

Comunicación LoRa con ESP32 y PlatformIO en macOS

En este tutorial aprenderás a comunicar dos placas [ESP32 con chip LoRa SX1276](#) y pantalla OLED integrada. A diferencia del WiFi o Bluetooth, LoRa permite alcances de kilómetros con un consumo mínimo.

1. Requisitos de Hardware

- [2x Placas: ESP32 LoRa \(Heltec V2 / TTGO / SX1276\).](#)
- **2x Antenas:** (Obligatorio conectarlas antes de encender para no quemar el chip).
- **Cables USB:** Conexión a tu Mac.

2. Configuración del Proyecto (PlatformIO)

Crea dos carpetas independientes: `lora_emisor` y `lora_receptor`. En **ambas**, el archivo `platformio.ini` debe ser idéntico para asegurar que las librerías coincidan:

`platformio.ini`

Ini, TOML

```
[env:heltec_wifi_lora_32_V2]
platform = espressif32
board = heltec_wifi_lora_32_V2
framework = arduino
monitor_speed = 115200

lib_deps =
    sandeepmistry/LoRa @ ^0.8.0
    olikraus/U8g2 @ ^2.35.19
```

3. El Código del Emisor (src/main.cpp)

Este código envía un contador que aumenta cada 2 segundos y lo muestra en el OLED.

C++

```
#include <Arduino.h>
#include <SPI.h>
#include <LoRa.h>
#include <U8g2lib.h>

// Definición de pines para el chip SX1276 interno
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI00 26

// Inicializar pantalla OLED (SDA 4, SCL 15, Reset 16)
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, 16, 15, 4);

int contador = 0;

void setup() {
  Serial.begin(115200);
  u8g2.begin();

  // Configuración de pines SPI para LoRa
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI00);

  // Iniciar LoRa (915MHz para América, 868MHz para Europa)
  if (!LoRa.begin(915E6)) {
    Serial.println("¡Fallo al iniciar LoRa!");
  }
}
```

```

    while (1);
}
Serial.println("Emisor listo.");
}

void loop() {
    String mensaje = "Paquete: " + String(contador);
    Serial.println("Enviando: " + mensaje);

    // Enviar por Radio
    LoRa.beginPacket();
    LoRa.print(mensaje);
    LoRa.endPacket();

    // Actualizar Pantalla
    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_ncenB08_tr);
    u8g2.drawStr(0, 20, "LORA EMISOR");
    u8g2.setCursor(0, 45);
    u8g2.print(mensaje);
    u8g2.sendBuffer();

    contador++;
    delay(2000);
}

```

4. El Código del Receptor (src/main.cpp)

Este código escucha el aire y, al recibir un paquete, muestra el contenido y la potencia de la señal (**RSSI**).

C++

```

#include <Arduino.h>
#include <SPI.h>
#include <LoRa.h>
#include <U8g2lib.h>

```

```

#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DI00 26

U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, 16, 15, 4);

void setup() {
    Serial.begin(115200);
    u8g2.begin();
    SPI.begin(SCK, MISO, MOSI, SS);
    LoRa.setPins(SS, RST, DI00);

    if (!LoRa.begin(915E6)) {
        Serial.println("Fallo al iniciar LoRa");
        while (1);
    }
    Serial.println("Receptor esperando...");
}

void loop() {
    int packetSize = LoRa.parsePacket();
    if (packetSize) {
        String recibido = "";
        while (LoRa.available()) {
            recibido += (char)LoRa.read();
        }

        int rssi = LoRa.packetRssi();

        // Mostrar en Monitor Serie
        Serial.printf("Recibido: %s | RSSI: %d dBm\n",
            recibido.c_str(), rssi);

        // Mostrar en OLED
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_ncenB08_tr);
        u8g2.drawStr(0, 15, "LORA RECIBIDO:");
        u8g2.setCursor(0, 35);
    }
}

```

```
    u8g2.print(recibido);  
    u8g2.setCursor(0, 55);  
    u8g2.printf("RSSI: %d dBm", rssi);  
    u8g2.sendBuffer();  
}  
}
```

5. Ejecución desde la Terminal (macOS)

Identifica tus puertos con `pio device list`. En este ejemplo:

- **Emisor:** `/dev/cu.usbserial-0001`
- **Receptor:** `/dev/cu.usbserial-6`

Paso A: Subir el Emisor

Abre una terminal en la carpeta `lora_emisor`:

Bash

```
pio run -t upload --upload-port /dev/cu.usbserial-0001  
pio device monitor --port /dev/cu.usbserial-0001 --baud 115200
```

Paso B: Subir el Receptor

Abre otra pestaña de terminal (`Cmd + T`) en la carpeta `lora_receptor`:

Bash

```
pio run -t upload --upload-port /dev/cu.usbserial-6  
pio device monitor --port /dev/cu.usbserial-6 --baud 115200
```

6. Resultado Esperado

En el monitor del **Receptor**, verás líneas como esta: Recibido:
Paquete: 42 | RSSI: -45 dBm

- **Nota sobre el RSSI:** Cuanto más cerca de 0 sea el número (ej: -30), más fuerte es la señal. Si llega a -120, la señal es muy débil y está al límite del alcance.