Gesture-Controlled Robot Car with WiFi Camera Streaming

EE186 Final Project Report

Ronny Majeed Abdullah

Fall 2025

Contents

1 Final Specification **2**
1.1 Project Description 2
1.2 Core Functions 2
1.3 Capabilities Demonstrated (Required Categories) 2

2 System Design **3**
2.1 Block Diagram 3
2.2 Hardware Components 4
2.3 Electrical Connections 5
2.4 Vehicle Photo 5

3 Implementation **6**
3.1 Microcontroller Firmware 6
3.2 Web Viewer Application 6
3.3 Timer/Interrupt Usage 6

4 Results **6**
4.1 Performance Summary 6
4.2 Demonstration 6

5 Challenges and Lessons Learned **7**
5.1 Challenges 7
5.2 Solutions 7

6 Future Improvements **7**

7 Conclusion **7**

1 Final Specification

1.1 Project Description

This project implements a gesture-controlled robot car with real-time wireless video feedback. A sensor glove with four Force Sensing Resistors (FSRs) allows control of the robot's directional movement over Bluetooth. A separate STM32 board interfaces with an OV2640 camera and ESP32 AirLift WiFi co-processor to provide live JPEG streaming via a web application.

1.2 Core Functions

- Read analog inputs from 4 FSRs to detect hand gestures
- Transmit directional commands over Bluetooth (HC-05)
- Drive dual DC motors for robot car movement
- Capture visual data using OV2640 camera
- Stream compressed frames over WiFi
- Display live camera feed via a Next.js web interface

1.3 Capabilities Demonstrated (Required Categories)

Our project successfully demonstrates skills in the following:

- **Serial Communication:** UART link with HC-05 Bluetooth modules
- **Interfacing with Peripherals:** OV2640 camera, motor driver (L298N), ESP32 WiFi module
- **Analog-to-Digital Conversion:** 4 FSR sensors read via ADC channels on STM32L4R5
- **Interrupts and DMA:** UART RX interrupts for commands, DMA for camera DCMI capture

2 System Design

2.1 Block Diagram

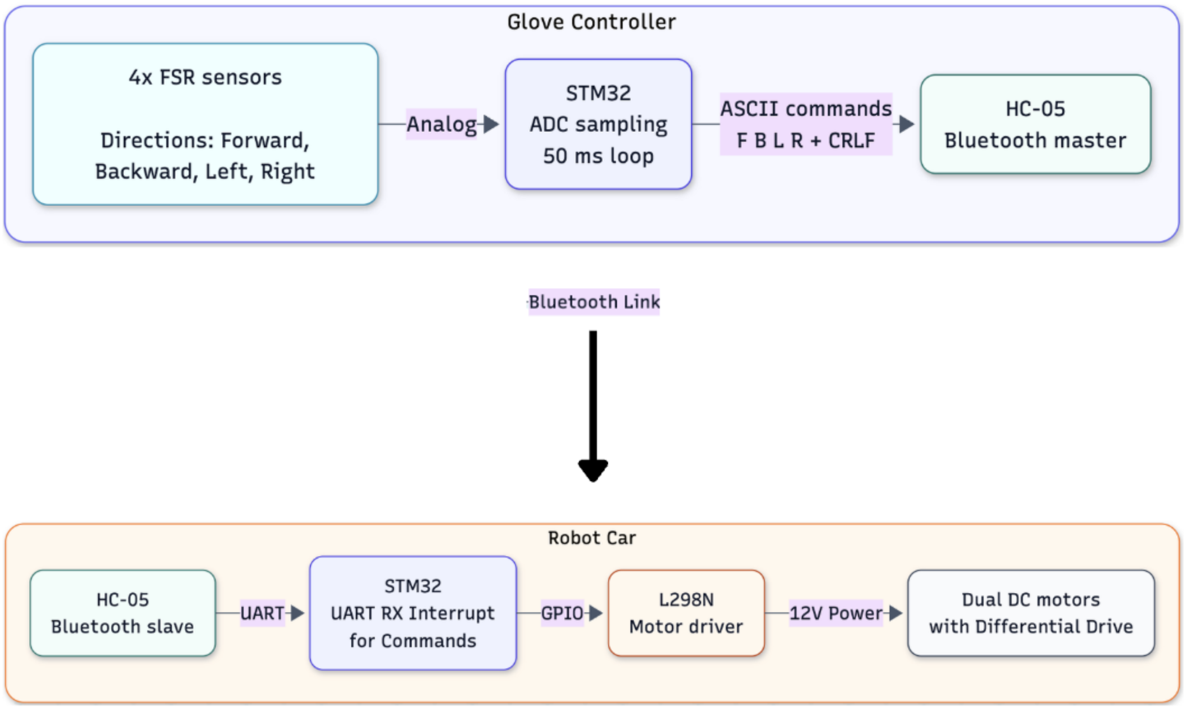

Figure 1: Controller Device to Target Device Block Diagram

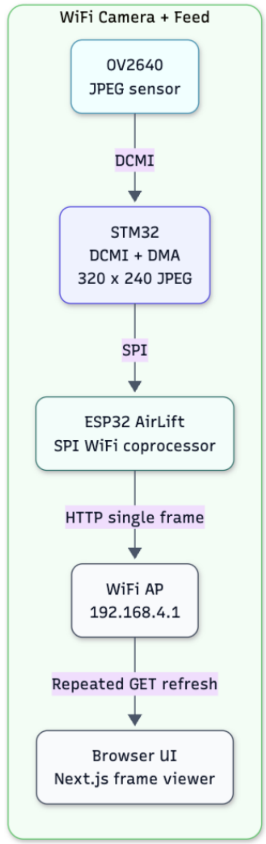

Figure 2: WiFi Camera + Feed Block Diagram

2.2 Hardware Components

- 3x STM32L4R5ZI Nucleo-144 boards
- 4x FSR pressure sensors
- 2x HC-05 Bluetooth modules
- OV2640 camera sensor
- ESP32 AirLift (WiFiNINA)
- Dual DC motors + L298N motor driver

2.3 Electrical Connections

Table 1 summarizes key pins:

Component	STM32 Pin(s)
FSR (Forward)	PC0 (ADC1_CH1)
FSR (Backward)	PC3 (ADC1_CH4)
FSR (Left)	PC1 (ADC1_CH2)
FSR (Right)	PC4 (ADC1_CH13)
Bluetooth TX/RX	USART2/USART3
Motor 1	PF13
Motor 2	PF14
OV2640 DCMI	PA4–PA6, PB6–PB9
ESP32 SPI	PB13–PB15 (SPI2)
ESP32 CS/RST/RDY	PG0, PG1, PF9

Table 1: Key electrical connections

2.4 Vehicle Photo

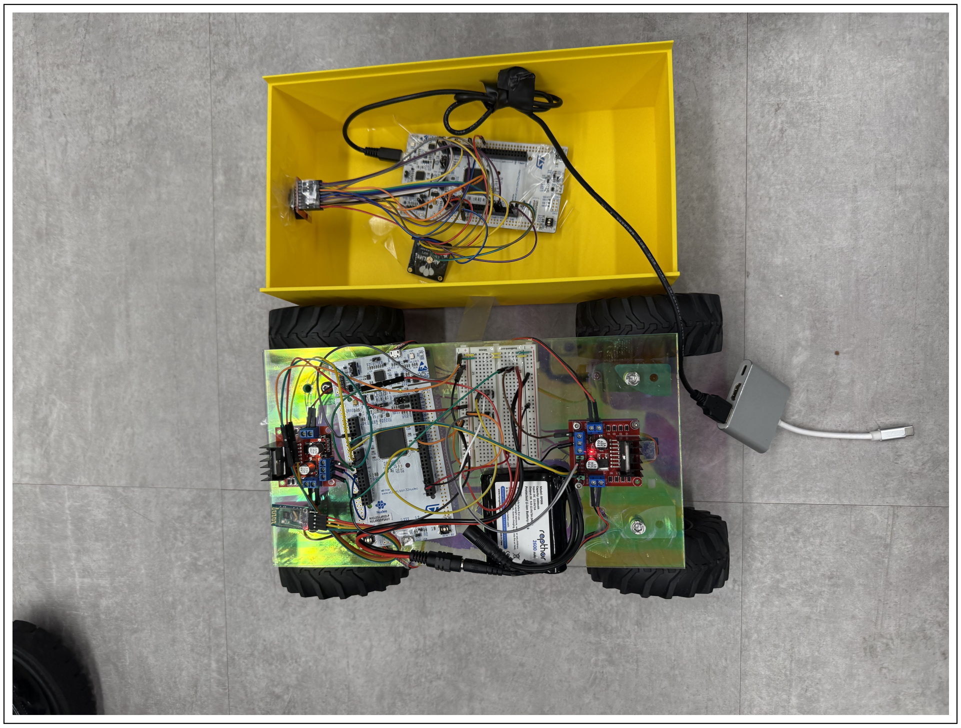

Figure 3: Internal Circuit for Motor Drivers and Camera

3 Implementation

3.1 Microcontroller Firmware

- Glove MCU: Reads ADC values every 50ms, sends ASCII commands (F,B,L,R) with `\r\n`
- Car MCU: UART interrupt handler decodes incoming commands and drives motor GPIOs
- Camera MCU: DCMI with DMA captures 320×240 JPEG frames; served via HTTP on port 80

3.2 Web Viewer Application

A lightweight Next.js UI fetches frames repeatedly using HTTP GET from WiFi AP `192.168.4.1`.

3.3 Timer/Interrupt Usage

- UART RX interrupt for motor direction commands
- DMA interrupt for camera frame capture complete
- Polling timer for ADC gesture sampling

4 Results

4.1 Performance Summary

Metric	Result
Gesture command latency	$\approx$ 80–120 ms
Camera frame rate	$\approx$ 2 FPS
WiFi streaming mode	HTTP single-frame refresh
Bluetooth operating range	About 6–10 meters
ADC threshold	1000 / 4095 (for gesture activation)

Table 2: Measured system performance

4.2 Demonstration

The system successfully enabled:

- Reliable Bluetooth control of robot movement
- Live camera preview on laptop browser
- Combined teleoperation experience

5 Challenges and Lessons Learned

5.1 Challenges

- Noise in analog FSR readings required threshold tuning
- Bluetooth sometimes dropped linking during motion current spikes
- Limited power for motor drivers and STM32 boards
- Achieving stable WiFi streaming with limited MCU memory

5.2 Solutions

- Enabled moving average filtering on ADC input
- Added a 12V Li-Po battery capable of 1A output
- Improved motor driver isolation and grounding
- JPEG compression + reduced resolution

6 Future Improvements

- Continuous MJPEG streaming for ¿10 FPS
- PWM speed control for differential steering
- IMU-based gesture recognition
- Mobile app integration
- Obstacle avoidance using ultrasonic sensors or computer vision

7 Conclusion

We successfully created a functional gesture-controlled robot car system with wireless video streaming, demonstrating our capabilities in embedded systems design across sensing, communication, motor control, and real-time imaging.