**ROBOTIC ARM USING JOYSTICK**

**A PROJECT BASED LEARNING REPORT**

# *Submitted by*

**HARIESH A (22EC023)**

**HARIHARAN T (22EC025)**

## **KUMAR** **T** (22EC046)

**MITHRESHWARAN S.M (22EC056)**

## 

*In partial fulfilment for the award of the degree*

*of*

# BACHELOR OF ENGINEERING

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**NANDHA ENGINEERING COLLEGE, ERODE**

**(AUTONOMOUS)**

**(Affiliated to Anna University, Chennai)**

**June 2024**

**BONAFIDE CERTIFICATE**

Certified that this project report “**ROBOTIC ARM USING JOYSTICK”** is the bonafide work of “**HARIESH.A (22EC023), HARIHARAN.T (22EC025), KUMAR.T (22EC046), MITHRESHWARAN S.M (22EC056)**”, who carried out the Project Based Learning work under my supervision.

|  |  |
| --- | --- |
|  | **SIGNATURE**  Mr. M.Srinevasan, M.E.,  Assistant Professor,  PBL COORDINATOR  Department of ECE,  Nandha Engineering College,  (Autonomous)  Erode-638052. |

## ACKNOWLEDGEMENT

The success of a work depends on the team work and co-operation various involved either directly or indirectly.

We express our thanks to our beloved Chairman of Sri Nandha Educational,Trust **Thiru.V.SHANMUGAN** and our beloved Secretary, **Thiru.S.THIRUMOORTHI** of Nandha Engineering College for providing us all the basic amenities to complete the course successfully.

We wish to express our deep sense of gratitude and thanks to our beloved Principal **Dr.N.RENGARAJAN** for the excellent facilties providing during the course study and project work.

We wish to express our hearty thanks to **Dr.S.KAVITHA,** Professor & Dean

**, Dr.C.N.MARIMUTHU**, Professor & Dean(R&D), Department of Electronics and Communication Engineering for providing support during the project work.

We are highly indebted to our project guide **Mr.M.SRINEVASAN**, Assistant Prefesser, Project Lab Incharge, Department of Electronics and Communication Engineering, for their guidance and motivation in completing the project.

We thank all the Teaching and non-Teaching staff members of Electronics and Communication Engineering Department for their support to our project work. we also express our thanks to our family members and friends for their encouragement timely help and suggestion.

**TITLE OF CONTENT**

**CHAPTER NO TITLE PAGE NO**

**ABSTRACT** 5

1. **INTRODUCTION** 6
2. **COMPONENTS REQUIRED** 7
   1. HARDWARE REQUIRED 7
      1. Arduino UNO 7
      2. Jumping wire 9
      3. Servo motor 10
   2. SOFTWARE REQUEIRED 11
      1. Arduino IDE 11
      2. Program Coding 11
3. **BLOCK DIAGRAM** 13
   1. Circuit diagram 13
   2. Working 14
4. **CONCLUSION** 15

**FUTURE SCOPE** 15

**REFERENCE** 16

**APPENDIX** 16

## ABSTRACT

This abstract presents a novel approach to controlling a robotic arm using a joystick interface. The integration of a joystick provides an intuitive and user-friendly method for manipulating the robotic arm's movements in real-time. The system comprises a robotic arm equipped with actuators and sensors, along with a joystick interface connected to a control unit. The control unit processes input signals from the joystick and translates them into corresponding commands for the robotic arm. Through the joystick's intuitive control scheme, users can effortlessly manipulate the robotic arm's movements, including rotation, translation, and gripping actions. This allows for precise and efficient control, facilitating a wide range of applications in fields such as manufacturing, healthcare and research.

**CHAPTER 1**

## INTRODUCTION

In In the realm of advanced robotics, the ability to manipulate objects with precision and dexterity is a hallmark of technological achievement. One such innovation that showcases this capability is the robotic arm controlled by a joystick. This sophisticated system combines the finesse of human control with the power and reliability of robotic mechanisms, offering a versatile solution across various industries. A joystick, traditionally associated with gaming consoles and aircraft control, serves as the interface between human intent and robotic action in this setup. By translating subtle movements of the joystick into complex manoeuvres of the robotic arm, operators can achieve tasks that range from delicate assembly operations in manufacturing to precise movements in medical procedures. The integration of sensors, actuators, and advanced algorithms ensures that the robotic arm responds swiftly and accurately to the joystick inputs, mimicking human arm movements with unparalleled precision. This synergy of human-guided control and robotic efficiency not only enhances productivity but also opens new possibilities in fields where precision and reliability are paramount. This introduction sets the stage for exploring how joystick-controlled robotic arms are revolutionizing industries, advancing automation, and reshaping the way we interact with technology.

Top of Form

Bottom of Form

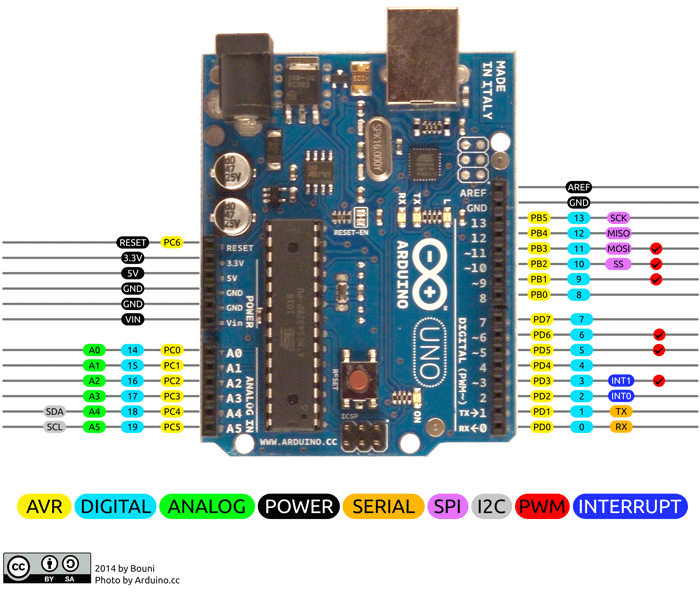
**CHAPTER 2**

**COMPONENTS REQUIRED**

## 2.1. HARDWARE REQUIRED

### 2.1.1 ARDUINO UNO MICROCONTROLLER

The Arduino Uno is a widely recognized and versatile microcontroller board that has become synonymous with DIY electronics and prototyping. Developed by Arduino.cc, it is based on the ATmega328P microcontroller and offers a user-friendly platform for both beginners and experienced makers to create interactive projects.

Key features of the Arduino Uno include its compact size, ease of use, and open-source nature. It features 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog input pins, and a USB connection for programming and power supply. The board can be powered either via the USB connection or an external power supply, making it adaptable for various applications.

**fig 2.1.1(i) Arduino UNO Pin Configuration**

Overall, the Arduino Uno serves as a gateway into the world of electronics and programming, empowering users to bring their ideas to life and explore the realms of automation, IoT (Internet of Things), robotics, and beyond.



**fig 2.1.1(ii) Arduino UNO**

## 1.3 JUMPING WIRES

A jump wire (also known as jumper, jumper wire, DuPont wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end

(or sometimes without them – simply “tinned”), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their “end connectors” into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.



**fig 2.1.2 Jumping wires**

## 1.4 SERVO MOTOR

One of the standout features of the ESP32 is its built-in Wi-Fi and Bluetooth connectivity. This makes it suitable for a wide range of applications where wireless communication is essential, such as IoT devices, home automation, and industrial applications.



**fig 2.1.3 Servo motor**

## 2.2 SOFTWARE REQUEIRED

### 2.2.1 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a software application that provides a user-friendly platform for programming and uploading code to Arduino microcontrollers. It is a crucial tool for hobbyists, students, and professionals working with Arduino boards to create a wide range of electronic projects. The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

### 2.2.2 PROGRAM CODING

#include <Servo.h>

Servo servo\_z\_axis;

Servo servo\_x\_axis;

Servo servo\_y\_axis;

Servo servo\_clamp;

int x\_axis\_degree = 90;

int y\_axis\_degree = 90;

int z\_axis\_degree = 85;

int clamp\_degree = 90;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//The defining of analog input pins tested with Arduino UNO and Arduino Nano.

//The other cards not tested

#define left\_joystick\_x A0

#define left\_joystick\_y A1

#define right\_joystick\_x A2

#define right\_joystick\_y A3

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void setup() {

Serial.begin(9600);

servo\_z\_axis.attach(3);

servo\_clamp.attach(5);

servo\_x\_axis.attach(6);

servo\_y\_axis.attach(9);

}

void loop() {

int left\_joystick\_x\_value = analogRead(left\_joystick\_x);

int left\_joystick\_y\_value = analogRead(left\_joystick\_y);

int right\_joystick\_x\_value = analogRead(right\_joystick\_x);

int right\_joystick\_y\_value = analogRead(right\_joystick\_y);

if(left\_joystick\_x\_value < 340) y\_axis\_degree -=7;

else if(left\_joystick\_x\_value > 680) y\_axis\_degree +=7;

if(left\_joystick\_y\_value < 340) clamp\_degree -=5;

else if(left\_joystick\_y\_value > 680) clamp\_degree +=5;

if(right\_joystick\_x\_value < 340) x\_axis\_degree -=7;

else if(right\_joystick\_x\_value > 680) x\_axis\_degree +=7;

if(right\_joystick\_y\_value < 340) z\_axis\_degree -=7;

else if(right\_joystick\_y\_value > 680) z\_axis\_degree +=7;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//You should decide the max/min angles.

z\_axis\_degree = min(145, max(15, z\_axis\_degree));

x\_axis\_degree = min(175, max(40, x\_axis\_degree));

y\_axis\_degree = min(150, max(5, y\_axis\_degree));

clamp\_degree = min(90, max(75, clamp\_degree));

Serial.print("x\_axis\_degree : ");

Serial.print(x\_axis\_degree);

Serial.print(", y\_axis\_degree : ");

Serial.print(y\_axis\_degree);

Serial.print(", z\_axis\_degree 4 : ");

Serial.print(z\_axis\_degree);

Serial.print(", clamp\_degree : ");

Serial.println(clamp\_degree);

servo\_clamp.write(clamp\_degree);

servo\_x\_axis.write(x\_axis\_degree);

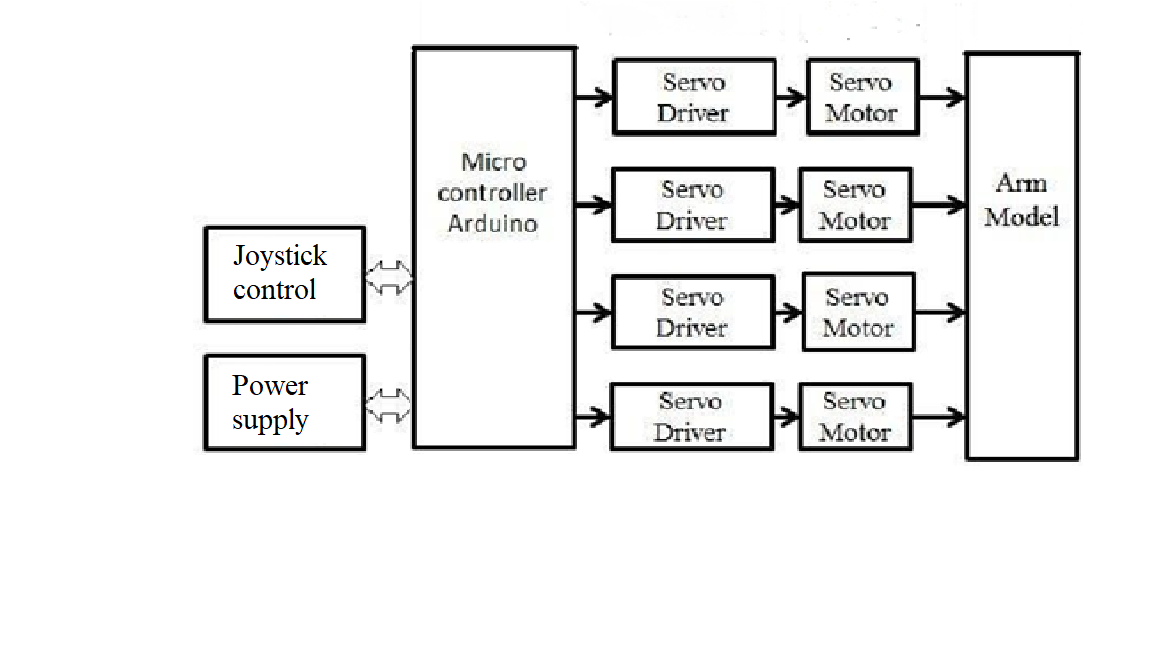
servo\_y\_axis.write(y\_axis\_degree);

servo\_z\_axis.write(z\_axis\_degree);

}

**CHAPTER 3**

## 3.1 BLOCK DIAGRAM OF PROPOSED SYSTEM

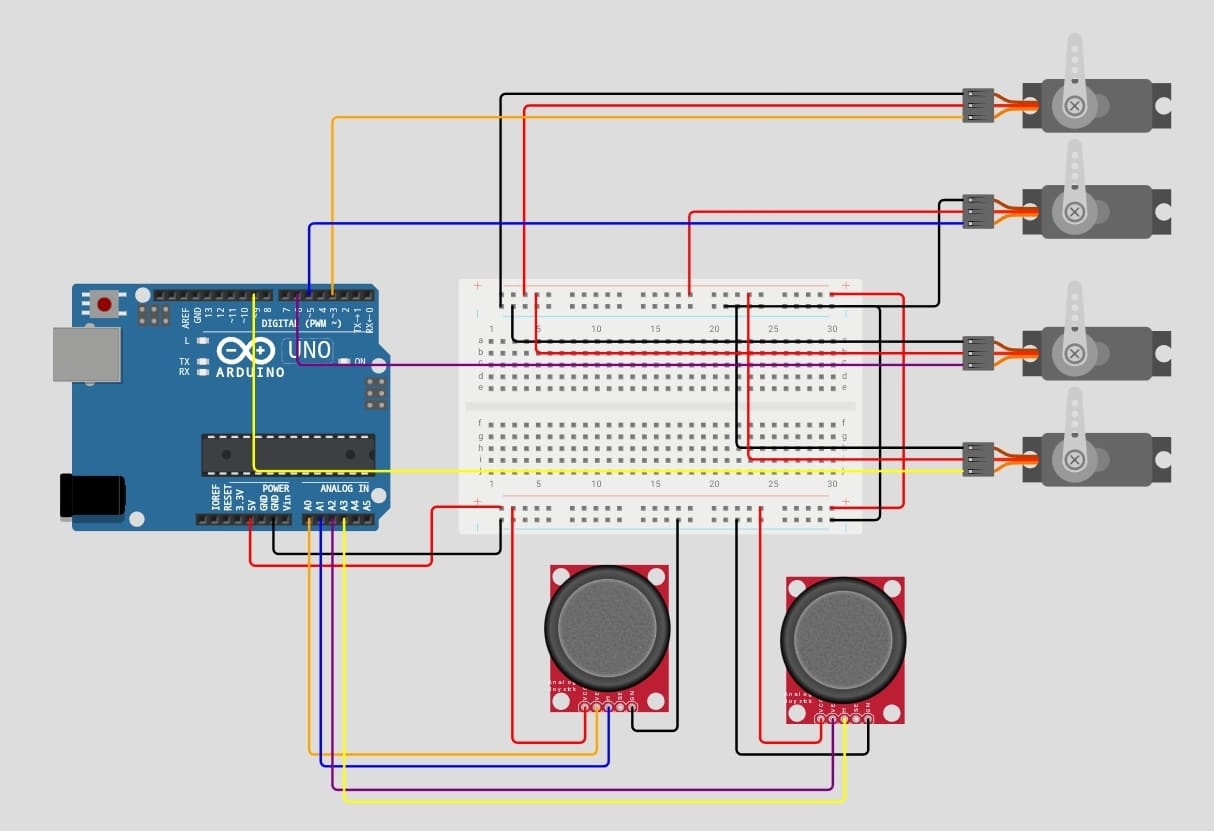
****

**fig 3.1 Block diagram of proposed system**

To control a robotic arm using a joystick, first connect the joystick to a microcontroller like Arduino. Write code to read the joystick's X and Y axis values, typically analog inputs, and map these to the range of motion of your robotic arm (e.g., degrees of rotation or positional values). Ensure the microcontroller can output control signals compatible with your robotic arm’s actuators (motors or servos).

Test and calibrate the system for smooth and accurate movement. Consider safety measures and integrate feedback mechanisms for enhanced control and reliability. This setup allows intuitive manual control of the robotic arm, suitable for applications ranging from hobbyist projects to industrial automation tasks.

3.2 CIRCUIT DIAGRAM OF PROPOSED SYSTEM



**fig 3.2 Circuit diagram of proposed system**

To advance your project of controlling a robotic arm with a joystick, consider integrating advanced features such as speed control for precise movements and fine-tuning adjustments to enhance operational accuracy. Incorporating sensors like proximity or force sensors can enable the arm to react intelligently to its environment, detecting obstacles or adjusting grip strength. Implementing wireless communication allows for remote operation, expanding the arm's usability across distances. Developing a graphical user interface (GUI) simplifies monitoring and control, improving user interaction. Additionally, automating tasks based on sensor inputs or predefined conditions enhances efficiency. Documenting your setup and maintenance procedures ensures scalability and future troubleshooting. By integrating these elements, you can create a sophisticated robotic arm system capable of versatile and autonomous operation, suitable for diverse applications from industrial automation to research and development environments.

Bottom of Form

**CHAPTER 4**

## CONCLUSION

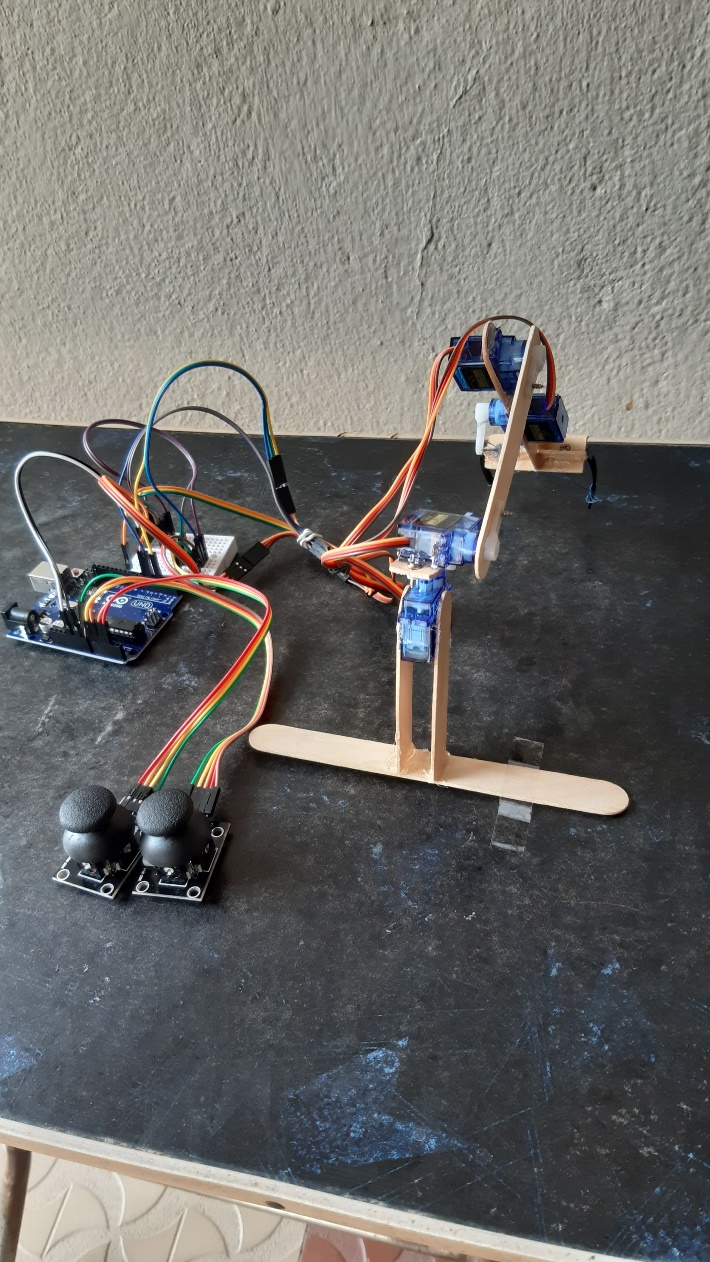
In conclusion, the integration of a robotic arm with the ardiuno UNO microcontroller represents a powerful and versatile solution for achieving precise and controlled motion in various applications. The combination of the robotic arm's mechanical components and the ardiuno UNO's computational capabilities opens up a wide range of possibilities for automation, remote control, and customization.

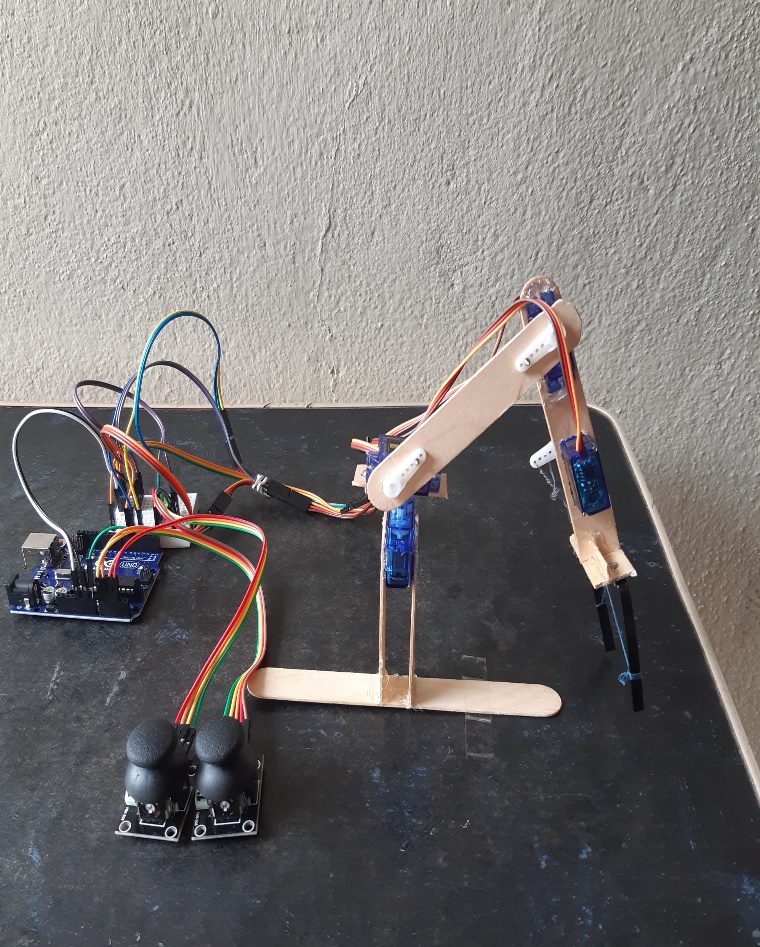
## FUTURE SCOPE

* The future scope for robotic arms using Ardiuno UNO envisions a paradigm shift in automation and robotics, with an emphasis on intelligence, adaptability, and collaboration.
* As technology advances, we anticipate the integration of more advanced sensors, such as computer vision and tactile sensors, to enhance the robotic arm's perception and interaction capabilities.
* AI-driven control algorithms will enable these arms to learn and optimize their movements, adapting to dynamic environments and complex tasks.
* Collaboration between humans and robotic arms will be a focal point, with enhanced safety features and intuitive interfaces allowing for seamless interaction.
* The use of multiple robotic arms working collaboratively, possibly in swarm configurations, is likely to become more prevalent, increasing efficiency in industrial applications.
* Cloud connectivity and edge computing integration will enable offloading computational tasks, facilitating remote monitoring, and execution of complex operations.

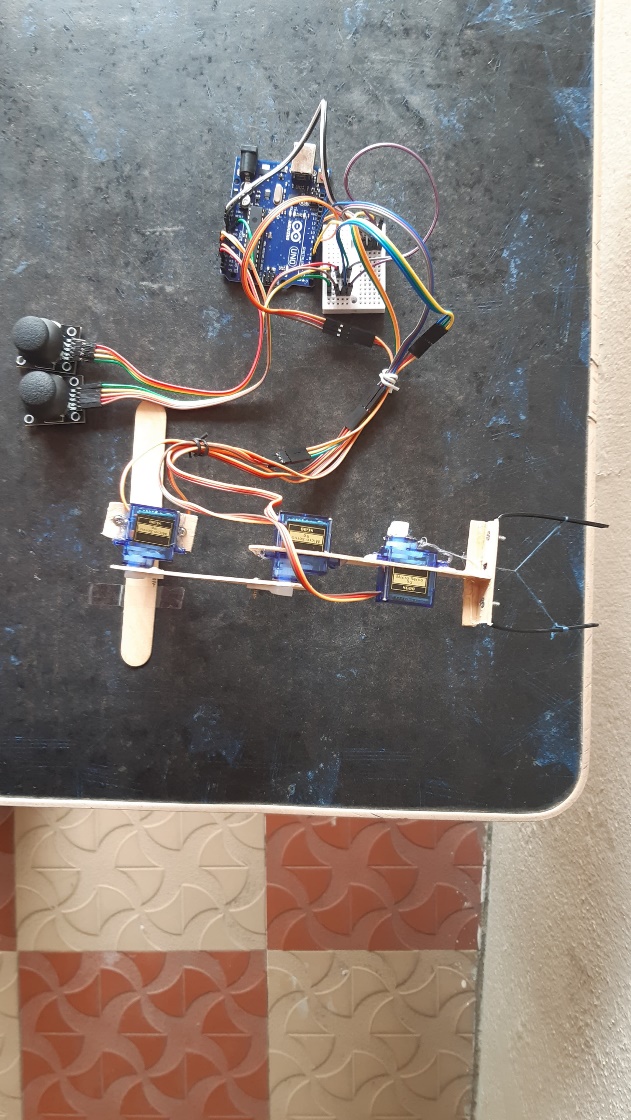
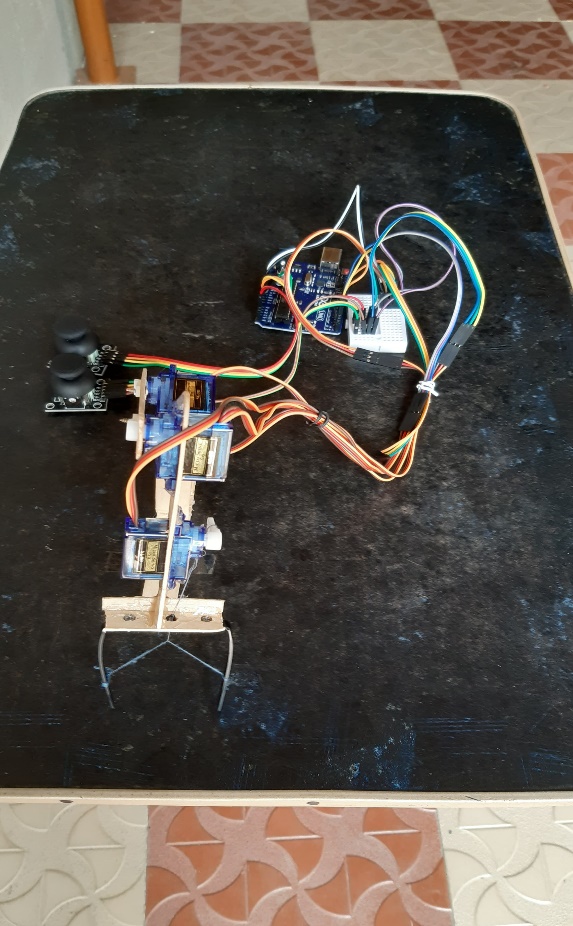
**APPENDIX**

**PROTOTYPE MODEL FOR ROBOTIC ARM USING JOYSTICK**





**fig 4.1 Back view fig 4.2 Side view**

Figure 4.1 shows the back view of Robetic armand figure 4.2 shows the slide view of Robetic arm.

**fig 4.3 Front view fig 4.4 Top view**

Figure 4.3 shows the front view of Robetic armand figure 4.4 shows the top view of Robetic arm.