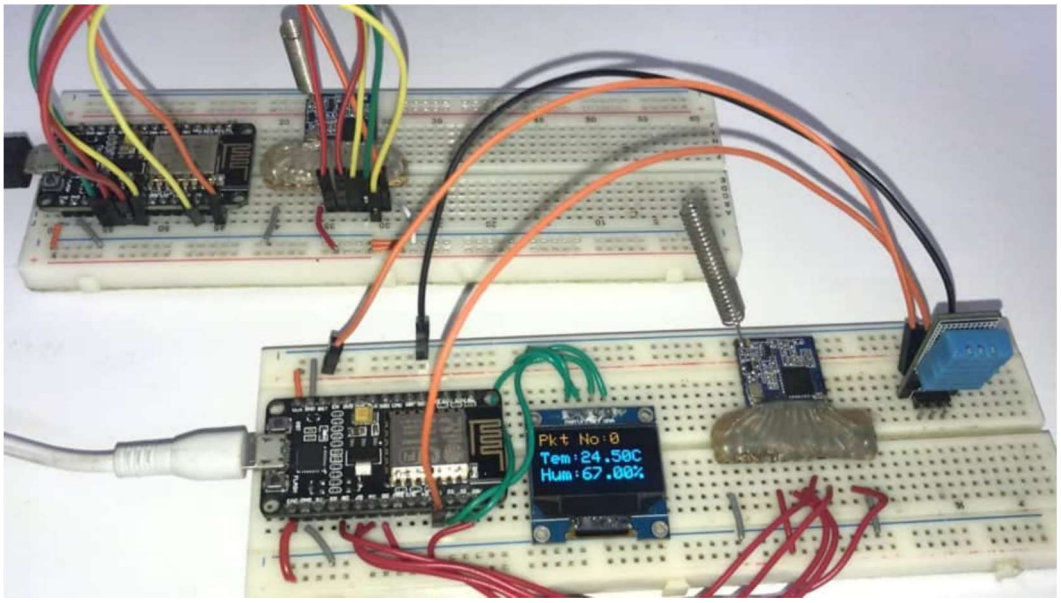
ESP8266 Projects IoT Projects LoRa Projects

ESP8266 & LoRa SX1278 Transmitter Receiver with DHT11

Admin Last Updated: August 21, 2022 15 26,464 4 minutes read



In this tutorial, we will make **Lora Transmitter & Receiver using Lora Module SX1278 & NodeMCU ESP8266 Wifi Module**. The communication type is a point to point and data will be transferred wirelessly from one end (transmitter) to another end (receiver).

Table of Contents [hide]

Overview

In the last couple of years, there is a number of communication technologies available for interaction between **IoT devices**. The most popular ones are the **Wi-Fi** Technology and **Bluetooth** Module. But they have few limitations like limited range, limited access points & high power consumption. So LoRa technology is introduced by Semtech to fix all these issue. Using a single battery the device operates over a year.

Display. We will take two examples for learning LoRa SX1278 & **ESP8266 Communication**. First, we will transmit a simple packet and check whether is the data is received at the receiver end or not. Then we will add **DHT11 Humidity & Temperature Sensor** to transmitter Circuit and send the humidity temperature value wirelessly to the receiver end.

Before getting started, you can visit the following posts as well:

1. *Interfacing SX1278 (Ra-02) LORA Module with Arduino*: Check Here
2. *Sending Sensor Data Wirelessly with LoRa SX1278 & Arduino*: Check Here
3. *ESP32 & LoRa SX1278/76 Transmitter Receiver with OLED*: Check Here
4. *ESP32 LoRa Sensor Data Monitoring on Web Server*: Check Here

Bill of Materials

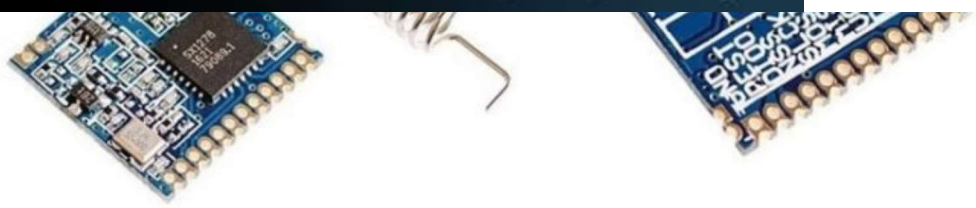
Following are the components required for making this project. All the components can easily be purchased from Amazon. The

1	NodeMCU	ESP8266 12E Board	2	https://amzn.t
2	LoRa Module	SX1278/76 Lora Module	2	https://amzn.t
3	DHT11 Sensor	Humidity & Temperature Sensor	1	https://amzn.t
4	OLED Display	0.96" I2C OLED Display	1	https://amzn.t
5	Connecting Wires	Jumper Wires	20	https://amzn.t
6	Breadboard	-	1	https://amzn.t

Semtech SX1278 LoRa Module

SX1278 Module

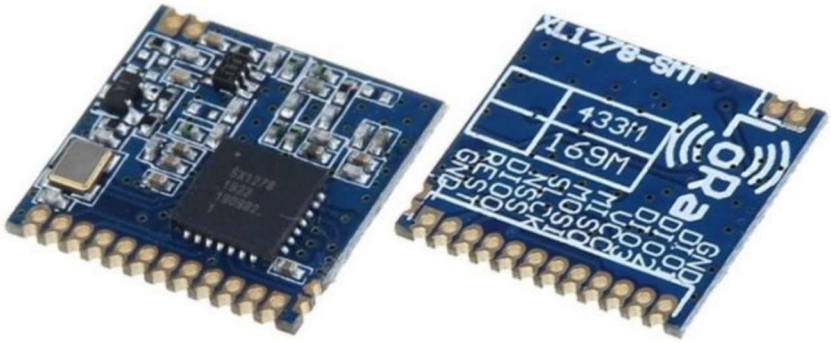
The **SX1276/77/78/79** transceivers feature the LoRa® long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption.



SX1278 can achieve a sensitivity of over **-148dBm** using a low-cost crystal. The high sensitivity combined with the integrated **+20dBm power amplifier** yields industry leading link budget making it optimal for any application requiring range or robustness. Lora SX1278 also provides significant advantages in both blocking and selectivity over conventional **modulation techniques**, solving the traditional design compromise between range, interference immunity and energy consumption. Learn more about it at: Semtech SX1278 Datasheet.

Semtech SX1278 Pinout

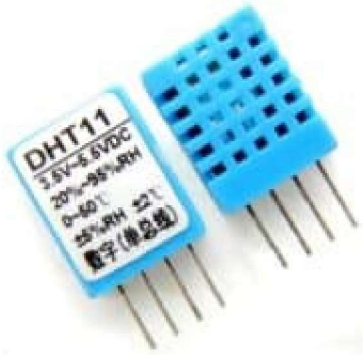
There are different versions and types of **SX1278 breakout board** available in market. But basically all of them has same pinout as **LoRa SX1278** is an **SPI module**. I am using this board as shown in photos below.



1	GND	Ground (0 V)
2	D101	Digital I/O
3	D102	Digital I/O
4	D103	Digital I/O
5	VCC	Power (3.6 V Maximum)
6	MISO	SPI Data Output
7	MOSI	SPI Data Input
8	SLCK	SPI Clock
9	NSS	SPI Chip Select
10	D100	Digital I/O
11	RESET	Reset
12	GND	Ground (0 V)

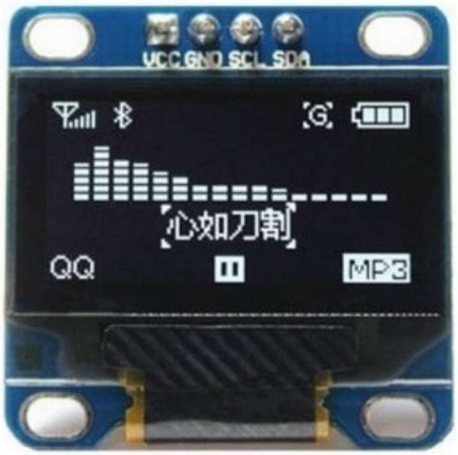
DHT11 Humidity & Temperature Sensor

The **DHT11** is a basic, ultra low-cost digital **temperature and humidity sensor**. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a **digital signal** on the **data pin** (no analog input pins needed).



Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using the **library, sensor** readings can be up to 2 seconds old.

having a resolution of **128x64**. The package includes display board, display,4 pin male header pre-soldered to board.



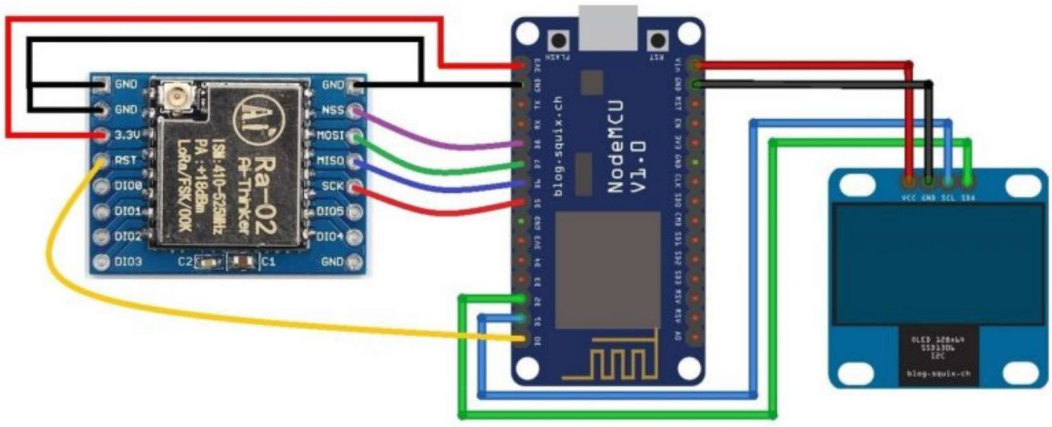
OLED (Organic Light-Emitting Diode) is a self light-emitting technology composed of a thin, multi-layered organic film placed between an anode and cathode. In contrast to LCD technology, OLED does not require a backlight. OLED possesses high application potential for virtually all types of displays and is regarded as the ultimate technology for the next generation of flat-panel displays.

Simple LoRa SX1278 & ESP8266 Transmitter Receiver

Let us understand from this basic example. We will first simply **interface Lora SX1278 Module with NodeMCU ESP8266 Board**. The schematic connections & code is given below.

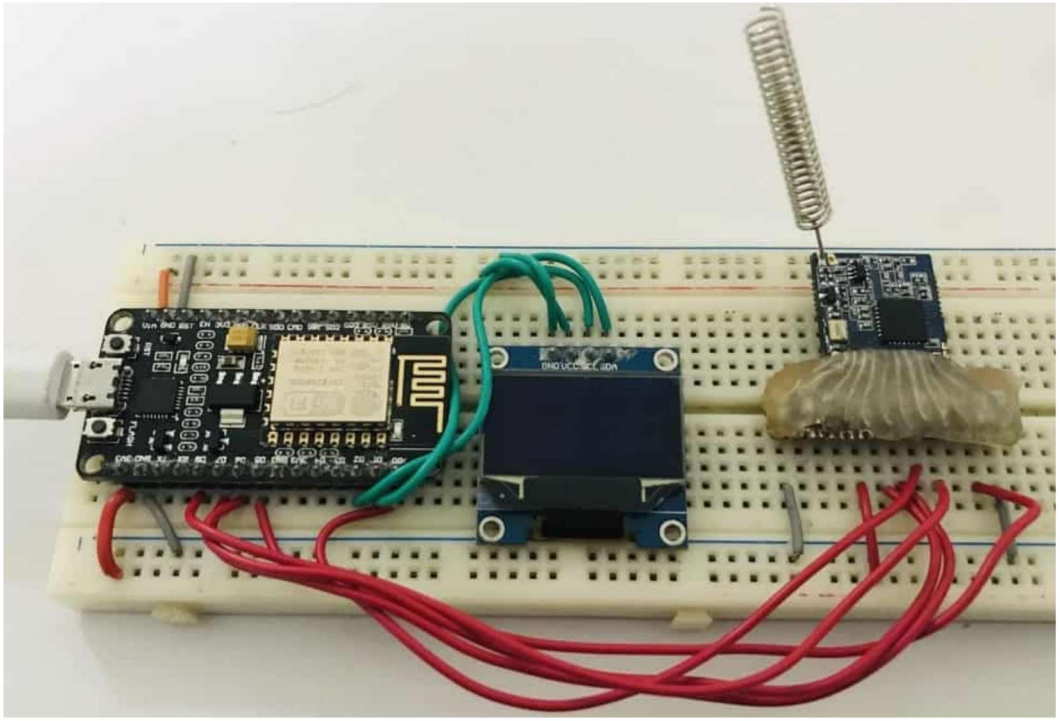
Trasnsmitter Circuit

Here is a simple **Transmitter Circuit**. Assemble the circuit as shown in the figure below.



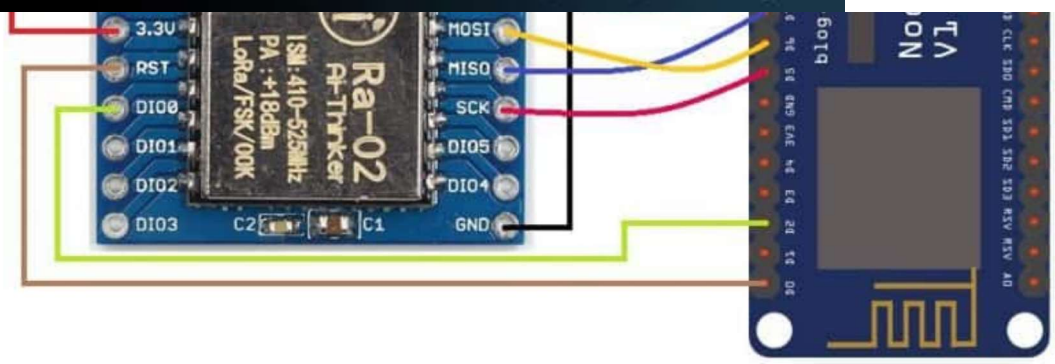
The connection is fairly simple. Connect the OLED SDA SCL pins to D2, D1 of Nodemcu respectively. Simlarly connect the Lora SX1278 & NodeMCU ESP8266 as follows.

D5	SCK
D0	RST



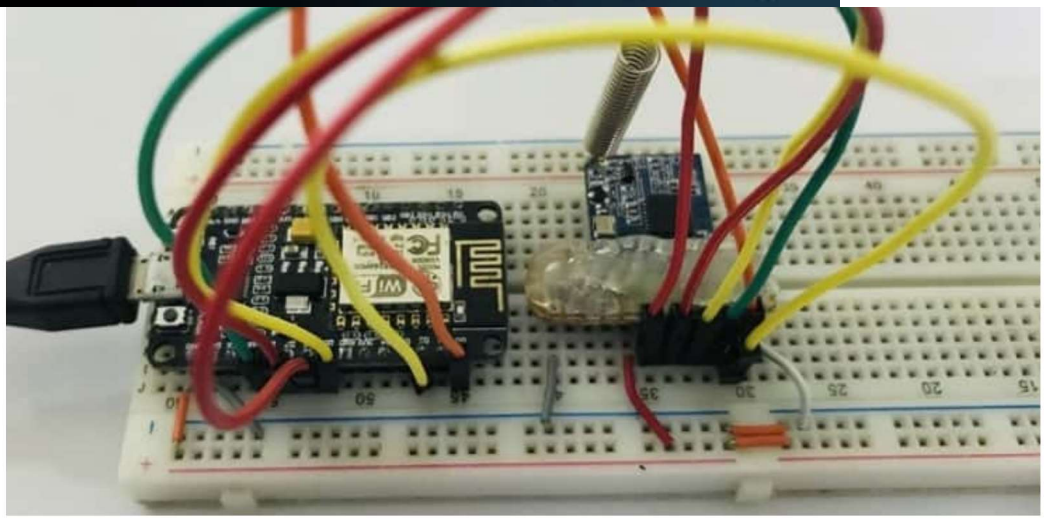
Receiver Circuit

Here is a simple **Receiver Circuit**. Assemble the circuit as shown in the figure below.



The connection is fairly simple. The Lora SX1278 & NodeMCU ESP8266 connection is as follows.

NodeMCU Pins	SX1278 Pins
GND	GND
3.3V	VCC
D8	NSS
D7	MOSI
D6	MISO
D5	SCK
D0	RST
D2	DIO0



Trasnsmitter Code

Before using this code add the following library to Arduino IDE:

1. Lora Library: Download
2. SSD1306 OLED Library: Download
3. Adafruit GFX Library: Download

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, S
#define OLED_RESET      -1  // Reset pin # (or -1 if sharing A
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, O


#define ss 15
#define rst 16
#define dio0 2
int counter = 0;


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

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  while (!Serial);
  Serial.println("LoRa Sender");
  LoRa.setPins(ss, rst, dio0);
    if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    delay(100);
    while (1);
  }

  display.display();
  delay(2);
  display.clearDisplay();
}
```

```
  display.setTextSize(2);             // Normal 1:1 pixel scal
  display.setTextColor(WHITE);        // Draw white text
  display.setCursor(0,0);             // Start at top-left cor
  display.print(F("Pkt No:"));
  display.print(counter);

  // send packet
  LoRa.beginPacket();
  LoRa.print(F("Pkt No:"));
  LoRa.println(counter);

  LoRa.endPacket();

  counter++;

  delay(3000);
}
```

Receiver Code

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

#define ss 15
#define rst 16
#define dio0 2

void setup() {
  Serial.begin(115200);
  while (!Serial);

  Serial.println("LoRa Receiver Callback");

  LoRa.setPins(ss, rst, dio0);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
```

```
  // put the radio into receive mode
  LoRa.receive();
}

void loop() {
  // do nothing
}

void onReceive(int packetSize) {
  // received a packet
  Serial.print("Received packet '");

  // read packet
  for (int i = 0; i < packetSize; i++) {
    Serial.print((char)LoRa.read());
  }

  // print RSSI of packet
  Serial.print("' with RSSI ");
  Serial.println(LoRa.packetRssi());
}
```

Results & Observations

Once the code is uploaded, the transmitter section will start sending the data and receiver section will start receiving the data.

Similarly you can open the serial monitor and observe the transmitted and received packet with numbers.

Sending DHT11 Sensor Humidity Temperature Data using SX1278 & ESP8266

Here is a **DHT11 Loara SX1278 ESP8266 Transmitter Circuit**. Assemble the circuit as shown in the figure below.

The connection is fairly simple. Connect the OLED SDA SCL pins to D2, D1 of Nodemcu respectively. Connect the DHT11 output pin to D3 od Nodemcu as shown in the photo below. Similarly connect the **Lora SX1278 & NodeMCU ESP8266** same as above.

The connection for the **receiver circuit** is the same as above. There is no need to do any change.

Trasnsmitter Code

```
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, S
#define OLED_RESET     LED_BUILTIN // Reset pin # (or -1 if sh
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, O

#define DHTPIN 0          //pin where the dht11 is connected
#define ss 15
#define rst 16
#define dio0 2
int counter = 0;

DHT dht(DHTPIN, DHT11);
void setup()
{
  Serial.begin(115200);
  dht.begin();
```

```
  while (!Serial);
  Serial.println("LoRa Sender");
  LoRa.setPins(ss, rst, dio0);
    if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    delay(100);
    while (1);
  }

  display.display();
  delay(2);
  display.clearDisplay();
}

void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t))
  {
  Serial.println("Failed to read from DHT sensor!");
  return;
  }
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" degrees Celcius, Humidity: ");
  Serial.println(h);
  Serial.println();
  Serial.print("Sending packet: ");
  Serial.println(counter);

  display.clearDisplay();
  display.setTextSize(2);             // Normal 1:1 pixel scal
  display.setTextColor(WHITE);        // Draw white text
  display.setCursor(0,0);             // Start at top-left cor
  display.print(F("Pkt No:"));
```

```
  display.setCursor(0,40);
  display.print("Hum:");
  display.print(h);
  display.print("%");
  display.display();

  // send packet
  LoRa.beginPacket();
  LoRa.print(F("Pkt No:"));
  LoRa.println(counter);

  LoRa.print("Temp: ");
  LoRa.print(t);
  LoRa.println("°C");

  LoRa.print("Hum:  ");
  LoRa.print(h);
  LoRa.print("%");
  LoRa.println("");

  LoRa.endPacket();

  counter++;

  delay(3000);
}
```

Receiver Code

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

#define ss 15
#define rst 16
#define dio0 4
```

```
  LoRa.setPins(ss, rst, dio0);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }

  // register the receive callback
  LoRa.onReceive(onReceive);

  // put the radio into receive mode
  LoRa.receive();
}

void loop() {
  // do nothing
}

void onReceive(int packetSize) {
  // received a packet
  Serial.println("Received packet '");

  // read packet
  for (int i = 0; i < packetSize; i++) {
    Serial.print((char)LoRa.read());
  }
}
```

Results & Observations

Once the code is uploaded, the **transmitter section** will start sending the data and receiver section will start receiving the data. The **humidity** and **temperature value** can be observed in OLED Screen.

Similarly, you can open the serial monitor and observe the **transmitted and received humidity temperature data** with packet numbers.

Video Tutorial & Demonstrattion

LoRa SX1278/76 & ESP8266 Transmitter Receiver | Send DHT11 Sensor Data Wirelessly

Watch this video on YouTube.

You can check this post as well: ESP32 & LoRa SX1278/76 Transmitter Receiver with OLED

15 Comments

Also a bme280 i2c sensor for temperature, humidity and pressure would be nice. It can just added on the existing i2c bus for the display!

Looking forward for more lora & esp8266 posts!

See you around

Loading...

Reply

Mr. Alam

May 18, 2020 at 7:04 PM

Check some other post with esp32 & LoRa. I have added OLED at receiver as well.

Loading...

Reply

It always boot loops with the following in the log:

LoRa Receiver Callback
ISR not in IRAM!

User exception (panic/abort/assert)
Abort called

```
stack>>>
```

ctx: cont
sp: 3ffffee0 end: 3fffffc0 offset: 0000
3ffffee0: 00000748 feefeffe feefeffe feefeffe
3ffffef0: 000000fe 00000000 00000000 00000000
3fffff00: 00000000 00000000 00000000 00ff0000
3fffff10: 5ffffe00 5ffffe00 00000001 00000000
3fffff20: 00000001 00000002 3ffee354 402028ba
3fffff30: 40100502 1d8e975a 402018e5 402028cc
3fffff40: 00000081 00000081 3ffee354 40202de1
3fffff50: 00000000 00000000 3ffee350 40201533
3fffff60: 007a1200 00000001 0000000f 3ffee410
3fffff70: 3fffdad0 3ffee3a8 3ffee354 40202e90
3fffff80: 3fffdad0 3ffee3a8 3ffee354 40201810
3fffff90: 3fffdad0 3ffee3a8 3ffee354 4020108a
3fffffa0: feefeffe 00000000 3ffee3d0 4020251c
3fffffb0: feefeffe feefeffe 3ffe84e0 40100e21

Loading...

Reply

LoRa Receiver Callback
ISR not in IRAM!

Can you help Mr. Alam?

Loading...

Reply

Transmitt0r

June 2, 2020 at 1:35 AM

Hello, transmitter code works but receiver gives error like "LoRa Receiver Callback; ISR not in IRAM!; User exception; (panic/abort/assert); Abort called"

Can you help with that? What is the problem?

Loading...

Reply

Mr. Alam

June 2, 2020 at 9:51 AM

Hi, in the receiver circuit, please reverse the MISO MOSI Pins. The connection is mistake. I will change the diagram soon.

Loading...

Reply

from the serial:
"LoRa Receiver Callback; Starting LoRa failed!"

Any more ideas?

Loading...

Reply

Widya Febriandaru

September 22, 2020 at 1:52 PM

Same with me, any progress bruh? please reply if u fixed that 🙂

Loading...

Reply

Xavier RP

May 2, 2021 at 1:37 PM

Thanks for the tutorial sir and useful guidance...

Loading...

Reply

Loading...

Reply

Alex Newton

May 30, 2021 at 8:04 PM

Yes it should be 434. Thanks

Loading...

Reply

Siddharth

May 30, 2021 at 7:44 PM

Hi… Can you help me on how to configure LORA and Arduino Nano to Soil Moisture Reading over a 1 Sq.km area from different areas…

Loading...

Reply

Helmi

May 30, 2021 at 10:17 PM

Alex, so please change it in Your code, as some authorities don't like if You transmit on a not allowed frequency.

Loading...

Reply

It must say: #define dio0 4 // It's because D2 is the GPIO4 in the NodeMCU

Loading...

Reply

Laszlo

August 19, 2022 at 9:19 AM

I'm a beginner in this sector and I'm attempting to connect a Raspberry Pi Pico to an RFM95 and use the LoRa.h library to establish peer to peer communication, but for some reason it keeps reporting "Starting LoRa failed." Is this because I would have to manually configure the SPI pins or because the Raspberry Pi Pico is not supported by the LoRa.h library? I would appreciate any assistance. Thanks

Loading...

Reply