Comunicación LoRa con ESP32: Nodo Central y Dos Nodos Emisores

# Introducción

Este documento describe cómo configurar una red básica LoRa entre tres módulos ESP32 con chips LoRa SX1278, donde dos dispositivos actúan como emisores (nodos A y B) y uno como receptor central. El sistema permite ingresar mensajes desde el terminal y recibir un acuse de recibido ("ACK") desde el nodo central.

# Esquema de Conexión del SX1278 al ESP32

|  |  |
| --- | --- |
| SX1278 | ESP32 GPIO |
| MISO | 19 |
| MOSI | 23 |
| SCK | 18 |
| NSS/CS | 5 |
| RST | 14 |
| DIO0 | 2 |

# Flujo de Comunicación

1. El usuario escribe un mensaje en el monitor serial de un nodo emisor.  
2. El mensaje se envía por LoRa al nodo central, incluyendo un identificador.  
3. El nodo central recibe, imprime el mensaje y responde con un acuse ("ACK").  
4. El nodo emisor muestra el acuse en el monitor serial.

# Código del Nodo Emisor A

Este código permite enviar mensajes desde el monitor serial y recibir un acuse de recibido.

// Nodo A  
// Identificador: "A"  
  
#include <SPI.h>  
#include <LoRa.h>  
  
// Pines de conexión  
#define SS 5  
#define RST 14  
#define DIO0 2  
#define NODE\_ID "A"  
  
// Mensaje ingresado por terminal  
String inputMessage = "";  
  
void setup() {  
 Serial.begin(9600);  
 while (!Serial);  
  
 // Configura pines y frecuencia  
 LoRa.setPins(SS, RST, DIO0);  
 if (!LoRa.begin(433E6)) {  
 Serial.println("Fallo al iniciar LoRa");  
 while (1);  
 }  
 Serial.println("Nodo A listo. Escribe un mensaje y presiona ENTER:");  
}  
  
void loop() {  
 // Leer entrada del usuario desde el terminal  
 if (Serial.available()) {  
 inputMessage = Serial.readStringUntil('\n');  
 String message = String(NODE\_ID) + ":" + inputMessage;  
  
 // Enviar el mensaje  
 LoRa.beginPacket();  
 LoRa.print(message);  
 LoRa.endPacket();  
 Serial.println("Mensaje enviado: " + message);  
 }  
  
 // Esperar un acuse de recibido del nodo central  
 int packetSize = LoRa.parsePacket();  
 if (packetSize) {  
 String ack = "";  
 while (LoRa.available()) {  
 ack += (char)LoRa.read();  
 }  
 Serial.println("Acuse recibido: " + ack);  
 }  
}

# Código del Nodo Emisor B

Este código permite enviar mensajes desde el monitor serial y recibir un acuse de recibido.

// Nodo B  
// Identificador: "B"  
  
#include <SPI.h>  
#include <LoRa.h>  
  
// Pines de conexión  
#define SS 5  
#define RST 14  
#define DIO0 2  
#define NODE\_ID "B"  
  
// Mensaje ingresado por terminal  
String inputMessage = "";  
  
void setup() {  
 Serial.begin(9600);  
 while (!Serial);  
  
 // Configura pines y frecuencia  
 LoRa.setPins(SS, RST, DIO0);  
 if (!LoRa.begin(433E6)) {  
 Serial.println("Fallo al iniciar LoRa");  
 while (1);  
 }  
 Serial.println("Nodo B listo. Escribe un mensaje y presiona ENTER:");  
}  
  
void loop() {  
 // Leer entrada del usuario desde el terminal  
 if (Serial.available()) {  
 inputMessage = Serial.readStringUntil('\n');  
 String message = String(NODE\_ID) + ":" + inputMessage;  
  
 // Enviar el mensaje  
 LoRa.beginPacket();  
 LoRa.print(message);  
 LoRa.endPacket();  
 Serial.println("Mensaje enviado: " + message);  
 }  
  
 // Esperar un acuse de recibido del nodo central  
 int packetSize = LoRa.parsePacket();  
 if (packetSize) {  
 String ack = "";  
 while (LoRa.available()) {  
 ack += (char)LoRa.read();  
 }  
 Serial.println("Acuse recibido: " + ack);  
 }  
}

# Código del Nodo Central (Receptor)

Este código escucha mensajes desde los nodos emisores y envía un acuse de recibido.

// Nodo central receptor (Gateway)  
  
#include <SPI.h>  
#include <LoRa.h>  
  
// Pines de conexión  
#define SS 5  
#define RST 14  
#define DIO0 2  
  
void setup() {  
 Serial.begin(9600);  
 while (!Serial);  
  
 // Inicializa LoRa  
 LoRa.setPins(SS, RST, DIO0);  
 if (!LoRa.begin(433E6)) {  
 Serial.println("Fallo al iniciar LoRa");  
 while (1);  
 }  
 Serial.println("Receptor central iniciado");  
}  
  
void loop() {  
 // Esperar mensaje  
 int packetSize = LoRa.parsePacket();  
 if (packetSize) {  
 String incoming = "";  
 while (LoRa.available()) {  
 incoming += (char)LoRa.read();  
 }  
  
 Serial.println("Mensaje recibido: " + incoming);  
  
 // Extraer ID del nodo (carácter inicial antes del ':')  
 String nodeId = incoming.substring(0, 1);  
 String ack = "ACK:" + nodeId;  
  
 // Esperar un momento y enviar acuse  
 delay(100);  
 LoRa.beginPacket();  
 LoRa.print(ack);  
 LoRa.endPacket();  
 Serial.println("Acuse enviado a Nodo " + nodeId);  
 }  
}