RC CAR WITH ESP32

Bluetooth Remote Control via Xbox Series X Controller

1. Project Description

This project consists of a four-wheeled vehicle remotely controlled via Bluetooth BLE. Communication between the Xbox Series X controller and the vehicle is handled by an ESP32, which reads the left joystick values and drives two DC motors through an H-bridge (L298N).

The system supports four directions of movement: forward, backward, turn right, and turn left. When the controller disconnects or the joystick is at rest, the motors stop automatically to ensure the safety of the device.

2. Hardware Components

Component	Quantity	Function
ESP32 (DevKit)	1	Main microcontroller + BLE
H-Bridge L298N	1	Motor direction control
DC Gear Motor	2	Vehicle traction
4WD Robot Chassis	1	Physical structure of the vehicle
Breadboard	1	Temporary wiring connections
Xbox Series X Controller	1	Remote control interface
Dupont Cables	Various	Connections between components
Power Supply	1	Power for motors and ESP32

3. Electrical Wiring

ESP32 to H-Bridge (L298N)

ESP32 Pin	L298N Pin	Description
GPIO 26	IN1	Motor A control — direction 1
GPIO 27	IN2	Motor A control — direction 2

GPIO 14	IN3	Motor B control — direction 1
GPIO 12	IN4	Motor B control — direction 2

Note: The ENA and ENB pins on the L298N are connected to 5V to enable both motors at full speed. Motor power is supplied directly to the H-bridge.

4. System Architecture

The system follows a three-layer architecture:

Layer	Component	Technology
Input	Xbox Series X Controller	Bluetooth BLE
Processing	ESP32 + Arduino Firmware	C++ / Arduino IDE
Actuation	L298N + DC Motors	Digital GPIO signals

5. Control Logic

The Xbox controller's left joystick returns integer values in the range [-32768, 32767] for each axis. A dead zone of ±8000 is applied to prevent unintended movement caused by joystick drift.

Condition	Movement	Pin state IN1/IN2/IN3/IN4
x > 0 (joystick up)	Forward	HIGH / LOW / HIGH / LOW
x < 0 (joystick down)	Backward	LOW / HIGH / LOW / HIGH
y > 0 (joystick right)	Turn right	HIGH / LOW / LOW / HIGH
y < 0 (joystick left)	Turn left	LOW / HIGH / HIGH / LOW
Dead zone / disconnected	Stop	LOW / LOW / LOW / LOW

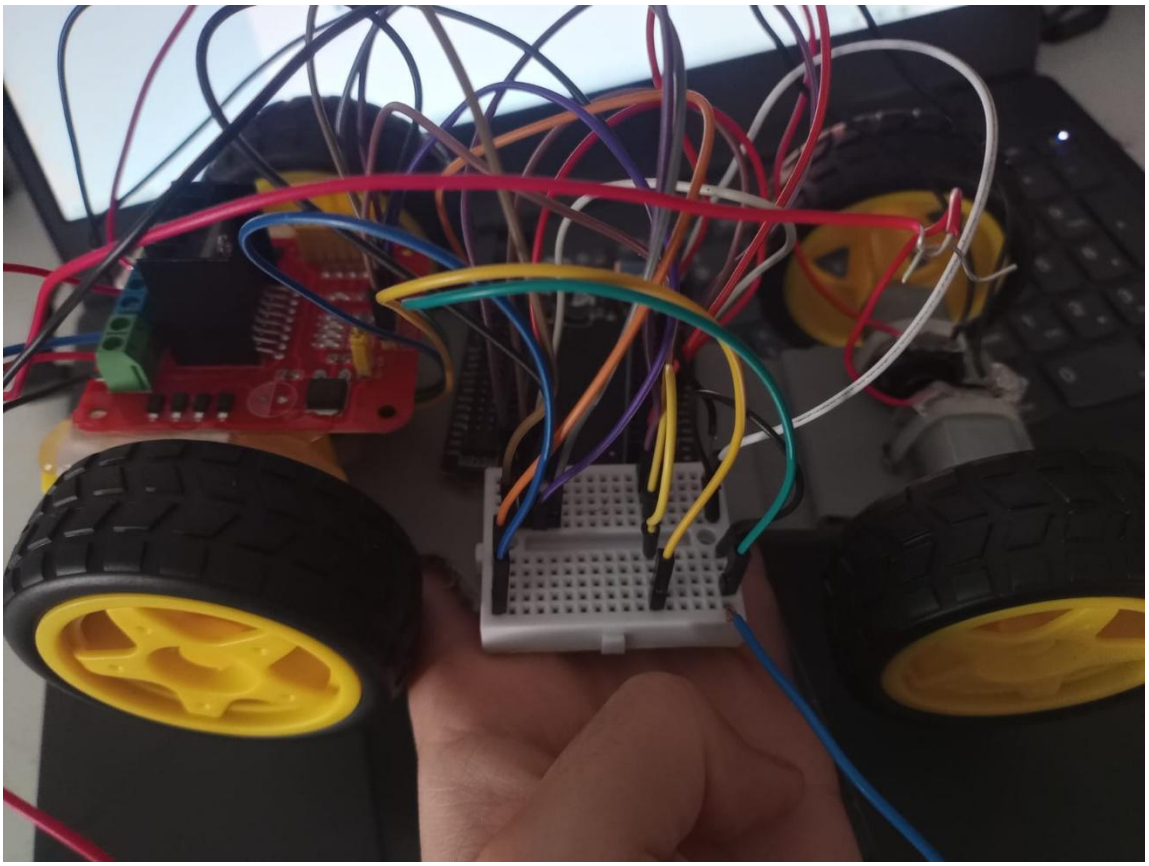
Movement priority: forward/backward takes precedence over lateral turns. The control loop runs every 20 ms (20 ms delay at the end of the loop).

6. Source Code

The firmware is written in C++ for Arduino IDE and uses the XboxSeriesXControllerESP32_asukiaaa library for BLE communication with the controller.

```
#include <XboxSeriesXControllerESP32_asukiaaa.hpp>

XboxSeriesXControllerESP32_asukiaaa::Core xboxController;

const int IN1 = 26;
const int IN2 = 27;
const int IN3 = 14;
const int IN4 = 12;

void setup() {
  Serial.begin(115200);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  xboxController.begin();
}

void loop() {
  xboxController.onLoop();
  if (xboxController.isConnected()) {
    int y = xboxController.xboxNotif.joyLHori;
    int x = xboxController.xboxNotif.joyLVert;
    if (abs(x) < 8000) x = 0;
    if (abs(y) < 8000) y = 0;
    if (x > 0) {
      digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
      digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
    } else if (x < 0) {
      digitalWrite(IN1, LOW);  digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);  digitalWrite(IN4, HIGH);
    } else if (y > 0) {
      digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);  digitalWrite(IN4, HIGH);
    } else if (y < 0) {
      digitalWrite(IN1, LOW);  digitalWrite(IN2, HIGH);
      digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
    } else { stopMotors(); }
  } else { stopMotors(); }
  delay(20);
}

void stopMotors() {
  digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
}
```

7. Prototype



8. Library Used

XboxSeriesXControllerESP32_asukiaaa

- Author: asukiaaa
- Platform: ESP32 (Arduino IDE)
- Protocol: Bluetooth Low Energy (BLE)
- Installation: Arduino Library Manager or GitHub

Main methods and properties used:

Method / Property	Description
xboxController.begin()	Initializes the BLE receiver
xboxController.onLoop()	Updates controller state on every cycle
xboxController.isConnected()	Returns true if the controller is connected

xboxNotif.joyLVert	Vertical value of the left joystick (logical X axis)
xboxNotif.joyLHori	Horizontal value of the left joystick (logical Y axis)

9. Possible Improvements

- Speed control using PWM (analogWrite) on the ENA/ENB pins of the L298N.
- Add an HC-SR04 ultrasonic sensor for obstacle detection and automatic braking.
- Implement proportional speed control based on joystick magnitude.
- Add LED indicators to signal movement direction.
- Use the right joystick for independent motor control (tank drive).
- Add an FPV camera for first-person view.

10. References

- XboxSeriesXControllerESP32_asukiaaa library: https://github.com/asukiaaa/XboxSeriesXControllerESP32_asukiaaa
- ESP32 Arduino documentation: https://docs.espressif.com/projects/arduino-esp32
- L298N Motor Driver datasheet: STMicroelectronics