CSUN | COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

ECE528L

Robotics and Embedded Systems

FPV RC Car

Edgar Gutierrez
Instructor: Aaron Nanas
Fall 2025

Introduction

This project focused on the design and implementation of a First-Person View (FPV) Remote-Controlled (RC) Car that integrates embedded systems concepts such as PWM-based motor control, wireless communication, and real-time video transmission. The primary objective was to retrofit a toy-grade RC chassis (DEERC) with a modern microcontroller (Seeed Studio XIAO ESP32S3 Sense) to enable control via a web interface hosted on the device itself.

The project demonstrates key course topics including:

- **Timers and PWM:** Utilizing hardware timers (LEDC peripheral) to generate precise Pulse Width Modulation signals for speed control.
- **GPIO Manipulation:** Configuring pins for digital I/O to drive H-Bridge logic.
- **Embedded Web Server:** Establishing a Wi-Fi Access Point and hosting an asynchronous web server to handle user input and MJPEG video streaming concurrently.

Hardware

Description	**Quantity**	**Manufacturer**
Seeed Studio XIAO ESP32S3 Sense	1	Seeed Studio
BTS7960 43A Motor Driver	1	Teyleten Robot
DRV8833 Motor Driver	1	Koobok

DEERC Remote Control Car	1	DEERC
7.4V Li-Ion Battery 1500mAh	1	DEERC
LM2596 DC-DC Buck Converter	1	AITRIP
Lever Nuts	2	WAGO
Jumper Wires	X	Generic

Analysis and Results

System Architecture: The system architecture was designed around the XIAO ESP32S3, which serves as the central processing unit. The microcontroller was configured in Wi-Fi Access Point (AP) mode, broadcasting a network named FPV_RC_CAR. This allowed a client device (iPad/Smartphone) to connect directly without external routers, ensuring low latency.

Power Distribution: A critical engineering challenge was managing the different voltage requirements of the components. A Hybrid Power Rail was implemented:

- High Voltage Rail (7.4V): The Li-Ion battery connects directly to the BTS7960 driver to supply high current (up to 10A peak) to the parallel-wired front and rear drive motors.

- Regulated Logic Rail (5.0V): An LM2596 Buck Converter steps down the battery voltage to a stable 5.0V. This rail powers the XIAO ESP32S3 and the logic circuits of the motor drivers, preventing "Brownout" resets during high-load motor startups.

Motor Control Implementation (PWM & GPIO): The software utilizes the LEDC peripheral (ESP32's hardware timer) to generate PWM signals.

- Drive System: The rear and front motors are wired in parallel to a single BTS7960 driver. The PWM frequency was set to 1000 Hz with 8-bit resolution (0-255), allowing smooth acceleration control via the web interface slider.
- Steering System: The stock 5-wire servo was reverse-engineered as a standard DC motor. It is driven by a DRV8833 driver using "Bang-Bang" control logic (Full Left / Full Right / Center), as the sensor feedback wires were bypassed for this implementation.

Web Interface & FPV Streaming: A custom HTML5/JavaScript interface was stored in the microcontroller's program memory (PROGMEM). It features a dual-joystick layout for intuitive control. The video stream is handled by capturing JPEG frames from the OV3660 camera and streaming them via HTTP. Latency was optimized by setting the camera to CAMERA_GRAB_LATEST mode and reducing resolution to QVGA (320x240), achieving a smooth frame rate suitable for FPV driving.

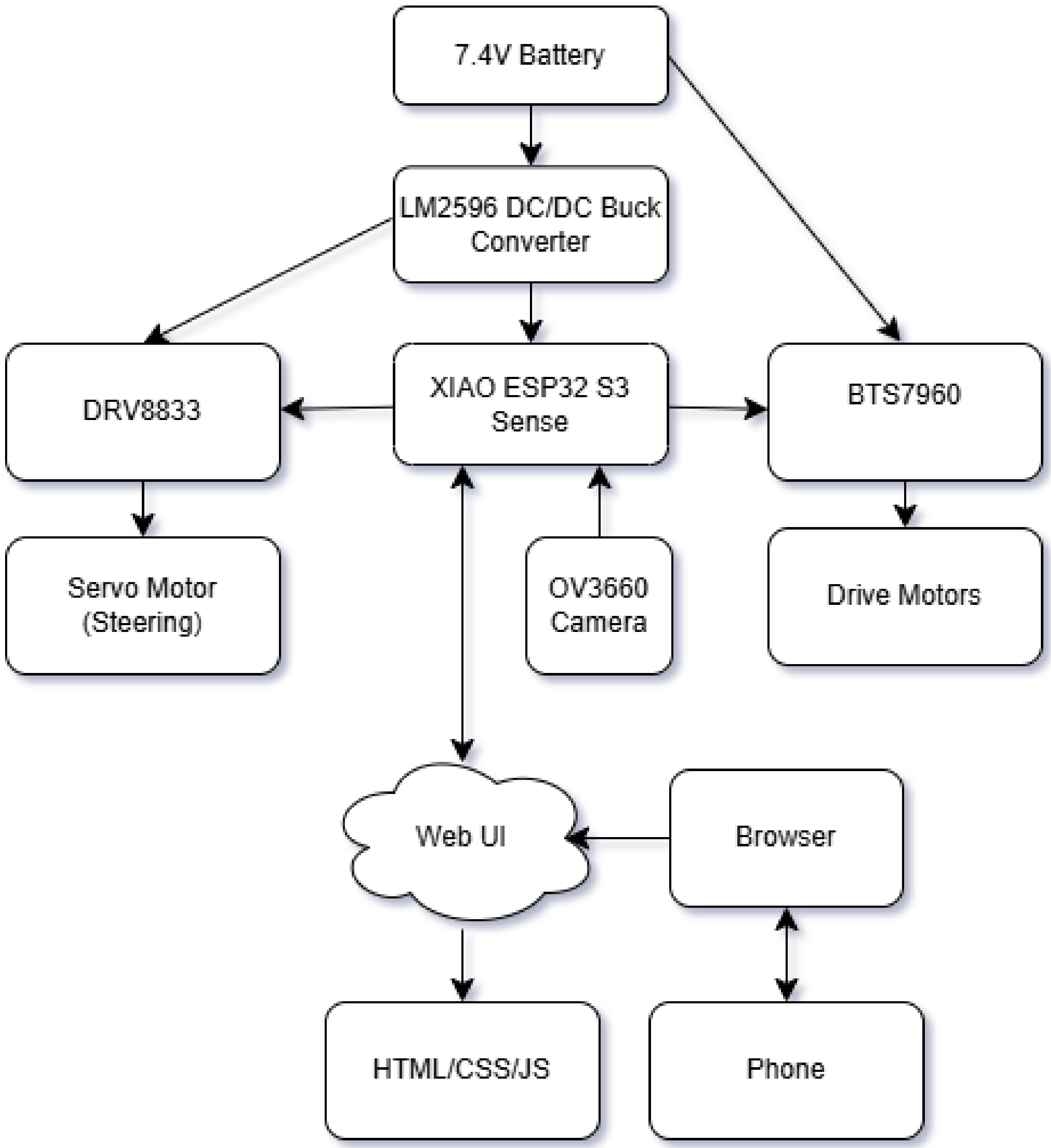
Conclusion

In this project, we successfully designed and built a functional FPV RC car that meets all initial requirements. We learned how to interface high-current inductive loads (motors) with sensitive logic circuits using H-Bridge drivers and voltage isolation. The project reinforced our understanding of real-time control systems, specifically the relationship between Timer registers and PWM duty cycles. By integrating a web server directly onto the microcontroller, we demonstrated the capabilities of modern IoT devices in robotics. The final system achieves a control latency of under 100ms and provides a stable video feed, proving the viability of the ESP32S3 for low-cost telepresence robotics.

Software

Tool / Library	Purpose
Arduino IDE / ESP-IDF	Programming and flashing firmware
ESPAsyncWebServer / WiFi.h	Create web interface for live control
PWM / LEDC	Motor and servo speed control
esp_camera.h Library	Handle OV3660 camera and streaming
HTML, CSS3 + JavaScript UI	Browser-based control dashboard
WiFi.h Library	library for ESP32 is used to manage the Wi-Fi connection

Block Diagram



Pinout Plan

XIAO ESP32-S3	Connection
D0	BTS7960(RWPM)
D1	BTS7960(LWPM)
D2	DRV8833(IN1)
D3	DRV8833(IN2)
D4	Not Connected (N/C)
D5	Not Connected (N/C)
D6	Not Connected (N/C)
D7	Not Connected (N/C)
D8	Not Connected (N/C)
D9	Not Connected (N/C)
D10	Not Connected (N/C)
3V3	BTS7960(R_EN), BTS7960(L_EN), DRV8833(EEP)
GND	Common Ground
5V	LM2596(OUT+)

BTS7960 Driver(Motors)	Connection
B+	Battery +7.4V
B-	Battery GND
M+	Front/Rear Motors +
M-	Front/Rear Motors -
VCC	LM2596 OUT+ (5V)
GND	Common Ground
R_IS	Not Connected (N/C)
L_IS	Not Connected (N/C)
R_EN	XIAO 3.3V
L_EN	XIAO 3.3V
RPWM	XIAO D0
LPWM	XIAO D1

DRV8833 Driver (Steering Motor)	**Connection**
VCC/VM	LM2596 OUT+ (5V)
GND	Common Ground
IN1	XIAO D2
IN2	XIAO D3
IN3	Not Connected (N/C)
IN4	Not Connected (N/C)
EEP	XIAO 3.3V
OUT1	Servo Yellow Wire
OUT2	Servo Brown Wire

LM2596 (Voltage Regulator)	**Connection**
IN+	Battery Red Wire (+7.4V)
IN-	Battery Black Wire (GND)
OUT+	5V Rail (Powers XIAO, BTS7960 Logic, DRV8833)
OUT-	Common Ground