Bluetooth Controlled Robotic Arm

Engineer: Alimul Fazal Nabil
Role Alignment: Infrastructure & Azure Cloud Systems Engineer | Robotics Researcher
Year: 2020 (Extended with 2026 Cloud Upgrade Design)

PART 1

1. Abstract

This thesis presents the design and implementation of a Bluetooth-controlled robotic arm using Arduino Mega 2560. The system demonstrates low-cost robotic manipulation using wireless communication, real-time embedded control, and modular mechanical design. The project served as a foundation for later work in ROS2 robotics, autonomous drones, and cloud-integrated IoT systems.

Key Contributions:

- Low-cost robotic manipulator architecture
- Stable servo control with dedicated power regulation
- Bluetooth command buffering to reduce latency
- Modular firmware design
- Future integration path with Azure IoT

2. Introduction

2.1 Motivation

Industrial robotic systems are expensive. Developing an affordable robotic arm enables education, research, and small-scale automation. This project aimed to create a prototype demonstrating real-time robotic manipulation with wireless control.

2.2 Objectives

• Build a 4-6 DOF robotic arm
• Enable mobile Bluetooth control
• Achieve stable servo motion
• Design scalable firmware
• Prepare system for ROS2 & cloud integration

3. Literature Review

Discuss:

- Industrial robot manipulators
- Arduino robotics research

- Bluetooth remote control systems
- Embedded servo control algorithms
- IoT robotics integration trends

Compare with existing robotic arms in education labs.

4. System Design Overview

4.1 Functional Architecture

Mobile App → Bluetooth → Arduino Controller → Servo Motors → Robotic Arm

4.2 Non-Functional Requirements

• Low cost (< $60)
• Reliable control latency <150ms
• Safe motion control
• Modular firmware
• Upgrade path to ROS2

5. Hardware Design

5.1 Component Selection

Arduino Mega chosen for high I/O count. HC-05 for reliable Bluetooth SPP communication.

5.2 Mechanical Design

Axis:

- Base rotation
- Shoulder lift
- Elbow bend
- Wrist rotate
- Gripper

Materials: Aluminum + Acrylic + 3D printed joints.

5.3 Electrical Design

Separate power rails for controller and motors to avoid jitter.
Add capacitor smoothing and fuse protection.

6. Software Architecture

6.1 Firmware Modules

- Command Parser
- Servo Controller
- Calibration Engine
- Safety Interrupt Handler

6.2 Mobile Application

Android app built for directional control and preset motion routines.

6.3 Communication Protocol

Command Format: J:ANGLE
Example: J2:90

7. Control Algorithms

Discuss PWM generation, smoothing, rate limiting, and interpolation of servo motion.

8. Engineering Challenges

Servo jitter caused by unstable voltage.
Bluetooth delay due to rapid command bursts.
Mechanical calibration errors.

Solutions implemented using power isolation, buffering algorithms, and calibration mapping tables.

9. Testing Methodology

Tests included:
• Continuous operation test
• Load test
• Bluetooth range test
• Accuracy measurement

Results showed stable operation with ±2° accuracy.

10. Results & Evaluation

System achieved reliable motion, stable wireless control, and cost efficiency. The robotic arm can be used as an educational manipulator.

---

11. Future Work

• ROS2 control integration
• Computer vision grasping
• Cloud telemetry with Azure IoT Hub
• Edge AI motion planning
• Digital twin simulation

12. Conclusion

This project provided foundational robotics engineering skills later applied to drone systems, computer vision pipelines, and hybrid cloud monitoring design.

PART 2 — GITHUB README (PROFESSIONAL VERSION)

Project Overview

Bluetooth Controlled Robotic Arm built with Arduino Mega. Demonstrates wireless robotic manipulation and embedded control engineering.

Tech Stack

Arduino Mega 2560
HC-05 Bluetooth
Servo Motors
Android App
Embedded C

Features

• Real-time Bluetooth control
• Multi-axis robotic motion
• Calibration system
• Power-stable design

Architecture Diagram

Mobile App → Bluetooth → Arduino → Servo Driver → Robotic Arm

Setup Instructions

1. Upload firmware using Arduino IDE
2. Pair HC-05 Bluetooth
3. Open mobile app
4. Send joint commands

Sample Command

J3:120

Future Roadmap

ROS2 integration
Azure IoT telemetry
Computer vision automation

PART 3 — AZURE IOT UPGRADE DESIGN

Objective

Upgrade robotic arm to cloud-connected telemetry system using Azure IoT Hub.

Architecture

Robotic Arm → ESP32 Gateway → Azure IoT Hub → Azure Functions → Power BI Dashboard

Features

• Real-time motion telemetry
• Remote firmware updates
• Predictive maintenance alerts
• Usage analytics dashboard

Azure Services Used

Azure IoT Hub
Azure Functions
Azure Digital Twins
Azure Storage
Power BI
GitHub Actions CI/CD

DevOps Pipeline

Firmware → GitHub Actions → OTA Update → IoT Device Deployment

This integrates robotics with your hybrid cloud engineering expertise.