
#define Fire_Sensor 2
#define Vibration_sensor 3
#define Relay 5
#define Buzzer 6
```

```
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distance;
void setup ()

{
  // put your setup code here, to run once:
 Serial.begin(9600);
 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
 pinMode(echoPin, INPUT); // Sets the echoPin as an Input
 pinMode(Fire_Sensor,INPUT);
 pinMode(Vibration_sensor,INPUT);
 pinMode(Relay,OUTPUT);
 pinMode(Buzzer,OUTPUT);
 digitalWrite(Relay ,HIGH);
 digitalWrite(Buzzer ,LOW);
 lcd.init();
 lcd.begin(16,2);
 lcd.backlight();
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("FOREST SECURITY");
 lcd.setCursor(0,1);
 lcd.print("TO SAVE TREES");
 delay(2000);
}

void loop()
{
  // put your main code here, to run repeatedly:

 digitalWrite(trigPin, LOW);
 delayMicroseconds(2); // Sets the trigPin on HIGH state for 10 micro seconds
```

```
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);// Calculating the distance
distance= duration*0.034/2;// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Distance:");
lcd.setCursor(0,1);
lcd.print(distance);
delay(1000);

int Fire_Sensor_state = digitalRead(Fire_Sensor);
Serial.print("Fire_Sensor:");
Serial.print(Fire_Sensor_state);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Fire_state:");
lcd.setCursor(0,1);
lcd.print(Fire_Sensor_state);
delay(1000);

int Vibration_sensor_state = digitalRead(Vibration_sensor);
Serial.print("Vibration_sensor_state:");
Serial.print(Vibration_sensor_state);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Vibration_state:");
lcd.setCursor(0,1);
lcd.print(Vibration_sensor_state);
delay(1000);
```

```
 if(Fire_Sensor_state == 0)
 {
  digitalWrite(Relay ,LOW);
  digitalWrite(Buzzer ,HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("FIRE DETECTED");
  SendMessage();
  delay(1000);
 }
 else if(Vibration_sensor_state == 1)
 {
  digitalWrite(Buzzer ,HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("VIBRATION DETECTED");
  SendMessage1();
  delay(1000);
 }

 else
 {
  digitalWrite(Relay ,HIGH);
  digitalWrite(Buzzer ,LOW);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NORMAL CONDITION");
  delay(1000);
 }
}
void Send Message()
{
 Serial.println("AT");
```

```
  delay(500);
  Serial.println("ATE0");
  delay(500);
  Serial.println("AT + CMGF = 1");
  delay(500);
  Serial.println("AT + CMGS = \"+919133758683\"\r");
  delay(500);
  Serial.println("FIRE DETECTED");
  delay(1000);
  Serial.write(26);
  Serial.println("Message sent..");
  delay(800);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MSG SENT");
  delay(1000);
}
void SendMessage1()
{
  Serial.println("AT");
  delay(500);
  Serial.println("ATE0");
  delay(500);
  Serial.println("AT + CMGF = 1");
  delay(500);
  Serial.println("AT + CMGS = \"+919133758683\"\r");
  delay(500);
  Serial.println("VIBRATION DETECTED");
  delay(1000);
  Serial.write(26);
  Serial.println("Message sent..");
  delay(800);
  lcd.clear();
  lcd.setCursor(0,0);
```

```
lcd.print("MSG SENT");
delay(1000);

  }
```

4.3 FLOW CHART

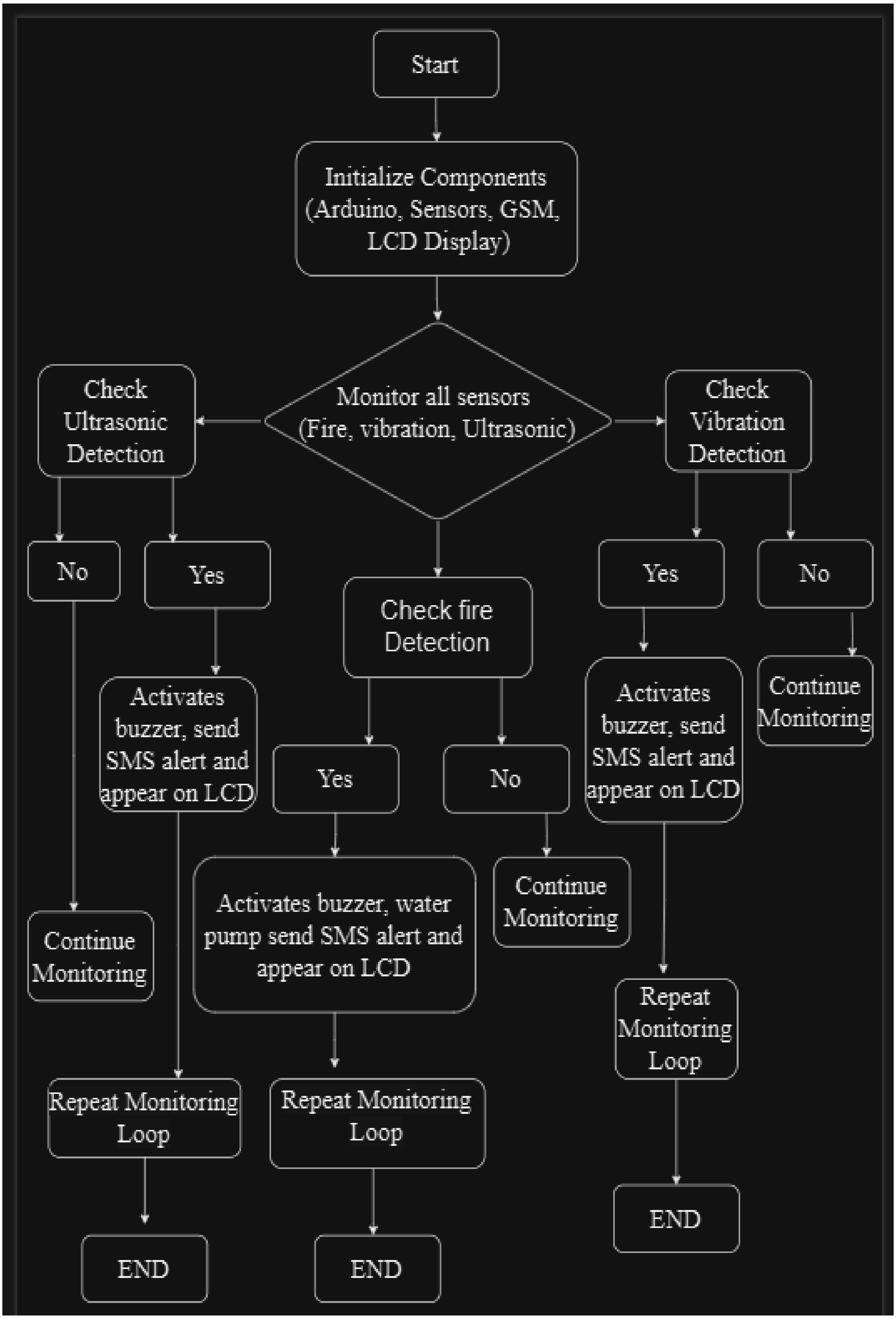


Fig:4.3 Flow chart

CHAPTER 5

SIMULATION AND DESIGN

5.1 OUTPUT SCREENSHOTS

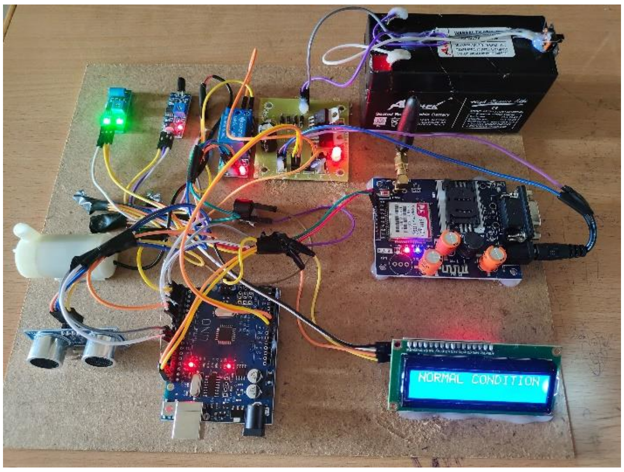


Figure shows that kit ready for monitoring

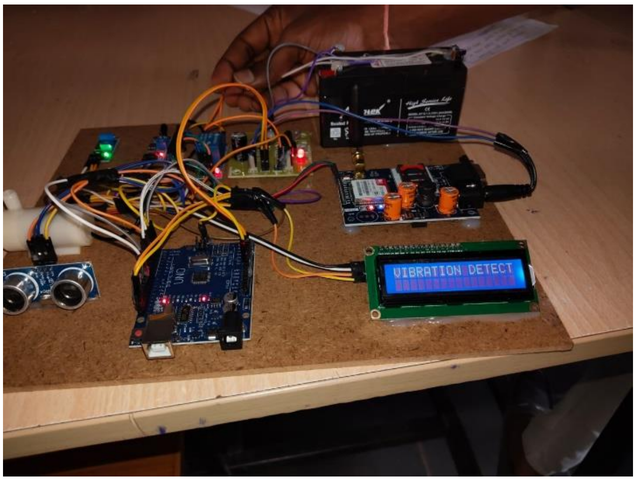


Figure shows vibrations detected and MSG sent

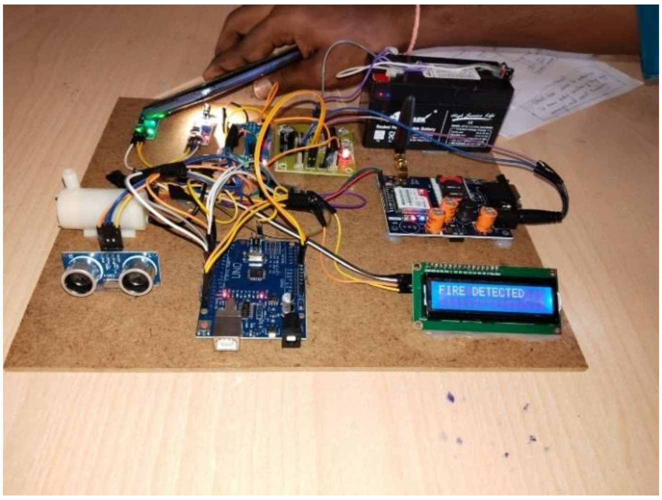


Figure shows Fire detected and MSG sent

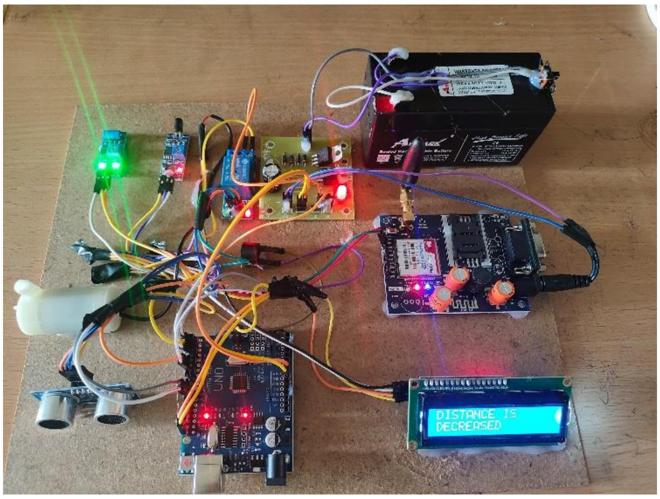


Figure shows distance is detected and MSG sent

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The “Forest Security to Save Trees” project offers a practical and efficient solution for protecting forest resources from threats such as illegal logging, deforestation, and fire hazards. By integrating advanced embedded systems with sensors like fire, vibration, and ultrasonic sensors, along with GSM communication modules, the system ensures real-time monitoring and immediate response to unauthorized activities or potential dangers. The Arduino-based microcontroller processes the sensor data and triggers alerts through SMS notifications, enabling swift action by forest authorities. This automated approach minimizes human dependency on manual patrolling, which is often inefficient and costly, while providing accurate and consistent results across large forest areas. The system’s ability to detect fire incidents and illegal activities significantly reduces the time to respond, thereby mitigating damage and preserving valuable and endangered tree species. Overall, this project demonstrates a robust and scalable solution that not only safeguards forest ecosystems but also contributes to environmental sustainability and biodiversity conservation.

6.2 FUTURE SCOPE

The "Forest Security to Save Trees" project has significant potential for future advancements to enhance its efficiency and scalability. Integrating **Internet of Things (IoT)** technology with cloud-based platforms can enable remote real-time monitoring and centralized data storage, allowing authorities to analyze patterns and predict threats more effectively. Incorporating **solar-powered energy systems** will ensure the system's continuous operation in remote forest areas, making it self-sustainable and reducing dependency on external power sources. Additionally, the use of **Artificial Intelligence (AI)** and **Machine Learning (ML)** can enable predictive analysis to identify high-risk areas and anticipate potential fire outbreaks or illegal activities based on historical and real-time data. The project can also be expanded by integrating **drone technology** for aerial surveillance and **advanced sensor networks** for monitoring additional parameters such as smoke, humidity, air quality, and wildlife movement. Furthermore, developing a **mobile application** for forest authorities will provide an accessible platform for real-time alerts, geolocation tracking, and faster response to incidents. With these enhancements, the system can offer a comprehensive, automated, and scalable solution for forest protection, contributing to global environmental sustainability and conservation efforts.

CHAPTER-7

REFERENCES

1. Yichang, China; Guangyu He; Junli Wan —Research on Zigbee wireless communication technology Wei Wang, In Electr.Eng. & Renewable Energy Sch., China Three Gorges University.
2. Chong gang Wang, Tao Jiang, Qian Zhang —ZigBee
3. Network Protocols and Applications
4. ZigBee specification version 2006, ZigBee document 064112, 2006. [5]ZigBee Alliance, ZigBee Specification. Version 1.0 ZigBee Document 053474r06.
5. Jiang, Y., Cao, J., & Du, Y. —Unmanned air vehicle landing based on Zigbee and vision guidance‖ WCICA 2006, 2, 10310 - 10314.
6. Muhammad Ali Mazidi, Roln D.Mckenley, "The 8051
7. Microcontroller and embedded system using assembly & C‖.
8. Hua Qian —API: GSM/GPRS Modem User Interface.
9. University of Texas at Dallas University of Texas at Dallas, 2007.