**SSD [20OE210] Course Project Report**

**On**

**Pothole Repair Machine**

**Developed By:**

**Team 5 Members**

**Hall Ticket No: Student Name:**

**2303A58031 Thaniskha**

**2303A58010 Neha**

**2303A58014 Chetnojwala**

**2303A58032 Anuvardhan**

**2303A58024 Charan Tej**

**CHAPTER 1**

**INTRODUCTION TO EMBEDDED SYSTEMS**

Embedded Technology is now in its prime and the wealth of knowledge available is mind blowing. However, most embedded systems engineers have a common complaint. There are no comprehensive resources available over the internet which deal with the various design and implementation issues of this technology. Intellectual property regulations of many corporations are partly to blame for this and also the tendency to keep technical know-how within a restricted group of researchers.

An embedded computer is frequently a computer that is implemented for a particular purpose. In contrast, an average PC computer usually serves a number of purposes: checking email, surfing the internet, listening to music, word processing, etc... However, embedded systems usually only have a single task, or a very small number of related tasks that they are programmed to perform.

Every home has several examples of embedded computers. Any appliance that has a digital clock, for instance, has a small embedded micro-controller that performs no other task than to display the clock. Modern cars have embedded computers onboard that control such things as ignition timing and anti-lock brakes using input from a number of different sensors.

Embedded computers rarely have a generic interface, however. Even if embedded systems have a keypad and an LCD display, they are rarely capable of using many different types of input or output. An example of an embedded system with I/O capability is a security alarm with an LCD status display, and a keypad for entering a password.

An embedded system can be defined as a control system or computer system designed to perform a specific task. Common examples of embedded systems include MP3 players, navigation systems on aircraft and intruder alarm systems. An embedded system can also be defined as a single purpose computer.

Most embedded systems are time critical applications meaning that the embedded system is working in an environment where timing is very important: the results of an operation are only relevant if they take place in a specific time frame. An autopilot in an aircraft is a time critical embedded system. If the autopilot detects that the plane for some reason is going into a stall then it should take steps to correct this within milliseconds or there would be catastrophic results.

* 1. **APPLICATIONS OF EMBEDDED SYSTEM**

Embedded systems are commonly found in consumer, cooking, industrial, automotive, medical, commercial and military applications.

Telecommunications systems employ numerous embedded systems from [telephone switches](https://en.wikipedia.org/wiki/Telephone_switch" \o "Telephone switch) for the network to [cell phones](https://en.wikipedia.org/wiki/Cell_phone" \o "Cell phone) at the end user. Computer networking uses dedicated [routers](https://en.wikipedia.org/wiki/Router_(computing)" \o "Router (computing)) and [network bridges](https://en.wikipedia.org/wiki/Network_bridge" \o "Network bridge) to route data.

[Consumer electronics](https://en.wikipedia.org/wiki/Consumer_electronics" \o "Consumer electronics) include [MP3 players](https://en.wikipedia.org/wiki/MP3_player" \o "MP3 player), mobile phones, [videogame consoles](https://en.wikipedia.org/wiki/Videogame_console" \o "Videogame console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera" \o "Digital camera), [GPS](https://en.wikipedia.org/wiki/Global_Positioning_System" \o "Global Positioning System) receivers, and [printers](https://en.wikipedia.org/wiki/Computer_printer" \o "Computer printer). Household appliances, such as [microwave ovens](https://en.wikipedia.org/wiki/Microwave_oven" \o "Microwave oven), [washing machines](https://en.wikipedia.org/wiki/Washing_machine" \o "Washing machine) and [dishwashers](https://en.wikipedia.org/wiki/Dishwashers" \o "Dishwashers), include embedded systems to provide flexibility, efficiency and features. Advanced [HVAC](https://en.wikipedia.org/wiki/HVAC" \o "HVAC) systems use networked [thermostats](https://en.wikipedia.org/wiki/Thermostat" \o "Thermostat) to more accurately and efficiently control temperature that can change by time of day and [season](https://en.wikipedia.org/wiki/Season" \o "Season). [Home automation](https://en.wikipedia.org/wiki/Home_automation" \o "Home automation) uses wired- and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices for sensing and controlling.

Transportation systems from flight to automobiles increasingly use embedded systems. New airplanes contain advanced [avionics](https://en.wikipedia.org/wiki/Avionics" \o "Avionics) such as [inertial guidance systems](https://en.wikipedia.org/wiki/Inertial_guidance_system" \o "Inertial guidance system) and [GPS](https://en.wikipedia.org/wiki/Global_Positioning_System" \o "Global Positioning System) receivers that also have considerable safety requirements. Various electric motors [brushless DC motors](https://en.wikipedia.org/wiki/Brushless_DC_motor" \o "Brushless DC motor), [induction motors](https://en.wikipedia.org/wiki/Induction_motor" \o "Induction motor) and [DC motors](https://en.wikipedia.org/wiki/DC_motor" \o "DC motor)  use electric/electronic [motor controllers](https://en.wikipedia.org/wiki/Motor_controller" \o "Motor controller). [Automobiles](https://en.wikipedia.org/wiki/Automobile" \o "Automobile), [electric vehicles](https://en.wikipedia.org/wiki/Electric_vehicle" \o "Electric vehicle), and [hybrid vehicles](https://en.wikipedia.org/wiki/Hybrid_vehicle" \o "Hybrid vehicle) increasingly use embedded systems to maximize efficiency and reduce pollution. Other automotive safety systems include [anti-lock braking system](https://en.wikipedia.org/wiki/Anti-lock_braking_system" \o "Anti-lock braking system) (ABS), [Electronic Stability Control](https://en.wikipedia.org/wiki/Electronic_Stability_Control" \o "Electronic Stability Control) (ESC/ESP), [traction control](https://en.wikipedia.org/wiki/Traction_control_system" \o "Traction control system) (TCS) and automatic [four-wheel drive](https://en.wikipedia.org/wiki/Four-wheel_drive" \o "Four-wheel drive).

[Medical equipment](https://en.wikipedia.org/wiki/Medical_equipment" \o "Medical equipment) uses embedded systems for [vital signs](https://en.wikipedia.org/wiki/Vital_signs" \o "Vital signs) monitoring, [electronic stethoscopes](https://en.wikipedia.org/wiki/Electronic_stethoscope" \o "Electronic stethoscope) for amplifying sounds, and various [medical imaging](https://en.wikipedia.org/wiki/Medical_imaging" \o "Medical imaging) ([PET](https://en.wikipedia.org/wiki/Positron_emission_tomography" \o "Positron emission tomography), [SPECT](https://en.wikipedia.org/wiki/Single_photon_emission_computed_tomography" \o "Single photon emission computed tomography), [CT](https://en.wikipedia.org/wiki/Computed_tomography" \o "Computed tomography), and [MRI](https://en.wikipedia.org/wiki/Magnetic_resonance_imaging" \o "Magnetic resonance imaging)) for non-invasive internal inspections. Embedded systems within medical equipment are often powered by industrial computers.[[9]](https://en.wikipedia.org/wiki/Embedded_system" \l "cite_note-9)

Embedded systems are used in transportation, fire safety, safety and security, medical applications and life critical systems, as these systems can be isolated from hacking and thus, be more reliable. For fire safety, the systems can be designed to have greater ability to handle higher temperatures and continue to operate. In dealing with security, the embedded systems can be self-sufficient and be able to deal with cut electrical and communication systems.

* 1. **CHARACTERISTICS OF EMBEDDED SYSTEM**

Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have [real-time](https://en.wikipedia.org/wiki/Real-time_computing" \o "Real-time computing) performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems consist of small parts within a larger device that serves a more general purpose. For example, the [Gibson Robot Guitar](https://en.wikipedia.org/wiki/Gibson_Robot_Guitar" \o "Gibson Robot Guitar) features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music. Similarly, an embedded system in an [automobile](https://en.wikipedia.org/wiki/Automobile" \o "Automobile) provides a specific function as a subsystem of the car itself.

Embedded systems range from [no user interface](https://en.wikipedia.org/wiki/Headless_system" \o "Headless system) at all, in systems dedicated only to one task, to complex [graphical user interfaces](https://en.wikipedia.org/wiki/Desktop_operating_system" \l "Graphical_user_interfaces" \o "Desktop operating system) that resemble modern computer desktop operating systems. Simple embedded devices use [buttons](https://en.wikipedia.org/wiki/Push-button" \o "Push-button), [LEDs](https://en.wikipedia.org/wiki/LED" \o "LED), graphic or character [LCDs](https://en.wikipedia.org/wiki/LCD" \o "LCD) ([HD44780 LCD](https://en.wikipedia.org/wiki/Hitachi_HD44780_LCD_controller" \o "Hitachi HD44780 LCD controller) for example) with a simple [menu system](https://en.wikipedia.org/wiki/Menu_(computing)" \o "Menu (computing)).

More sophisticated devices which use a graphical screen with [touch](https://en.wikipedia.org/wiki/Touch_screen" \o "Touch screen) sensing or screen-edge buttons provide flexibility while minimizing space used: the meaning of the buttons can change with the screen, and selection involves the natural behavior of pointing at what is desired. [Handheld systems](https://en.wikipedia.org/wiki/Mobile_device" \o "Mobile device) often have a screen with a "joystick button" for a pointing device.

Some systems provide user interface remotely with the help of a serial (e.g. [RS-232](https://en.wikipedia.org/wiki/RS-232" \o "RS-232), [USB](https://en.wikipedia.org/wiki/USB" \o "USB), [I²C](https://en.wikipedia.org/wiki/I%C2%B2C" \o "I²C), etc.) or network (e.g. [Ethernet](https://en.wikipedia.org/wiki/Ethernet" \o "Ethernet)) connection. This approach gives several advantages: extends the capabilities of embedded system, avoids the cost of a display, simplifies [BSP](https://en.wikipedia.org/wiki/Board_support_package" \o "Board support package) and allows one to build a rich user interface on the PC. A good example of this is the combination of an [embedded web server](https://en.wikipedia.org/wiki/Embedded_HTTP_server" \o "Embedded HTTP server) running on an embedded device (such as an [IP camera](https://en.wikipedia.org/wiki/IP_camera" \o "IP camera)) or a [network router](https://en.wikipedia.org/wiki/Router_(computing)" \o "Router (computing)). The user interface is displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser" \o "Web browser) on a PC connected to the device, therefore needing no software to be installed.

* 1. **PROCESSORS IN EMBEDDED SYSTEMS**

Embedded processors can be broken into two broad categories. Ordinary microprocessors Embedded processors can be broken into two broad categories. Ordinary microprocessors (μP) use separate integrated circuits for memory and peripherals. Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost. In contrast to the personal computer market, many different basic [CPU architectures](https://en.wikipedia.org/wiki/CPU_architecture" \o "CPU architecture) are used, since software is custom-developed for an application and is not a commodity product installed by the end user. Both [Von Neumann](https://en.wikipedia.org/wiki/Von_Neumann_architecture" \o "Von Neumann architecture) as well as various degrees of [Harvard architectures](https://en.wikipedia.org/wiki/Harvard_architecture" \o "Harvard architecture) are used. [RISC](https://en.wikipedia.org/wiki/RISC" \o "RISC) as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bits and beyond, although the most typical remain 8/16-bit. Most architectures come in a large number of different variants and shapes, many of which are also manufactured by several different companies.

[Numerous microcontrollers](https://en.wikipedia.org/wiki/List_of_common_microcontrollers" \o "List of common microcontrollers) have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but generally require more support circuitry than microcontrollers.

(μP) use separate integrated circuits for memory and peripherals. Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost. In contrast to the personal computer market, many different basic [CPU architectures](https://en.wikipedia.org/wiki/CPU_architecture" \o "CPU architecture) are used, since software is custom-developed for an application and is not a commodity product installed by the end user. Both [Von Neumann](https://en.wikipedia.org/wiki/Von_Neumann_architecture" \o "Von Neumann architecture) as well as various degrees of [Harvard architectures](https://en.wikipedia.org/wiki/Harvard_architecture" \o "Harvard architecture) are used. [RISC](https://en.wikipedia.org/wiki/RISC" \o "RISC) as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bits and beyond, although the most typical remain 8/16-bit. Most architectures come in a large number of different variants and shapes, many of which are also manufactured by several different companies.

[Numerous microcontrollers](https://en.wikipedia.org/wiki/List_of_common_microcontrollers" \o "List of common microcontrollers) have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but generally require more support circuitry than microcontrollers.

**1.4 DEBUGGING IN EMBEDDED SYSTEMS**

Embedded [debugging](https://en.wikipedia.org/wiki/Debugging" \o "Debugging) may be performed at different levels, depending on the facilities available. The different metrics that characterize the different forms of embedded debugging are: does it slow down the main application, how close is the debugged system or application to the actual system or application, how expressive are the triggers that I can set for debugging (e.g., I want to inspect the memory when a particular [program counter](https://en.wikipedia.org/wiki/Program_counter" \o "Program counter) value is reached), and what can I inspect in the debugging process (such as, only memory, or memory and registers, etc.).

From simplest to most sophisticated they can be roughly grouped into the following areas:

Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)

External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the [Remedy Debugger](https://en.wikipedia.org/wiki/Remedy_Debugger" \o "Remedy Debugger) which even works for heterogeneous [multicore](https://en.wikipedia.org/wiki/Multi-core_processor" \o "Multi-core processor) systems.

An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a [JTAG](https://en.wikipedia.org/wiki/JTAG" \o "JTAG) or [Nexus](https://en.wikipedia.org/wiki/Nexus_(standard)" \o "Nexus (standard)) interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.

An [in-circuit emulator](https://en.wikipedia.org/wiki/In-circuit_emulator" \o "In-circuit emulator) (ICE) replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.

A complete [emulator](https://en.wikipedia.org/wiki/Emulator" \o "Emulator) provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified, and allowing debugging on a normal PC. The downsides are expense and slow operation, in some cases up to 100 times slower than the final system.

For SoC designs, the typical approach is to verify and debug the design on an FPGA prototype board. Tools such as Certus[[11]](https://en.wikipedia.org/wiki/Embedded_system" \l "cite_note-12) are used to insert probes in the FPGA RTL that make signals available for observation. This is used to debug hardware, firmware and software interactions across multiple FPGA with capabilities similar to a logic analyzer.

Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as [HLL](https://en.wikipedia.org/wiki/High-level_programming_language" \o "High-level programming language) [source-code](https://en.wikipedia.org/wiki/Source-code" \o "Source-code), [assembly code](https://en.wikipedia.org/wiki/Assembly_code" \o "Assembly code) or mixture of both.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software- (and microprocessor-) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, and co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a [logic analyzer](https://en.wikipedia.org/wiki/Logic_analyzer" \o "Logic analyzer), for instance.

**1.5 RELIABILITY**

Embedded systems often reside in machines that are expected to run continuously for years without errors, and in some cases recover by themselves if an error occurs. Therefore, the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

* The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
* The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals.
* The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

A variety of techniques are used, sometimes in combination, to recover from errors—both software bugs such as [memory leaks](https://en.wikipedia.org/wiki/Memory_leak" \o "Memory leak), and also [soft errors](https://en.wikipedia.org/wiki/Soft_error" \o "Soft error) in the hardware:

* [watchdog timer](https://en.wikipedia.org/wiki/Watchdog_timer" \o "Watchdog timer) that resets the computer unless the software periodically notifies the watchdog subsystems with redundant spares that can be switched over to software "limp modes" that provide partial function
* Designing with a [Trusted Computing Base](https://en.wikipedia.org/wiki/Trusted_Computing_Base" \o "Trusted Computing Base) (TCB) architecture[[12]](https://en.wikipedia.org/wiki/Embedded_system" \l "cite_note-13) ensures a highly secure & reliable system environment
* A [hypervisor](https://en.wikipedia.org/wiki/Hypervisor" \o "Hypervisor) designed for embedded systems, is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
* [Immunity Aware Programming](https://en.wikipedia.org/wiki/Immunity_Aware_Programming" \o "Immunity Aware Programming)

**1.6 TRACING**

Real-time operating systems ([RTOS](https://en.wikipedia.org/wiki/RTOS" \o "RTOS)) often supports [tracing](https://en.wikipedia.org/wiki/Tracing_(software)" \o "Tracing (software)) of operating system events. A graphical view is presented by a host PC tool, based on a recording of the system behavior. The trace recording can be performed in software, by the RTOS, or by special tracing hardware. RTOS tracing allows developers to understand timing and performance issues of the software system and gives a good understanding of the high-level system behaviors. Commercial tools like [RTXC Quadros](https://en.wikipedia.org/wiki/RTXC_Quadros" \o "RTXC Quadros) or [IAR Systems](https://en.wikipedia.org/wiki/IAR_Systems" \o "IAR Systems) exists.

**CHAPTER 2**

**WIRELESS POTHOLE REPAIRING ROBOT**

**INTRODUCTION**

Potholes on roads are a common issue causing damage to vehicles, accidents, and inconvenience. Manual repair methods are time-consuming, labor-intensive, and expose workers to unsafe conditions. The "Wireless Pothole Repairing Robot" is an innovative solution that automates the process of repairing potholes. Controlled wirelessly, the robot uses advanced hardware components to simplify and accelerate the repair process, ensuring safety and efficiency.

**SYSTEM DESIGN**

The system design consists of the following components:

1. **Arduino Uno**: Acts as the brain of the system, controlling all other components.
2. **HC-05 Bluetooth Module**: Provides wireless communication for controlling the robot remotely.
3. **Li-Ion Batteries**: Power source for the entire robot.
4. **L298N Motor Driver Module**: Controls the motors for mobility.
5. **Motors (Motor 1 & Motor 2)**: Responsible for moving the robot to the pothole location.
6. **Relay:** For operating the pump
7. **9 volts Battery:** For extra volts to support pump
8. **Pump**: Used for dispensing repair material into the pothole.
9. **Servo Motor:** For working of Wiper.
10. **Wiper:** It is used to level the road after pumping
11. **Wireless Control**: Commands are sent via a smartphone or another Bluetooth-enabled device.

**OBJECTIVES**

 Automate the process of pothole repair using a robotic system.

 Enhance worker safety by minimizing manual labor in hazardous conditions.

 Ensure faster and more efficient road repair.

 Reduce traffic disruptions caused by road maintenance.

 Provide a cost-effective and scalable solution for infrastructure maintenance.

**WORKING**

 The vehicle is powered by Li-Ion batteries and is wirelessly controlled using the HC-05 Bluetooth module. A user sends commands through a smartphone or remote device to navigate the robot to the target pothole. The motors, controlled by the L298N motor driver, ensure smooth movement of the robot. Once positioned over the pothole, the L298N controls the pump to dispense repair material (e.g., asphalt mix or filler). After the dispensing is completed the wiper is turned on and then it used for leveling the road. The repair process is completed, and the robot moves to the next location, ensuring minimal human intervention.

**ADVANTAGES**

 **Safety**: Reduces risks to workers involved in road repair.

 **Efficiency**: Speeds up the pothole repair process.

 **Cost-Effective**: Reduces manpower and operational costs in the long run.

 **Precision**: Ensures proper and consistent filling of potholes.

 **Scalable**: Can be adapted for larger-scale road maintenance operations.

**APPLICATIONS**

 Road maintenance by municipal corporations.

 Highway repair tasks to ensure smooth traffic flow.

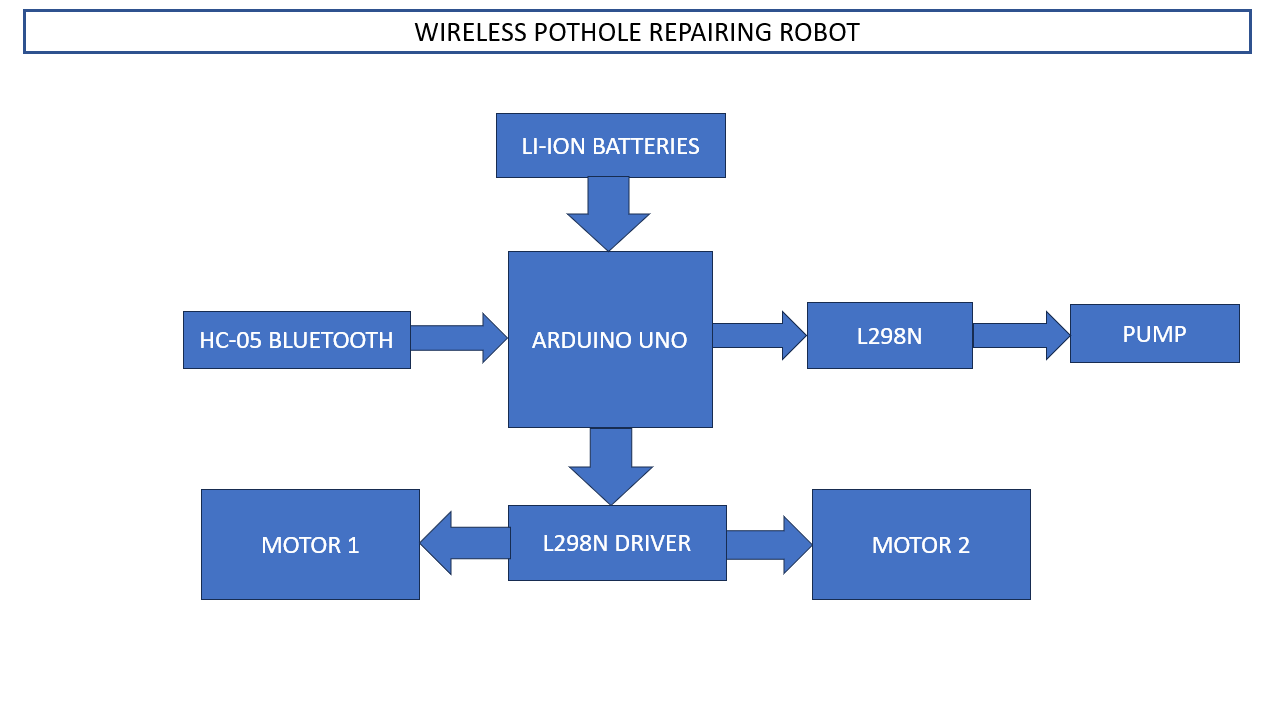
 Industrial and private estate infrastructure repair.

 Repair of parking lots and internal roads in housing societies.

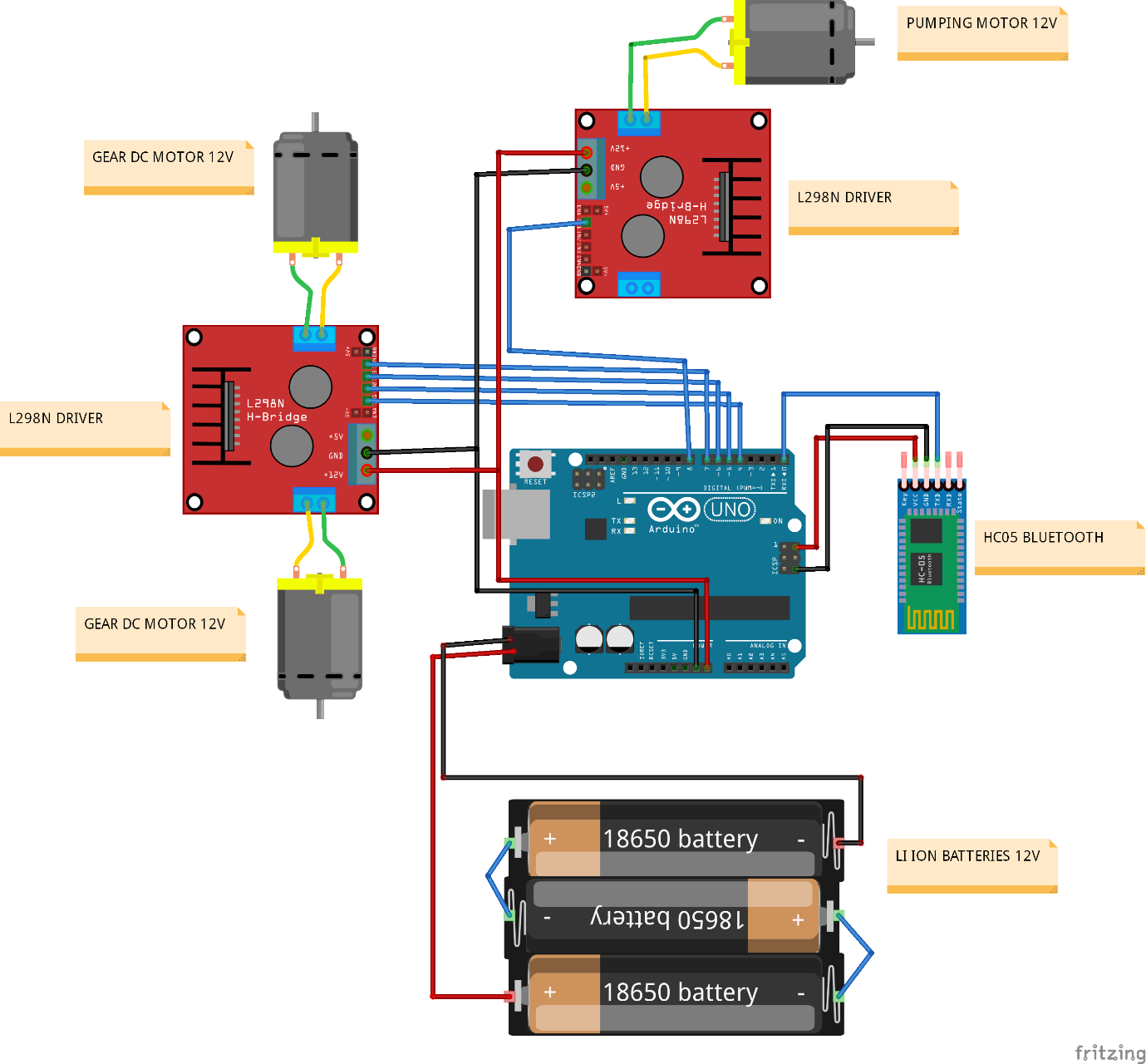
**FUTURE SCOPE**

1. **Integration with AI**: Use of artificial intelligence for pothole detection and autonomous navigation.
2. **Enhanced Materials**: Automatic mixing and dispensing of advanced repair materials.
3. **Improved Control Systems**: Use of Wi-Fi or GSM for long-distance control.
4. **Modular Design**: Incorporation of additional features like surface leveling and quality assurance.
5. **Solar-Powered Operation**: Increasing energy efficiency and environmental sustainability.

**BLOCK DIAGRAM**

****

**CIRCUIT DIAGRAM**

****

**CHAPTER 3**

**HARDWARE REQUIREMENTS**

Hardware Components of this project are

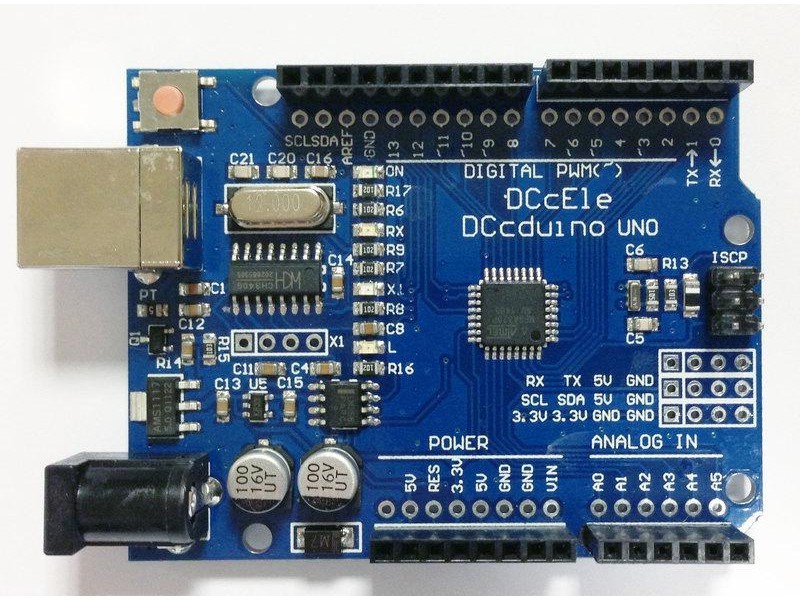
1. ARDUINO UNO
2. L298N
3. WATER PUMP
4. DC GEAR MOTOR
5. HC-05 BLUETOOTH
6. LI-ION BATTERIES

**4.1 INTRODUCTION TO ARDUINO UNO**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/" \t "_blank)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/" \t "_blank).

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of [accessible knowledge](http://forum.arduino.cc/) that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The [software](https://www.arduino.cc/en/Main/Software), too, is open-source, and it is growing through the contributions of users worldwide.



**4.1.1 WHY ARDUINO**

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

**4.1.2 ADVANTAGES OF ARDUINO**

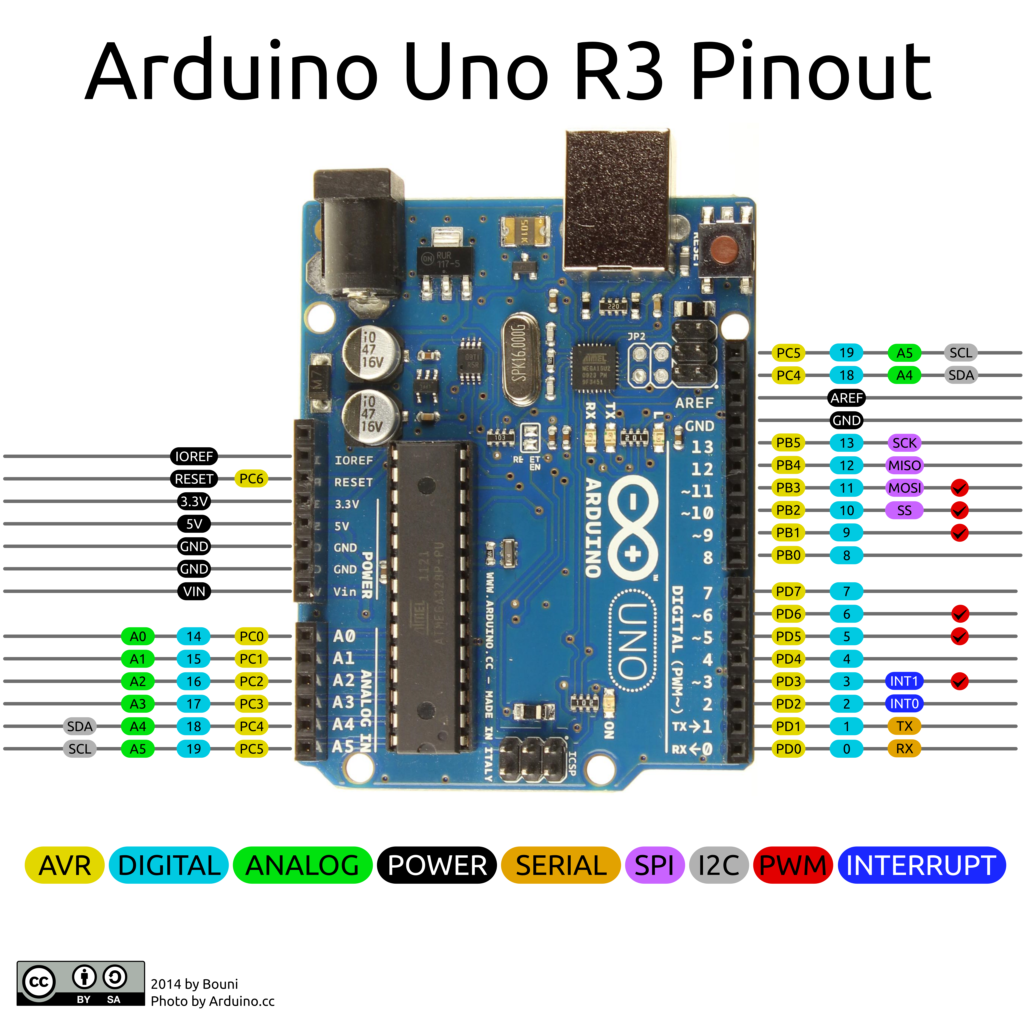
* **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
* **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
* **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
* **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
* **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the [breadboard version of the module](https://www.arduino.cc/en/Main/Standalone) in order to understand how it works and save money.

**4.1.3 FEATURES OF ARDUINO UNO**

The **Arduino Uno** is a microcontroller board based on the ATmega328. Arduino is an open-source, prototyping platform and its simplicity makes it ideal for  hobbyists to use as well as professionals. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

**Features of the Arduino UNO:**

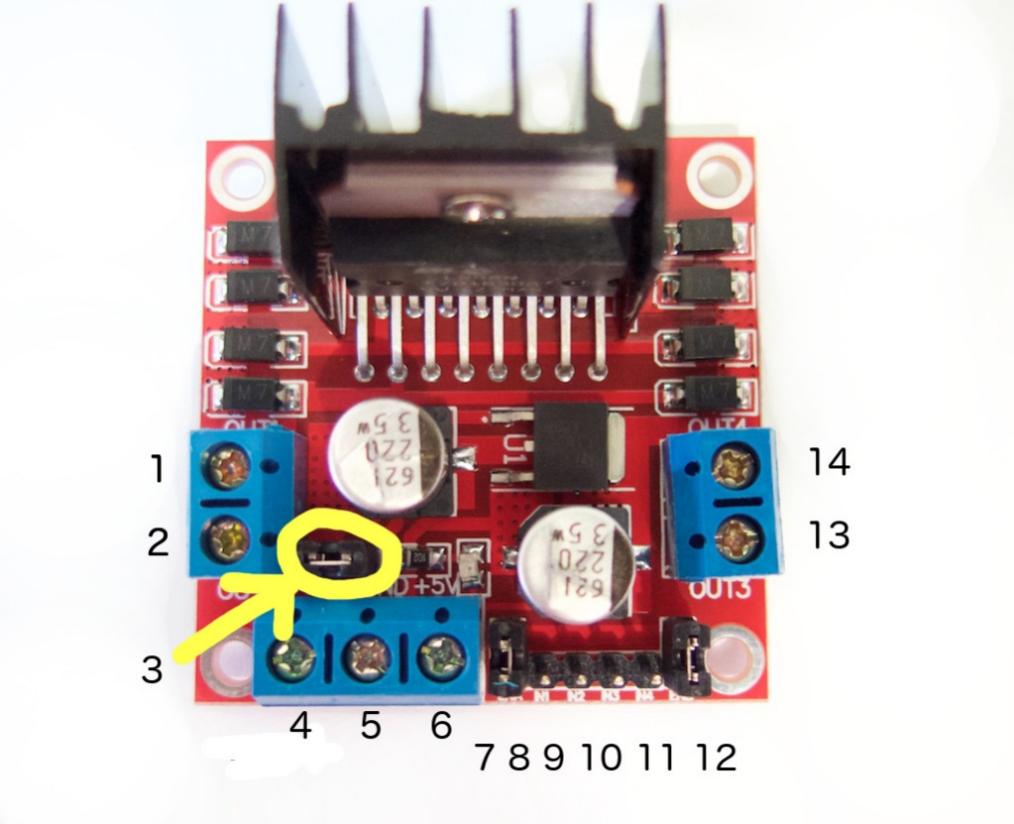
* Microcontroller: ATmega328
* Operating Voltage: 5V
* Input Voltage (recommended): 7-12V
* Input Voltage (limits): 6-20V
* Digital I/O Pins: 14 (of which 6 provide PWM output)
* Analog Input Pins: 6
* DC Current per I/O Pin: 40 mA
* DC Current for 3.3V Pin: 50 mA
* Flash Memory: 32 KB of which 0.5 KB used by bootloader
* SRAM: 2 KB (ATmega328)
* EEPROM: 1 KB (ATmega328)
* Clock Speed: 16 MHz



**4.2 INTRODUCTION TO L293D MOTOR DRIVER**

[Dual Motor Controller](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/) Module 2A with Arduino. This allows you to control the speed and direction of two DC motors, or control one bipolar stepper motor with ease. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC.

There is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board. These L298 H-bridge dual motor controller modules [.](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/)



**4.2.1 PIN DESCRIPTION**

1. DC motor 1 "+" or stepper motor A+
2. DC motor 1 "-" or stepper motor A-
3. 12V jumper - remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
13. DC motor 2 "+" or stepper motor B+
14. DC motor 2 "-" or stepper motor B-

**4.2.2 CONTROLLING DC MOTOR**

To control one or two DC motors is quite easy. First connect each motor to the A and B connections on the L298N  [module](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/). If you're using two motors for a robot (etc) ensure that the polarity of the motors is the same on both inputs. Otherwise you may need to swap them over when you set both motors to forward and one goes backwards!

Next, connect your power supply - the positive to pin 4 on the module and negative/GND to pin 5. If you supply is up to 12V you can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module. This can be fed to your Arduino's 5V pin to power it from the motors' power supply. Don't forget to connect Arduino GND to pin 5 on the module as well to complete the circuit.

Now you will need six digital output pins on your Arduino, two of which need to be PWM (pulse-width modulation) pins. PWM pins are denoted by the tilde ("~") next to the pin number, for example: Finally, connect the Arduino digital output pins to the driver module. In our example we have two DC motors, so digital pins D9, D8, D7 and D6 will be connected to pins IN1, IN2, IN3 and IN4 respectively. Then connect D10 to module pin 7 (remove the jumper first) and D5 to module pin 12 (again, remove the jumper).

The motor direction is controlled by sending a HIGH or LOW signal to the drive for each motor (or channel). For example for motor one, a HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and  a LOW and HIGH will cause it to turn in the other direction. However the motors will not turn until a HIGH is set to the enable pin (7 for motor one, 12 for motor two). And they can be turned off with a LOW to the same pin(s). However if you need to control the speed of the motors, the PWM signal from the digital pin connected to the enable pin can take care of it.

**4.2 INTRODUCTION TO WATER PUMP**

A **pump** is a device that moves fluids ([liquids](https://en.wikipedia.org/wiki/Liquid" \o "Liquid) or [gases](https://en.wikipedia.org/wiki/Gas" \o "Gas)), or sometimes [slurries](https://en.wikipedia.org/wiki/Slurry" \o "Slurry), by mechanical action. Pumps can be classified into three major groups according to the method they use to move the fluid: direct lift, displacement, and gravity pumps.

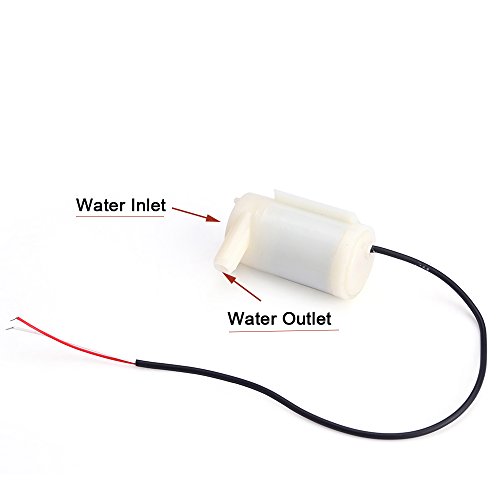
Pumps operate by some mechanism (typically [reciprocating](https://en.wikipedia.org/wiki/Reciprocating_motion" \o "Reciprocating motion) or [rotary](https://en.wikipedia.org/wiki/Rotation" \o "Rotation)), and consume [energy](https://en.wikipedia.org/wiki/Energy" \o "Energy) to perform [mechanical work](https://en.wikipedia.org/wiki/Mechanical_work" \o "Mechanical work) by moving the fluid. Pumps operate via many energy sources, including manual operation, [electricity](https://en.wikipedia.org/wiki/Electricity" \o "Electricity), [engines](https://en.wikipedia.org/wiki/Engines" \o "Engines), or [wind power](https://en.wikipedia.org/wiki/Wind_power" \o "Wind power), come in many sizes, from microscopic for use in medical applications to large industrial pumps.

Mechanical pumps serve in a wide range of applications such as [pumping water from wells](https://en.wikipedia.org/wiki/Water_well_pump" \o "Water well pump), [aquarium filtering](https://en.wikipedia.org/wiki/Aquarium_filter" \o "Aquarium filter), [pond filtering](https://en.wikipedia.org/wiki/Pond" \o "Pond) and [aeration](https://en.wikipedia.org/wiki/Aeration" \o "Aeration), in the [car industry](https://en.wikipedia.org/wiki/Car_industry" \o "Car industry) for [water-cooling](https://en.wikipedia.org/wiki/Water_cooling" \o "Water cooling) and [fuel injection](https://en.wikipedia.org/wiki/Fuel_injection" \o "Fuel injection), in the [energy industry](https://en.wikipedia.org/wiki/Energy_industry" \o "Energy industry) for [pumping oil](https://en.wikipedia.org/wiki/Pumping_(oil_well)" \o "Pumping (oil well)) and [natural gas](https://en.wikipedia.org/wiki/Natural_gas" \o "Natural gas) or for operating [cooling towers](https://en.wikipedia.org/wiki/Cooling_tower" \o "Cooling tower). In the [medical industry](https://en.wikipedia.org/wiki/Medical_industry" \o "Medical industry), pumps are used for biochemical processes in developing and manufacturing medicine, and as artificial replacements for body parts, in particular the [artificial heart](https://en.wikipedia.org/wiki/Artificial_heart" \o "Artificial heart) and [penile prosthesis](https://en.wikipedia.org/wiki/Penile_prosthesis" \o "Penile prosthesis).

Single stage pump - When in a casing only one impeller is revolving then it is called single stage pump.

Double/ Multi stage pump - When in a casing two or more than two impellers are revolving then it is called double/ multi stage pump.

In biology, many different types of chemical and bio-mechanical pumps have [evolved](https://en.wikipedia.org/wiki/Evolutionary_biology" \o "Evolutionary biology), and [biomimicry](https://en.wikipedia.org/wiki/Biomimicry" \o "Biomimicry) is sometimes used in developing new types of mechanical pumps.



**4.3 INTRODUCTION TO DC GEAR MOTOR**

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The [universal motor](https://en.wikipedia.org/wiki/Universal_motor" \o "Universal motor) can operate on direct current but is a lightweight motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and hoists, or in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with [AC motors](https://en.wikipedia.org/wiki/AC_motors" \o "AC motors) possible in many applications.



**4.3.1 BRUSHED DC MOTOR**

The [brushed DC electric motor](https://en.wikipedia.org/wiki/Brushed_DC_electric_motor" \o "Brushed DC electric motor) generates torque directly from DC power supplied to the motor by using internal commutation, stationary magnets ([permanent](https://en.wikipedia.org/wiki/Magnet" \o "Magnet) or [electromagnets](https://en.wikipedia.org/wiki/Electromagnet" \o "Electromagnet)), and rotating electrical magnets.

Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Disadvantages are high maintenance and low life-span for high intensity uses. Maintenance involves regularly replacing the carbon brushes and springs which carry the electric current, as well as cleaning or replacing the [commutator](https://en.wikipedia.org/wiki/Commutator_(electric)" \o "Commutator (electric)). These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor. Brushes consist of conductors.

**4.3.2 BRUSHLESS DC MOTOR**

Typical brushless DC motors use one or more permanent magnets in the rotor and [electromagnets](https://en.wikipedia.org/wiki/Electromagnet" \o "Electromagnet) on the motor housing for the stator. A motor controller converts DC to [AC](https://en.wikipedia.org/wiki/Alternating_current" \o "Alternating current). This design is mechanically simpler than that of brushed motors because it eliminates the complication of transferring power from outside the motor to the spinning rotor. The motor controller can sense the rotor's position via [Hall effect](https://en.wikipedia.org/wiki/Hall_effect" \o "Hall effect) sensors or similar devices and can precisely control the timing, phase, etc., of the current in the rotor coils to optimize torque, conserve power, regulate speed, and even apply some braking. Advantages of brushless motors include long life span, little or no maintenance, and high efficiency. Disadvantages include high initial cost, and more complicated motor speed controllers. Some such brushless motors are sometimes referred to as "synchronous motors" although they have no external power supply to be synchronized with, as would be the case with normal AC synchronous motors.

**4.3.3 PERMANENT MAGNET STATOR**

A PM motor does not have a field winding on the stator frame, instead relying on PMs to provide the magnetic field against which the rotor field interacts to produce torque. Compensating windings in series with the armature may be used on large motors to improve commutation under load. Because this field is fixed, it cannot be adjusted for speed control. PM fields (stators) are convenient in miniature motors to eliminate the power consumption of the field winding. Most larger DC motors are of the "dynamo" type, which have stator windings. Historically, PMs could not be made to retain high flux if they were disassembled; field windings were more practical to obtain the needed amount of flux. However, large PMs are costly, as well as dangerous and difficult to assemble; this favors wound fields for large machines.

To minimize overall weight and size, miniature PM motors may use high energy magnets made with [neodymium](https://en.wikipedia.org/wiki/Neodymium" \o "Neodymium) or other strategic elements; most such are neodymium-iron-boron alloy. With their higher flux density, electric machines with high-energy PMs are at least competitive with all optimally designed [singly fed](https://en.wikipedia.org/wiki/DC_motor" \l "Singly_fed_electric_motor) synchronous and induction electric machines. Miniature motors resemble the structure in the illustration, except that they have at least three rotor poles (to ensure starting, regardless of rotor position) and their outer housing is a steel tube that magnetically links the exteriors of the curved field magnets.

**4.3.4 SERIES CONNECTION**

A series DC motor connects the [armature](https://en.wikipedia.org/wiki/Armature_(electrical_engineering)" \o "Armature (electrical engineering)) and [field windings](https://en.wikipedia.org/wiki/Field_coil" \o "Field coil) in [series](https://en.wikipedia.org/wiki/Series_circuits" \o "Series circuits) with a [common](https://en.wikipedia.org/wiki/Battery_(electricity)" \o "Battery (electricity)) D.C. power source. The motor speed varies as a non-linear function of load torque and armature current; current is common to both the stator and rotor yielding current squared (I^2) behavior. A series motor has very high starting torque and is commonly used for starting high inertia loads, such as trains, elevators or hoists.[[2]](https://en.wikipedia.org/wiki/DC_motor" \l "cite_note-2) This speed/torque characteristic is useful in applications such as [dragline excavators](https://en.wikipedia.org/wiki/Dragline_excavator" \o "Dragline excavator), where the digging tool moves rapidly when unloaded but slowly when carrying a heavy load.

A series motor should never be started at no load. With no mechanical load on the series motor, the current is low, the counter-EMF produced by the field winding is weak, and so the armature must turn faster to produce sufficient counter-EMF to balance the supply voltage. The motor can be damaged by overspeed. This is called a runaway condition.

Series motors called [universal motors](https://en.wikipedia.org/wiki/Universal_motor" \o "Universal motor) can be used on [alternating current](https://en.wikipedia.org/wiki/Alternating_current" \o "Alternating current). Since the armature voltage and the field direction reverse at the same time, torque continues to be produced in the same direction. However they run at a lower speed with lower torque on AC supply when compared to DC due to [reactance](https://en.wikipedia.org/wiki/Electrical_reactance" \o "Electrical reactance) voltage drop in AC which is not present in DC.[[3]](https://en.wikipedia.org/wiki/DC_motor" \l "cite_note-3)Since the speed is not related to the line frequency, universal motors can develop higher-than-synchronous speeds, making them lighter than induction motors of the same rated mechanical output. This is a valuable characteristic for hand-held power tools. Universal motors for commercial [utility](https://en.wikipedia.org/wiki/Utility_frequency" \o "Utility frequency) are usually of small capacity, not more than about 1 kW output. However, much larger universal motors were used for electric locomotives, fed by special low-frequency [traction power networks](https://en.wikipedia.org/wiki/Traction_power_network" \o "Traction power network) to avoid problems with commutation under heavy and varying loads.

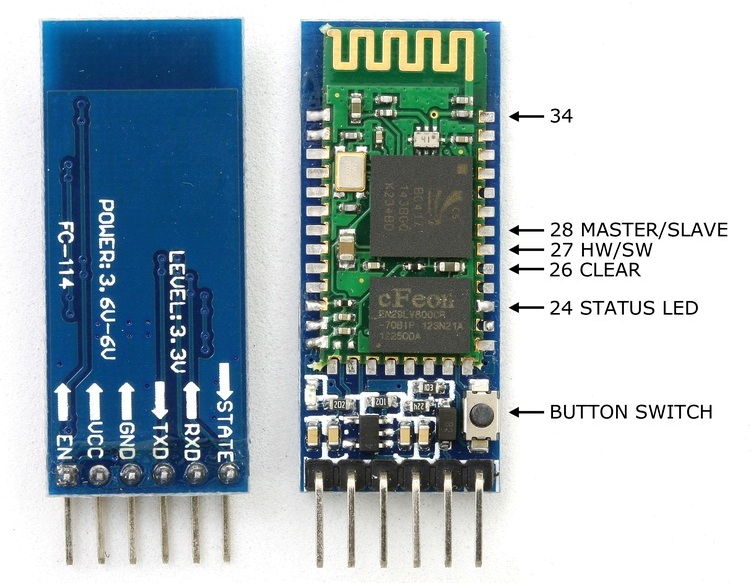
**4.3.5 SHUNT CONNECTION**

A shunt DC motor connects the armature and field windings in parallel or shunt with a common D.C. power source. This type of motor has good speed regulation even as the load varies, but does not have the starting torque of a series DC motor.[[4]](https://en.wikipedia.org/wiki/DC_motor" \l "cite_note-4) It is typically used for industrial, adjustable speed applications, such as machine tools, winding/unwinding machines and tensioners.



**4.8.1 BLUETOOTH MODULE HC-05**

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default, the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.



**4.8.2 HARDWARE FEATURES**

* Typical ‐80dBm sensitivity.
* Up to +4dBm RF transmit power.
* 3.3 to 5 V I/O.
* PIO (Programmable Input/Output) control.
* UART interface with programmable baud rate.
* With integrated antenna.
* With edge connector.

**4.8.3 SOFTWARE FEATURES**

* Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
* Auto‐connect to the last device on power as default.
* Permit pairing device to connect as default.
* Auto‐pairing PINCODE:”1234” as default.

**4.8.4 PIN DESCRIPTION**

The HC-05 Bluetooth Module has 6pins. They are as follows:

**ENABLE:**

When enable is pulled **LOW**, the module is disabled which means the module will **not turn on** and it **fails to communicate**.When enable is **left open or connected to 3.3V**, the module is enabled i.e the module **remains on**and **communication also takes place**.

**Vcc:**

Supply Voltage 3.3V to 5V

**GND:**

Ground pin

**TXD & RXD:**

These two pins acts as an UART interface for communication

**STATE:**

It acts as a status indicator.When the module is **not connected to / paired** with any other bluetooth device,signal goes **Low**.At this **low state**,the **led flashes continuously** which denotes that the module is **not paired** with other device.When this module is **connected to/paired** with any other bluetooth device,the signal goes **High**.At this **high state**,the **led blinks with a constant delay** say for example 2s delay which indicates that the module is **paired**.

**BUTTON SWITCH:**

This is used to switch the module into AT command mode.To enable AT command mode,press the button switch for a second.With the help of AT commands,the user can change the parameters of this module but only when the module is not paired with any other BT device.If the module is connected to any other bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

**4.6 INTRODUCTION TO 18650 BATTERY**



A **lithium-ion battery** or **Li-ion battery** is a type of [rechargeable battery](https://en.wikipedia.org/wiki/Rechargeable_battery" \o "Rechargeable battery) in which [lithium](https://en.wikipedia.org/wiki/Lithium" \o "Lithium) [ions](https://en.wikipedia.org/wiki/Ion" \o "Ion) move from the negative [electrode](https://en.wikipedia.org/wiki/Electrode" \o "Electrode) through an [electrolyte](https://en.wikipedia.org/wiki/Electrolyte" \o "Electrolyte) to the positive electrode during discharge, and back when charging. Li-ion batteries use an [intercalated](https://en.wikipedia.org/wiki/Intercalation_(chemistry)" \o "Intercalation (chemistry)) lithium [compound](https://en.wikipedia.org/wiki/Chemical_compound" \o "Chemical compound) as the material at the positive electrode and typically [graphite](https://en.wikipedia.org/wiki/Graphite" \o "Graphite) at the negative electrode.

Li-ion batteries have a high [energy density](https://en.wikipedia.org/wiki/Energy_density" \o "Energy density), no [memory effect](https://en.wikipedia.org/wiki/Memory_effect" \o "Memory effect) (other than [LFP cells](https://en.wikipedia.org/wiki/Lithium_iron_phosphate_battery" \o "Lithium iron phosphate battery))[[9]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-9) and low [self-discharge](https://en.wikipedia.org/wiki/Self-discharge" \o "Self-discharge). Cells can be manufactured to either prioritize energy or power density. [[10]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-10) They can however be a safety hazard since they contain flammable electrolytes, and if damaged or incorrectly charged can lead to explosions and fires. [Samsung](https://en.wikipedia.org/wiki/Samsung" \o "Samsung) was forced to recall [Galaxy Note 7](https://en.wikipedia.org/wiki/Samsung_Galaxy_Note_7" \l "Battery_faults" \o "Samsung Galaxy Note 7) handsets following lithium-ion fires,[[11]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-s7NYT-11) and there have been several incidents involving batteries on [Boeing 787s](https://en.wikipedia.org/wiki/Boeing_787_Dreamliner_battery_problems" \o "Boeing 787 Dreamliner battery problems).

A prototype Li-ion battery was developed by [Akira Yoshino](https://en.wikipedia.org/wiki/Akira_Yoshino" \o "Akira Yoshino) in 1985, based on earlier research by [John Goodenough](https://en.wikipedia.org/wiki/John_Goodenough" \o "John Goodenough), [M. Stanley Whittingham](https://en.wikipedia.org/wiki/M._Stanley_Whittingham" \o "M. Stanley Whittingham), [Rachid Yazami](https://en.wikipedia.org/wiki/Rachid_Yazami" \o "Rachid Yazami) and [Koichi Mizushima](https://en.wikipedia.org/wiki/Koichi_Mizushima_(scientist)" \o "Koichi Mizushima (scientist)) during the 1970s–1980s,[[12]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-ieee-12)[[13]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-nobel-13)[[14]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-NIMS-14) and then a commercial Li-ion battery was developed by a [Sony](https://en.wikipedia.org/wiki/Sony" \o "Sony) and [Asahi Kasei](https://en.wikipedia.org/wiki/Asahi_Kasei" \o "Asahi Kasei) team led by Yoshio Nishi in 1991.[[15]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-NAE-15)

Lithium-ion batteries are commonly used for [portable electronics](https://en.wikipedia.org/wiki/Portable_electronics" \o "Portable electronics) and [electric vehicles](https://en.wikipedia.org/wiki/Electric_vehicle" \o "Electric vehicle) and are growing in popularity for military and [aerospace](https://en.wikipedia.org/wiki/Aerospace" \o "Aerospace) applications. Chemistry, performance, cost and safety characteristics vary across types of lithium-ion batteries. Handheld electronics mostly use [lithium polymer batteries](https://en.wikipedia.org/wiki/Lithium_polymer_battery" \o "Lithium polymer battery) (with a polymer gel as electrolyte), a [lithium cobalt oxide](https://en.wikipedia.org/wiki/Lithium_cobalt_oxide" \o "Lithium cobalt oxide) (LiCoO3-based lithium rich layered materials, LMR-NMC), and [lithium nickel manganese cobalt oxide](https://en.wikipedia.org/wiki/Lithium_nickel_manganese_cobalt_oxide" \o "Lithium nickel manganese cobalt oxide) (LiNiMnCoO2 or NMC) may offer longer lives and may have better rate capability. Such batteries are widely used for electric tools, medical equipment, and other roles. NMC and its derivatives are widely used in electric vehicles.

Research areas for lithium-ion batteries include extending lifetime, increasing energy density, improving safety, reducing cost, and increasing charging speed,[[19]](https://en.wikipedia.org/wiki/Lithium-ion_battery" \l "cite_note-19) among others. Research has been under way in the area of non-flammable electrolytes as a pathway to increased safety based on the flammability and volatility of the organic solvents used in the typical electrolyte. Strategies include [aqueous lithium-ion batteries](https://en.wikipedia.org/wiki/Aqueous_lithium-ion_battery" \o "Aqueous lithium-ion battery), ceramic solid electrolytes, polymer electrolytes, ionic liquids, and heavily fluorinated systems

**CHAPTER 5**

**PROGRAM CODE**

#include <Servo.h> // Include the Servo library

char t;

Servo myServo; // Create a Servo object

void setup() {

pinMode(13, OUTPUT); // Left motors forward

pinMode(12, OUTPUT); // Left motors reverse

pinMode(11, OUTPUT); // Right motors forward

pinMode(10, OUTPUT); // Right motors reverse

pinMode(9, OUTPUT); // LED

myServo.attach(6); // Attach the servo to pin 6

myServo.write(0); // Initialize servo at 0 degrees

Serial.begin(9600);

}

void loop() {

if (Serial.available()) {

t = Serial.read();

Serial.println(t); // Print received command for debugging

}

if (t == 'F') { // Move forward

digitalWrite(13, HIGH);

digitalWrite(11, HIGH);

Serial.println("Moving Forward");

t = '\0'; // Clear command

}

else if (t == 'B') { // Move reverse

digitalWrite(12, HIGH);

digitalWrite(10, HIGH);

Serial.println("Moving Backward");

t = '\0';

}

else if (t == 'L') { // Turn right

digitalWrite(11, HIGH);

digitalWrite(13, LOW);

Serial.println("Turning Right");

t = '\0';

}

else if (t == 'R') { // Turn left

digitalWrite(13, HIGH);

digitalWrite(11, LOW);

Serial.println("Turning Left");

t = '\0';

}

else if (t == 'W') { // Turn LED on

digitalWrite(9, HIGH);

Serial.println("LED ON");

t = '\0';

}

else if (t == 'w') { // Turn LED off

digitalWrite(9, LOW);

Serial.println("LED OFF");

t = '\0';

}

else if (t == 'S') { // STOP all motors

digitalWrite(13, LOW);

digitalWrite(12, LOW);

digitalWrite(11, LOW);

digitalWrite(10, LOW);

Serial.println("STOP");

t = '\0';

}

else if (t == 'V') { // Activate servo

myServo.write(90); // Move servo to 90 degrees

Serial.println("Servo Activated");

t = '\0';

}

else if (t == 'v') { // Deactivate servo

myServo.write(0); // Move servo back to 0 degrees

Serial.println("Servo Deactivated");

t = '\0';

}

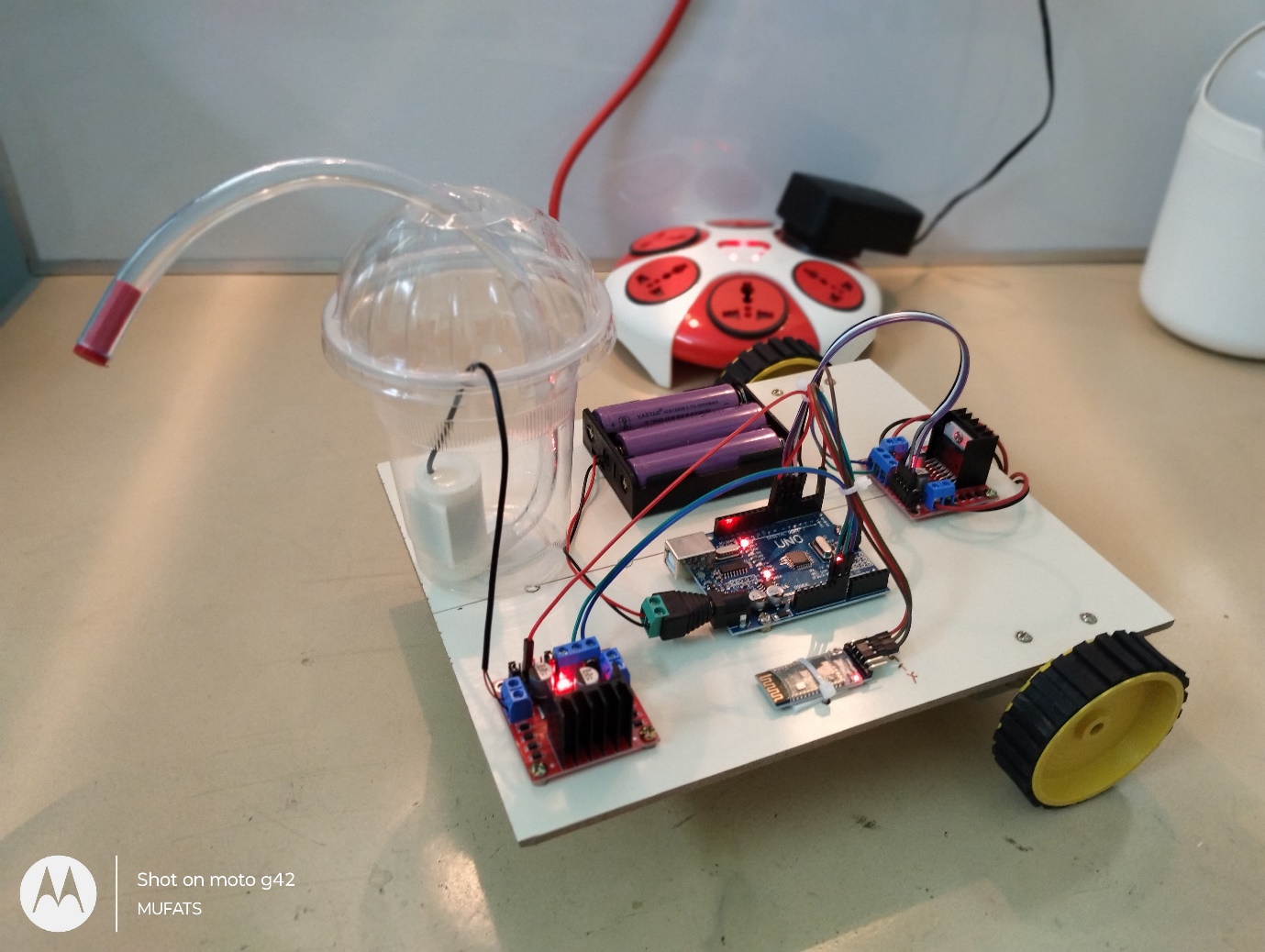
// Optional: reduce delay to 5 ms or remove it entirely for faster response

delay(5);

}

**CHAPTER 5**

**RESULT**

****

**CHAPTER 7**

**CONCLUSION**

The "Wireless Pothole Repairing Robot" is a transformative step toward automating road maintenance, addressing one of the most persistent problems faced by urban and rural infrastructure. By leveraging wireless control, robotic precision, and efficient repair mechanisms, this system ensures safer and quicker pothole repairs, reducing manual labor and traffic disruptions. With its cost-effectiveness and scalability, the robot holds great potential for widespread adoption in road maintenance projects. Future enhancements, such as integrating AI for autonomous operations and advanced materials for durable repairs, can further revolutionize the way we approach infrastructure maintenance, paving the way for smarter and safer cities.