ESP-to-ESP Messaging System using ESP-NOW

# Introduction

This project demonstrates a simple, local messaging system using two ESP32 microcontrollers communicating over the ESP-NOW protocol. It operates without the need for a central Wi-Fi network or router. The system includes a predefined set of messages sent by pressing buttons and displayed on an OLED screen, along with a buzzer notification on message reception.

# Problem Statement

In environments where internet connectivity is unavailable or impractical—such as remote field operations, temporary setups, or secure lab environments—there’s often a need for direct device-to-device communication. This project provides a reliable, short-range, offline communication solution using ESP-NOW to transmit short predefined messages between ESP32 boards.

# Features

- Offline communication using ESP-NOW (no Wi-Fi or router required)

- 5 predefined message buttons

- OLED screen with separate Sent and Received message display

- Buzzer alert on receiving a message

- Each device can both send and receive messages

# Technical Details

• Microcontroller: ESP32 (x2)  
• Display: 0.96" I2C OLED (SDA: GPIO21, SCL: GPIO22)  
• Buttons: GPIO 13, 12, 14, 27, 26 (with INPUT\_PULLUP)  
• Buzzer: Active Buzzer on available GPIO  
• Communication Protocol: ESP-NOW  
• No internet or router required

# Use Cases

- Quick messaging between devices in a smart home

- Emergency or status alert system in small workspaces

- Local communication in IoT deployments without Wi-Fi infrastructure

- Temporary networks for disaster-prone or outdoor field setups

# Limitations

- Limited range (typically ~30m indoors, ~100m line-of-sight)

- Message transmission is MAC address specific

- Not suitable for high-bandwidth or long-range communication

# Future Enhancements

- Adding more ESP32 nodes to create a mesh-like alert network

- Integrating message logging or timestamping

- Upgrading to encrypted ESP-NOW for secure messaging

- Adding custom message composition with a keypad or touchscreen