**4.Hardware and Software Requirements**

**Arduino UNO.**

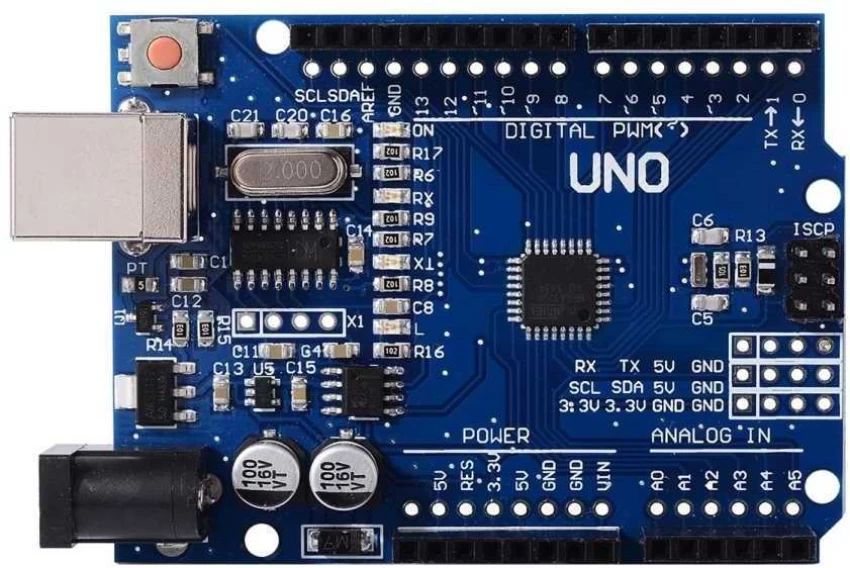
****

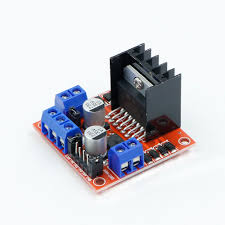
Fig.

Arduino UNO

The Arduino UNO is one of the most popular and widely used microcontroller boards in the Arduino family, based on the ATmega328P microcontroller. It features 14 digital input/output pins, numbered from 0 to 13, out of which 6 pins (3, 5, 6, 9, 10, 11) can be used as PWM (Pulse Width Modulation) outputs, allowing for control over devices like motors and LEDs with variable brightness or speed. Additionally, the UNO provides 6 analog input pins (A0 to A5), which can read variable voltage signals from sensors, converting them into values the microcontroller can process. The board also has a 5V pin and a 3.3V pin for powering external components, along with several GND (Ground) pins to complete circuits. It includes a reset pin to manually restart the program. For communication, the TX (pin 1) and RX (pin 0) pins handle serial communication, allowing the UNO to talk to other devices like computers or other microcontrollers. The power jack, USB port, and ICSP header provide different ways to power, program, or upload code to the board. With its easy-to-use design, rich pin layout, and strong support community, the Arduino UNO is ideal for beginners and advanced users alike.

The Arduino UNO operates at 5V logic level and can be powered through the USB connection (5V), the DC power jack (7-12V), or the Vin pin (7-12V). The onboard voltage regulator ensures stable operation, providing 5V and 3.3V outputs for connected components. It also features onboard LEDs—the "L" LED (connected to pin 13) for quick testing, as well as TX and RX LEDs that blink during serial communication. The board supports I2C (pins A4 - SDA and A5 - SCL) and SPI (pins 10-13) communication protocols, allowing it to interface with a variety of sensors, displays, and other modules. The Arduino UNO can be programmed using the Arduino IDE with simple C/C++-based code and has a bootloader that allows for easy code uploading without needing an external programmer. Due to its open-source nature, affordable price, and strong ecosystem of shields (expansion boards) and libraries, the Arduino UNO remains a top choice for robotics, IoT, automation, and DIY electronics projects.

**L298N Motor Driver.**



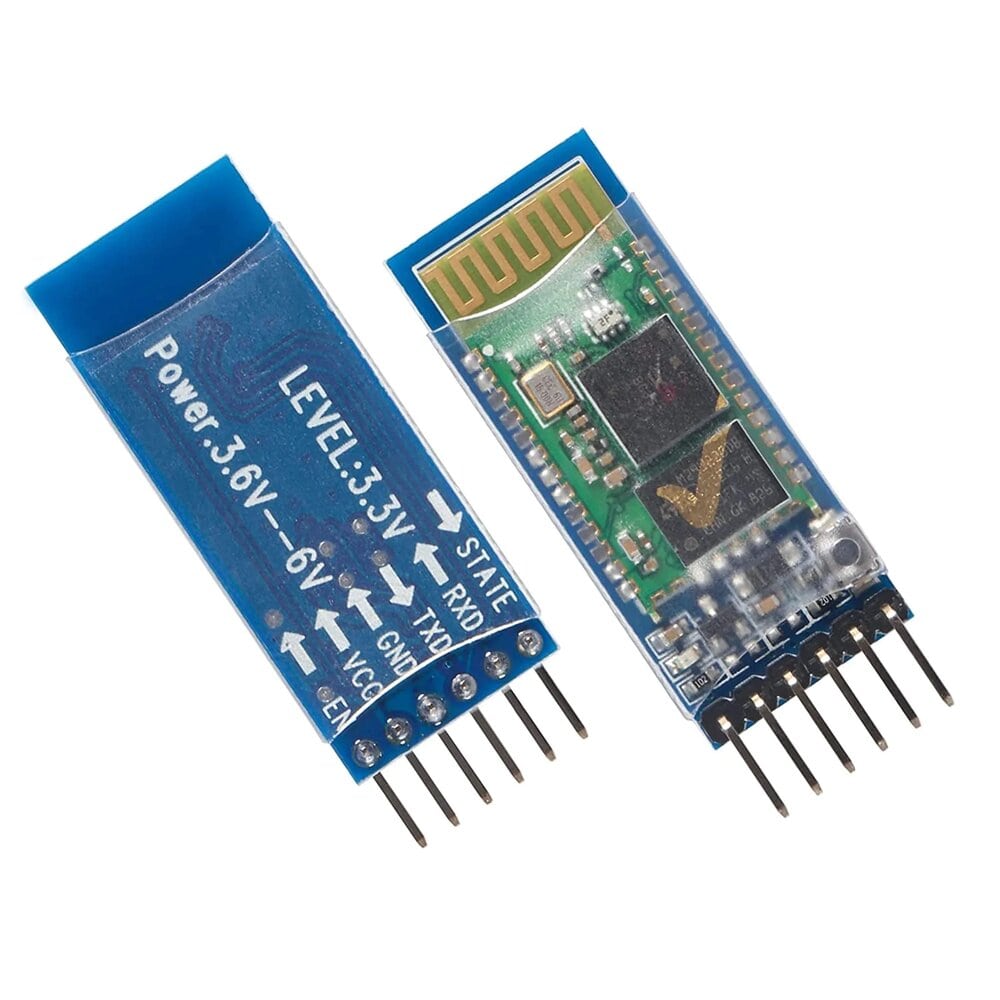
The L298N Motor Driver Module is a versatile and widely used motor driver designed to control the speed and direction of two DC motors or one stepper motor. It is based on the L298N dual H-Bridge motor driver IC, which allows motors to rotate both forward and backward by reversing polarity. The module supports a wide operating voltage range from 5V to 35V, making it compatible with various motors, including low-voltage toy motors and higher-power motors. Each channel can handle up to 2A of continuous current, though a large heatsink is provided to help dissipate heat during high-power operation.

The L298N module features two motor output channels, each controlled by two input pins (IN1, IN2 for motor A and IN3, IN4 for motor B) that set the rotation direction. Speed control is achieved using PWM (Pulse Width Modulation) signals applied to the ENA and ENB pins. The module also includes a 5V regulator, which can power external circuits like an Arduino, provided the input voltage is within limits. Built-in flyback diodes protect against voltage spikes caused by motor switching. With its reliability, low cost, and ease of use, the L298N Motor Driver Module is a popular choice for robotics, automation, and DIY electronics projects.

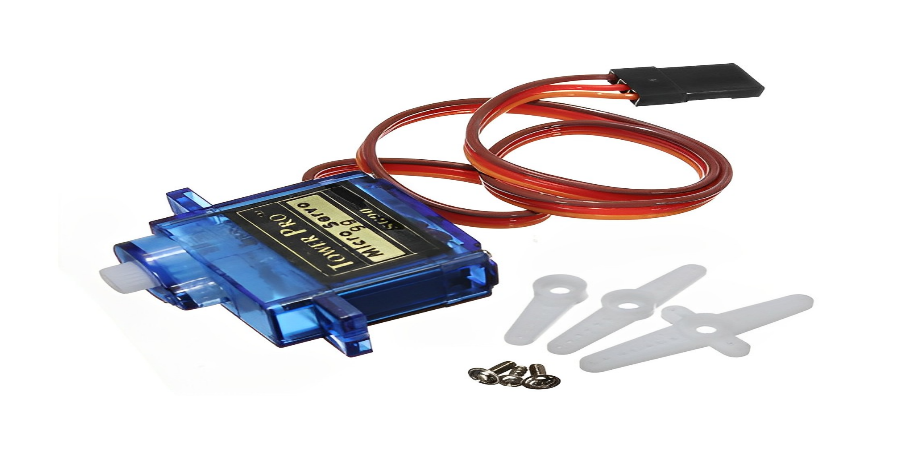
Bluetooth Module HC-05.

The HC-05 Bluetooth Module is a popular and affordable Bluetooth device used for wireless communication between microcontrollers, such as Arduino, and smartphones, computers, or other Bluetooth-enabled devices. It operates using Bluetooth 2.0 technology, with a typical communication range of about 10 meters in open space. The HC-05 module supports both Master and Slave modes, meaning it can either initiate a connection to another Bluetooth device or wait to be discovered and connected to. This flexibility makes it suitable for projects requiring two-way wireless communication.

The module operates at 3.3V logic levels, although its power supply can range from 3.6V to 6V, making it compatible with 5V microcontrollers like Arduino. It uses serial communication (UART) to interface with microcontrollers, allowing devices to send and receive data wirelessly through standard TX and RX pins. The key pin can be used to enter AT Command mode, where configuration settings like baud rate, device name, and pairing code can be adjusted.

The HC-05 module is widely used in robotics, home automation, remote control systems, and wireless sensor networks. Its ease of use, low cost, and reliable performance make it an ideal solution for adding Bluetooth connectivity to electronics projects.

**Servo Motors.**



Servo motors are precision-controlled motors commonly used in robotics, automation systems, and control applications requiring accurate positioning. Unlike regular DC motors, servo motors are designed to rotate to a specific angle rather than continuously spinning. They are equipped with an internal control circuit and a position feedback mechanism, typically a potentiometer, which continuously monitors the shaft’s position to ensure the motor reaches and maintains the desired angle.

Servo motors usually have three wires — power (Vcc), ground (GND), and control signal (PWM). The control signal is a Pulse Width Modulation (PWM) signal, where the pulse duration determines the angle the servo moves to, typically ranging from 0° to 180° in standard hobby servos. These motors are highly favored in applications such as robotic arms, remote-controlled vehicles, and automated systems, where precise angle control is critical.

There are different types of servo motors, including continuous rotation servos, which can rotate fully like DC motors but still accept position-based commands, and standard servos, which rotate to a specified angle. Their compact size, precise control, and ease of interfacing with microcontrollers like Arduino make them essential components in robotics, drones, and industrial control systems.

Water Pump.



A water pump is a mechanical device used to move water from one location to another, typically by converting mechanical energy into hydraulic energy. Water pumps are widely used in agriculture, households, industrial processes, and electronic projects requiring water flow. They come in various types, including centrifugal pumps, diaphragm pumps, submersible pumps, and peristaltic pumps, each suited for different applications.

In electronics and DIY projects, small DC water pumps are commonly used. These pumps are powered by low-voltage DC motors, usually operating between 3V and 12V, making them compatible with microcontrollers like Arduino or Raspberry Pi. They can be controlled using relay modules or motor drivers, allowing automation in projects such as automatic plant watering systems, fountains, or water cooling setups.

Most small water pumps consist of an inlet and outlet port, with an impeller inside that rotates to push water through the system. Some pumps are designed for submersion, allowing them to operate underwater, while others are meant for inline use, drawing water from a reservoir. Their compact size, low cost, and ease of integration make them popular for home automation, aquariums, hydroponics, and irrigation systems, where efficient water movement is essential.

**4.2 Working of Each Component**

**Arduino UNO**

The Arduino UNO works as a microcontroller board that reads inputs from sensors, processes the data, and controls connected devices like LEDs, motors, or displays. It uses an ATmega328P microcontroller, which acts as the brain of the board. The UNO reads analog and digital inputs from sensors (like temperature or motion sensors) connected to its input pins. Based on the uploaded code, it processes this data and sends commands to output devices such as motors, relays, or displays through output pins.

The Arduino IDE is used to write programs in C/C++, which are uploaded to the board via a USB connection. The UNO communicates with external devices using protocols like UART (Serial), I2C, and SPI. Power can be supplied through USB, a DC power jack, or the Vin pin. Its easy-to-use hardware and software make it ideal for beginners and professionals in electronics and IoT projects.

**L298N Motor Driver**

The L298N Motor Driver works by using an H-Bridge configuration to control the direction and speed of two DC motors independently. It receives control signals from a microcontroller, such as Arduino, through input pins (IN1, IN2 for motor A and IN3, IN4 for motor B). These inputs determine the polarity applied to the motors, allowing them to rotate forward, backward, or stop.

The speed of the motors is controlled using PWM (Pulse Width Modulation) signals sent to the ENA and ENB pins. These signals vary the average voltage supplied to the motors, adjusting their speed smoothly.

The module supports a wide operating voltage range (5V to 35V) and can supply up to 2A per motor channel, making it suitable for small to medium-sized motors. It also includes a heatsink for heat dissipation and flyback diodes to protect the circuit from voltage spikes, ensuring safe and reliable motor control.

**Bluetooth Module HC-05**

The HC-05 Bluetooth module works as a wireless communication device, enabling a microcontroller, such as an Arduino, to send and receive data wirelessly to and from other Bluetooth-enabled devices like smartphones, laptops, or other microcontrollers. It operates using Bluetooth 2.0 technology and supports both Master and Slave modes, allowing it to either initiate a connection or wait for pairing requests.

Communication with a microcontroller is done via serial communication (UART) using the TX and RX pins. Data sent from the microcontroller is transmitted wirelessly to the paired device, and data received wirelessly is forwarded to the microcontroller.

The module operates on a 3.3V logic level but can be powered with 3.6V to 6V. It also features an AT Command mode, which allows configuration of settings such as baud rate, device name, and password. Its ease of use, low cost, and reliable performance make it popular for wireless control and data logging projects.

**Servo Motors.**

Servo motors work by receiving control signals that determine their position rather than continuous rotation. They have an internal control circuit and a position sensor (usually a potentiometer) connected to the motor shaft. This allows the motor to continuously monitor its position and adjust itself to match the commanded angle.

A PWM (Pulse Width Modulation) signal is used to control the motor’s position. The pulse width (typically ranging between 1ms to 2ms) corresponds to a specific angle, usually within a 0° to 180° range for standard hobby servos. The servo’s internal controller compares the actual position with the desired position from the PWM signal and moves the motor until they match.

Servo motors have three wires: power, ground, and signal. They are commonly used in robotic arms, RC vehicles, and automated systems because they offer precise position control, compact size, and easy integration with microcontrollers like Arduino.

**Water Pump**.

A water pump works by converting electrical energy into mechanical energy to move water from one place to another. In small DC water pumps commonly used in electronics and DIY projects, a DC motor drives an impeller inside the pump housing. When the motor spins, the impeller creates centrifugal force, drawing water into the inlet and pushing it out through the outlet, creating continuous water flow.

The pump can be powered by a low-voltage power source, usually 3V to 12V, and can be controlled by a relay module or motor driver connected to a microcontroller like Arduino for automation. Some pumps are designed to operate submerged in water (submersible pumps), while others draw water from an external source (inline pumps).

These pumps are commonly used in automatic plant watering systems, fountains, aquariums, and cooling systems where efficient, controlled water flow is needed, making them essential for both home and industrial projects.

**4.3 Software and Programming Approach**

The software and programming approach for the smart floor cleaner involves developing a control program using the Arduino IDE, integrating various components such as the Arduino UNO, L298N motor driver, HC-05 Bluetooth module, servo motor, and water pump. The Arduino acts as the central controller, receiving commands via Bluetooth from a smartphone app through the HC-05 module. These commands, such as forward, backward, left, right, start spraying, or stop, are processed by the Arduino to control the L298N motor driver, which drives the two DC motors responsible for moving the cleaner. The servo motor is used to adjust the spray nozzle angle, ensuring proper water distribution across the floor. The water pump is controlled via a relay or transistor, switching it on or off based on Bluetooth commands. The program is designed to continuously monitor incoming data from the HC-05 and execute the corresponding motor and pump actions in real-time. This flexible and modular software design allows easy future enhancements, such as adding obstacle sensors for autonomous cleaning. The use of PWM signals enables precise speed control of the motors, while the servo motor ensures accurate water spraying direction, making the system efficient and responsive for floor cleaning applications.

The software and programming approach for the smart floor cleaner integrates various components, including the Arduino UNO, L298N motor driver, HC-05 Bluetooth module, servo motor, and water pump, all working together under a single Arduino sketch developed in the Arduino IDE. The Arduino acts as the central controller, processing commands received wirelessly via the HC-05 Bluetooth module from a smartphone app. These commands control the movement, direction, and speed of the cleaner using the L298N motor driver, which drives the two DC motors for navigation. Commands like 'F' for forward, 'B' for backward, 'L' for left, and 'R' for right determine how the cleaner moves across the floor.

Additionally, the servo motor is programmed to adjust the spray nozzle angle, ensuring water or cleaning solution covers the intended area evenly. The water pump, controlled through a relay module or transistor circuit, is switched on or off using Bluetooth commands to spray water during cleaning. The program also handles PWM signals for precise motor speed control and accurate servo positioning. The loop() function in the Arduino sketch constantly checks for incoming Bluetooth commands, processes them, and triggers the appropriate motor actions, spray control, or servo adjustments. This modular, real-time control system allows easy future upgrades, such as adding sensors for obstacle detection, automated cleaning patterns, or feedback mechanisms to monitor cleaning progress, making the software scalable, flexible, and efficient for smart floor cleaning applications.