Day 3 Wireless Communication Techniques with ESP32

Dipesh Bhattarai

4 july 2025

Introduction

In today's robotics training, we explored various wireless communication techniques, focusing on the ESP32 microcontroller. We learned how to connect the ESP32 to Wi-Fi and Bluetooth, integrate it with mobile devices, and implement IoT concepts. This knowledge culminated in a hands-on project where we controlled an LED remotely using Bluetooth and Wi-Fi.

ESP32 Microcontroller

The ESP32 is a low-cost, energy-efficient microcontroller developed by Espressif Systems. It features a dual-core 32-bit processor, typically the Tensilica Xtensa LX6, capable of operating at up to 240MHz. Integrated within the ESP32 are both Wi-Fi (802.11 b/g/n) and Bluetooth (v4.2 BR/EDR and BLE) capabilities, enabling seamless wireless communication. Additionally, it offers a wide array of peripherals, including 34 programmable GPIOs, 12-bit ADCs, 8-bit DACs, PWM, SPI, I^{2}C, and UART interfaces, making it suitable for a broad spectrum of IoT and robotics projects. The microcontroller also supports advanced security features like secure boot, flash encryption, and cryptographic hardware acceleration, ensuring data integrity and protection against unauthorized access.

Wi-Fi Connectivity

The ESP32's built-in Wi-Fi support allows it to connect to existing wireless networks as a client (Station mode) or create its own network (SoftAP mode). This dual-mode capability facilitates the development of applications such as remote monitoring systems, web servers, and IoT devices that require internet connectivity. The microcontroller supports the 802.11 b/g/n standards, ensuring reliable and fast wireless communication. By leveraging the Wi-Fi connectivity, developers can implement features like over-the-air (OTA) updates, cloud data synchronization, and remote control of devices, enhancing the functionality and versatility of their projects.

Bluetooth Classic

Bluetooth Classic on the ESP32 enables continuous, bidirectional data streaming, making it ideal for applications that require real-time communication, such as audio transmission or serial data exchange. Utilizing the BluetoothSerial library, developers can establish Bluetooth connections to other devices, like smartphones or computers, facilitating remote control and monitoring functionalities. The ESP32's Bluetooth Classic support is compatible with a wide range of Bluetooth-enabled devices, ensuring broad interoperability and ease of integration into existing systems.

Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a power-efficient wireless communication protocol designed for short-range data exchange. The ESP32's BLE capabilities allow it to serve as both a peripheral and a central device, supporting a wide range of applications, from health monitoring systems to proximity-based services. BLE's low power consumption is particularly beneficial for battery-operated devices, extending their operational lifespan. The ESP32's BLE support includes features like advertising, scanning, and GATT (Generic Attribute Profile) services, enabling the creation of complex and efficient wireless communication networks.

Simultaneous Wi-Fi & Bluetooth Operation

The ESP32's architecture allows for the simultaneous operation of both Wi-Fi and Bluetooth radios, enabling devices to handle multiple communication tasks concurrently. This capability is achieved through time-division multiplexing, where the microcontroller allocates time slots to each radio, ensuring efficient and non-interfering operation. While the simultaneous use of both radios is supported, it's essential to consider the application's requirements and the potential impact on performance, as heavy real-time operations on both radios may lead to brief interruptions or reduced throughput.

LED Control Project

In our hands-on project, we connected an LED (with a current-limiting resistor) to a General Purpose Input/Output (GPIO) pin on the ESP32 using jumper wires. We then implemented two methods to control the LED remotely:

- **Bluetooth Classic:** By pairing the ESP32 with a smartphone or computer, we utilized the BluetoothSerial library to send commands ("1" to turn on, "0" to turn off) to the ESP32, which in turn controlled the LED's state.
- **Wi-Fi Web Server:** The ESP32 was configured to host a simple web server. By accessing the ESP32's IP address through a web browser, we presented a user interface with ON/OFF buttons that allowed us to control the LED remotely over the local network.

This project demonstrated the practical application of wireless communication techniques, showcasing how the ESP32 can serve as a bridge between hardware components and various communication protocols, enabling remote control and monitoring functionalities.