**Project Report**

**on**

**"Robusta 2.0"**

**A Bluetooth Control Robotic-Arm-Vehicle**

**Course No: CSE-3104**

**Course Title – Peripherals and Interfacing Laboratory**

**PREPARED BY:**

**1. George Tonmoy Roy**

**Roll:1907114**

**&**

**2. Masudur Rahman Rabby**

**Roll:1907113**

**PREPARED FOR:**

**Md. Repon Islam**

**Lecturer**

**CSE, KUET**

**&**

**Md. Badiuzzaman Shuvo**

**Lecturer**

**CSE, KUET**

**Table of Content**

1. Objective

2. Introduction

3. Apparatus Required

4. Implementation

5. Photo

6. Pseudocode

7. Video

8. Applications

9. Limitations

10. Future Plan

11. Conclusion

**OBJECTIVE**

1. Develop a Bluetooth-controlled robotic arm vehicle controlled via a mobile app, providing precise movement control for both the vehicle and the robotic arm.
2. Implement a reliable wireless communication system between the app and the vehicle, enabling seamless control and synchronization of various vehicle and robotic arm actions.
3. Design an intuitive and user-friendly app interface for easy navigation and manipulation of the vehicle and robotic arm functionalities.
4. Optimize the Bluetooth connection to ensure stable and responsive control, minimizing latency and ensuring a smooth user experience.
5. Create a modular and expandable project structure, allowing for future enhancements and additions to incorporate additional features or improve existing functionalities.

**INTRODUCTION**

The Bluetooth Control Robotic-Arm-Vehicle project combines Bluetooth technology and robotics to provide users with a remote-control platform for operating a robotic vehicle and manipulating objects using a robotic arm. With an intuitive mobile app interface, users can navigate the vehicle and control the arm's movements wirelessly, promoting engagement, creativity, and problem-solving skills. This project showcases the potential of Bluetooth and robotics in real-world applications, offering a cost-effective and scalable solution that can be customized for various purposes.

**Project Components**

1. Arduino UNO

2. Motor Driver L298N

3. Bluetooth Module HC05

4. Buck Converter

5. Motor

6. Wheel

7. Servo motor-MG995

8. Servo motor-SG90

9. Mini Breadboard

10. Cardboard Chassis

12. Li-po battery single cell (3.7V)

12. Battery Case

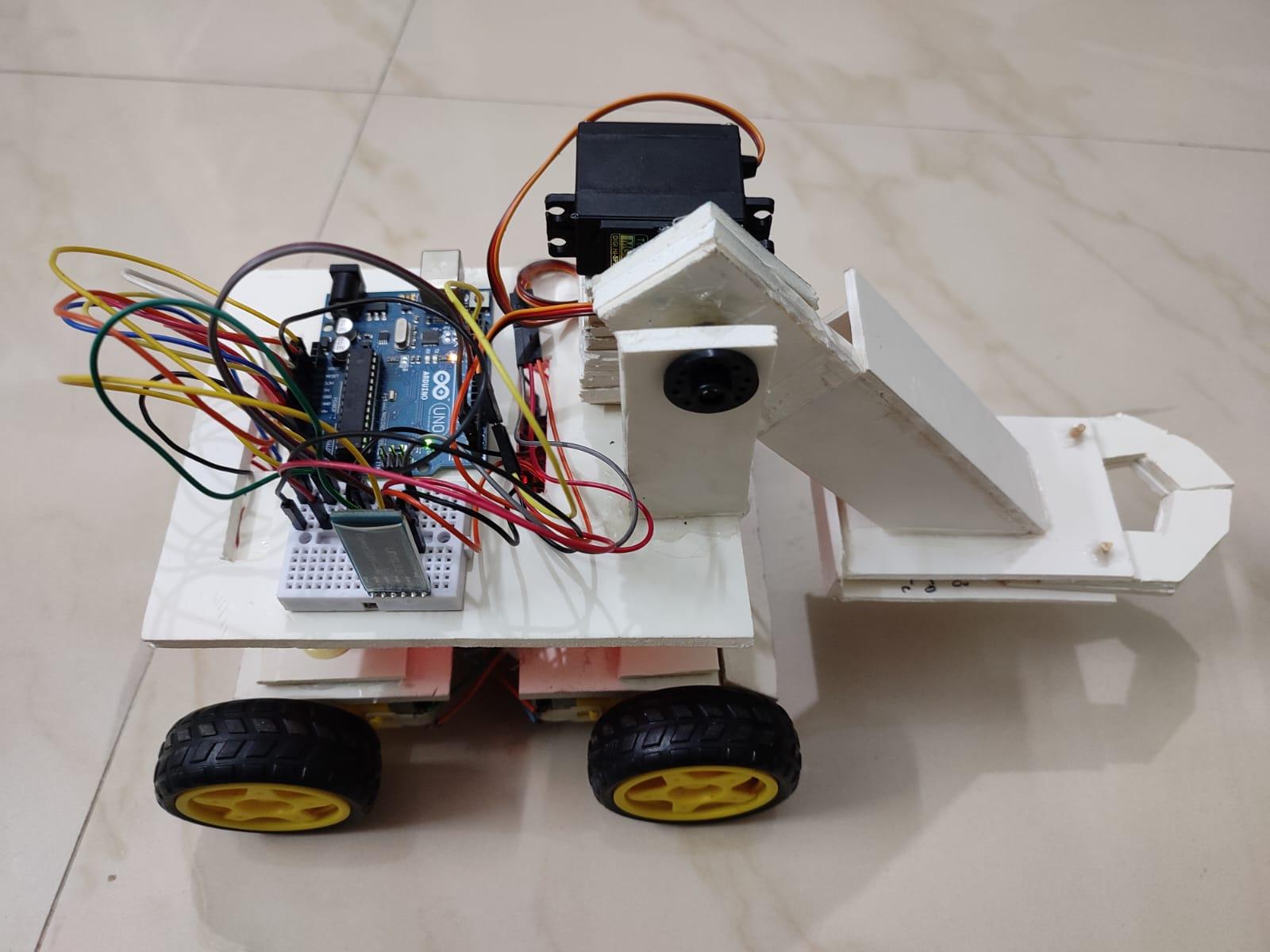
13. Connecting wire (M2F & M2M)

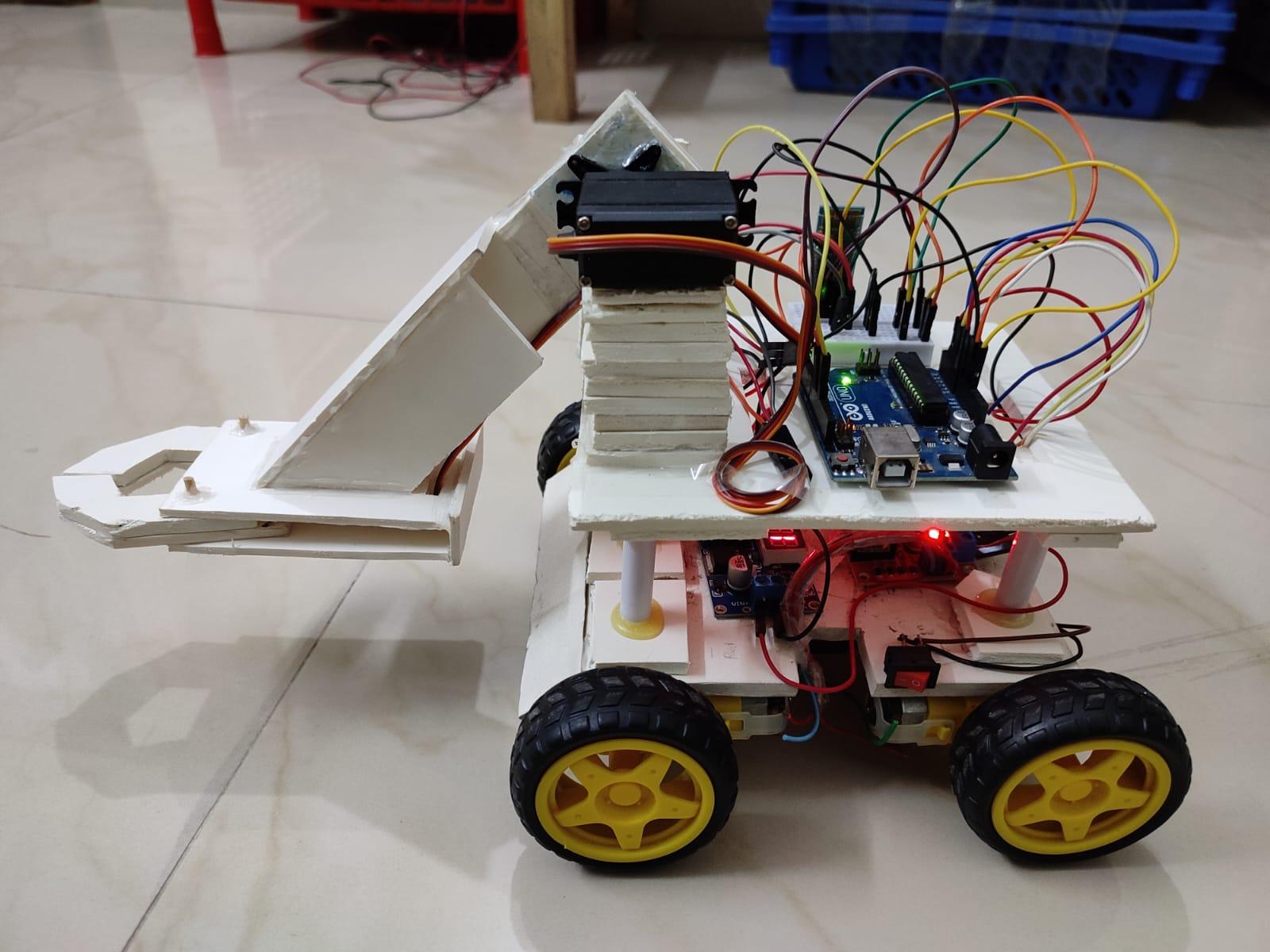
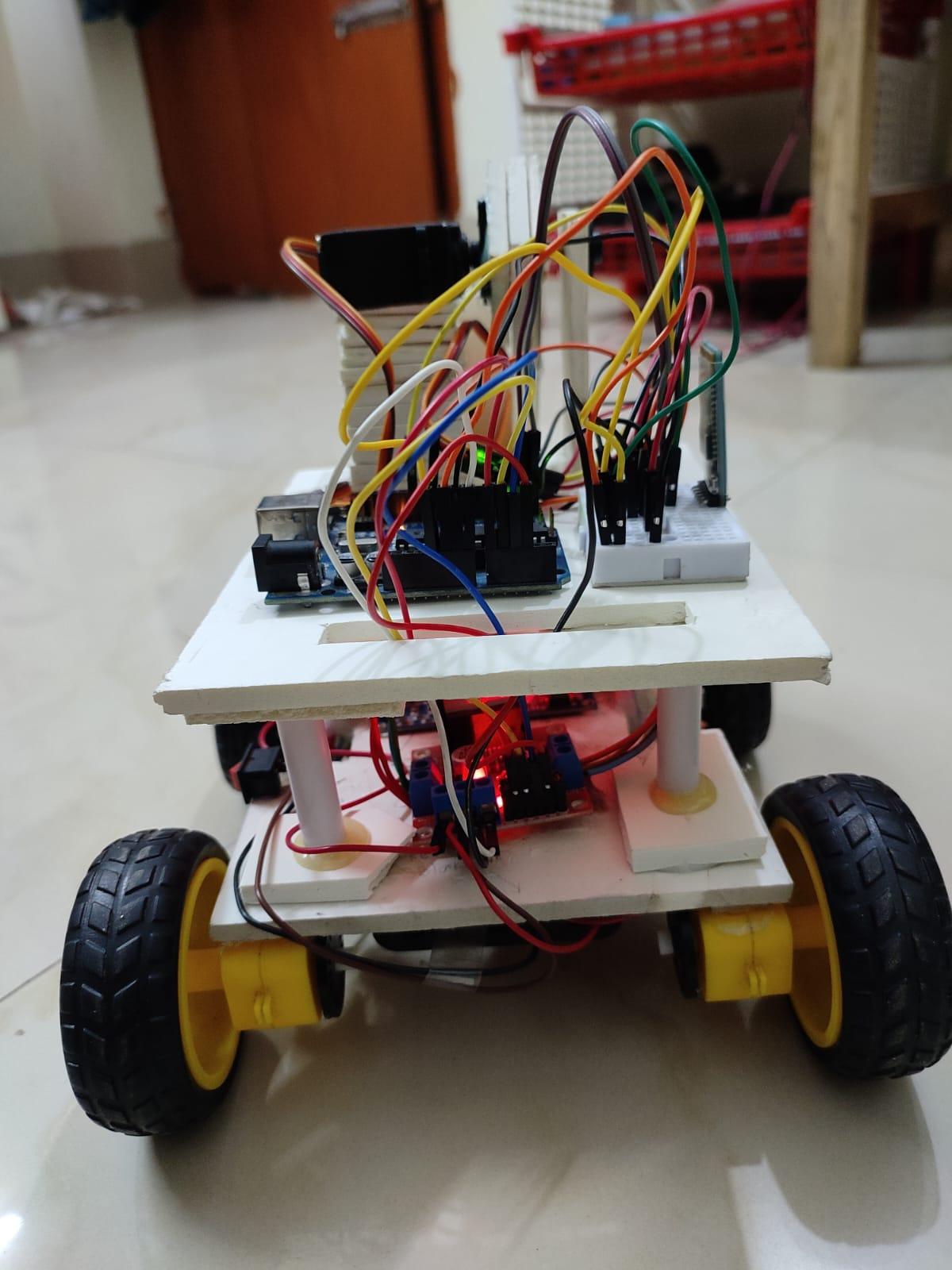
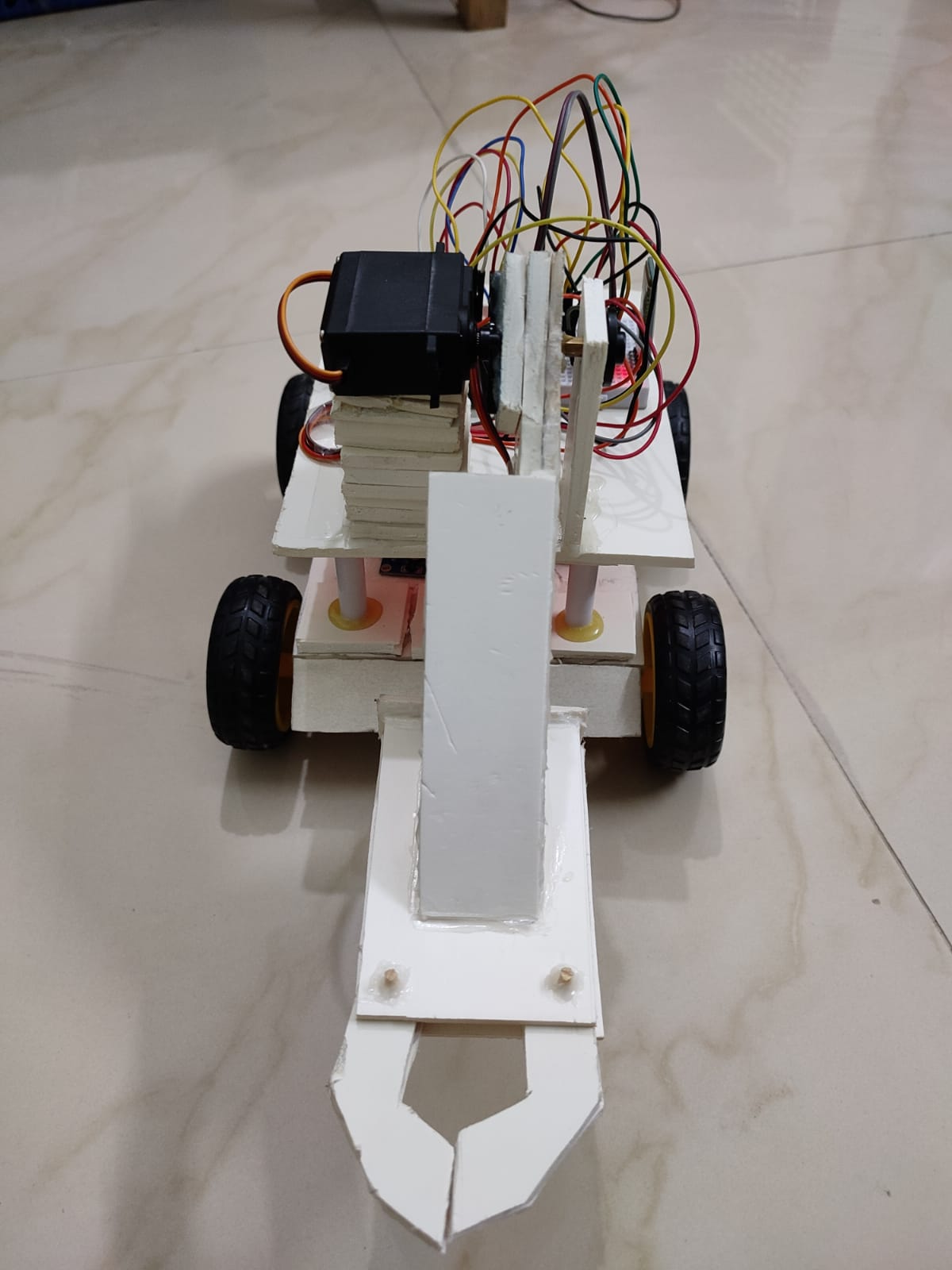
**IMPLEMENTATION**

The project can be implemented by following these steps:

* We connect the servo motor MG995 in pin 10 and servo motor SG90 in pin 12 on the Arduino UNO
* Connect the L298N with Arduino in A0 – A3 pin.
* For the motor, we connect the positive and negative wires to the IN1 -IN4 pin to the L298N motor driver.
* Attach the Bluetooth module’s TX pin to Arduino’s RX pin and RX pin to TX pin respectively and VCC and Ground in the VCC and GND of Arduino.
* For the servomotor and L298N motor driver, external power has to use. We use 3 batteries of 3.7 volts and connect with the DC to DC Buck Converter for a stable connection between the servo motor and motor.
* We program the Arduino Uno using the Arduino IDE, utilizing libraries such as the Servo library for controlling the servo motor.
* We write the necessary code logic and the code is uploaded to the Arduino Uno to ensure that all components are functioning correctly.
* We assemble the hardware components on the breadboard and connect them to the Arduino Uno.

**PHOTO**





**PSEUDOCODE**

Declare Servo motor\_3

Declare Servo motor\_5

Declare variables: servo3, servo5, bt\_data, Speed

Setup():

Initialize serial communication

Attach motor\_3 to pin 10

Attach motor\_5 to pin 12

Set initial positions for motor\_3 and motor\_5

Set pin modes for motor control

Delay for 1 second

Loop():

If serial data is available:

Read and store the data in bt\_data

Print bt\_data

If bt\_data is greater than 20:

Update Speed with bt\_data

// Motor control based on bt\_data

If bt\_data equals 1:

Call forword() function

Else if bt\_data equals 2:

Call backword() function

Else if bt\_data equals 3:

Call turnLeft() function

Else if bt\_data equals 4:

Call turnRight() function

Else if bt\_data equals 5:

Call Stop() function

Else if bt\_data equals 6:

Call turnLeft() function

Delay for 400 milliseconds

Set bt\_data to 5

Else if bt\_data equals 7:

Call turnRight() function

Delay for 400 milliseconds

Set bt\_data to 5

Else if bt\_data equals 12:

If servo3 is less than 180:

Increment servo3 by 1

Update motor\_3 position with servo3

Else if bt\_data equals 13:

If servo3 is greater than 0:

Decrement servo3 by 1

Update motor\_3 position with servo3

Else if bt\_data equals 16:

If servo5 is greater than 150:

Decrement servo5 by 1

Update motor\_5 position with servo5

Else if bt\_data equals 17:

If servo5 is less than 180:

Increment servo5 by 1

Update motor\_5 position with servo5

Delay for 30 milliseconds

forword():

Set appropriate pin values for forward motion

backword():

Set appropriate pin values for backward motion

turnRight():

Set appropriate pin values for right-turn motion

turnLeft():

Set appropriate pin values for left turn motion

Stop():

Set pin values to stop motor motion

**VIDEO**

****

**APPLICATIONS**

* The project can be used as an educational tool to teach robotics, programming, and automation concepts.
* It finds applications in industrial settings for tasks like material handling, assembly, and inspection.
* The vehicle can be utilized for remote surveillance and monitoring in areas that are inaccessible.
* It offers an interactive and fun experience for enthusiasts and hobbyists interested in robotics and remote-controlled vehicles.
* The project can be used as a platform for conducting experiments and research in robotics, control systems, and human-robot interaction.

**LIMITATIONS**

* The control range of the Bluetooth technology used may be limited, restricting the distance over which the vehicle can be operated.
* The project relies on the availability of a Bluetooth connection, which may be subject to interference or connectivity issues.
* The project focuses on the control of the vehicle and arm through the app but lacks features such as obstacle detection and avoidance.
* Depending on the power source used, the runtime of the vehicle may be limited, requiring frequent recharging or battery replacement.
* The project may involve a certain level of complexity in terms of assembly, programming, and troubleshooting, requiring technical expertise or assistance for beginners.

**FUTURE PLAN**

* Adding obstacle detection, autonomous navigation, and intelligent decision-making capabilities to enhance the vehicle's functionality.
* Exploring alternative communication methods like Wi-Fi or RF for broader control range and flexibility.
* Leveraging artificial intelligence algorithms to enable the vehicle to learn, adapt, and make intelligent decisions.

**CONCLUSION**

The Bluetooth Control Robotic-Arm-Vehicle project highlights the use of mobile app-based control for a robotic vehicle with an arm. Although there are limitations in obstacle detection and avoidance, the project offers potential for future enhancements such as advanced features, alternative control methods, and AI integration. With further research and development, it can contribute to advancements in robotic control systems and automation technologies.