Bachelor of Technology in Electronics & Communication Engineering PRAVEEN MANNE 21E15A0410 Under the Supervision of

Thesis · July 2024

DOI: 10.13140/RG.2.2.26324.62083

Praveen Manne
Bharat Institute of Engineering and Technology

TRIO CLEANING ROBOT USING ARDUINO UNO ARTIFICIAL INTELLIGENCE

Industrial Oriented Major Project Th

Submitted to

Jawaharlal Nehru Technological University Hyderabad

in Partial Fulfilment of the Requirements

for the Award of the Degree of

Bachelor of Technology

in

Electronics & Communication Engineering

Submitted by

PRAVEEN MANNE **21E15A0410**

Under the Supervision of

Mr. Dasari Sankara Reddy

Associate Professor

Department of Electronics & Communication Engineering

BHARAT INSTITUTE OF ENGINEERING & TECHNOLOGY

Mangalpally(v), Ibrahimpatnam (M), Rangareddy -5015

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to Sri **CH. Venugopal Reddy**, Chairman & Secretary of BIET, for providing congenial atmosphere and encouragement.

We would like to thank **Prof. G. Kumaraswamy Rao**, Former Director & O.S. of DLRL Ministry of Defence, Sr. Director, BIET, **Dr V. Kannan**, Principal, and Dr Sanjay K, College Admi In charge for having provided all the facilities and support.

We would like to thank our Department Coordinator/ HOD, **Mr. Dr. H.P.Singh**. Admin In charge, Electronics and Communication Engineering.

We are thankful to our Project Coordinator **Mr. Satyabrata Singha**, Assistant Professor, Electronics and Communication Engineering for his support and cooperation throughout the process of this project

We are thankful to our Guide **Mr. Dasari Sankara Reddy.**, Electronics and Communication Engineering Dept., for his sustained inspiring Guidance and cooperation throughout the process of this project.

We express our deep sense of gratitude and thanks to all the Teaching and Non-Teaching staff of our college who stood with us during the project and helped us to make it a successful venture. We place highest regards to our Parent, our Friends and Well-wishers who helped a lot in making the report of this project.

ABSTRACT

This project involves creating a versatile cleaning robot using Arduino, capable of sweeping, mopping, and drying functionalities. The system integrates motors for sweeping and mopping, obstacle detection sensors for navigation, and a drying mechanism, such as a fan or a drying cloth. The Arduino board serves as the central control unit, managing the interactions between various components. The connections are established to enable efficient power distribution, motor control, sensor input, and optional remote control through interface like Bluetooth or Wi-Fi. This abstract outline the fundamental aspects of the robot's design and functionality.

Outlines to design a versatile cleaning robot using Arduino.

- •Arduino Control Unit.
- •Motorized Sweeping and Mopping Mechanisms.
- •Obstacle Detection Sensors.
- •Drying Mechanism.
- •Efficient Power Distribution.
- •Optional Remote-Control Interfaces

TABLE OF CONTENTS

ABBREVATION

LCD: LIQUIDE CRYSTAL DIODE

LDR: LIGHT DEPENDENT RESISTOR

CPU: CENTRAL PROCESSING UNIT

LED: LIGHT EMITTING DIODE

RAM: RANDOM ACCESS MEMORY

USB: UNIVERSAL SERIAL BUS

ARDUINO IDE: Arduino Integrated Development Environment

DC: Direct Current

BLDC: Brushless Direct Current

CHAPTER 1

INTRODUCTION

A cleaning system refers to a methodical approach or set of procedures designed to remove dirt, dust, germs, and other undesirable substances from surfaces, objects, or environments. Cleaning systems are employed in various contexts, including household cleaning, commercial cleaning, industrial cleaning, healthcare facilities, and more. Here are some key aspects typically associated with cleaning systems:

- Cleaning Agents: These are substances or chemicals used to loosen and remove dirt, stains, and contaminants. Cleaning agents can include detergents, solvents, disinfectants, and abrasive cleaners.
- Tools and Equipment: Cleaning systems often require specialized tools and equipment tailored to specific cleaning tasks. These may include vacuum cleaners, mops, brooms, scrub brushes, microfiber cloths, sponges, and steam cleaners, among others.
- Techniques and Procedures: Effective cleaning systems involve following established techniques and procedures to ensure thorough and consistent cleaning. This may include dusting, sweeping, mopping, scrubbing, wiping, and sanitizing surfaces using appropriate methods.
- Safety Measures: Safety is paramount in cleaning systems, especially when using potentially hazardous chemicals or operating heavy machinery. Proper training, protective gear, and adherence to safety protocols are essential components of any cleaning system.
- Environmental Considerations: Many modern cleaning systems emphasize environmentally friendly practices, such as using eco-friendly cleaning agents, reducing water consumption, and minimizing waste generation.

- Regulatory Compliance: Depending on the setting, cleaning systems may need to adhere to various regulations and standards set by government agencies or industry organizations to ensure hygiene, safety, and sanitation.
- Quality Control: Regular inspections and quality control measures are often implemented to assess the effectiveness of cleaning systems and ensure that cleanliness standards are met consistently.

In India, cleaning systems encompass a wide range of practices and methods tailored to address the unique environmental, cultural, and economic factors of the country. Here are some key aspects of cleaning systems in India:

- Traditional Practices: India has a rich tradition of manual cleaning methods, including sweeping with brooms made from natural materials like grass or coconut fibers. Traditional practices often emphasize the importance of cleanliness for spiritual, cultural, and health reasons.
- Modern Cleaning Equipment: With urbanization and industrialization, there has been a proliferation of modern cleaning equipment and technology in India. This includes vacuum cleaners, floor scrubbers, high-pressure washers, and mechanized sweepers, which are increasingly used in commercial and industrial settings.
- Household Cleaning Products: Indian households typically use a variety of cleaning products, including detergents, floor cleaners, dishwashing liquids, and multipurpose

cleaners. Many Indian households also rely on natural or homemade cleaning solutions, such as vinegar, baking soda, and lemon juice, due to cultural preferences and affordability.

- Public Sanitation Initiatives: The Indian government has launched several sanitation initiatives aimed at improving public hygiene and cleanliness. One notable example is the Swachh Bharat Abhiyan (Clean India Mission), which aims to eliminate open defecation, improve waste management, and promote cleanliness and sanitation practices across the country.
- Community-Based Cleaning Programs: In many Indian communities, local residents participate in community-based cleaning programs such as "shramdaan" (voluntary labor) and "clean-up drives" to clean streets, public spaces, and water bodies. These initiatives often involve collaboration between government agencies, NGOs, and community groups.
- Waste Management: Waste management is a significant aspect of cleaning systems in India. Efforts to improve waste collection, segregation, recycling, and disposal are essential for maintaining cleanliness and mitigating environmental pollution.

- Challenges and Opportunities: Despite progress, India faces challenges such as inadequate infrastructure, limited access to clean water and sanitation facilities, and cultural attitudes toward cleanliness. However, there are also opportunities for innovation and collaboration to address these challenges and improve cleaning systems nationwide.

Overall, cleaning systems in India are diverse and evolving, encompassing a mix of traditional practices, modern technology, public initiatives, and community participation aimed at promoting cleanliness, hygiene, and environmental sustainability.

CHAPTER – 2
LITERATURE REVIEW

2.1 HARDWARE DESCRIPTION

A hardware description typically refers to a detailed explanation or documentation of the physical components, specifications, and characteristics of a particular piece of hardware, such as a computer, electronic device, or machinery. This description may include information such as:

- Components: A list and description of the various physical parts that make up the hardware, including processors, memory modules, storage devices, input/output ports, and other integrated circuits or electronic components.

- Specifications: Detailed technical specifications for each component, including parameters such as speed, capacity, voltage, power consumption, dimensions, weight, and interface standards (e.g., USB, HDMI, Ethernet).

- Architecture: An overview of the hardware's overall architecture and how its components are interconnected and function together to perform specific tasks or operations.

- Manufacturing Details: Information about the manufacturing process, materials used, assembly methods, and quality control measures employed in producing the hardware.

- Performance Characteristics: Data and metrics related to the hardware's performance, including processing speed, throughput, latency, response times, and other relevant performance indicators.

- Compatibility: Compatibility information indicating which operating systems, software applications, peripherals, and other hardware components are compatible with the described hardware.

- Environmental Considerations: Details about environmental factors such as operating temperature range, humidity tolerance, and resistance to shock, vibration, dust, and other environmental hazards.

- Safety and Regulatory Compliance: Information regarding safety features, certifications, and regulatory compliance standards that the hardware conforms to, ensuring safe and legal operation.

- Maintenance and Support: Guidelines and recommendations for maintenance, troubleshooting, and technical support services available for the hardware, including warranty coverage and repair procedures.

Overall, a hardware description provides a comprehensive understanding of the physical attributes, capabilities, and specifications of a particular piece of hardware, serving as a valuable resource for engineers, technicians, users, and other stakeholders involved in its design, development, deployment, and maintenance.

CHAPTER – 3

HARDWARE AND SOFTWARE DESCRIPTION

HARDWARE COMPONENTS

The following are the list of hardware components used in project-

1. Arduino UNO
2. L293D Motor Drive
3. UltraSonic Sensor
4. Bluetooth Module(HC-05)

3.1 ARDUINO UNO:

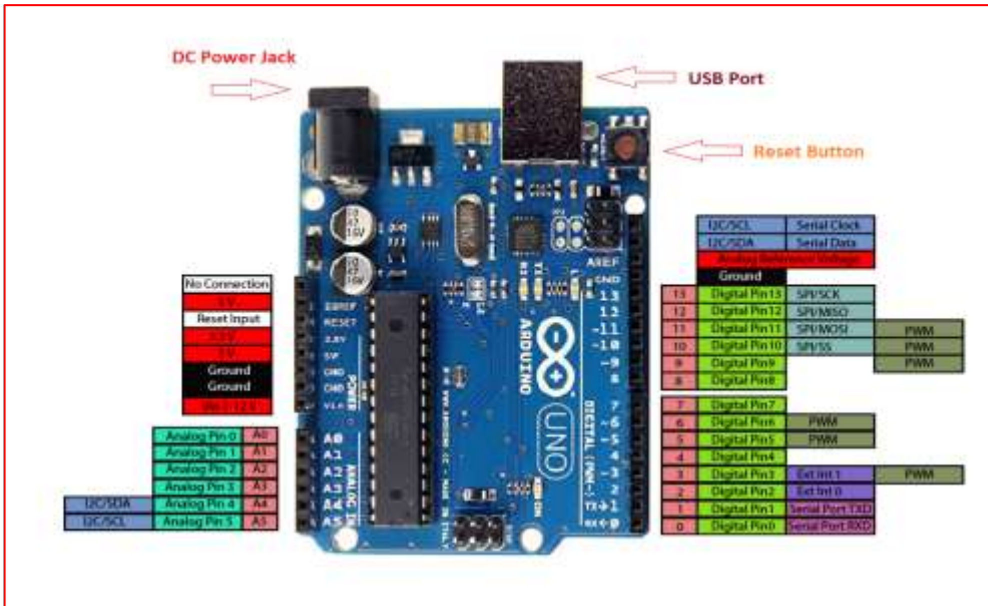

Figure:3.1. Pin diagram of Arduino UNO

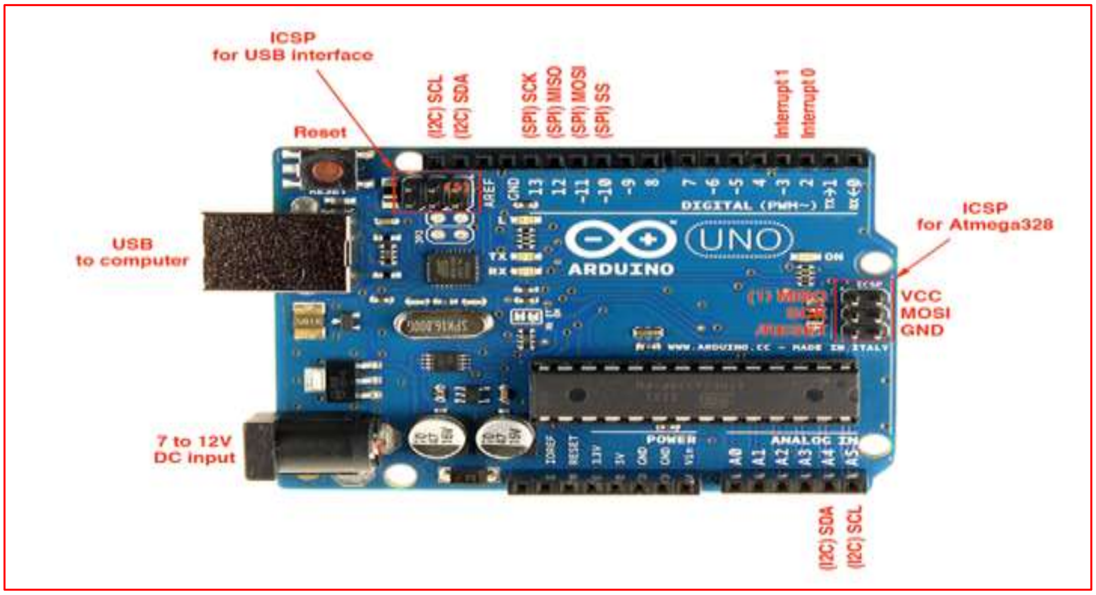

Figure:3.2. On board components of Arduino UNO

3.1.1. Pin description

Pin Category	Pin Name	Details
Power	**Vin, 3.3V, 5V, GND**	**Vin:** Input voltage to Arduino when using an external power source (6-12V). **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 measure analog voltage in the range of 0-5V
Input/ Output Pins	**Digital Pins D0 - D13**	Can be used as input or output pins. 0V (low) and 5V (high)
Serial	Rx, **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.
IIC	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	**AREF**	To provide reference voltage for input voltage.

3.1.2. Memory in Arduino Uno

- ⇨ Flash memory of Arduino Uno is 32Kb.
- ⇨ It has preinstalled boot loader on it, which takes a flash memory of 2kb.
- ⇨ SRAM memory of this Microcontroller board is 8kb.
- ⇨ It has an EEPROM memory of 1kb.

The Arduino board is designed in such a way that it is very easy for beginners to get started with microcontrollers. This board especially is breadboard friendly is very easy to handle the connections. Let’s start with powering the Board.

3.1.3. Input/output:

There are totally 14 digital Pins and 8 Analog pins on your Uno board. The digital pins can be used to interface sensors by using them as input pins or drive loads by using them as output pins. A simple function like pinMode () and digitalWrite () can be used to control their operation. The operating voltage is 0V and 5V for digital pins. The analog pins can measure analog voltage from 0V to 5V using any of the 8 Analog pins using a simple function liken analogRead ()

These pins apart from serving their purpose can also be used for special purposes which are discussed 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 analogWrite () 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 an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, it’s off.
- **I2C A4 (SDA) and A5 (SCA):** Used for IIC communication using Wire library.
- **AREF:** Used to provide reference voltage for analog inputs with analogReference () function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

3.1.4. ATMEGA328P

ATMEGA328P is high performance, low power controller from Microchip. ATMEGA328P is an 8-bit microcontroller based on AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards.

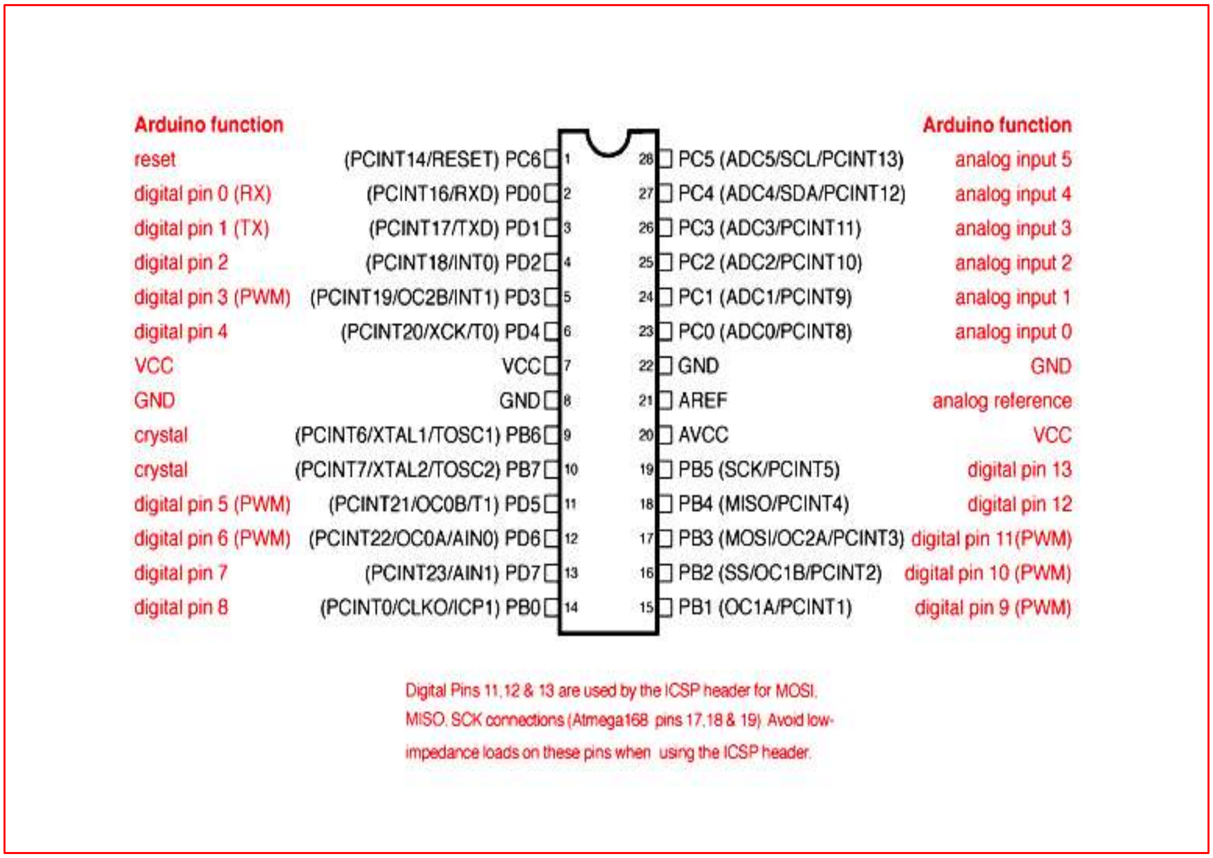

Figure:3.3. ATMEGA328P Pin diagram

3.1.5. ATMega328 Pin Configuration

ATMEGA328P is a 28 pin chip as shown in pin diagram above. Many pins of the chip here have more than one function. We will describe functions of each pin in below table.

Pin No.	Pin name	Description	Secondary Function
1	PC6 (RESET)	Pin6 of PORTC	Pin by default is used as RESET pin. PC6 can only be used as I/O pin when RSTDISBL Fuse is programmed.
2	PD0 (RXD)	Pin0 of PORTD	RXD (Data Input Pin for USART)

			USART Serial Communication Interface [Can be used for programming]
3	PD1 (TXD)	Pin1 of PORTD	TXD (Data Output Pin for USART) USART Serial Communication Interface [Can be used for programming] INT2(External Interrupt 2 Input)
4	PD2 (INT0)	Pin2 of PORTD	External Interrupt source 0
5	PD3 (INT1/OC2B)	Pin3 of PORTD	External Interrupt source1 OC2B(PWM - Timer/Counter2 Output Compare Match B Output)
6	PD4 (XCK/T0)	Pin4 of PORTD	T0(Timer0 External Counter Input) XCK (USART External Clock I/O)
7	VCC		Connected to positive voltage
8	GND		Connected to ground
9	PB6 (XTAL1/TOSC1)	Pin6 of PORTB	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
10	PB7 (XTAL2/TOSC2)	Pin7 of PORTB	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
11	PD5	Pin5 of PORTD	T1(Timer1 External Counter Input)

	(T1/OC0B)		OC0B(PWM - Timer/Counter0 Output Compare Match B Output)
12	PD6 (AIN0/OC0A)	Pin6 of PORTD	AIN0(Analog Comparator Positive I/P) OC0A(PWM - Timer/Counter0 Output Compare Match A Output)
13	PD7 (AIN1)	Pin7 of PORTD	AIN1(Analog Comparator Negative I/P)
14	PB0 (ICP1/CLKO)	Pin0 of PORTB	ICP1(Timer/Counter1 Input Capture Pin) CLKO (Divided System Clock. The divided system clock can be output on the PB0 pin)
15	PB1 (OC1A)	Pin1 of PORTB	OC1A (Timer/Counter1 Output Compare Match A Output)
16	PB2 (SS/OC1B)	Pin2 of PORTB	SS (SPI Slave Select Input). This pin is low when controller acts as slave. [Serial Peripheral Interface (SPI) for programming] OC1B (Timer/Counter1 Output Compare Match B Output)
17	PB3 (MOSI/OC2A)	Pin3 of PORTB	MOSI (Master Output Slave Input). When controller acts as slave, the data is received by this pin. [Serial Peripheral Interface (SPI) for programming]

			OC2 (Timer/Counter2 Output Compare Match Output)
18	PB4 (MISO)	Pin4 of PORTB	MISO (Master Input Slave Output). When controller acts as slave, the data is sent to master by this controller through this pin. [Serial Peripheral Interface (SPI) for programming]
19	PB5 (SCK)	Pin5 of PORTB	SCK (SPI Bus Serial Clock). This is the clock shared between this controller and other system for accurate data transfer. [Serial Peripheral Interface (SPI) for programming]
20	AVCC		Power for Internal ADC Converter
21	AREF		Analog Reference Pin for ADC
22	GND		GROUND
23	PC0 (ADC0)	Pin0 of PORTC	ADC0 (ADC Input Channel 0)
24	PC1 (ADC1)	Pin1 of PORTC	ADC1 (ADC Input Channel 1)
25	PC2 (ADC2)	Pin2 of PORTC	ADC2 (ADC Input Channel 2)
26	PC3 (ADC3)	Pin3 of PORTC	ADC3 (ADC Input Channel 3)
27	PC4 (ADC4/SDA)	Pin4 of PORTC	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/output Line)

28	PC5 (ADC5/SCL)	Pin5 of PORTC	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)

3.2 L293D MOTOR DRIVE SHIELD:

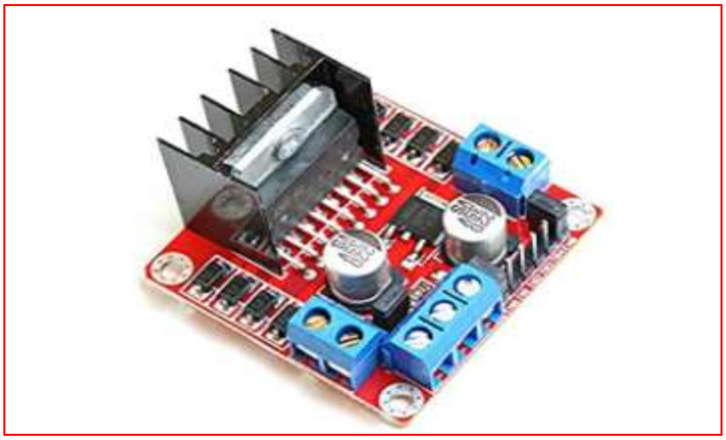

Fig.3.4 L293D motor driver IC

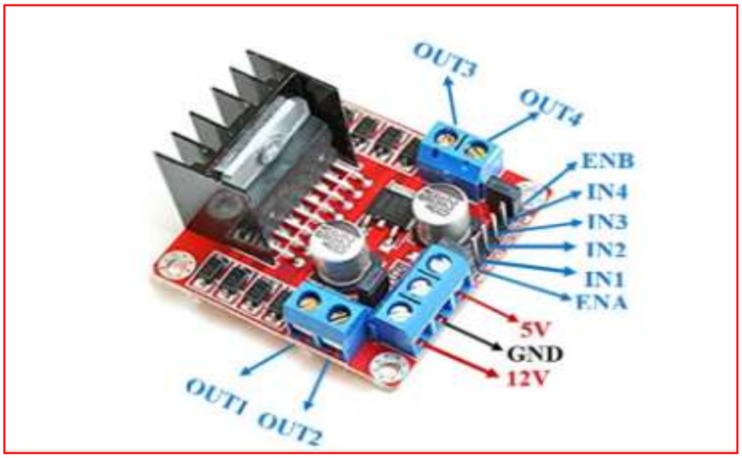

Fig.3.5: on board components of L293D motor driver IC

This **L293D Motor Driver Module** is a high-power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. **L293D Module** can control up to 4 DC motors, or 2 DC motors with directional and speed control.

3.2.1. L293D Module Features & Specifications:

- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Driver Model: L298N 2A
- Current Sense for each motor
- Heat sinks for better performance
- Power-On LED indicator

3.2.2. Brief about L293D Module

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

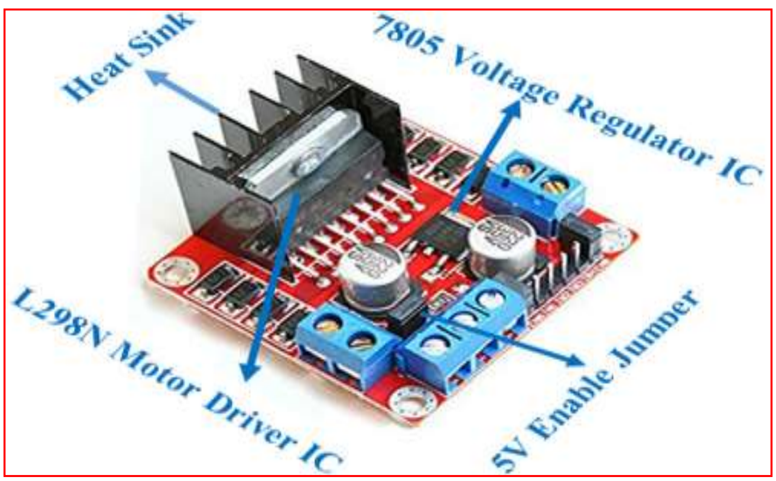

Fig.3.6: on board regulators of L293D

78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry. ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B. Internal circuit diagram of L298N Motor Driver module is given below

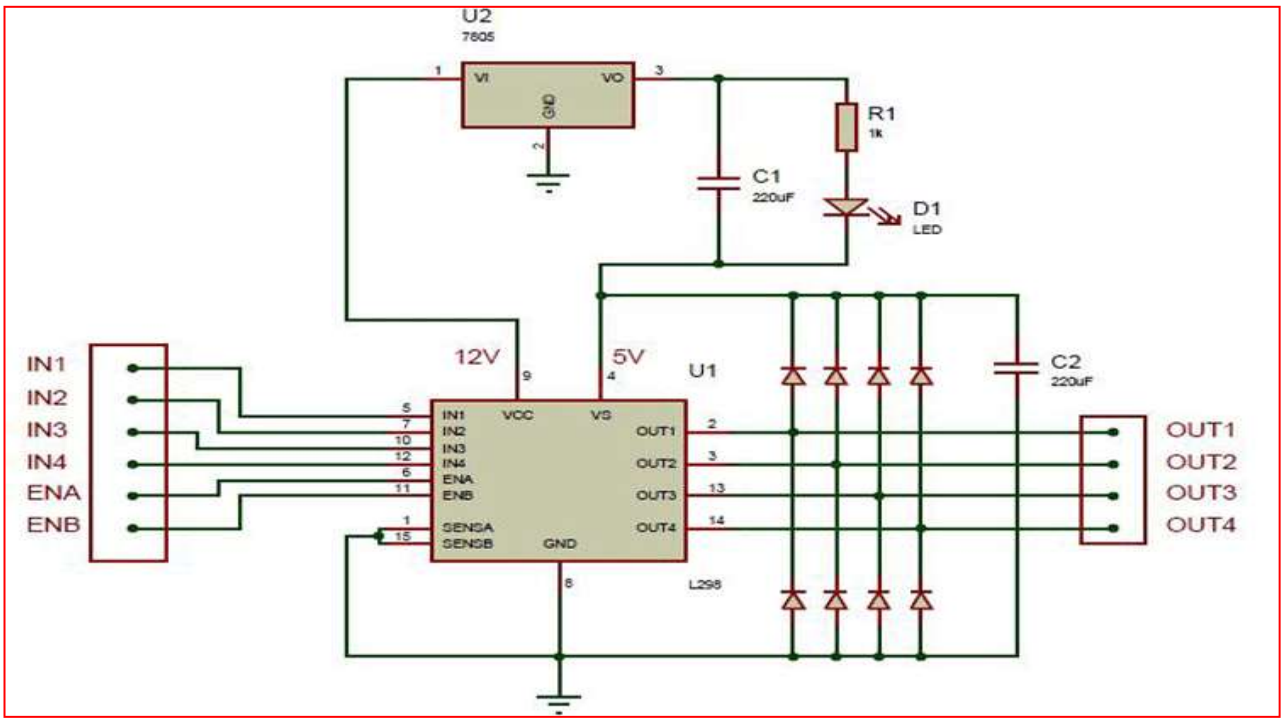

Fig.3.7: Internal circuit diagram of L293D Motor Driver module is given below

3.2.3. Applications of L293D Module

- Drive DC motors
- Drive stepping motors
- In Robotics

3.3 ULTRASONIC SENSOR:

An ultrasonic sensor emits sound waves toward an object and determines its distance by detecting reflected waves. (Getty Images)
An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is D = ½ T x C (where D is the distance, T is(where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be:
D = 0.5 x 0.025 x 343
or about 4.2875 meters.

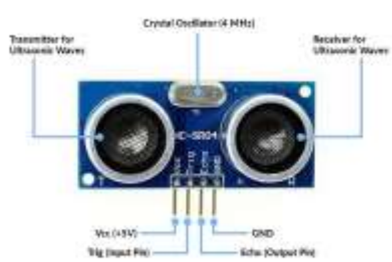

Fig.3.8:Ultrasonic sensor

3.4 DC MOTOR:

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs.

3.4.1. Working Principle of a DC Motor

The DC motor is the device which converts the direct current into the mechanical work. It works on the principle of Lorentz Law, which states that.. "The current-carry conductor placed in a magnetic and electric field experience a force". The experienced force is called the Lorentz force. The Flemming left-hand rule gives the direction of the force.

Fig.3.9: DC Motor

3.4.2: Fleming Left Hand Rule:

If the thumb, middle finger and the index finger of the left hand are displaced from each other by an angle of 90°, the middle finger represents the direction of the magnetic field. The index finger represents the direction of the current, and the thumb shows the direction of forces acting on the conductor.

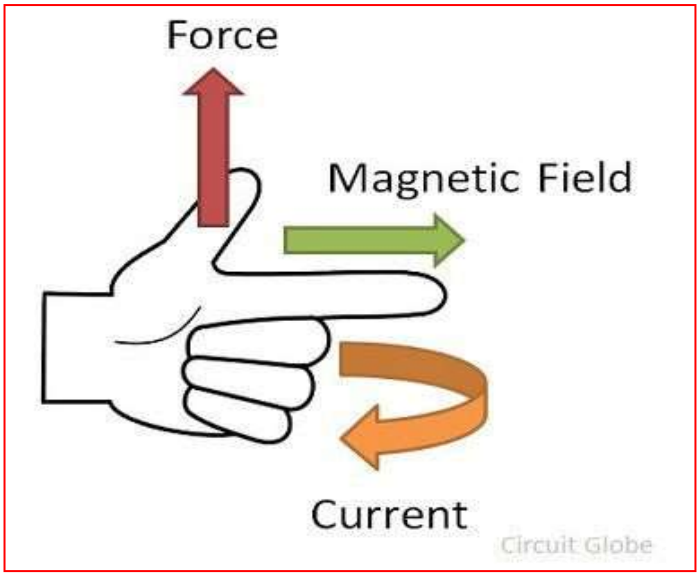

Fig.3.9.1 Left Hand Rule

The formula calculates the magnitude of the force,

$$\boxed{F = BIl \quad \text{newton}}$$

Before understanding the working of DC motor, first, we have to know about its construction. The armature and stator are the two main parts of the DC motor. The armature is the rotating part, and the stator is their stationary part. The armature coil is connected to the DC supply.

The armature coil consists the commutators and brushes. The commutators convert the AC induced in the armature into DC and the brushes transfer the current from rotating part of the motor to the stationary external load. The armature is placed between the north and south pole of the permanent or electromagnet.

For simplicity, consider that the armature has only one coil which is placed between the magnetic field shown below in the figure A. When the DC supply is given to the armature coil the current starts flowing through it. This current develops its own field around the coil.

Figure B shows the field induces around the coil:

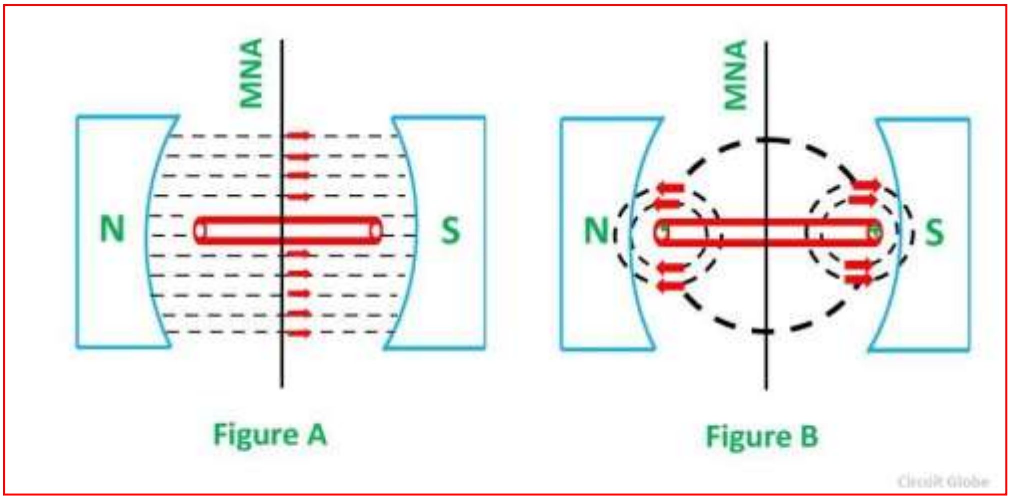

Fig.3.9.2 Working Principle of DC Motor

By the interaction of the fields (produced by the coil and the magnet), the resultant field develops across the conductor. The resultant field tends to regain its original position, i.e., in the axis of the main field. The field exerts the force at the ends of the conductor, and thus the coil starts rotating.

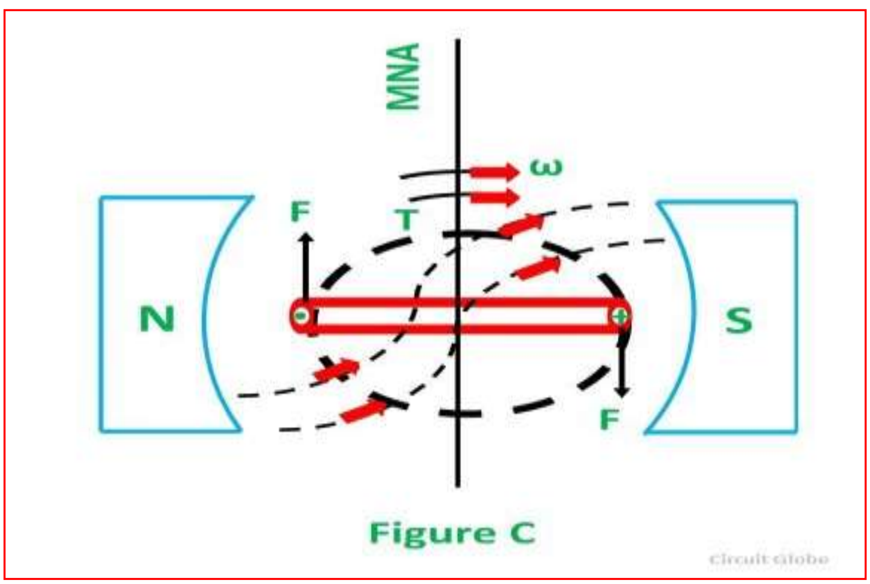

Fig:3.9.3 Working Principle of DC Motor

Let the field produced by the main field be F_m, and this field rotates in the clockwise direction. When the current flows in the coil, they produce their own magnetic field say, F_r. The field F_r tries to come in the direction of the main field. Thereby, the torque act on the armature.

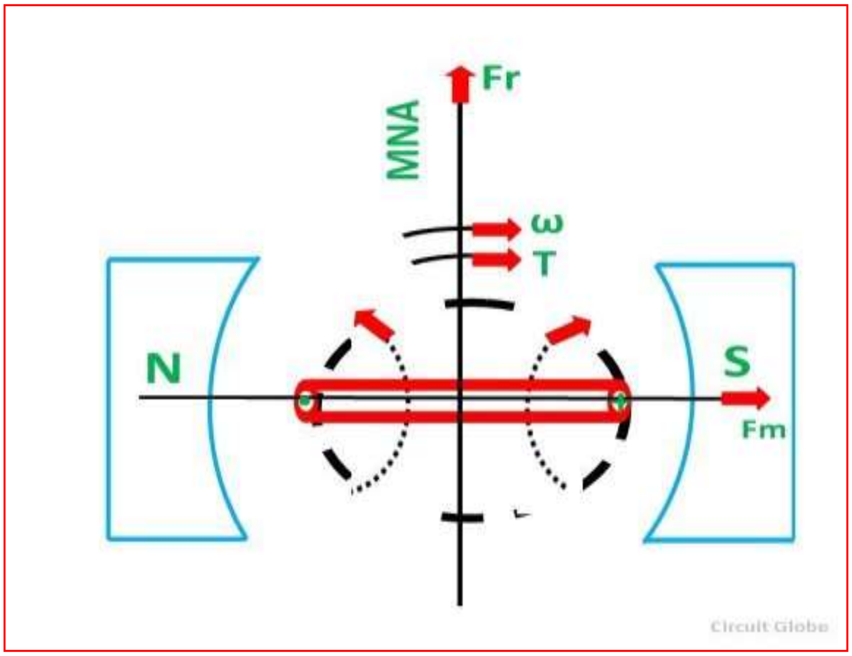

Fig.3.9.4: Working Principle of DC Motor

The actual DC motor consists of a large number of armature coils. The speed of the motor is directly proportional to the number of coils used in the motor These coils are kept under the impact of the magnetic field.

The one end of the conductors is kept under the influence of the north pole, and the other end is kept under the influence of the south pole. The current enters into the armature coil through the north pole and move outwards through the south pole.

When the coil moves from one brush to another, at the same time the polarity of the coil also changes. Thus, the direction of the force or torque acting on the coil remains the same.

The torque induces in the coil become zero when the armature coil is perpendicular to the main field. The zero torque means the motor stops rotating. For solving this problem, the number of armature coil is used in the rotor. So, if one of their coils is perpendicular to the field, then the other coils induce the torque. And the rotor moves continuously.

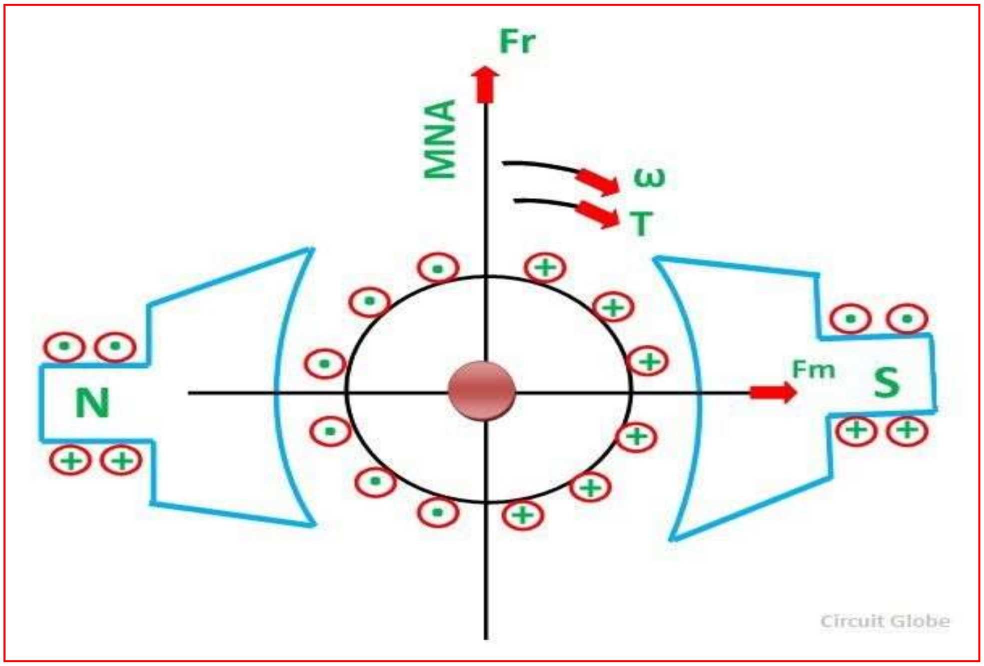

Fig.3.9.5: DC Machine

Also, for obtaining the continuous torque, the arrangement is kept in such a way that whenever the coils cut the magnetic neutral axis of the magnet the direction of current in the coils become reversed. This can be done with the help of the commutator.

3.5 BLDC MOTOR:

Brushless DC Motors or BLDC Motors have become a significant contributor of the modern drive technology. Their rapid gain in popularity has seen an increasing range of applications in the fields of Consumer Appliances, Automotive Industry, Industrial Automation, Chemical and Medical, Aerospace and Instrumentation**.**

Even though they have been used for drives and power generation for a long time, the sub kilowatt range, which has been dominated by Brushed DC Motors, has always been a grey area. But the modern power electronics and microprocessor technology has allowed the small Brushless DC

Motors to thrive, both in terms price and performance. A Brushless DC Motor is similar to a Brushed DC Motor but as the name suggests, a BLDC doesn't use brushes for commutation but rather they are electronically commutated. In conventional Brushed DC Motors, the brushes are used to transmit the power to the rotor as they turn in a fixed magnetic field. suggests, a BLDC doesn't use brushes for commutation but rather they are electronically commutated. In conventional Brushed DC Motors, the brushes are used to transmit the power to the rotor as they turn in a fixed magnetic field.

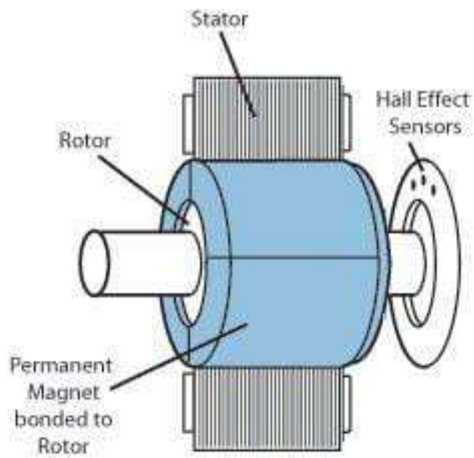

Fig.3.5.1:BLDC MOTOR

}

3.6 SERVO MOTOR:

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the **DC servo motor working**. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

3.6.1. Servo Motor Working Mechanism

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

3.6.2. Servo Motor Working Principle

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

3.6.3. Interfacing Servo Motors with Microcontrollers:

Interfacing hobby Servo motors like s90 servo motor with MCU is very easy. **Servos have three wires coming out of them**. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. An **MG995 Metal Gear Servo Motor** which is most commonly used for RC cars humanoid bots etc. The picture of MG995 is shown below:

Figur:3.6.1. Servo motor

The colour coding of your servo motor might differ hence check for your respective datasheet. All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

3.6.4. Controlling Servo Motor:

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

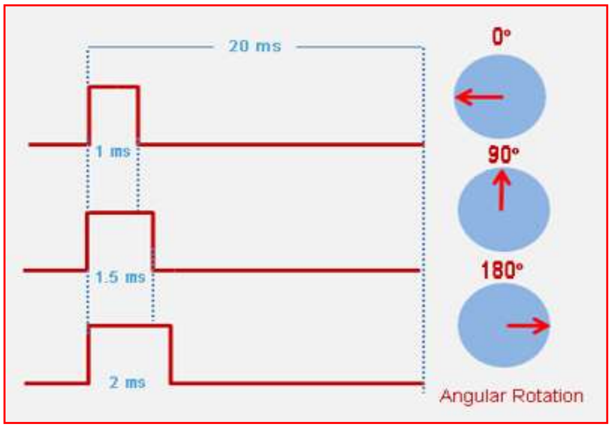

Fig:3.6.2. PWM waves for servo motor

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

3.7 BLUETOOTH MODULE(HC-05):

Fig : Bluetooth module(HC-05)

Bluetooth module (Bluetooth module) refers to the basic circuit set of the chip with integrated Bluetooth function, used for short-range 2.4G wireless communication module. For the end user, the Bluetooth module is a semi-finished product so this is called as Bluetooth module.

3.8 ARDUINO IDE CODE FOR ROBOT:

```
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>

#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 200
#define MAX_SPEED 190
#define MAX_SPEED_OFFSET 20

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
Servo myservo;

boolean goesForward=false;
int distance = 100;
int speedSet = 0;

void setup() {

  myservo.attach(10);
  myservo.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
}

void loop() {
 int distanceR = 0;
 int distanceL =  0;
 delay(40);

 if(distance<=15)
 {
  moveStop();
  delay(100);
  moveBackward();
  delay(300);
  moveStop();
```

```
  delay(200);
  distanceR = lookRight();
  delay(200);
  distanceL = lookLeft();
  delay(200);

  if(distanceR>=distanceL)
  {
    turnRight();
    moveStop();
  }else
  {
    turnLeft();
    moveStop();
  }
 }else
 {
  moveForward();
 }
 distance = readPing();
}

int lookRight()
{
    myservo.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(115);
    return distance;
}

int lookLeft()
{
    myservo.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    myservo.write(115);
    return distance;
    delay(100);
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if(cm==0)
  {
    cm = 250;
  }
  return cm;
}
```

```
void moveStop() {
 motor1.run(RELEASE);
 motor2.run(RELEASE);
 motor3.run(RELEASE);
 motor4.run(RELEASE);
 }

void moveForward() {

 if(!goesForward)
 {
  goesForward=true;
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2)
  {
  motor1.setSpeed(speedSet);
  motor2.setSpeed(speedSet);
  motor3.setSpeed(speedSet);
  motor4.setSpeed(speedSet);
  delay(5);
  }
 }
}

void moveBackward() {
  goesForward=false;
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
 for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2)
 {
  motor1.setSpeed(speedSet);
  motor2.setSpeed(speedSet);
  motor3.setSpeed(speedSet);
  motor4.setSpeed(speedSet);
  delay(5);
 }
}

void turnRight() {
 motor1.run(FORWARD);
 motor2.run(FORWARD);
 motor3.run(BACKWARD);
 motor4.run(BACKWARD);
 delay(500);
 motor1.run(FORWARD);
 motor2.run(FORWARD);
```

```
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}

void turnLeft() {
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(500);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
```

3.9 ARDUINO IDE CODE FOR MOTORS:

```
#include <SoftwareSerial.h>
SoftwareSerial bluetooth(0, 1); // RX, TX pins for Bluetooth module
int sweep = 3;
int clean = 4;
int dry = 5;

void setup() {
  Serial.begin(9600); // For debugging
  bluetooth.begin(9600);
  pinMode(sweep, OUTPUT);
  pinMode(clean, OUTPUT);
  pinMode(dry, OUTPUT);
}

char c;
String voice;

void loop() {
  if (bluetooth.available() > 0) {
    voice = "";
    voice = bluetooth.readString();
```

```
    Serial.println(voice); // For debugging
  }

  if (voice == "sweep") {
    analogWrite(sweep, 255); // Full speed
  } else if (voice == "sweep off") {
    analogWrite(sweep, 0); // Stop
  }

  if (voice == "clean") {
    analogWrite(clean, 255);
  } else if (voice == "clean off") {
    analogWrite(clean, 000);
  }

  if (voice == "dry") {
    analogWrite(dry, 255);
  } else if (voice == "dry off") {
    analogWrite(dry, 000);
  }
}
```

CHAPTER-4
METHODOLOGIES AND MODELLING

The methodology adopted for this project involves the Cleaning system which involve requirement selections, Technology selection, Data collection, Data processing, Data management, Integration with existing system, user interface design, Security measures, Testing, Deployment, Monitoring and maintenance, compliance, feedback loop, documentation these steps can develop agriculture system.

4.1 BLOCK DIAGRAM

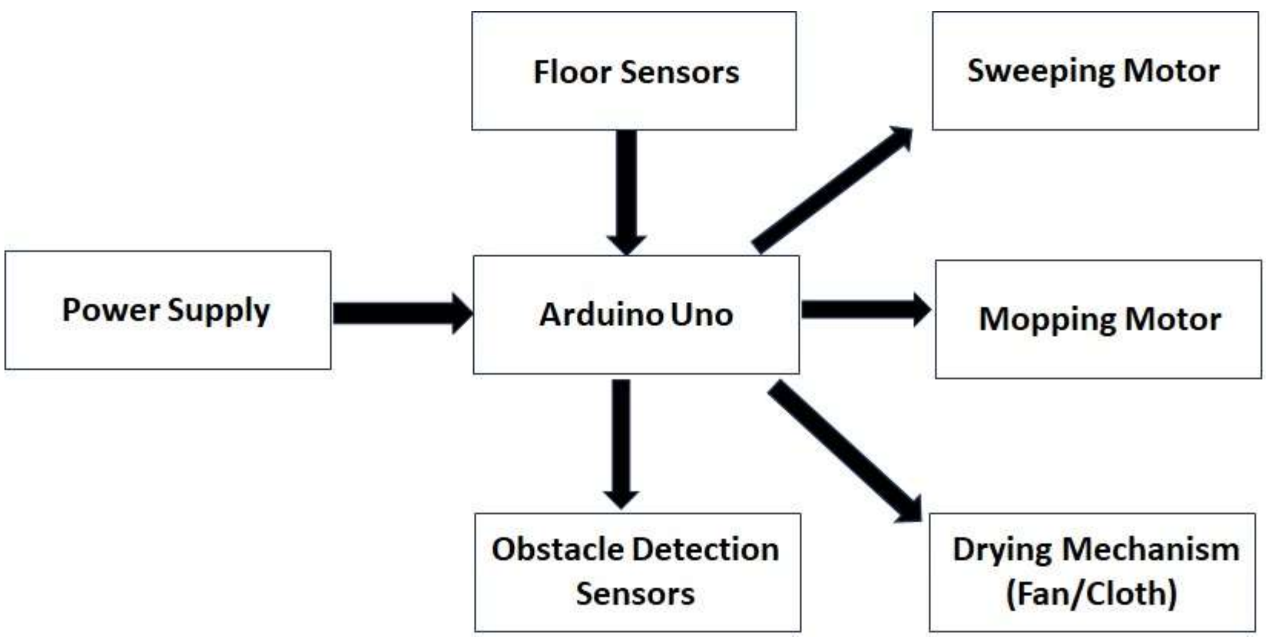

Fig.4.1: Block Diagram

4.2 CIRCUIT DIAGRAM:

A circuit diagram of an Cleaning robot using artificial intelligence would show the connections between the different components of the robot. Here's an example of what the circuit diagram used for this project. The L293D Motor Driver Shield is connected to the top of the Arduino UNO Board. The 4 DC motors are connected to the Motor Driver shield which is controlled via digital pins D1, D2, D3, and D4 of the Arduino UNO Board. The Servo Motor is controlled via PWM Pin D10 of Arduino. The Ultrasonic Sensor HC-SR04 has TRIGN & ECHO Pins which are connected to A0 & A1 of the Arduino Board. To control the BLDC, Motor an ESC Module Servo Testes is used. The entire circuit is powered by an 11.1V Lithium-Ion Battery. The Arduino has a 5V regulator which converts the 11.1V to 5V Supply. The BLDC Motor directly operates at 11.1V.

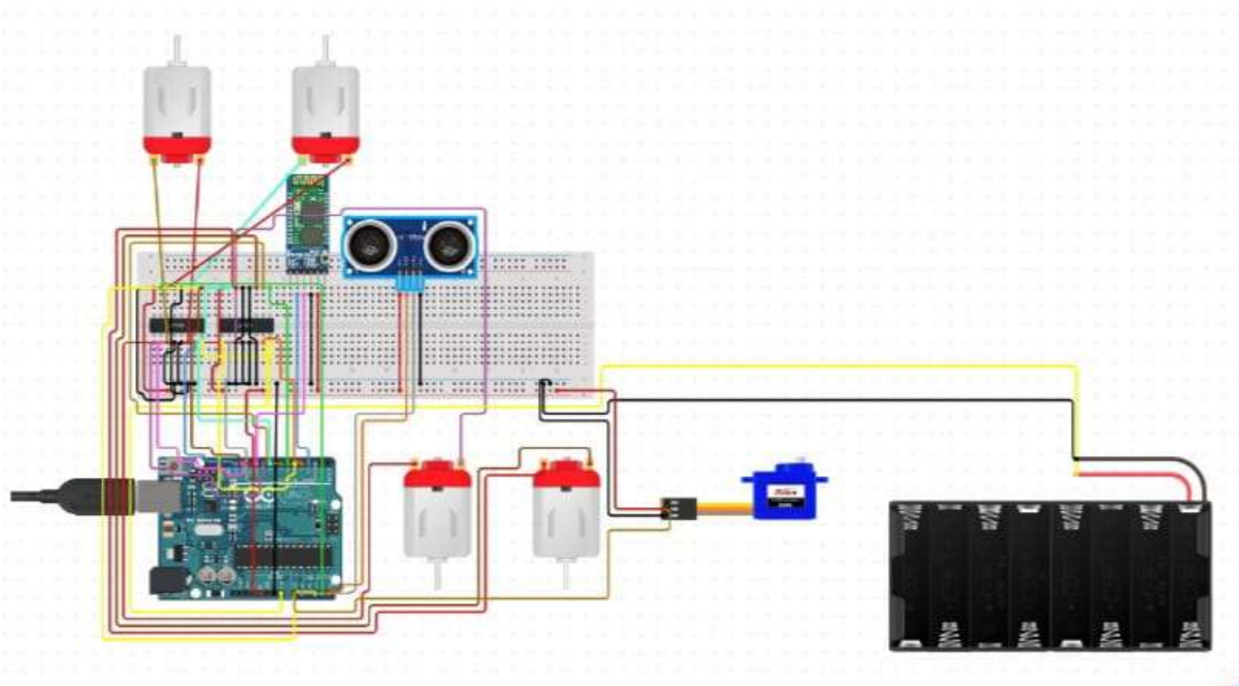

Figure:4.2 Circuit Diagram

CHAPTER-5
EXPERIMENTAL RESULTS

5.1 EXPLANATION:

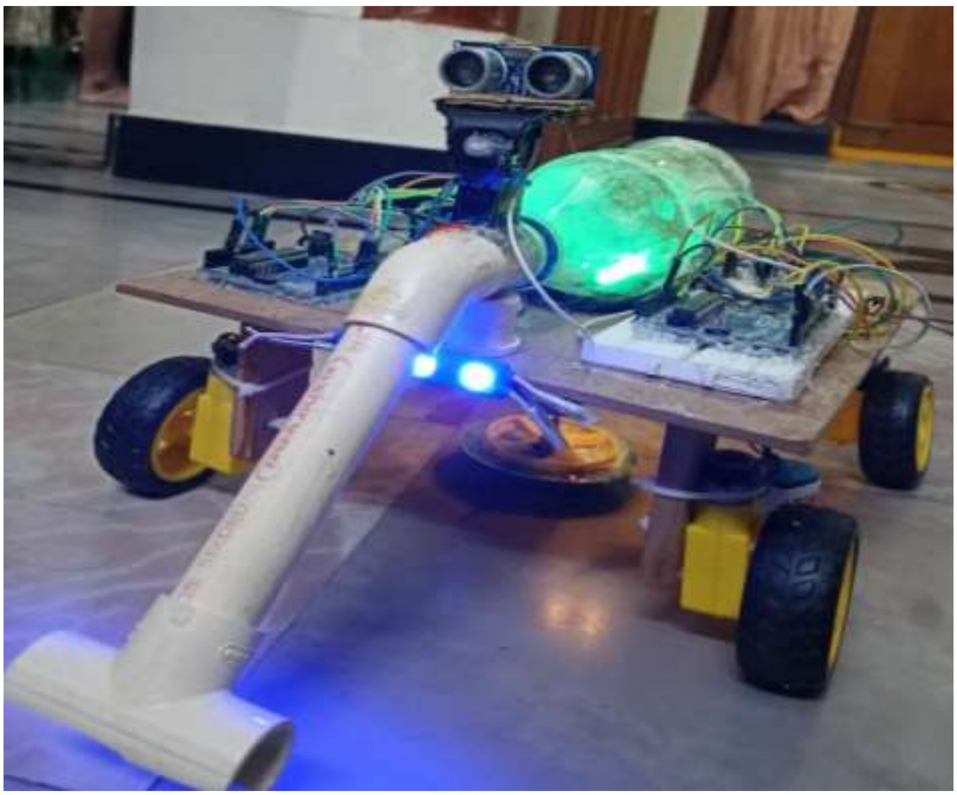

Figure:5.1 Trio Cleaning Robot

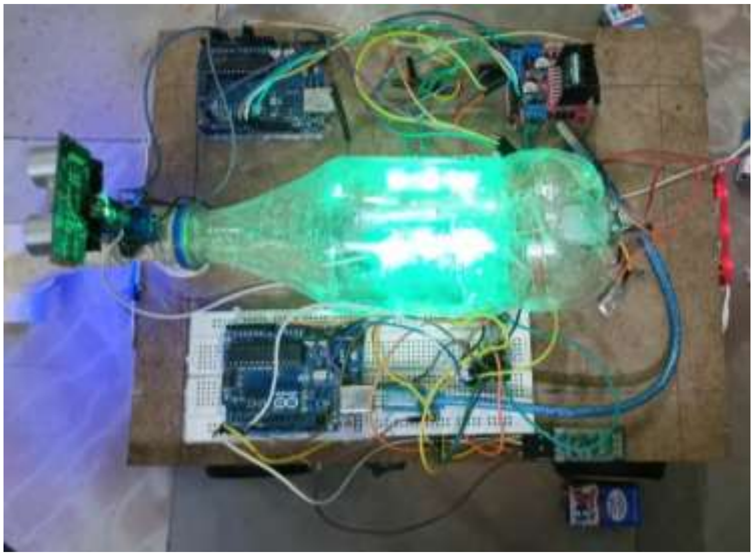

Figure:5.2 Front view Trio Cleaning Robot

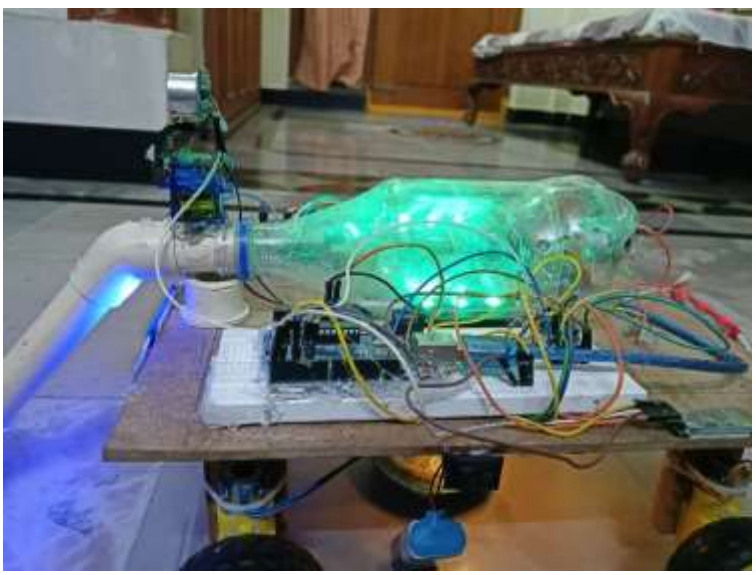

Figure:5.3 Side view Trio Cleaning Robot

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6.1 ADVANTAGES:

- Time Saving.
- Multi Functionality.
- Cost-effective.
- Eco-Friendly.
- Automated Process

6.2 DISADVANTAGES:

- Prohibitively expensive to buy and maintain.
- Large, heavy and complicated.
- Operator learning curve.

6.3 APPLICATIONS:

- Remote Monitoring and Control
- Household Cleaning
- Commercial Cleaning
- Industrial Cleaning
- Multi-floor Cleaning

6.4 CONCLUSION:

The Trio Cleaning Robot project combines the power of Arduino microcontrollers and AI to create a sophisticated cleaning solution that not only automates the cleaning process but also adapts to different environments. This innovative project aims to simplify household chores and contribute to the advancement of smart home technologies. Ongoing maintenance and updates will ensure continued success in meeting customer needs and industry demands. The project successfully addressed key objectives, improving efficiency, sustainability, and user experience. With a robust design and eco-friendly features, it aligns with modern environmental standards. User feedback and performance metrics indicate a positive impact on lawn care practices. Ongoing maintenance and updates will ensure continued success in meeting customer needs and industry demands.

6.5 FUTURE SCOPE:

- It includes Environmental Considerations.
- It includes Automated Cleaning robots.
- It includes Precision Cleaning.
- Able to edit the Battery Technology.

REFERENCES

[1] H.Leamer,'RFID design and principles, 'second edition, Artech house,inc,2008

[2] B.m.Stephen and R.W john 'RFID technology and applications', Cambridge university press,2008

[3] Aiyana and M. Alias 'attendance system using RFID system 'faculty of electronics and computer engineering, India.

[4] Environmental Protection Agency. (2021). Environmental Monitoring Handbook: Guidelines for the Installation and Maintenance of Monitoring Sensors. Retrieved from [URL].

[5] Smart, T., & Green, L. (2017). Wireless Sensor Networks in Environmental Monitoring: A Review. Journal of Sensor and Actuator Networks, 6(3), 21