PESU I/O slot-15

PES UNIVERSITY

PROJECT REPORT

ROBOTIC ARM

Course: Introduction to Microcontrollers
Instructor: Aniruddhan (2022)
Student Name: Chandan M
Student ID: PESUG22EC059

1. Introduction

1.1 Project Overview

This project aims to design and develop a robotic arm that can perform the basic task of picking and placing objects. The primary focus is on automating this process using an ESP32 microcontroller, which serves as the brain of the robotic arm. The arm is designed to mimic the movements of a human arm, making it capable of performing tasks that require precision and repeatability, which are crucial in industrial automation.

1.2 Objectives

The main objectives of the project are:

- To understand the fundamental working principles of microcontrollers and their applications in robotics.
- To design a mechanical system that can replicate the human arm's functions.
- To implement both manual and automatic control methods for the robotic arm using an ESP32 microcontroller.
- To develop skills in interfacing microcontrollers with motors, sensors, and other peripherals.

2. Project Description

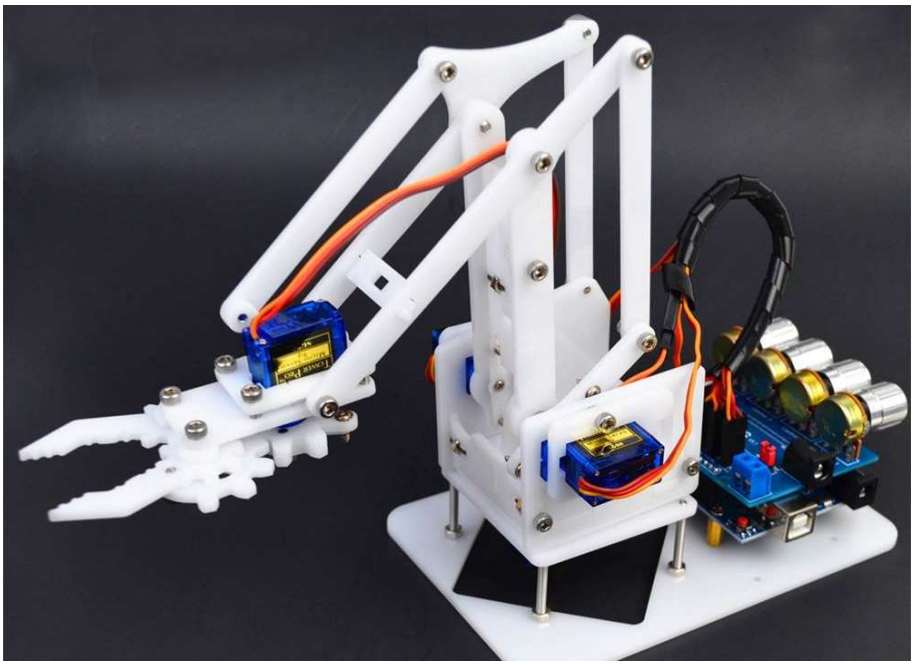
2.1 Components Used

The project utilizes the following key components:

- **Microcontroller:** ESP32, a powerful and versatile microcontroller with built-in Wi-Fi and Bluetooth capabilities.
- **Motors:** Three stepper motors are used to control the base rotation, arm movement, and the gripper for picking up objects.
- **Grippers:** Mechanical grippers are attached to the end of the arm to handle objects.
- **Power Supply:** A 5V power supply is used to power the ESP32 and motors.
- **Additional Components:** Resistors, capacitors, diodes, and a breadboard are used for circuit stability and prototyping.

2.2 Circuit Diagram

The circuit is designed to interface the ESP32 with the stepper motors and grippers. The ESP32 controls the motors using pulse width modulation (PWM) signals, which dictate the speed and position of the motors. The circuit includes motor drivers to handle the high current required by the stepper motors. The connections are carefully planned to ensure that the robotic arm operates smoothly and without interference.



2.3 Working Principle

The robotic arm is programmed to follow a specific sequence of operations: picking up an object, rotating to a designated position, and then placing the object. The ESP32 microcontroller sends control signals to the stepper motors based on the predefined instructions coded in the program. The stepper motors control the movement of the base, arm, and gripper, allowing for precise positioning and handling of objects. The system can operate in both manual mode, where a user controls the movements via an interface, and automatic mode, where the arm follows a preset routine.

3. Software Design

3.1 Programming Environment

The software for controlling the robotic arm was developed using the following tools:

- **IDE:** Arduino IDE was used for writing and uploading the code to the ESP32 microcontroller.
- **Programming Language:** The project was programmed in C/C++.
- **Libraries:** The `Stepper.h` library was used to control the stepper motors, and the `WiFi.h` library was included for potential wireless control.

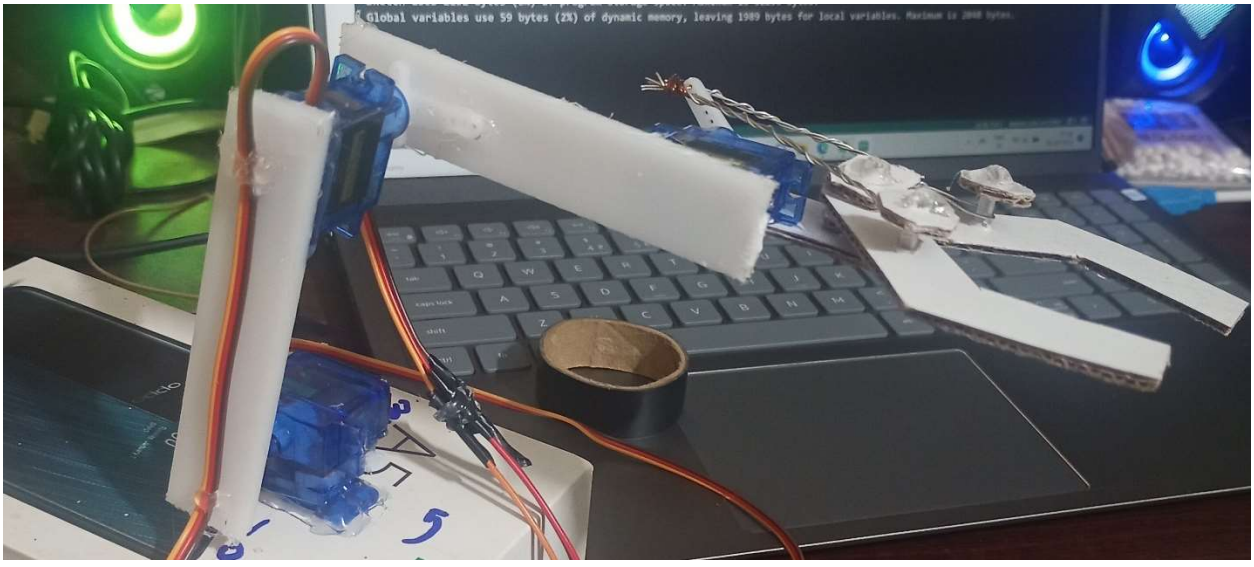
3.2 Code Explanation

The code is organized into various sections:

- **Initialization:** This section initializes the ESP32, sets up the motor pins, and configures the PWM signals.
- **Manual Control:** Functions in this section allow the user to control the arm via an interface. Commands are sent to the ESP32, which then adjusts the motor positions accordingly.
- **Automatic Control:** In this mode, the arm follows a predetermined sequence. The code uses loops and conditional statements to move the arm through the pick-and-place operation.
- **Error Handling:** Basic error handling is implemented to ensure the arm stops if an unexpected condition occurs, such as a motor stalling.

3.3 Flowchart

The flowchart for the robotic arm's operation starts with system initialization. It then checks whether the system is in manual or automatic mode. In manual mode, it waits for user input and then moves the arm accordingly. In automatic mode, it executes a series of movements to pick and place an object. The process repeats until the task is completed or the system is turned off.



4. Testing and Results

4.1 Testing

Testing involved verifying the movement accuracy of the robotic arm and ensuring that the ESP32 could effectively control the stepper motors. Each motor's calibration was done to ensure precise control. Tests were conducted for both manual and automatic modes. The arm's ability to pick and place objects of different weights and sizes was evaluated.

4.2 Results

The robotic arm successfully performed the pick-and-place operations as expected. It demonstrated reliable and repeatable movements in both manual and automatic modes. Challenges such as motor synchronization and power issues were encountered but resolved by fine-tuning the PWM signals and improving the circuit's power management.

5. Conclusion

5.1 Summary

This project provided valuable insights into the practical application of microcontrollers in robotics. The robotic arm was successfully designed, built, and programmed to perform basic pick-and-place tasks. The project met all objectives, including understanding microcontroller operations, interfacing with motors, and implementing control logic.

5.2 Future Work

Future improvements could include adding more degrees of freedom to the arm for increased versatility, integrating sensors for feedback control, and exploring advanced control algorithms. Additionally, enhancing the interface for easier manual control or adding wireless capabilities for remote operation would be beneficial.