

# SENSORES Y ACTUADORES – TST 2024



# TP9-e2- Implementar la comunicación RF entre y RX -TX

### Comunicación LoRa entre dos Arduino usando el Módulo LoRa SX1278

Se envían valores de temperatura y humedad de un Arduino a otro utilizando el módulo LoRa SX1278 El sensor DHT11 está conectado al lado transmisor, Arduino. Este Arduino obtendrá los valores de temperatura y humedad de DHT11 para luego enviar al otro Arduino a través del módulo LoRa SX1278.

Estos valores de humedad y temperatura se imprimirán en la pantalla LCD conectada al segundo Arduino.

# **Componentes:**

- 1 Arduino Uno (Transmisor)
- 1 Arduino Uno (Receptor)
- 2 módulos LoRa SX1278
- 1 Sensor DHT11
- 1 Módulo LCD 16×2

Cables de puente

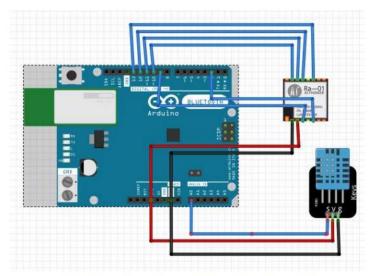
Transmisión: (Sx)

Usaremos un Arduino UNO con módulo LoRa y sensor DHT11.

La interfaz de Arduino UNO con LoRa y DHT11 se muestra a continuación:



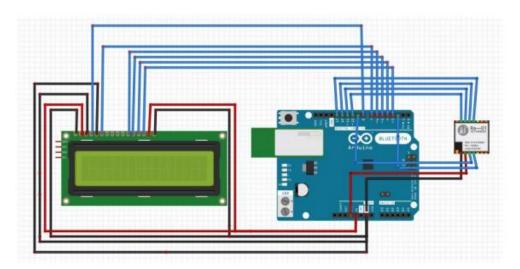
El diagrama del circuito para conectar el Arduino con LoRa y el módulo DHT11.



El diagrama del circuito para conectar el Arduino con LoRa y el módulo DHT11.

# Recepción: (Tx)

Utilizar un Arduino Uno con módulo LoRa y un módulo de pantalla LCD de 16 × 2.



El diagrama del circuito para conectar el Arduino con LoRa y el módulo LCD .

Se puede usar el módulo LoRa SX1278 con ESP32. LoRa es una tecnología de radiofrecuencia inalámbrica introducida por una empresa llamada Semtech destinada a transmitir información bidireccional a larga distancia sin consumir mucha energía. Aquí usaremos dos módulos LoRa, uno con ESP32 y otro con placa Arduino. La configuración ESP32 LoRa solicitará los datos meteorológicos de una API para cualquier ciudad



determinada y los enviará a la configuración Arduino LoRa. Luego, Arduino LoRa recibirá los datos y los mostrará en una pantalla LCD de 16x2. Entonces ESP32 LoRa actuará como transmisor y Arduino LoRa como receptor. También se conecta un OLED al ESP32 para mostrar la temperatura y la humedad en el lado del remitente. Aprenda cómo podemos utilizar el módulo LoRa SX1278 con la placa de desarrollo ESP32.

Este sistema consta de dos ESP32: uno que actúa como emisor y el otro como receptor. La comunicación entre ambos se realizará mediante LoRa, y cada uno tendrá una pantalla OLED para mostrar los datos.

Para implementar un sistema de monitoreo de datos utilizando ESP32 y el módulo LoRa SX1278 (con pantalla OLED en el emisor y receptor), se puede seguir estos pasos.

### **Componentes necesarios**

- 1. 2 x ESP32 : uno para el emisor y otro para el receptor.
- 2. 2 x Módulo LoRa SX1278 : para la comunicación de largo alcance.
- 3. 2 x Pantalla OLED (SSD1306): para mostrar los datos en ambos nodos.
- 4. Cables y protoboard.

### Librerías necesarias

En el IDE de Arduino, necesitas las siguientes librerías:

- LoRa.h: para manejar la comunicación LoRa.
- Adafruit\_SSD1306 y Adafruit\_GFX : para manejar la pantalla OLED.

Se puede instalar estas librerías desde el gestor de librerías del IDE de Arduino.

### **Conexiones**

Conexión entre ESP32 y LoRa SX1278

### PIN ESP32 Pin de LoRa

	V		C١	

Tierra Tierra

5 CCC

19 MISO

27 MOSI

#### PIN ESP32 Pin de LoRa

18 NSS/CS

14 Primera vez

26 Dio0

### Conexión entre ESP32 y OLED SSD1306

### **PIN ESP32 Pin OLED**

3,3 V CCV

Tierra Tierra

21 Adventista del Séptimo Día

22 LCC

**Nota**: Verifique que los pinos se ajusten a su hardware. Los pinos pueden variar según el modelo del ESP32 y el módulo LoRa que utiliza.

### Código para el Emisor (Sender)

Este código envía datos de ejemplo (como la temperatura y humedad) desde el emisor hacia el receptor.

```
CPP-P
Copiar código
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN WIDTH 128
#define SCREEN HEIGHT 64
Adafruit SSD1306 display(SCREEN WIDTH, SCREEN HEIGHT, &Wire, -1);
#define SCK
#define MISO
                19
#define MOSI
                27
#define SS
                18
#define RST
                14
#define DIO0
void setup() {
  Serial.begin(115200);
  while (!Serial);
```

```
// Inicializar OLED
  if (!display.begin(SSD1306 SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("No se pudo iniciar OLED"));
    while (true);
  }
  display.clearDisplay();
  // Inicializar LoRa
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(433E6)) { // Configurado para 433 MHz }
    Serial.println("Error al inicializar LoRa");
    while (true);
  Serial.println("LoRa iniciado");
  // Mostrar mensaje de inicio
  display.setTextSize(1);
  display.setTextColor(SSD1306 WHITE);
  display.setCursor(0, 10);
  display.println("Iniciando Emisor...");
  display.display();
  delay(1000);
}
void loop() {
  int temperatura = random(20, 30); // Valor de temperatura de ejemplo
  int humedad = random(40, 60);
                                     // Valor de humedad de ejemplo
  // Enviar datos
  LoRa.beginPacket();
  LoRa.print("Temp: ");
  LoRa.print(temperatura);
  LoRa.print("C, Hum: ");
  LoRa.print(humedad);
  LoRa.println("%");
  LoRa.endPacket();
  // Mostrar en OLED
  display.clearDisplay();
  display.setCursor(0, 10);
  display.print("Temp: ");
  display.print(temperatura);
  display.println("C");
  display.print("Hum: ");
  display.print(humedad);
  display.println("%");
  display.display();
  delay(2000);
}
```

# Código para el Receptor (Receiver)

El código del receptor recibe los datos y los muestra en la pantalla OLED.

```
CPP-P
Copiar código
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN WIDTH 128
#define SCREEN HEIGHT 64
Adafruit SSD1306 display(SCREEN WIDTH, SCREEN HEIGHT, &Wire, -1);
#define SCK
#define MISO
                19
#define MOSI
#define SS
                18
#define RST
                14
#define DIOO
void setup() {
  Serial.begin(115200);
  while (!Serial);
  // Inicializar OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("No se pudo iniciar OLED"));
    while (true);
  display.clearDisplay();
  // Inicializar LoRa
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO0);
  if (!LoRa.begin(433E6)) {
    Serial.println("Error al inicializar LoRa");
    while (true);
  Serial.println("LoRa iniciado");
  // Mensaje inicial en pantalla
  display.setTextSize(1);
  display.setTextColor(SSD1306 WHITE);
  display.setCursor(0, 10);
  display.println("Iniciando Receptor...");
  display.display();
  delay(1000);
}
void loop() {
  // Verificar si hay datos entrantes
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    String receivedData = "";
```

```
// Leer datos recibidos
while (LoRa.available()) {
    receivedData += (char)LoRa.read();
}

// Mostrar datos en serial y en OLED
Serial.print("Datos recibidos: ");
Serial.println(receivedData);

display.clearDisplay();
display.setCursor(0, 10);
display.println("Datos recibidos:");
display.setCursor(0, 30);
display.println(receivedData);
display.println(receivedData);
display.display();
}
```

## Explicación del Código

- Inicialización del OLED: En ambos códigos, se inicializa la pantalla OLED en la dirección 0x3C. Si tu pantalla OLED está en una dirección diferente, asegúrate de actualizarla.
- Configuración de LoRa: Ambos códigos inicializan LoRa en la frecuencia 433E6, adecuada para el módulo SX1278. Asegúrese de que la frecuencia sea la misma en ambos dispositivos.
- 3. **Envío de Datos** : El emisor envía datos de ejemplo, como temperatura y humedad, a intervalos de 2 segundos.
- 4. **Recepción y visualización de datos** : El receptor muestra los datos recibidos tanto en el monitor serial como en la pantalla OLED.

# Prueba y Solución de Problemas

- 1. **Pruebe el emisor y el receptor por separado** para asegurarte de que cada uno funcione correctamente con sus componentes.
- 2. Verifique las conexiones y las frecuencias en ambos códigos, deben coincidir.
- 3. **Revise el monitor serial** para ver si se reciben los datos correctamente.

Con estos pasos, se debería poder implementar un sistema de monitoreo de datos con ESP32 y LoRa.



# REFERENCIAS BIBLIOGRAFICAS

- 1. Adafruit. (Dakota del Norte). \*Biblioteca de gráficos Adafruit GFX\*. Adafruta. https://github.com/adafruit/Adafruit-GFX-Library
- 2. Adafruit. (Dakota del Norte componentes principales, librerías y dispositivos utilizados en el desarrollo del sistema de monitoreo de datos con ESP32 y LoRa SX1278: