Oblivion
ROBOTIC CLUB

Sep, 5, 2025

ARM PROJECT

Presented By
Oussama lechhab
Mohamed taleb

I- Indroduction

Context

Robotics and embedded systems are essential in modern technology, supporting applications in automation, industry, and research. ARM-based microcontrollers are widely used because of their efficiency, flexibility, and ability to handle real-time tasks. Building a robotic arm using ARM technology provides a practical way to apply embedded system concepts while addressing real engineering challenges.

Objectives

The main objectives of this project are:

- To design and implement a robotic arm system controlled by an ARM-based microcontroller.
- To integrate sensors and actuators for precise and reliable movement.
- To apply programming and embedded system principles in a real-world application.
- To evaluate the performance of the system through testing and experimentation.

Scope

This project covers the design, programming, and implementation of a robotic arm prototype. It focuses on the control system, movement accuracy, and integration of hardware and software. it remains within the scope of a functional prototype suitable for educational and demonstration purposes.

II- Background & Theoretical Concepts

ARM Architecture
The ARM architecture is one of the most widely used in embedded systems due to its high performance and low power consumption. It provides an efficient platform for real-time applications, making it suitable for robotics and automation. ARM microcontrollers are particularly effective for controlling actuators, reading sensors, and handling communication tasks simultaneously.

Servo Motors
Servo motors are a key component in robotic arms, as they allow precise control of angular position. Each joint of the arm is driven by a servo motor, enabling coordinated movements such as rotation, lifting, and gripping. By sending PWM (Pulse Width Modulation) signals from the ARM microcontroller, the position of each servo can be accurately adjusted.

3D Printing
To build the mechanical structure of the robotic arm, 3D printing technology was used. This method allowed us to design and manufacture custom parts that are lightweight, low-cost, and adaptable to the servo motors. The use of 3D printing ensured that the design could be easily modified and improved during the development process.

Web-Based Control
A web interface was developed to remotely control the robotic arm. This interface communicates with the ARM system and allows users to send movement commands directly from a computer or mobile device. By combining embedded hardware with web technologies, the project demonstrates how robotics can be integrated with the Internet of Things (IoT) concept.

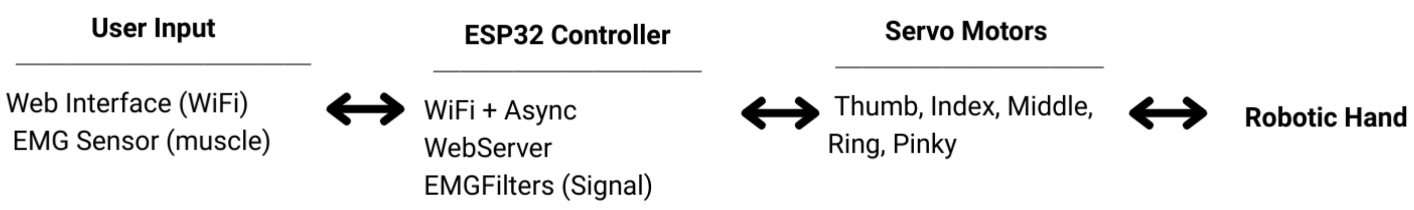
Technologies Used

- ARM-based microcontroller for system control.
- Servo motors for joint actuation.
- 3D printing for the robotic arm structure.
- Web development tools for the control interface.

III- System Design & Implementation

1. Architecture Overview

The system integrates user inputs, processing, and actuation to control a robotic hand. The high-level architecture is depicted below:



2. Program Flow / State Machine

1. **Initialization Phase**
    - Configure ServoManager and attach all servo motors.
    - Initialize EMG sensor and filtering pipeline.
    - Establish WiFi Access Point and launch the AsyncWebServer.
2. **Input Acquisition**
    - **Web Interface**: Listen for HTTP GET requests to toggle individual fingers.
    - **EMG Sensor**: Continuously read analog signals, filter them, and detect muscle activity exceeding a calibrated threshold.
3. **Control Logic (ServoManager)**
    - Determine the target finger(s) and movement (open/close).
    - Smoothly transition servo to the target angle using moveTo() function.
    - Update lastAngle for accurate state tracking.
4. **Actuation**
    - Send PWM signals to servo motors to actuate joints.
    - Translate servo motion into mechanical movement of the 3D-printed hand.
5. **Continuous Monitoring Loop**
    - Concurrently monitor web and EMG inputs.
    - Execute appropriate servo commands in real-time.
    - Provide status feedback via Web UI and Serial Monitor.

3. Module Descriptions

- **Communication Module**:
    - Implements WiFi AP mode on ESP32.

    - Hosts a non-blocking web server for finger control and status feedback.
- **Control Module**:
    - ServoManager class abstracts servo operations.
    - Supports individual finger control, all-finger commands, and toggle functionality.
    - Ensures smooth angular transitions and prevents abrupt movements.
- **Sensor Module**:
    - EMG sensor captures muscle signals for gesture-based control.
    - EMGFilters library filters noise and detects activation thresholds.
    - Provides robust input for dynamic hand movements.
- **Actuator Module**:
    - Five servo motors correspond to each finger.
    - PWM signals control position, velocity, and smoothness.
    - Actuators directly drive the 3D-printed hand joints.

4. Design Decisions

- **Hardware**:
    - **ESP32**: Chosen for integrated WiFi, sufficient GPIOs, and real-time PWM control.
    - **Servo Motors**: Lightweight, precise, and suitable for articulated finger control.
    - **3D-Printed Hand**: Customizable design, rapid prototyping, and mechanical flexibility.
- **Software**:
    - **Arduino Framework**: Simplifies embedded development and library integration.
    - **AsyncWebServer**: Provides non-blocking, real-time web control of the hand.
    - **EMGFilters**: Facilitates accurate and reliable processing of EMG signals.
- **Control Strategy**:
    - Modular separation of sensors, actuators, and communication ensures maintainability.
    - Supports dual control modes: web-based individual finger commands and EMG-driven full-hand gestures.
    - Smooth transitions and threshold-based EMG detection prevent mechanical stress and ensure safe operation.

V- Appendices

1- Source code

https://github.com/SENECA-1EE7/ARM-PROJEC

The repository includes:

- Sensor Code: Logic for reading and processing EMG sensor signals.
- Server Code: ESP32 web server handling communication between the robotic hand and the web interface.
- Component Control Code: ServoManager class controlling all servo motors of the robotic hand.
- Web Interface: HTML, CSS, and JavaScript files to operate the robotic hand remotely.
- the 3d components

2- EMG Sensor lirary and documents

https://www.mediafire.com/file_premium/5mdxiwpixvclsxu/Dry_Electrode_muscle_electrical_sensor.zip/file