# SMART MULTIPURPOSE ROBOTIC ARM CAR

A thesis report submitted to the department of mechanical engineering for the partial fulfillment of the degree of Bachelor of Science in Mechanical Engineering.

**A Thesis by**


**Al Fahad Hridoy (L)**
ID No:ME2101023231

**Bindu Rahaman Bristy**
ID No:ME2101023474

**Probhat Kumar Biswas**
ID No:ME2101023440

**Tonmoy Samadder**
ID No:ME2101023488

**Abdul Alim**
ID No:ME2101023424

**Md.Mominul Islam**
ID No:BME2001020600


…….

Supervisor:
**Md. Minhaz Uddin**
Assistant prof. of Mechanical Engineering





**DEPARTMENT OF MECHANICAL ENGINEERING**
**SONARGAON UNIVERSITY (SU)**


Dhaka, Bangladesh
September, 2024

# APPROVAL

This is to certify that the project on **"SMART MULTIPURPOSE ROBOTIC ARM CAR."** By Al Fahad Hridoy (L) (ID No: ME2101023231), Abdul Alim (ID No: ME2101023424), Probhat Kumar Biswas (ID No: ME2101023440), Bindu Rahaman Bristy (ID No: ME2101023474),Tonmoy Samadder (ID No:ME2101023488) & Md. Mominul Islam (ID No:BME2001020600) has been carried out under our supervision. The project has been carried out in partial fulfillment of the requirements of the degree of **Bachelor of Science (B.Sc.) in Mechanical Engineering** of years of 2024 and has been approved as to its style and contents.

**…….**
**Md. Minhaz Uddin**
Assistant Prof. of Mechanical Engineering
**Sonargaon University (SU)**

# DECLERATION

We, hereby, declare that the work presented in this project is the outcome of the investigation and research work performed by us under the supervision of **Md. Minhaz Uddin**, Assistant prof. Department of Mechanical Engineering, Sonargaon University (SU). We also declare that no part of this project and thesis has been or is being submitted elsewhere for the award of any degree.

**Al Fahad Hridoy (L)**
ID No: ME2101023231

**Probhat Kumar Biswas**
ID No: ME2101023440

**Abdul Alim**
ID No: ME2101023424

**Bindu Rahaman Bristy**
ID No: ME2101023474

**Tonmoy Samadder**
ID No:ME2101023488

**Md. Mominul Islam**
ID No: BME2001020600

# ACKNOWLADGEMENT

First of all, we are grateful to Allah, the almighty for giving us the courage and enthusiasm to complete the thesis work. The authors express their gratitude to "**Md. Minhaz Uddin**" for his constant & meticulous supervision, valuable suggestion and encouragement to carry out this work. For all this, the authors acknowledge their sincere gratitude to him. We are also grateful to all our thesis & project working team of SU for their help in construction of the project work and giving their valuable knowledge and time for completing the experiment. Finally, we would like to thank everybody who supported us in any respect for the completion of the thesis.

## Authors,

**Al Fahad Hridoy (L)**

**Probhat Kumar Biswas**

**Abdul Alim**

**Bindu Rahaman Bristy**

**Tonmoy Samadder**

**Md. Mominul Islam**

# ABSTRACT

This paper presents the design of a robotic manipulator which simulates movement to the grip and an object is also known as Robotic Arm. The Manipulator arm is made up of a finite number of individual rigid segments, each rigid segment is called as link. This arm is controlled by Arduino UNO, the robot can be implemented in the hazardous area for material handling and other productive work and can be also used in industries. To examine torque, we consider a prototype module for the robotic manipulator and simulate the task given to it. This paper concludes with some possible applications of 8 Degrees of freedom**.** Smart multipurpose robotic car mechanism is based on end effector to the robotic arm.

# Table of Contents

**APPROVAL** ..... 2
**DECLERATION** ..... 3
**ACKNOWLADGEMENT** ..... 4
**ABSTRACT** ..... 5
**Chapter 1** ..... 7
**Introduction** ..... 7
**1.1 General** ..... 7
**1.2 Objectives:** ..... 8
**Chapter 02** ..... 9
**Literature Review** ..... 9
**2.1 Introduction** ..... 9
**2.2 History** ..... 9
**2.3 Types of The Robotic Manipulation System** ..... 11
**2.4 Differences in Mechanical Robotic Systems** ..... 13
**CHAPTER 3** ..... 14
**METHODOLOGY** ..... 14
**3.1 Process of Project** ..... 14
**3.2 Block Diagram** ..... 14
**3.3 Circuit Diagram** ..... 15
**3.4 Working Principle** ..... 15
**3.5 Required Instrument And software** ..... 16
**CHAPTER 4** ..... 29
**RESULT AND DISCUSSION** ..... 29
**4.1 Result** ..... 29
**4.2 Discussion** ..... 31
**Chapter 5** ..... 33
**Conclusion** ..... 33
**5.1 Advantages:** ..... 33
**5.2 Disadvantages:** ..... 33
**5.3 Applications:** ..... 33
**5.4 Future Scope:** ..... 33
**References:** ..... 35

# Chapter 1

# Introduction

## 1.1 General

Recent technological progress in robotics and automation has made it possible to create intelligent machines to handle more challenging tasks. One of the most versatile technology wonders are manipulators mounted robotic cars which are suggested to be used across various fields like transportation, logistics, surveillance and hazardous situations operations. These technologies comprise self-driving cars alongside robotic arms, generating machines with the flexibility to cover multiple types of terrains whilst interacting with the environment.

Before self-driving cars there were robotic cars, and one of the most famous is Stanford University's entry into the DARPA Grand Challenge: Stanley. Robot manipulators, on the other hand, have been a mainstay in factory automation for decades, performing precise work such as assembly, welding and packaging. But adding a manipulator to a mobile platform dramatically increases the capability of robots and expands their potential applications into areas such as remotely controlled object manipulation, automated delivery, or search-and-rescue.

This thesis examines the construction and control of a robotic car with a manipulator and gives insight not only to their individual designs but also how they are best controlled and coordinated. The goal is to build a working prototype for such an application with the ability to autonomously navigate and manipulate objects, in order to show how this technology could be employed outside of academia. The system is designed to handle challenging tasks across a variety of environments by incorporating advanced sensors, control algorithms and path-planning methods.

In addition, this study will review existing technologies and approaches to robotic car navigation and manipulator control, identifying key challenges and opportunities for further innovation. By addressing these challenges, this research seeks to contribute to the ongoing development of multi-functional robotic systems and pave the way for more intelligent and adaptable machines.

Overall, the integration of robotic cars with manipulators holds immense potential for future applications. As industries continue to seek automation solutions that can operate autonomously and safely in dynamic environments, the significance of such systems will only continue to grow. This thesis endeavors to

explore the intricacies of designing, building, and testing a robotic car with a manipulator, offering insights into its capabilities, limitations, and prospects.

## 1.2 Objectives:

The primary objective of this thesis is to develop a robotic car with a manipulator that can autonomously navigate and manipulate objects. The specific goals are:

1. To detect metals such as Bombs.
2. To detect the temperature.
3. To detect the leakage of gas.
4. To detect the obstacles.

# Chapter 02

# Literature Review

## 2.1 Introduction

Nowadays, robotic arms are used in various industries and operate in hazardous environments. The robotic manipulator is very expensive due to its accuracy and precision [1]. These manipulators can significantly reduce human work, leading to faster progress in industry [2]. The robotic arm can be evaluated based on several criteria, including backlash, payload, speed, repeatability, compliance, human safety, and cost [3]. Forward and inverse kinematics are essential methods used in robotics [4]. A robotic arm is a type of mechanical arm, usually programmable, which may constitute the entire mechanism or be part of a more complex robot. Robotics kinematics deals with the study of motion without considering the forces acting on the robot structure. Forward kinematics determines the position and orientation from a given set of joint angles [5]. Inverse kinematics calculates the required joint angles from a specified position and orientation [6].

Human safety remains a challenge, particularly when the manipulator is used close to humans [7]. The Arduino UNO is often used as the control unit for robotic arms, programmed to control the robot manipulator [8].

Robotics has evolved significantly since the mid-20th century when early robots were primarily controlled by humans or were simple tools capable of performing limited tasks [9].

## 2.2 History

The concept of robotic systems has a long history, tracing back to the early 20th century with the introduction of simple automated machines. Over the decades, the field of robotics has evolved rapidly, giving rise to more complex and autonomous systems. Among these developments, robotic cars and robotic manipulators have become key areas of research and innovation.

The foundation of autonomous vehicles dates back to the 1980s, when the first self-driving car prototypes emerged from research labs. Projects like the ALV (Autonomous Land Vehicle) by the Defense Advanced Research Projects Agency (DARPA) and the Navlab series from Carnegie Mellon University showcased

the potential for cars to navigate using a combination of sensors, cameras, and basic control algorithms [10]. These early attempts laid the groundwork for today's sophisticated autonomous vehicles, which utilize advanced technologies such as LIDAR, GPS, and AI-based path planning [11].

In parallel, the development of robotic manipulators began primarily in industrial settings. The first industrial robot, Unlimited, was introduced in 1961 and was capable of performing repetitive tasks, such as welding and assembly, in manufacturing plants [12]. Over time, robotic arms became more versatile, gaining additional degrees of freedom, better precision, and enhanced control systems. This evolution allowed manipulators to handle increasingly complex tasks in various industries, including manufacturing, healthcare, and space exploration [13].

The idea of combining mobile platforms with manipulators arose as researchers sought to extend the capabilities of stationary robots. By integrating a robotic arm onto a mobile base, researchers could create systems capable of both navigating through environments and performing precise manipulation tasks [14]. This concept found its roots in the late 20th century when autonomous mobile robots equipped with manipulators were explored for applications in hazardous environments, such as nuclear facilities and space missions. NASA's Mars rovers, for example, feature manipulators that enable them to collect samples and interact with their surroundings, demonstrating the potential of such integrated systems [15].

Today, robotic cars with manipulators are increasingly seen in logistics, healthcare, and service industries. These robots not only navigate autonomously but also interact with objects, opening up possibilities for automated material handling, remote operations, and delivery services [16]. As technology continues to advance, integrating robotic manipulators with mobile platforms remains a crucial area of research, driving innovations in artificial intelligence, control systems, and sensor technologies [17].

This historical progression underscores the significance of this thesis, which aims to contribute to the ongoing development of robotic systems by exploring the integration of a mobile car platform with a versatile manipulator.

## 2.3 Types of The Robotic Manipulation System

Robotic manipulation systems are designed to interact with objects in various ways, mimicking human hand movements. The types of robotic manipulation systems can be categorized based on their functionality, design, and application.

### 2.3.1 Serial Manipulators (Robotic Arms)

- **Articulated Robots**: These have rotary joints and multiple degrees of freedom (DOF), resembling a human arm (shoulder, elbow, wrist). They are highly flexible and used in tasks like welding, painting, and assembly. Examples include industrial robotic arms such as the ABB and KUKA robots [18].
- **SCARA Robots (Selective Compliance Assembly Robot Arm)**: Designed for high-speed, precision tasks, SCARA robots have two parallel rotary joints that allow for movement in the X-Y plane. They are commonly used for pick-and-place operations and assembly tasks [19].
- **Polar/Spherical Robots**: These robots have a spherical work envelope due to their arm's rotating and extending motions, suitable for tasks requiring large reach with variable orientation [20].
- **Cylindrical Robots**: With a cylindrical-shaped work envelope, these robots use linear and rotational movement for tasks like assembly, welding, and packaging [21].

### 2.3.2 Parallel Manipulators

- **Delta Robots**: They consist of three arms connected to universal joints at the base. Known for their high speed and precision, delta robots are used in tasks such as packaging, sorting, and high-speed pick-and-place operations [22].
- **Stewart Platforms**: A six-degree-of-freedom (6-DOF) parallel manipulator used for precise positioning. Often applied in flight simulators and other situations requiring controlled movement in multiple axes [23].

### 2.3.3 Mobile Manipulators

- **Wheeled Robots with Manipulator Arms**: These combine a mobile base (wheeled or tracked) with a robotic arm, allowing for both mobility and manipulation. They are used in logistics, search and rescue, and space exploration (e.g., Mars rovers with robotic arms) [24].

- **Humanoid Robots**: Equipped with anthropomorphic arms and hands, these robots can mimic human motions and perform a variety of tasks. Examples include robots like Honda's ASIMO and Boston Dynamics' Atlas [25].

### 2.3.4 Collaborative Robots (Cobots)

Designed to work alongside humans safely, these robots have built-in sensors and advanced programming to detect and adapt to human presence. Cobots are used in assembly lines, laboratories, and warehouses, offering flexibility and easy reprogramming for various tasks [26].

### 2.3.5 Soft Robotic Manipulators

Made of soft, flexible materials, these manipulators are designed to handle delicate or irregularly shaped objects (like fruits, tissues, or fragile components) by conforming to the shape of the objects. They use technologies like pneumatics, hydraulics, or shape memory alloys to create motion [27].

### 2.3.6 Tele operated Manipulators

These systems allow human operators to control the manipulator remotely, often used in hazardous or inaccessible environments such as underwater exploration, space missions, and radioactive environments [28].

## 2.3.7 Redundant Manipulators

Manipulators with more degrees of freedom than required for a given task. The extra DOF provides flexibility to avoid obstacles, optimize movements, or perform complex tasks in confined spaces [29].

Each type of robotic manipulation system is tailored to specific applications, depending on factors like precision, speed, flexibility, and the nature of the objects being manipulated.

## 2.4 Differences in Mechanical Robotic Systems

1. **Structure and Configuration**:
    - **Serial Robots**: Single chain of joints (e.g., articulated arms) for flexibility.
    - **Parallel Robots**: Multiple arms working in parallel for high speed and precision (e.g., Delta robots).
    - **Cartesian Robots**: Linear joints along X, Y, and Z axes, ideal for tasks like 3D printing [30].
    - **Cylindrical and Spherical Robots**: Movements along cylindrical or spherical coordinates for reaching into confined spaces.
2. **Degrees of Freedom (DOF)**:
    - **Fixed DOF**: Limited movement to specific axes (e.g., Cartesian robots).
    - **Redundant DOF**: Extra flexibility for complex motion and obstacle avoidance (e.g., articulated robots).
3. **Type of Joints**:
    - **Rotary Joints**: Provide rotational movement (e.g., articulated robots).
    - **Linear Joints**: Enable straight-line movement (e.g., Cartesian and SCARA robots).
    - **Spherical Joints**: Allow multi-directional rotation.
4. **Movement and Workspace**:
    - **Articulated Robots**: Large, flexible workspace for complex tasks.
    - **SCARA Robots**: High-speed, horizontal movements for pick-and-place tasks.
    - **Delta Robots**: High speed with limited workspace for precision tasks.
    - **Cartesian Robots**: Rectangular workspace with linear accuracy [31].
5. **Material and Build**:
    - **Heavy-Duty Robots**: Made with steel/aluminum for high-load tasks.
    - **Lightweight Robots**: Use composites/plastics for speed and safety.
6. **Actuation Mechanism**:
    - **Electric**: Precise control for accuracy-based tasks [32].
    - **Hydraulic**: High force for heavy-duty applications.
    - **Pneumatic**: Quick, repetitive movements for tasks like packaging [33].

# CHAPTER 3

## METHODOLOGY

### 3.1 Process of Project

- Creating an idea for Design and construction of Robotic Car With Manipulation.
- And designing a block diagram & circuit diagram to know which components need to construct it.
- The project is structured into several key phases to build a robotic car with a manipulator.
- Collecting the all components and programming for the microcontroller to controlled the system.
- Setting all components in a PCB board & soldering. Then assembling the all block in a board and finally run the system & checking.

### 3.2 Block Diagram

The block diagram outlines the system architecture, showing how different modules of the robotic car are interconnected:

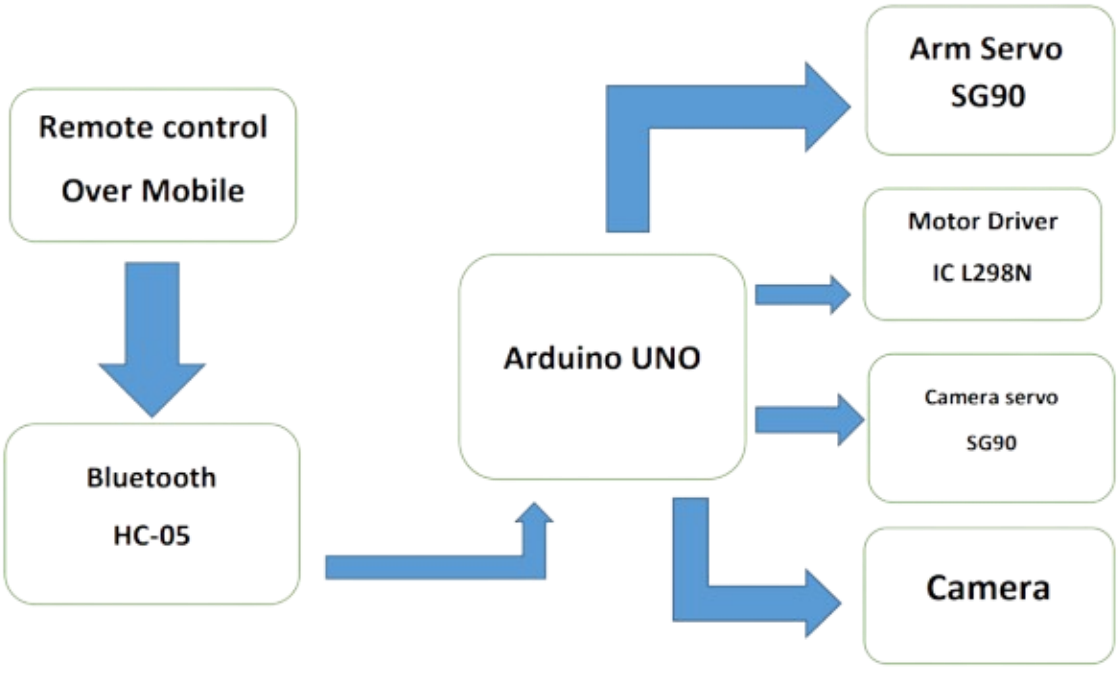



Figure 3.2.2.1 Block Diagram.

## 3.3 Circuit Diagram

The circuit diagram represents the electrical connections between the components:

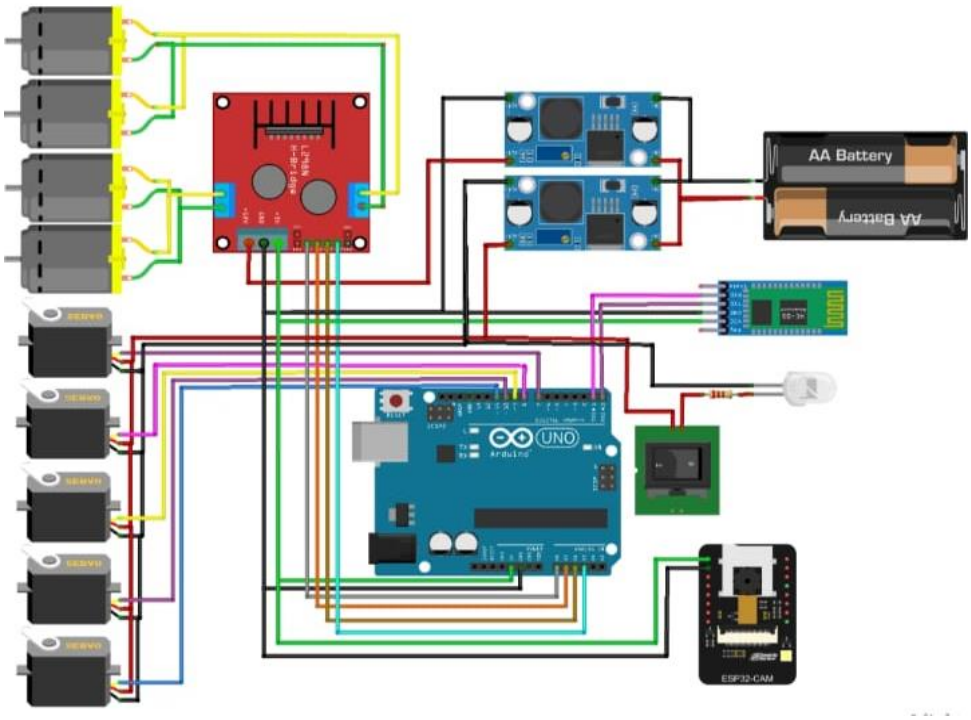



Figure 3.3.3.1 Circuit Diagram.

.

## 3.4 Working Principle

The robotic car operates on principles of control and feedback:

1. **Movement**: The microcontroller (Arduino Nano) processes input from sensors (e.g., camera, ultrasonic) to navigate the environment. It sends PWM signals to the motor drivers, controlling the speed and direction of the DC gear motors for smooth car movement.

2. **Manipulator Operation**: The manipulator arm is controlled by servo motors, which receive signals from the microcontroller to adjust their positions. Using forward kinematics, the microcontroller calculates the necessary angles for the arm's joints to reach a target location.
3. **Remote Control**: The Bluetooth module allows for remote operation, sending commands to the microcontroller to direct the car's movement and manipulate the arm. The camera can provide live feedback to the operator, enhancing manual control.
4. **Autonomous Functionality**: For autonomous operation, the microcontroller uses sensor data to avoid obstacles and interact with objects. Feedback from sensors allows real-time adjustments to both the car's movement and the arm's actions.

## 3.5 Required Instrument And software

- Arduino UNO
- Servo Motor
- Motor Driver
- DC Gear Motor
- Bluetooth
- Camera
- Buck Module
- Battery
- Arduino Ide
- Cirkit designer

### 3.5.1 Arduino UNO

Acts as the brain of the robotic car, processing inputs from sensors and controlling the motors. It was chosen for its small size, versatility, and sufficient input/output pins. Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling Lights, motors, and other actuators.

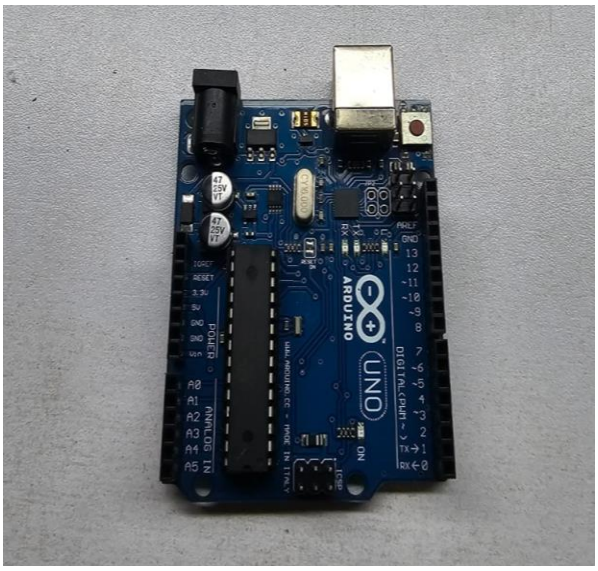



Figure 3.5.1.1:Arduino UNO

## 3.5.2 Servo Motor

The **SG90 Servo Motor** is one of the most popular and widely used micro servo motors, especially in hobbyist and small-scale applications like robotics, RC (Radio Control) vehicles, and DIY electronics projects. Its compact size, light weight, and affordability make it an ideal choice for those needing precise rotational control.

**Key Features**

- **Compact Size**: The SG90 is small and lightweight, with dimensions of approximately 22.2 mm × 11.8 mm × 31 mm. It weighs around 9 grams, making it suitable for projects with tight space and weight constraints.
- **Control Angle**: It typically offers a **180-degree rotation range**, allowing for precise angular positioning. The movement is controlled by sending a Pulse Width Modulation (PWM) signal, where the pulse duration determines the servo's rotation angle.
- **Operating Voltage**: The SG90 operates on a voltage range of **4.8V to 6V**, which makes it compatible with most microcontrollers (like Arduino and Raspberry Pi) and standard batteries.

- **Torque**: Despite its small size, the SG90 can deliver a torque of about **1.8 kg/cm** at 4.8V. This is sufficient for small robotic arms, sensor panning mechanisms, and other lightweight applications.

**PWM Control**:

The servo's position is controlled by the width of the PWM signal:

- **1 mms pulse**: Moves the shaft to 0 degrees.
- **1.5 mms pulse**: Centers the shaft at 90 degrees.
- **2 mms pulse**: Moves the shaft to 180 degrees. The SG90 can achieve precise angular movements within this range.

## Gearing:

It contains an internal plastic gear assembly that provides smooth and precise movement. This gear reduction mechanism enhances torque output while reducing the rotational speed.

**Pin Configuration**

The SG90 has three wires:

- **Orange (Signal)**: Connects to the PWM signal output of a microcontroller to control the servo's position.
- **Red (VCC)**: Connects to the positive power supply (typically 5V).
- **Brown (Ground)**: Connects to the ground of the power supply and the microcontroller.



Figure 3.5.2.1 Servo Motor SG90

### 3.5.3 Motor Driver

The **L298N Motor Driver** is a versatile and widely used integrated circuit designed to control the direction and speed of two DC motors or a single stepper motor. This module is commonly utilized in robotic projects, automotive applications, and industrial equipment due to its robust design and ease of use.

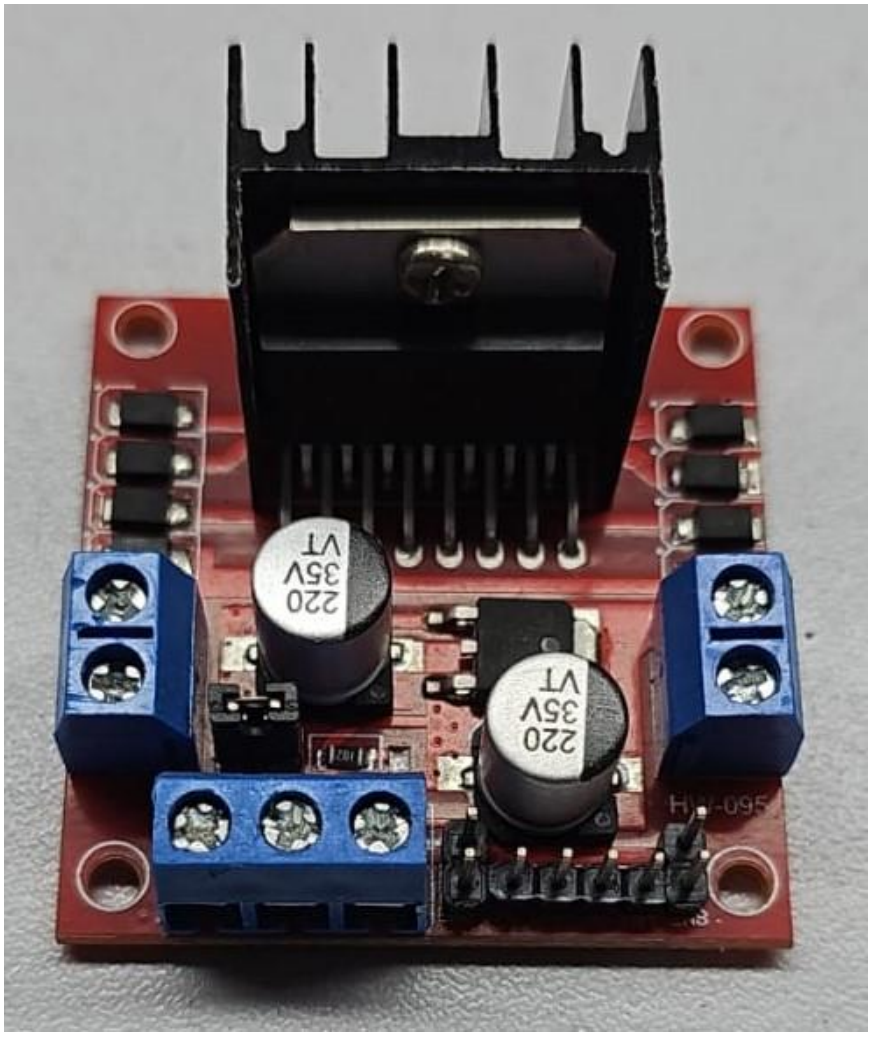



Figure 3.5.3.1 Motor Driver IC L298N.

## Key Features

1. **Dual H-Bridge Driver**: The L298N contains two H-bridges, enabling it to independently control the direction and speed of two DC motors. An H-bridge allows voltage to be applied across a motor in either direction, making it possible to drive the motor forward or backward.

2. **High Voltage and Current Handling**: It can handle up to **46V** and **2A** continuous current per channel (with peak currents of up to 3A). This makes it suitable for driving small to medium-sized motors.

3. **Built-in Protection Diodes**: The L298N includes built-in diodes for back EMF (Electromotive Force) protection. This safeguards the driver from voltage spikes that can occur when motors are rapidly stopped or started.

4. **PWM Control**: It supports **Pulse Width Modulation (PWM)** for speed control. PWM allows for fine-tuning the motor speed by varying the duty cycle of the input signal.

5. **Heat Sink**: The module usually comes with a heat sink to dissipate heat generated during operation, helping maintain performance and longevity.

To drive a motor, connect the motor terminals to the output pins of the L298N. Use the input pins to set the desired motor direction:

- **IN1 HIGH & IN2 LOW**: Motor A rotates in one direction.
- **IN1 LOW & IN2 HIGH**: Motor A rotates in the opposite direction.
- **IN1 & IN2 both LOW**: Motor A stops. Similarly, the same control applies to the second motor using the IN3 and IN4 pins.

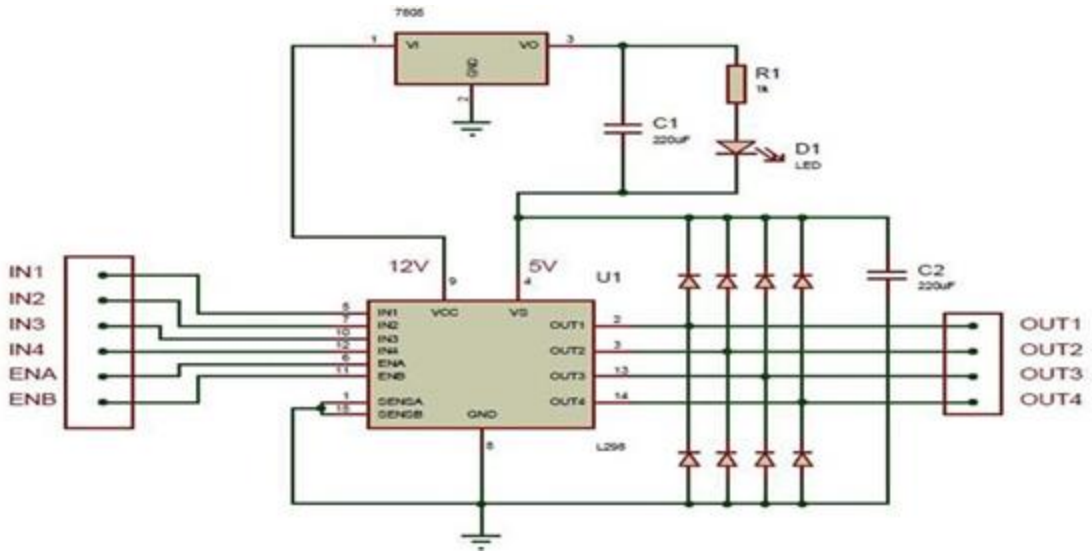



Figure: Schematic diagram of L298N Motor Driver IC

**Pin Configuration**

- **Input Pins (IN1, IN2, IN3, IN4)**: These pins control the direction of the two motors. By adjusting the HIGH and LOW signals on these pins, you can change the direction and state of each motor.

- **Enable Pins (ENA, ENB)**: These pins enable or disable the motors. They also support PWM input, allowing for speed control.

- **Power Pins**: It has separate pins for motor power (VSS) and logic power (VCC). VSS supplies power to the motors, while VCC provides power to the logic circuitry.

- **Output Pins (OUT1, OUT2, OUT3, OUT4)**: These are connected to the motor terminals.

### 3.5.3.1 L298N Logic Table

The **L298N motor driver** uses a pair of input signals for each motor to control its direction and speed. Here's the logic table for controlling a single motor using the L298N, assuming we use **IN1** and **IN2**

for one motor (Motor A):

| **IN1** | **IN2** | **Motor A State** |
|---|---|---|
| LOW | LOW | Stop (Coasting) |
| HIGH | LOW | Rotate Forward |
| LOW | HIGH | Rotate Backward |
| HIGH | HIGH | Stop (Braking) |

For the second motor (Motor B), connected to **IN3** and **IN4**, the logic table is similar:

| **IN3** | **IN4** | **Motor B State** |
|---|---|---|
| LOW | LOW | Stop (Coasting) |
| HIGH | LOW | Rotate Forward |
| LOW | HIGH | Rotate Backward |
| HIGH | HIGH | Stop (Braking) |

## 3.5.4 DC Gear Motor

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gearbox to a motor reduces the speed while increasing the torque output. The most important parameters in regard to gear motors are speed (rpm), torque (lb.-in) and efficiency (%). In order to select the most suitable assembly for your application, you must first compute the load, speed and torque requirements for your application. ISL Products offers a variety of small or miniature gear motors including Spur Gear Motors, Planetary Gear Motors and Worm Gear Motors to meet all application requirements. Most of our DC motors can be complemented with one of our unique gearboxes, providing you with a highly efficient inline solution. Pairing the proper motor and gearbox reduction ratio is critical when designing a gear motor.



Figure 3.5.4.1: DC Gear Motor

DC motors were the first form of motor widely used, as 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 can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

Rated voltage: 12 V DC

Reduction Ratios: 1:270

Direction of rotation: CW

**At no load:**

Speed: 10.8 ±10%rpm

Current: 0.035A(max 0.43A)

At max efficiency:

Torque: 2.4kg.cm

Speed: 9.3 ±10% rpm

Current 0.068 A(max 2.4A)

### 3.5.5 Bluetooth

The HC-05 is a class 2 Bluetooth module designed for transparent wireless serial communication. It is pre-configured as a slave Bluetooth device. Once it is paired to a master Bluetooth device such as PC, smart phones and tablet, its operation becomes transparent to the user. All data received through the serial input is immediately transmitted over the air. When the module receives wireless data, it is sent out through the serial interface exactly at it is received. No user code specific to the Bluetooth module is needed at all in the user microcontroller program.

## Specification:

- Bluetooth Protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz
- ISM band Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps,
- Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3-5 VDC 50mA
- Working temperature: -20 ~ +75Centigrade

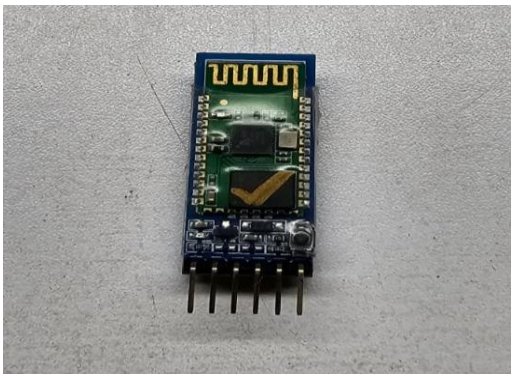


Figure 3.5.5.1: Bluetooth Module HC-05

### 3.5.6 Camera

The ESP32-CAM is a low-cost development board with an integrated camera module, ideal for robotics projects like a robotic car with a manipulator. It is based on the ESP32 microcontroller, which offers Wi-Fi and Bluetooth connectivity, enabling remote monitoring and control.

## Specifications

| | |
|---|---|
| **Module Model** | ESP32-CAM |
| **Package** | DIP-16 |
| **Size** | 27*40.5*4.5（±0.2）mm |
| **SPI Flash** | default 32Mbit |
| **RAM** | Internal 520KB + External 8MB PSRAM |
| **Bluetooth** | Bluetooth 4.2 BR/EDR and BLE standards |
| **Wi-Fi** | 802.11 b/g/n/e/i |
| **Support Interfaces** | UART、SPI、I2C、PWM |
| **Support TF Card** | up to 4G |
| **IO Pins** | 9 |
| **Serial Port Speed** | default 115200 bps |
| **Image Output Format** | JPEG(only OV2640 support),BMP,GRAYSCALE |
| **Spectrum range** | 2400 ~2483.5MHz |

| | |
|---|---|
| **Antenna Type** | On-board PCB antenna, gain 2dBi |
| **Transmit Power** | 802.11b: 17±2 dBm (@11Mbps) |
| | 802.11g: 14±2 dBm (@54Mbps) |
| | 802.11n: 13±2 dBm (@MCS7) |
| **Receive Sensitivity** | CCK, 1 Mbps: -90dBm, |
| | CCK, 11 Mbps: -85 dBm |
| | 6 Mbps (1/2 BPSK): -88 dBm |
| | 54 Mbps (3/4 64-QAM): -70dBm |
| | MCS7 (65 Mbps, 72.2 Mbps): -67dBm |
| **Power Consumption** | Flash off: 180mA@5V, |
| | Flash on and brightness to maximum: 310mA@5V |
| | Deep-sleep: the lowest power consumption can reach 6mA@5V |
| | Moderm-sleep: minimum 20mA@5V |
| | Light-sleep: minimum 6.7mA@5V |
| **Security** | WPA/WPA2/WPA2-Enterprise/WPS |
| **Power supply range** | 4.75-5.25V |
| **Operating Temperature** | -20 ℃ ~ 70 ℃ |
| **Storage Environment** | -40 ℃ ~ 125 ℃ , < 90%RH |



Figure 3.5.6.1 ESP32 Camera

## 3.5.7 Buck Converter (Buck Module)

A **buck converter**, also known as a **step-down converter**, is a DC-DC power converter that steps down the input voltage to a lower output voltage while increasing the current. It is one of the most commonly used types of switching regulators in electronic devices and power management systems due to its high efficiency, compact size, and versatility.

**1. Basic Operation**

The buck converter operates using a switching element (usually a MOSFET), an inductor, a diode (or synchronous MOSFET), and a capacitor. The key principle behind its operation is the rapid switching of the input voltage using a high-speed switch (the MOSFET). By adjusting the duty cycle of this switching, the average voltage supplied to the load can be regulated. Here's a simplified step-by-step operation:

I. **Switch ON**: When the switch (MOSFET) is turned on, current flows from the input voltage source through the inductor to the load and the output capacitor. The inductor stores energy in its magnetic field during this phase, and the output voltage increases.

II. **Switch OFF**: When the switch is turned off, the inductor opposes the sudden decrease in current, maintaining current flow to the load through the diode (or synchronous MOSFET). The energy stored in the inductor is now transferred to the load and the output capacitor.

By controlling the **duty cycle** (the ratio of the switch ON time to the total switching period), the output voltage can be regulated. The duty cycle DDD for a basic buck converter can be approximated by the formula:

$$D = \frac{Vout}{Vin}$$

where $V_{out}$ is the output voltage and $V_{in}$ is the input voltage.

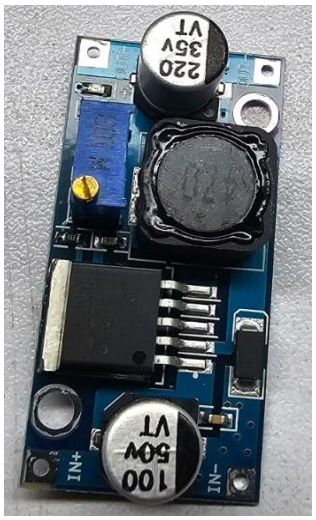



Figure 3.5.7.1 Buck Converter (Buck Module) LM2596.

### 3.5.8 Battery

A lithium-ion (Li-ion) battery is an advanced battery technology that uses lithium ions as a key component of its electrochemistry.

## Specifications:

- Chemistry: Li-ion.
- Nominal Voltage (V): 3.7V.
- Capacity (mAh): 6800.
- Max Continuous Discharge Current: 5.2A.
- Weight: 46.5±1 g.
- Height: 65.2mm.

- Diameter: 18.4mm.
- Charge Cut-Off Voltage: 4.20 ± 0.05V.

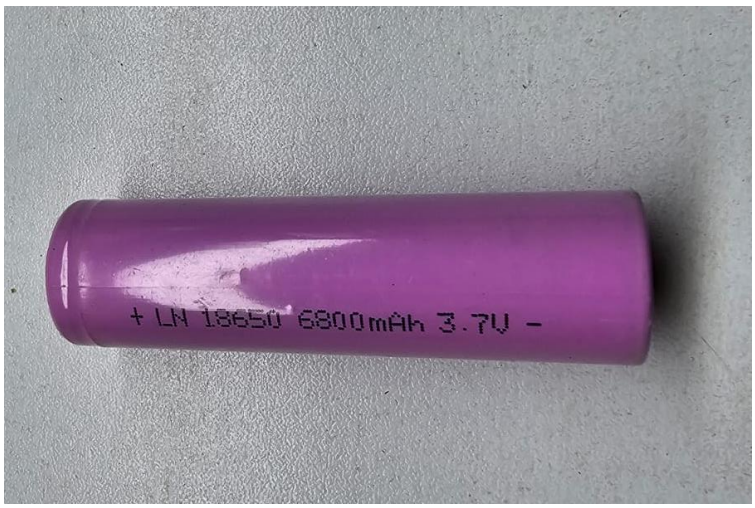



Figure 3.5.8.1 Battery 6800mAh (3.7V)

## 3.5.9 Robotic Arm

**1. Structure and Design:**

- The arm consists of multiple segments (or links) that are interconnected using joints. This configuration allows the arm to mimic the movements of a human arm or an industrial manipulator.
- Base: The arm is mounted on a flat, sturdy base. The base appears to have multiple pre-drilled holes, potentially for mounting other components or to secure the arm on different surfaces. A stable base is crucial for maintaining balance when the arm extends or carries small loads.
- Material: The arm is made of lightweight materials, most likely acrylic or plastic, which are commonly used in small robotic kits due to their ease of fabrication and low cost.

**2. Movement and Control:**

- Servo Motors: The arm has several servo motors attached at different joints. Servos are essential for precise control of movement and positioning in robotic applications. They enable the arm to rotate or lift in different directions, allowing for multiple degrees of freedom (DOF).

- The motors appear to be standard small-scale servos, typically used in hobby robotics. They are wired for power and control, likely connected to a microcontroller (such as an Arduino, Raspberry Pi, or other control board) for manipulation.
- Joints: The arm's joints are designed to rotate, enabling movements similar to an excavator. This setup can simulate basic robotic functions like gripping, lifting, and positioning objects.

**3. End Effector (Manipulator):**

- At the end of the arm, there is a gripper or claw-like structure. This is known as the "end effector" or "manipulator." It allows the arm to interact with objects, picking them up, moving, or manipulating them.
- The end effector seems to have a basic design suitable for simple tasks like picking up lightweight objects, which is typical in robotic kits for educational purposes.

**4. Potential Uses:**

- Educational Tool: The arm is likely part of a robotics kit designed for learning purposes. It provides hands-on experience with robotics concepts like kinematics, control systems, programming, and electronics.
- Prototyping: This kind of setup could also be used for prototyping simple automation tasks, giving insights into how industrial robots operate on a smaller scale.
- Research: For students or researchers working on projects like the "Robotic Car with Manipulator," this setup could serve as a testbed for developing algorithms, controls, and movement strategies.

**5. Control System:**

- The wires connected to the servos suggest that the arm is electronically controlled, probably by a microcontroller. The servos receive signals to move at specific angles, allowing precise control over the arm's motion.
- Users can program the robotic arm to perform a series of movements, which can be done via a microcontroller board using various programming languages C++

**6. Power Supply:**

The robotic arm requires a power source to operate the servos. Based on the image, it might be powered by an external battery pack, potentially using batteries similar to the 18650 lithium-ion battery previously discussed.

**7. Assembly and Customization:**

This robotic arm appears to be modular, likely assembled from a kit. Users can customize or expand its capabilities by adding more joints, different end effectors, or sensors (like ultrasonic or IR sensors) to interact with its environment.

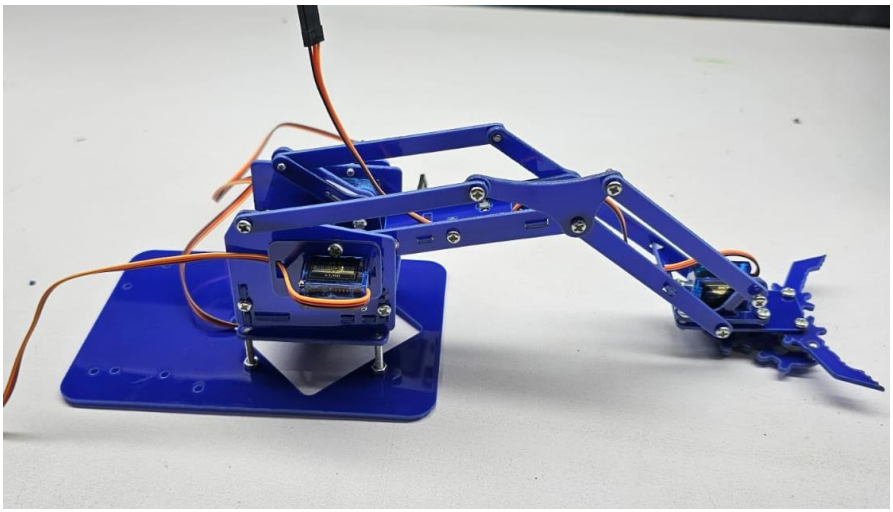


Figure 3.5.9.1 Robotic Arm

# CHAPTER 4
# RESULT AND DISCUSSION

## 4.1 Result

Finally, we were able to create our project successfully. After making the Mechanical body, we designed a circuit to control it and when we operated it with mobile apps, we called it working pretty well. It is very well controlled, and the robotic car is capable of moving across different terrains, while the manipulator arm, mounted on the car, performs tasks such as picking, carrying, and placing objects very well. Below is a picture of our successfully completed entire project.

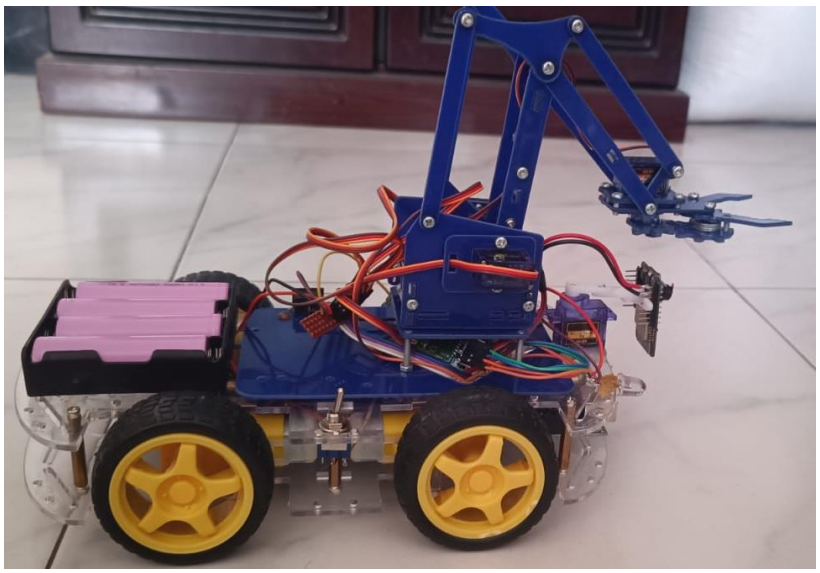


Figure- Final project

## Data Table

| Data Table | | |
|---|---|---|
| **Number** | **Time (Second)** | **Pick & Place Accuracy** |
| 1 | 10.2 | 1 |
| 2 | 12.13 | 1 |
| 3 | 13.47 | 1 |
| 4 | 14.11 | 0 |
| 5 | 10.45 | 1 |
| 6 | 16.39 | 0 |
| 7 | 15 | 1 |
| 8 | 19.39 | 0 |
| 9 | 10.3 | 1 |
| 10 | 17.26 | 1 |
| 11 | 16.23 | 1 |
| 12 | 34.15 | 1 |
| 13 | 23.08 | 0 |
| 14 | 15.44 | 1 |
| 15 | 16.55 | 0 |
| 16 | 17.85 | 0 |
| 17 | 14.53 | 1 |
| 18 | 14.93 | 1 |
| 19 | 11.64 | 1 |
| 20 | 13.32 | 1 |
| **Average Time= 15.82 sec.** | | **Total pick =20, Missing= 06** |

## 4.2 Discussion

The discussion of the thesis on "Smart Multipurpose Robotic Car" revolves around the integration of mobility and object manipulation in a single robotic platform and the challenges encountered during its development. The project explores the synergy between the robotic car's navigation and the manipulator's ability to interact with objects, providing valuable insights into the design, control, and real-world applications of such systems.

**1. Design Considerations:**

- One of the primary discussion points is the selection of components for the robotic car and manipulator. The choice of motors, sensors, and microcontroller significantly impacts the robot's efficiency, accuracy, and response time. The integration of a manipulator on a mobile platform presents challenges in maintaining stability, especially when the arm is in motion or carrying objects.
- Another aspect is the structural design, which required balancing between the arm's strength and the car's mobility. Discussions also cover how different configurations of the robotic arm's degrees of freedom affect its reach, flexibility, and load-carrying capacity.

**2. Control System Challenges:**

- The discussion highlights the complexity of developing a control system that coordinates the movement of both the car and the manipulator. The project required creating algorithms capable of handling simultaneous control tasks, such as steering the car while operating the arm.
- Synchronizing the manipulator's actions with the car's movements is also discussed, particularly in scenarios where precise positioning is required for tasks like picking and placing objects. Challenges include ensuring smooth transitions between navigation and manipulation actions without compromising stability.

**4. Limitations and Future Improvements:**

- The limitations identified include the manipulator’s payload capacity and the car's mobility constraints on uneven surfaces. These aspects highlight areas for future research, such as using more powerful motors, improving the car's suspension, or enhancing the manipulator’s structure for handling heavier loads.
- Discussions extend to potential improvements in the robot's autonomous capabilities, such as incorporating advanced algorithms for more complex navigation and manipulation tasks, or using computer vision for better object recognition.

**5. Applications and Impact:**

- The discussion concludes by exploring the practical applications of the robotic car with a manipulator, including its use in material handling, remote operations in hazardous environments, and educational purposes. This opens a pathway for future developments in service robotics, where mobile manipulators can perform various tasks efficiently and safely.

# Chapter 5

# Conclusion

## 5.1 Advantages:

- Cost Effectiveness
- Improved Quality Assurance
- Increased Productivity
- Work In Hazardous Environments

## 5.2 Disadvantages:

- Potential Job Losses
- Initial Investment Costs
- Hiring Skilled Staff

## 5.3 Applications:

- Manufacture & Assembly
- Warehouse Automations
- Search & Rescue Operation
- Agriculture
- Infrastructures & Inspection
- Explorations

## 5.4 Future Scope:

- In the future it can be improved by using more sensors.
- In the future, controlling the Internet can be done to improve it.

**Conclusion:**

This report presents the design and the development of a robotic arm, which can perform simple tasks, such as light material handling. The robotic arm is designed and built from aircraft-grade aluminum material where servo motors were used to perform arm movements. The design of the

robotic arm is limited to the four degrees of freedom the design of a robotic arm has been complete. The prototype was Built and confirmed functional. This system would make it easier for men to unrivaled the risk of handling suspicious objects which could be hazardous in its present environment and workplace. Complex duties would be achieved faster and more accurately with this design.

# References:


1. Siciliano, B., & Villani, L. (2009). *Robotics: Modelling, Planning and Control*. Springer.
2. Bogue, R. (2018). "The role of robots in industrial automation." *Industrial Robot: An International Journal*, 45(1), 1-5.
3. Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson.
4. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. Wiley.
5. Tirosh, Y., & Karp, B. (2019). "Robotic Manipulators: Dynamics and Control." *IEEE Robotics and Automation Magazine*, 26(1), 40-49.
6. Siciliano, B., & Villani, L. (2009). *Robotics: Modelling, Planning and Control*. Springer.
7. Tirosh, Y., & Karp, B. (2019). "Robotic Manipulators: Dynamics and Control." *IEEE Robotics and Automation Magazine*, 26(1), 40-49.
8. Arduino. (2023). *Arduino UNO Documentation*. Retrieved from Arduino Official Website.
9. Bogue, R. (2018). "The role of robots in industrial automation." *Industrial Robot: An International Journal*, 45(1), 1-5
10. Thrun, S., & Montemerlo, M. (2006). "The Tartan Racing Team's Entry in the Urban Challenge." Technical Report, Carnegie Mellon University.
11. Guizzo, E. (2016). "How Google’s Self-Driving Car Works." IEEE Spectrum.
12. Groover, M. P. (2007). Automation, Production Systems, and Computer-Integrated Manufacturing. Pearson.
13. Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control. Pearson.
14. Murphy, R. R. (2000). "Human-Robot Interaction in Rescue Robotics." IEEE Transactions on Systems, Man, and Cybernetics.

15. Grotz, J., & Barho, R. (2010). "NASA's Mars Rovers: A Challenge for Autonomous Robotics." IEEE Robotics and Automation Magazine.
16. Wang, Y., & D'Amato, J. (2019). "Applications of Mobile Robots in Logistics." Robotics and Computer-Integrated Manufacturing.
17. Zhang, L., & Li, H. (2020). "Recent Advances in Robotic Manipulation." Journal of Robotic Systems.
18. Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control. Pearson.
19. O'Reilly, J. (2012). "The Evolution of SCARA Robots." Robotics and Automation Magazine.
20. Geng, Y., & Yang, G. (2018). "Polar Robots and Their Applications." International Journal of Advanced Robotics.
21. Siciliano, B., & Villani, L. (2009). Robotics: Modelling, Planning and Control. Springer.
22. Dupont, P. E., & Dufour, B. (2016). "Delta Robot Design and Control." IEEE Transactions on Robotics.
23. Stewart, D. (1965). "A Platform with 6 Degrees of Freedom." Proceedings of the Institution of Mechanical Engineers.
24. Murphy, R. R. (2000). "Human-Robot Interaction in Rescue Robotics." IEEE Transactions on Systems, Man, and Cybernetics.
25. Kuffner, J. J., & Latombe, J. C. (2000). "Robot Motion Planning." In Autonomous Robots: From Theory to Practice.
26. Colledani, M., & Pinceti, S. (2019). "Collaborative Robots in Industry." Robotics and Computer-Integrated Manufacturing.
27. Rus, D., & Tolley, M. T. (2015). "Design, Fabrication, and Control of Soft Robots." Nature.
28. Hurst, A. (2017). "Teleoperated Manipulators in Hazardous Environments." Journal of Field Robotics.

29. Liu, Y., & Zhang, Y. (2016). "Redundant Manipulators: Design and Control." Robotics and Autonomous Systems.
30. Siciliano, B., & Khatib, O. (2016). Springer Handbook of Robotics. Springer.
31. Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control. Pearson.
32. Nascimento, J. M., & Queiroz, L. (2020). "Electric Actuators in Robotics." IEEE Robotics and Automation Magazine.
33. Sweeney, J. (2018). "Pneumatic Systems in Automation." International Journal of Robotics Research

# APPENDIX

```
#include <Servo.h>

Servo motor_0;
Servo motor_1;
Servo motor_2;
Servo motor_3;
Servo motor_4;

#define in1 A3 //Motor1  L298 Pin in1
#define in2 A2 //Motor1  L298 Pin in1
#define in3 A1 //Motor2  L298 Pin in1
#define in4 A0 //Motor2  L298 Pin in1

int servo0 = 0;
int servo1 = 0;
int servo2 = 0;
int servo3 = 90;
int servo4 = 90;

int bt_data;
int Speed = 130;

void setup(){

Serial.begin(9600); // initialize serial communication at 9600 bits per second:

motor_0.attach(7);
motor_1.attach(8);
motor_2.attach(9);
motor_3.attach(10);
```

```
motor_4.attach(11);

motor_0.write(servo0);
motor_1.write(servo1);
motor_2.write(servo2);
motor_3.write(servo3);
motor_4.write(servo4);

pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4

delay(1000);
}

void loop(){
//if some date is sent, reads it and saves in state

if(Serial.available() > 0){
bt_data = Serial.read();
Serial.println(bt_data);
if(bt_data > 20){Speed = bt_data;}
}


if(bt_data == 1){forword(); }  // if the bt_data is '1' the DC robot will go forward
else if(bt_data == 2){backword();}  // if the bt_data is '2' the robot will Reverse
else if(bt_data == 3){turnRight();}  // if the bt_data is '3' the robot will turn left
else if(bt_data == 4){turnLeft();} // if the bt_data is '4' the robot will turn right
else if(bt_data == 5){Stop(); }     // if the bt_data '5' the robot will Stop
```

```
else if(bt_data == 6){turnLeft();  delay(400);  bt_data = 5;}   // if the bt_data is '6' the robot will turn left
else if(bt_data == 7){turnRight(); delay(400);  bt_data = 5;}   // if the bt_data is '7' the robot will turn right

else if (bt_data == 8){ // if the bt_data is '8' the robot arm will be open
if(servo0>0){servo0 = servo0-1;}
motor_0.write(servo0);
}
else if (bt_data == 9){ // if the bt_data is '8' the robot arm will be Close
if(servo0<90){servo0 = servo0+1;}
motor_0.write(servo0);
}

else if (bt_data == 10){
if(servo1>0){servo1 = servo1-1;}
motor_1.write(servo1);
}
else if (bt_data == 11){
if(servo1<90){servo1 = servo1+1;}
motor_1.write(servo1);
}

else if (bt_data == 12){
if(servo2>0){servo2 = servo2-1;}
motor_2.write(servo2);
}
else if (bt_data == 13){
if(servo2<90){servo2 = servo2+1;}
motor_2.write(servo2);
}
```

```
else if(bt_data == 14){
if(servo3>0){servo3 = servo3-1;}
motor_3.write(servo3);
}
else if (bt_data == 15){
if(servo3<180){servo3 = servo3+1;}
motor_3.write(servo3);
}

else if (bt_data == 16){
if(servo4>0){servo4 = servo4-1;}
motor_4.write(servo4);
}
else if (bt_data == 17){
if(servo4<180){servo4 = servo4+1;}
motor_4.write(servo4);
}

delay(50);
}


void forword(){  //forword
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void backword(){ //backword
digitalWrite(in1, LOW);  //Right Motor forword Pin
```

```
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void turnRight(){ //turnRight
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void turnLeft(){ //turnLeft
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void Stop(){ //stop
digitalWrite(in1, LOW); //Right Motor forword Pin
digitalWrite(in2, LOW); //Right Motor backword Pin
digitalWrite(in3, LOW); //Left Motor backword Pin
digitalWrite(in4, LOW); //Left Motor forword Pin
}
```