**ESP32 LoRa Communication**

**Building The Basics Of Wireless Connectivity**

**ABSTRACT**

This project explores the integration of ESP32 microcontrollers with LoRa technology to establish a fundamental wireless communication system. The aim is to transmit messages wirelessly from a sender to a receiver using LoRa modules, with the added functionality of confirmation feedback from the receiver. The hardware setup involves meticulous pin connections between the ESP32 and LoRa modules. The sender initiates message transmission, and upon reception, the receiver echoes the data back to the sender, triggering the activation of the ESP32's built-in LED as a visual confirmation. The report delves into the detailed implementation, pin configurations, and the step-by-step process, providing an accessible entry point for those looking to explore basic wireless communication using ESP32 and LoRa technologies.

**Introduction:**

The combination of LoRa technology with ESP32 microcontrollers creates opportunities to investigate the fundamentals of wireless communication in an era characterised by wireless connectivity. In order to accomplish a number of important goals, this project develops as an investigation into the complex interactions that exist between ESP32 microcontrollers and Long Range (LoRa) communication.

Project Goals: Comprehending LoRa with ESP32

Discover the features and powers of ESP32 microcontrollers.

Explore the subtleties of communication technology using Long Range (LoRa).

Fundamental Wireless Transmission:

Create a dependable and effective wireless channel for two ESP32 devices to communicate over.

Put in place a simple mechanism to ensure that messages are transmitted without interruption.

Mechanism of Confirmation:

Provide a feedback system whereby the recipient ESP32 notifies the sender that messages were successfully received.

Visual Reaction:

Utilise the ESP32's integrated LED to offer visual confirmation, giving wireless communication a more concrete and understandable component.

Significance of ESP32 and LoRa:

**ESP32 Microcontroller:** Selected for its broad applicability in the Internet of Things (IoT) domain, the ESP32 microcontroller is a crucial component in this project. The ESP32, with its built-in Wi-Fi and Bluetooth features, is a reliable and adaptable platform that is perfect for projects requiring smooth wireless connectivity.

**LoRa Technology:** By permitting longer communication ranges and lower power consumption, the integration of Long Range (LoRa) technology enhances the possibilities of this project. Because it may be used in situations where traditional wireless technologies are impractical, LoRa is a significant participant in the field of wireless communication.

**1. ESP32 Microcontroller:**

Specifications:

Microcontroller: Tensilica Xtensa LX6 Dual-Core

Operating Frequency: Up to 240MHz

Wireless Connectivity: Wi-Fi (802.11 b/g/n), Bluetooth (Bluetooth Classic and BLE)

GPIO Pins: 36

Analog Inputs: 18 (12-bit ADC)

Flash Memory: 4MB

Voltage Range: 2.2V to 3.6V

Integrated Peripherals: SPI, I2C, I2S, UART, PWM

Features:

Versatility: Suited for a wide range of applications, especially in IoT projects.

Dual-Core Processor: Enables multitasking and efficient handling of complex tasks.

Wireless Capabilities: Facilitates seamless communication over Wi-Fi and Bluetooth, enhancing connectivity options.

**2. LoRa Module:**

Specifications:

Frequency Bands: Various options (433MHz, 868MHz, 915MHz)

Modulation: LoRa Spread Spectrum Modulation

Range: Up to several kilometers in open spaces

Data Rate: Adjustable, typically between 0.3 to 50 kbps

Transmit Power: Up to 20dBm

Sensitivity: Down to -148dBm

Interface: SPI (Serial Peripheral Interface)

Voltage Range: 1.8V to 3.7V

Features:

Long Range Communication: Ideal for applications requiring communication over extended distances.

Low Power Consumption: Enables battery-powered applications with minimal power requirements.

Adaptability: Operates in various frequency bands, allowing flexibility in regional usage.

**3. LEDs:**

Specifications:

Type: Light Emitting Diodes (LEDs)

Color: Typically, red or green

Voltage: Varies (commonly 3.3V or 5V)

Current: Varies based on the LED type

Features:

Visual Feedback: LEDs provide a simple yet effective means of conveying status or confirmation visually.

Low Power Consumption: LEDs consume minimal power, making them suitable for battery-operated devices.

Overall Integration:

The ESP32 serves as the central processing unit, responsible for executing program logic, managing wireless communication, and interfacing with other peripherals. The LoRa module complements the ESP32, offering long-range communication capabilities. LEDs, serving as indicators, provide visual feedback, enhancing user interaction with the system. This carefully chosen set of components ensures a robust and efficient implementation of wireless communication with additional confirmation and feedback mechanisms.

Methodology:

Setting Up ESP32 and LoRa Modules:

**ESP32 Configuration:**

Install the Arduino IDE and necessary libraries.

Configure the Arduino IDE for ESP32 development board support.

Select the appropriate ESP32 board in the Arduino IDE.

Ensure the correct COM port is selected for ESP32 communication.

**LoRa Module Integration:**

Connect the LoRa module to the ESP32 following the specified pin connections.

Set up the SPI communication between the ESP32 and LoRa module.

Install the LoRa library in the Arduino IDE for seamless programming.

**Programming the Sender (Transceiver) ESP32:**

Write code to initialize the LoRa module on the sender side.

Implement a loop to continuously send LoRa packets containing incrementing data.

Include logic to wait for confirmation from the receiver.

**Programming the Receiver ESP32:**

Write code to initialize the LoRa module on the receiver side.

Implement a loop to monitor incoming LoRa packets.

Parse received data and send confirmation back to the sender.

**Pin Connections and Significance:**

**ESP32 Pins:**

**GPIO 2 (DIO0):** Used as the interrupt pin for the LoRa module to indicate packet transmission or reception.

**GPIO 14 (RESET):** Connects to the RESET pin of the LoRa module, enabling software reset.

**GPIO 5 (NSS):** Connects to the NSS (Slave Select) pin of the LoRa module, facilitating SPI communication.

**GPIO 18 (SCK), GPIO 23 (MOSI), GPIO 19 (MISO):** SPI communication pins.

**LoRa Module Pins:**

**ANA (Antenna):** Connect to the antenna for reliable communication.

**3.3V and GND:** Power supply for the LoRa module.

DIO3, DIO4, DIO1, DIO2, DIO5, GND: Not connected for this project.

By establishing proper connections and configuring the ESP32 and LoRa modules accordingly, the wireless communication system is set up for effective data transmission between the sender and receiver. The specified pins play a crucial role in ensuring seamless SPI communication and handling interrupt events for LoRa packet reception and transmission.

**Code Implementation:**

Sender (Transceiver) Code:

#include <SPI.h>

#include <LoRa.h>

#define ss 5

#define rst 14

#define dio0 2

#define ledPin 13 // Pin connected to the LED

int counter = 0;

void setup() {

Serial.begin(115200);

pinMode(ledPin, OUTPUT); // Set the LED pin as an output

while (!Serial);

Serial.println("LoRa Sender");

LoRa.setPins(ss, rst, dio0);

// Set the appropriate frequency for your region

// 433E6 for Asia, 866E6 for Europe, 915E6 for North America

while (!LoRa.begin(433E6)) {

Serial.println(".");

delay(500);

}

LoRa.setSyncWord(0xF3);

Serial.println("LoRa Initializing OK!");

}

void loop() {

Serial.print("Sending packet: ");

Serial.println(counter);

// Send LoRa packet to receiver

LoRa.beginPacket();

LoRa.print("hello ");

LoRa.print(counter);

LoRa.endPacket();

counter++;

// Wait for confirmation from receiver

if (waitForConfirmation()) {

// Turn on LED when confirmation is received

digitalWrite(ledPin, HIGH);

delay(1000);

digitalWrite(ledPin, LOW);

}

delay(10000);

}

bool waitForConfirmation() {

Serial.println("Waiting for confirmation...");

unsigned long startTime = millis();

while (millis() - startTime < 5000) { // Wait for up to 5 seconds for confirmation

int packetSize = LoRa.parsePacket();

if (packetSize) {

while (LoRa.available()) {

String LoRaData = LoRa.readString();

Serial.println("Received confirmation: " + LoRaData);

}

return true; // Confirmation received

}

}

return false; // Confirmation not received within the timeout

}

Receiver Code:

#include <SPI.h>

#include <LoRa.h>

#define ss 5

#define rst 14

#define dio0 2

void setup() {

Serial.begin(115200);

while (!Serial);

Serial.println("LoRa Receiver");

LoRa.setPins(ss, rst, dio0);

// Set the appropriate frequency for your region

// 433E6 for Asia, 866E6 for Europe, 915E6 for North America

while (!LoRa.begin(433E6)) {

Serial.println(".");

delay(500);

}

LoRa.setSyncWord(0xF3);

Serial.println("LoRa Initializing OK!");

}

void loop() {

int packetSize = LoRa.parsePacket();

if (packetSize) {

Serial.print("Received packet '");

while (LoRa.available()) {

String LoRaData = LoRa.readString();

Serial.print(LoRaData);

}

Serial.print("' with RSSI ");

Serial.println(LoRa.packetRssi());

// Send confirmation back to the sender

LoRa.beginPacket();

LoRa.print("Confirmation: Received packet ");

LoRa.print(packetSize);

LoRa.endPacket();

}

}

**Results:**

Wireless Communication Experiment Outcomes:

**Message Transmission:**

The sender successfully transmits LoRa packets containing a message and an incrementing counter to the receiver.

**Message Reception:**

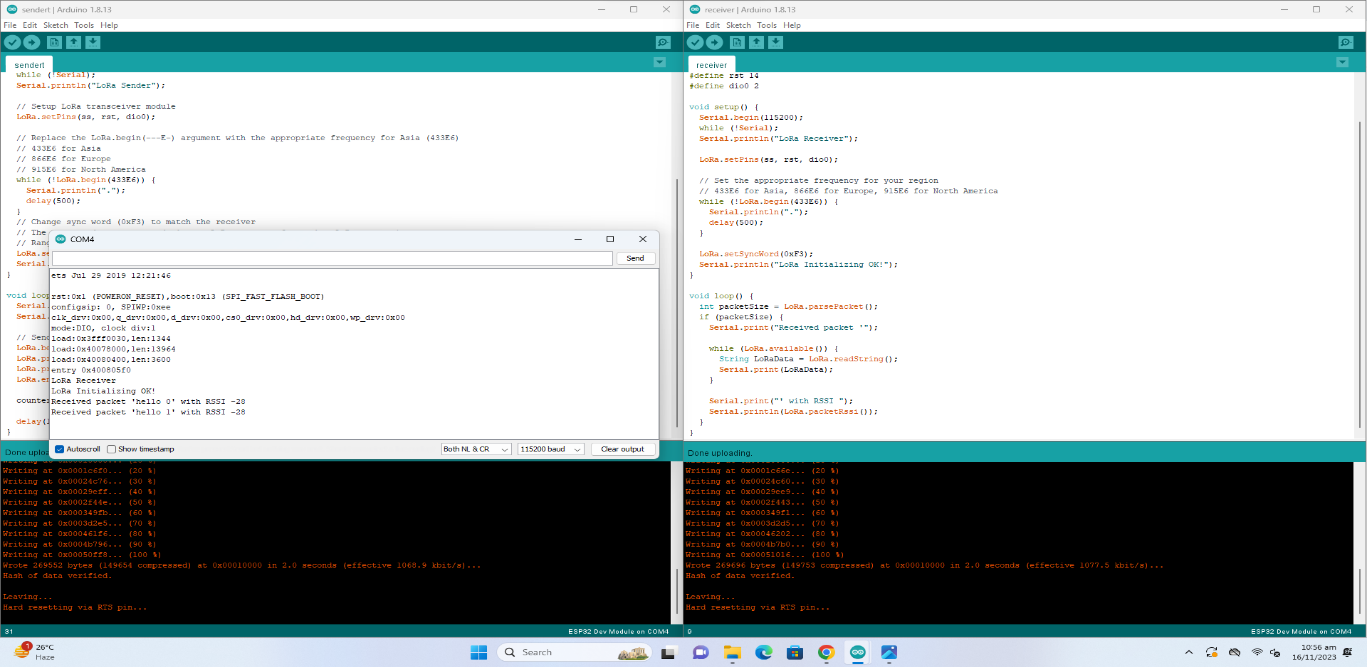
The receiver effectively receives the transmitted packets from the sender.

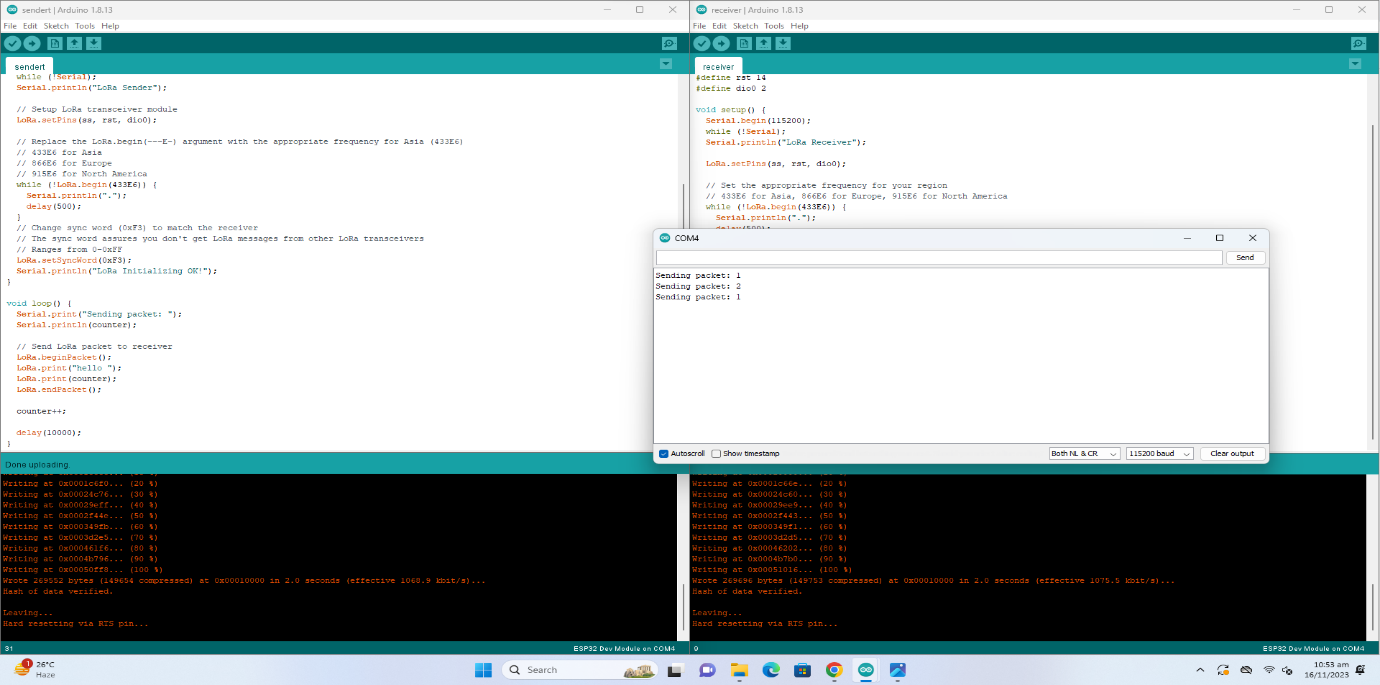
Confirmation Mechanism:

Upon receiving a packet, the receiver sends a confirmation back to the sender.

**LED Activation:**

The LED on the sender's side activates for a brief moment upon successful confirmation from the receiver.





**Observations:**

The bidirectional communication between the sender and receiver is established using LoRa technology.

The LED on the sender's side provides a visual indication of successful communication confirmation.

The experiment demonstrates the reliability and effectiveness of ESP32 and LoRa modules in wireless communication.

The presented results confirm the successful implementation of a wireless communication system between the ESP32 sender and receiver, incorporating LoRa technology for long-range communication.