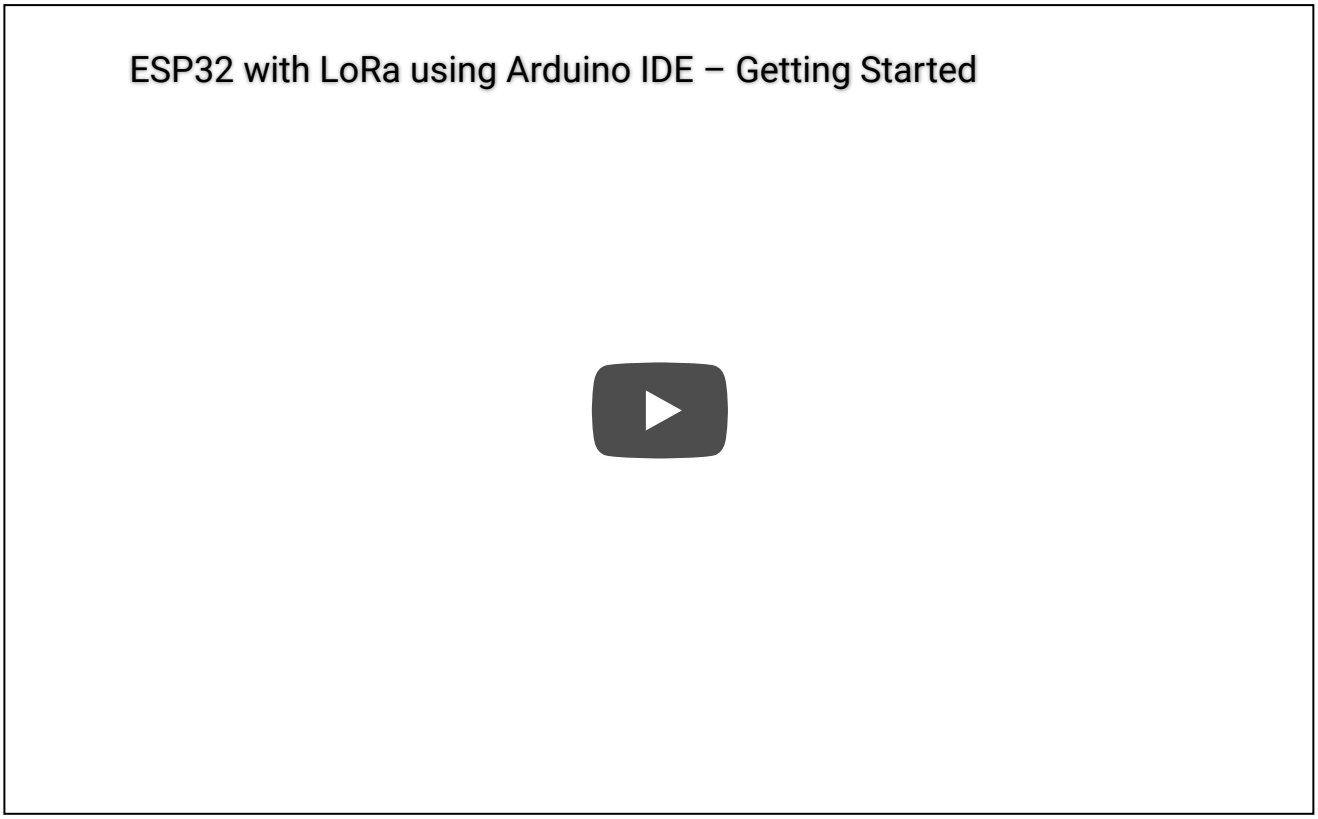
ESP32 with LoRa using Arduino IDE – Getting Started

In this tutorial we'll explore the basic principles of LoRa, and how it can be used with the ESP32 for IoT projects using the Arduino IDE. To get you started, we'll also show you how to create a simple LoRa Sender and LoRa Receiver with the RFM95 transceiver module.

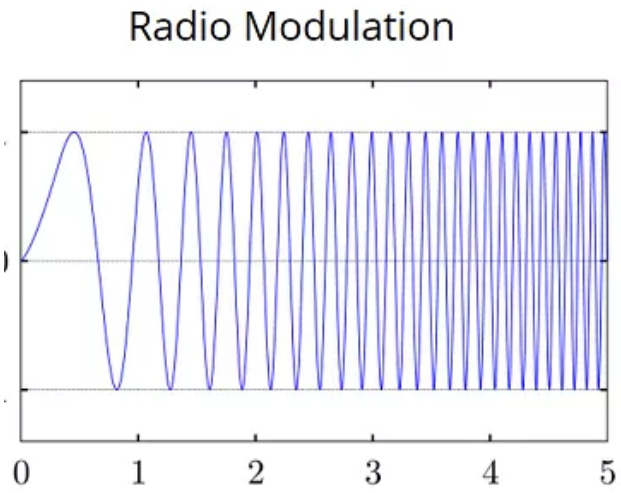
Introducing LoRa

For a quick introduction to LoRa, you can watch the video below, or you can scroll down for a written explanation.



What is LoRa?

LoRa is a wireless data communication technology that uses a radio modulation technique that can be generated by Semtech LoRa transceiver chips.



This modulation technique allows long range communication of small amounts of data (which means a low bandwidth), high immunity to interference, while minimizing power consumption. So, it allows long distance communication with low power requirements.

LoRa Frequencies

LoRa uses unlicensed frequencies that are available worldwide. These are the most widely used frequencies:

- 868 MHz for Europe
- 915 MHz for North America
- 433 MHz band for Asia

Because these bands are unlicensed, anyone can freely use them without paying or having to get a license. Check the frequencies used in your country.

LoRa long range and low power features, makes it perfect for battery-operated sensors and low-power applications in:

- Internet of Things (IoT)
- Smart home
- Machine-to-machine communication
- And much more...

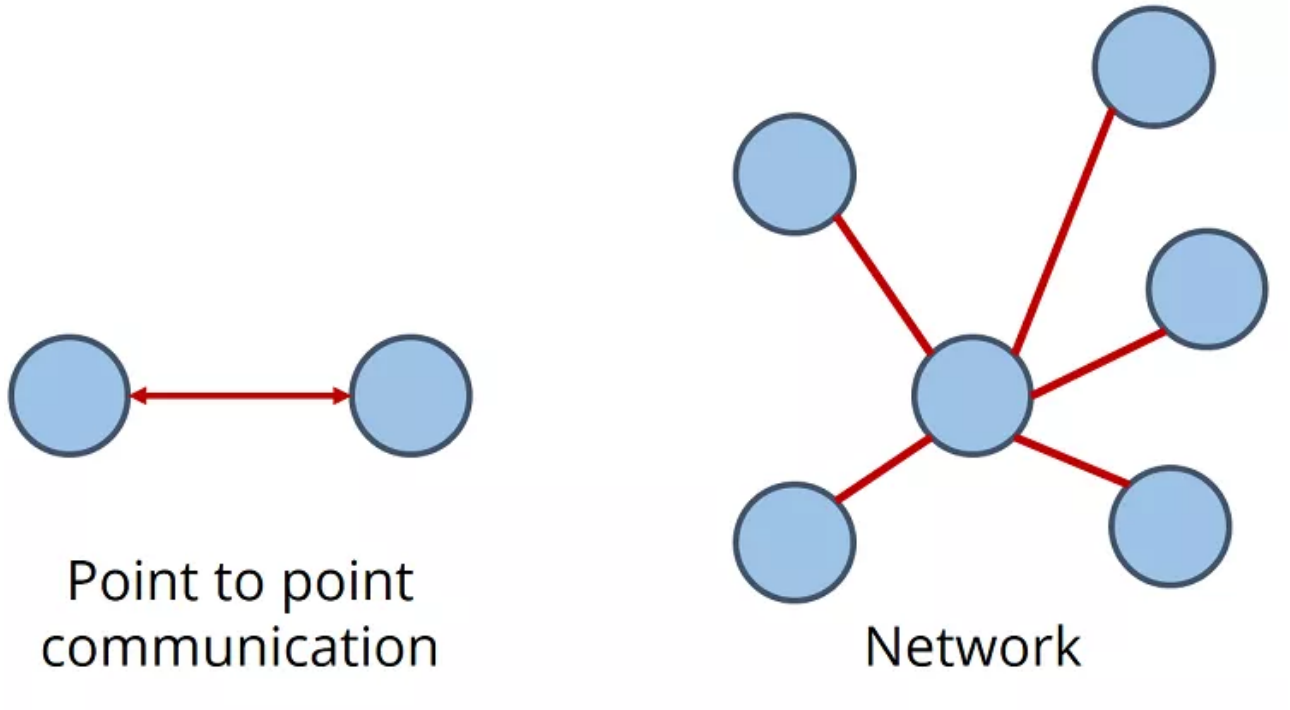
So, LoRa is a good choice for sensor nodes running on a coil cell or solar powered, that transmit small amounts of data.

Keep in mind that LoRa is not suitable for projects that:

- Require high data-rate transmission;
- Need very frequent transmissions;
- Or are in highly populated networks.

LoRa Topologies

You can use LoRa in:



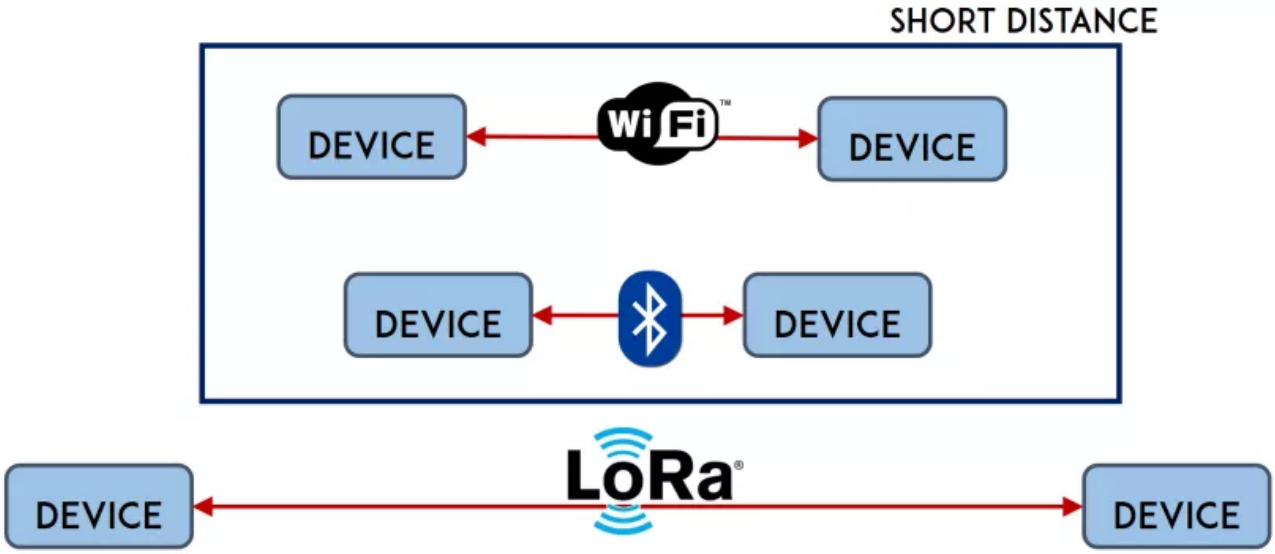
- Point to point communication
- Or build a LoRa network (using LoRaWAN for example)

Point to Point Communication

In point to point communication, two LoRa enabled devices talk with each other using RF signals.

For example, this is useful to exchange data between two ESP32 boards equipped with LoRa transceiver chips that are relatively far from each other or in environments without Wi-Fi coverage.

Unlike Wi-Fi or Bluetooth that only support short distance communication, two LoRa devices with a proper antenna can exchange data over a long distance.



You can easily configure your ESP32 with a LoRa chip to transmit and receive data reliably at more than 200 meters distance (you can get better results depending on your enviroment and LoRa settings). There are also other LoRa solutions that easily have a range of more than 30Km.

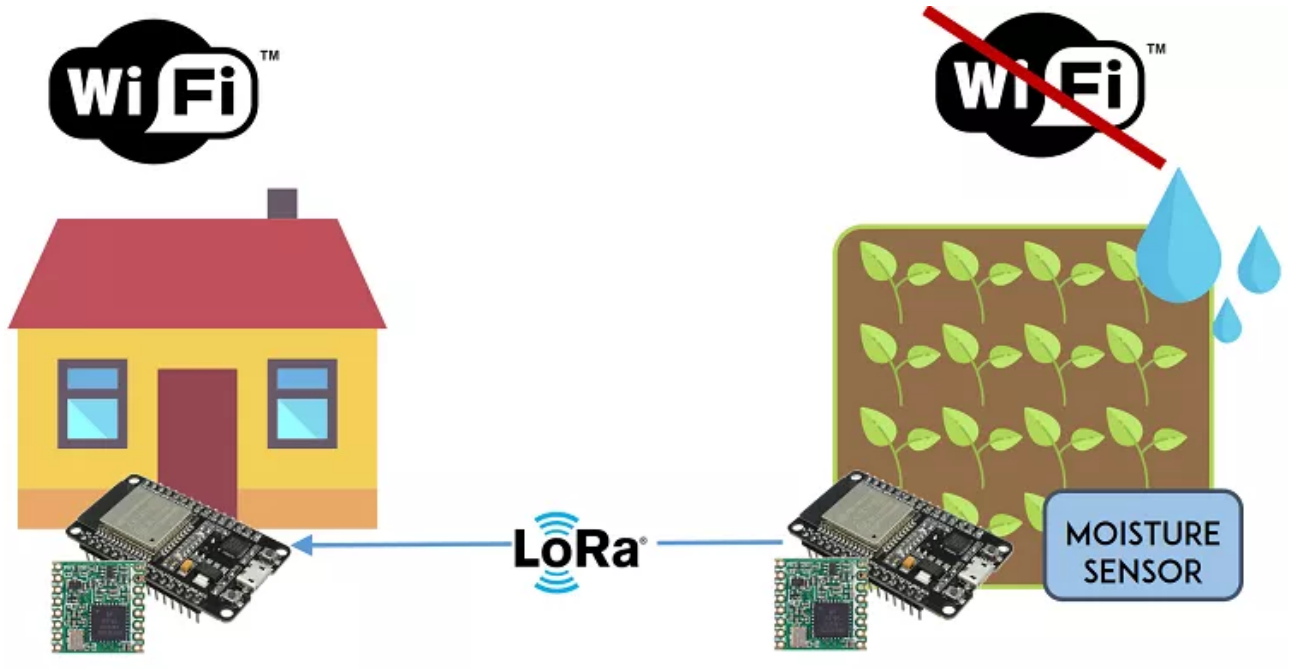
LoRaWAN

You can also build a LoRa network using LoRaWAN.

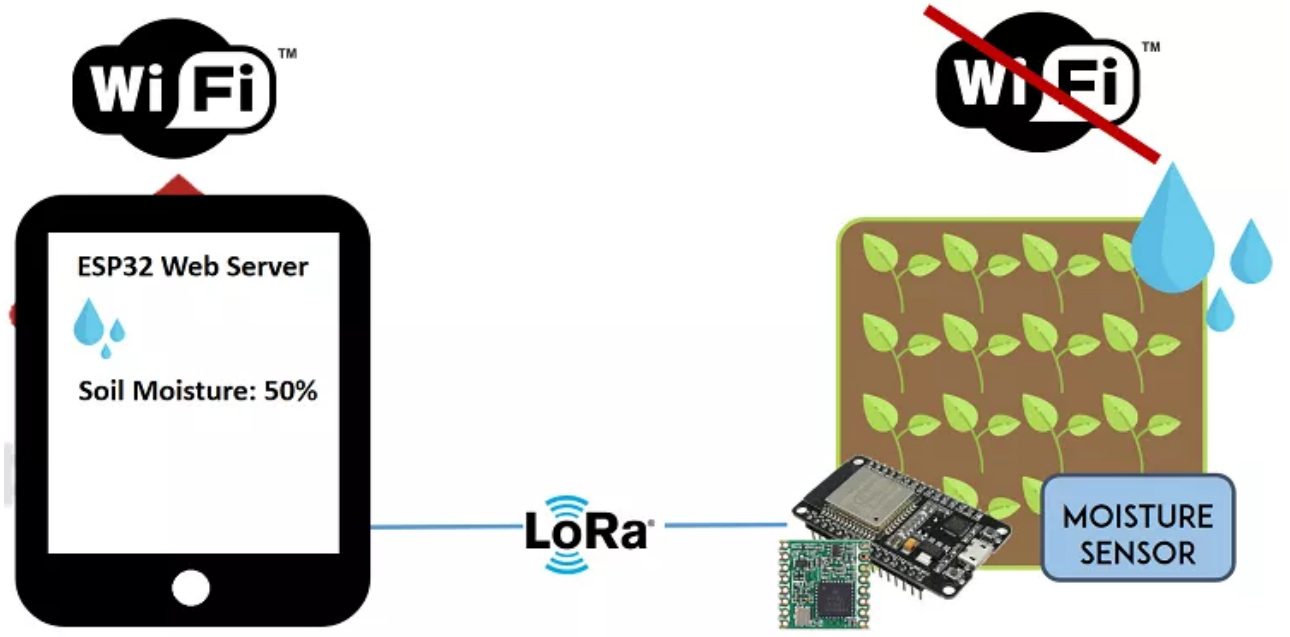
The LoRaWAN protocol is a Low Power Wide Area Network (LPWAN) specification derived from LoRa technology standardized by the LoRa Alliance. We won't explore LoRaWAN in this tutorial, but for more information you can check the LoRa Alliance and The Things Network websites.

How can LoRa be useful in your home automation projects?

Imagine that you want to measure the moisture in your field. Although, it is not far from your house, it probably doesn't have Wi-Fi coverage. So, you can build a sensor node with an ESP32 and a moisture sensor, that sends the moisture readings once or twice a day to another ESP32 using LoRa.



The later ESP32 has access to Wi-Fi, and it can run a web server that displays the moisture readings.



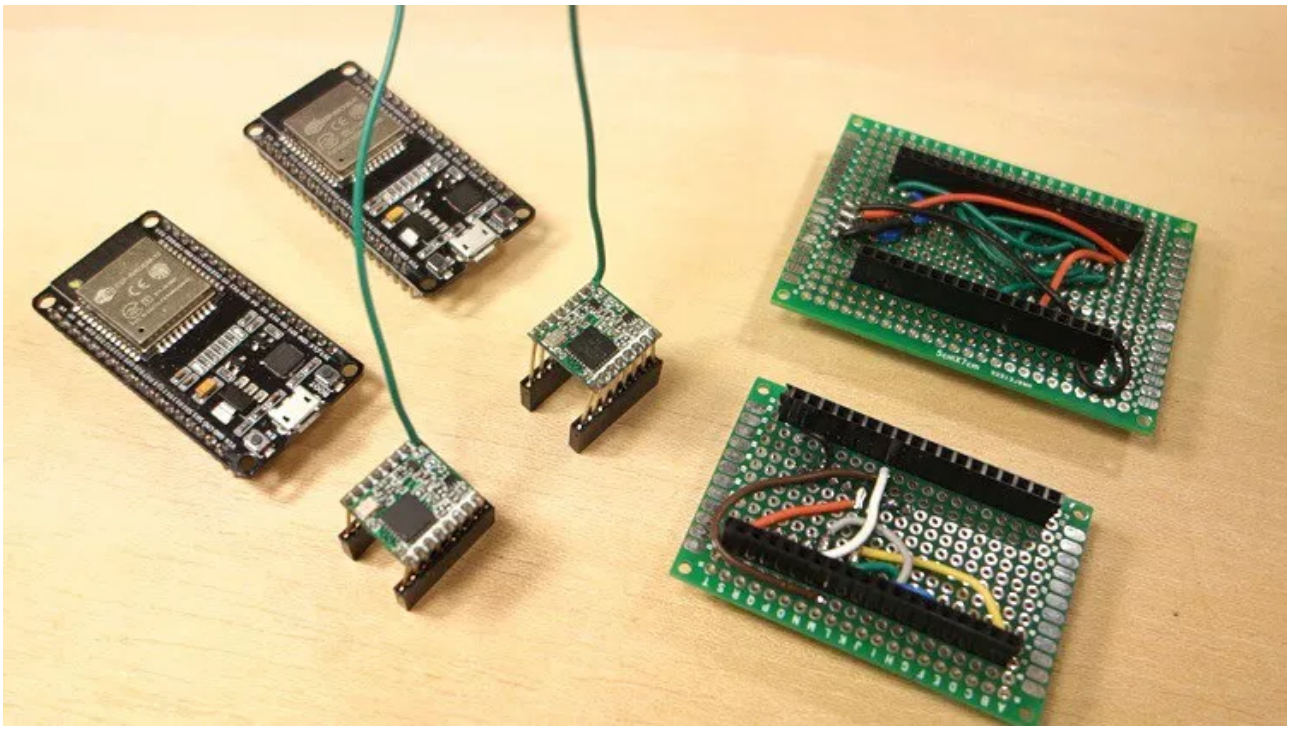
This is just an example that illustrates how you can use the LoRa technology in your ESP32 projects.

Monitoring – Reporting Sensor Readings from Outside: Soil Moisture and Temperature. Check the course page for more details.

ESP32 with LoRa

In this section we'll show you how to get started with LoRa with your ESP32 using Arduino IDE. As an example, we'll build a simple LoRa Sender and a LoRa Receiver.

The LoRa Sender will be sending a "hello" message followed by a counter for testing purposes. This message can be easily replaced with useful data like sensor readings or notifications.



To follow this part you need the following components:

- 2x ESP32 DOIT DEVKIT V1 Board
- 2x LoRa Transceiver modules (RFM95)
- RFM95 LoRa breakout board (optional)
- Jumper wires
- Breadboard or stripboard

You can use the preceding links or go directly to MakerAdvisor.com/tools to find all the parts for your projects at the best price!

Preparing the Arduino IDE

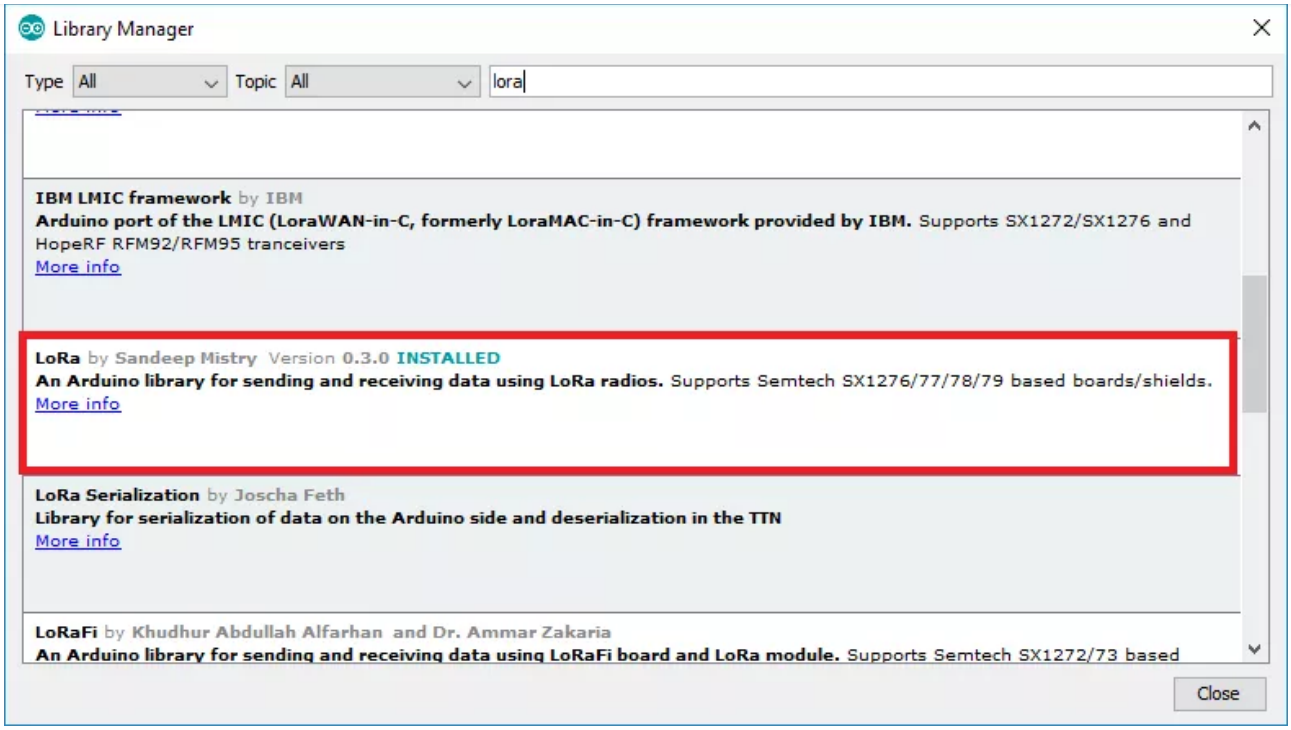
There's an add-on for the Arduino IDE that allows you to program the ESP32 using the Arduino IDE and its programming language. Follow one of the next tutorials to prepare your Arduino IDE to work with the ESP32, if you haven't already.

- **Windows** instructions – ESP32 Board in Arduino IDE
- **Mac and Linux** instructions – ESP32 Board in Arduino IDE

Installing the LoRa Library

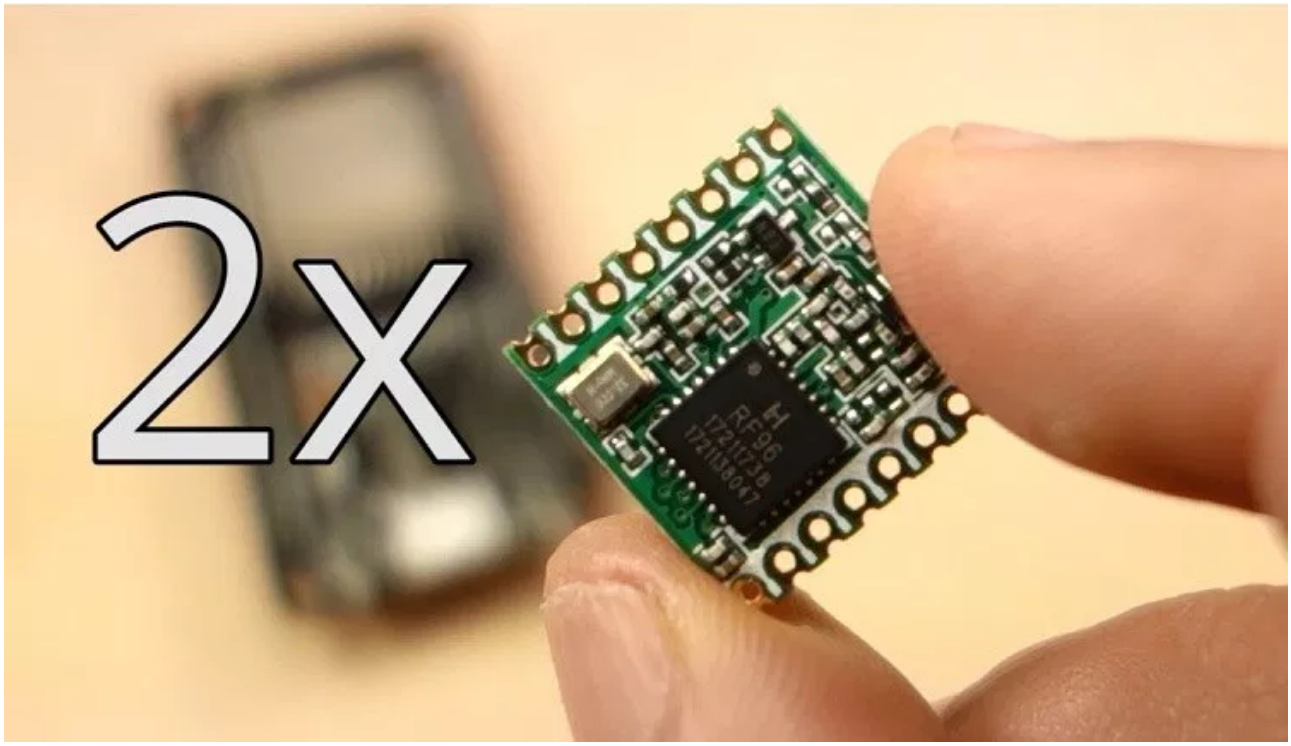
There are several libraries available to easily send and receive LoRa packets with the ESP32. In this example we'll be using the arduino-LoRa library by sandeep mistry.

Open your Arduino IDE, and go to **Sketch** > **Include Library** > **Manage Libraries** and search for "**LoRa**". Select the LoRa library highlighted in the figure below, and install it.



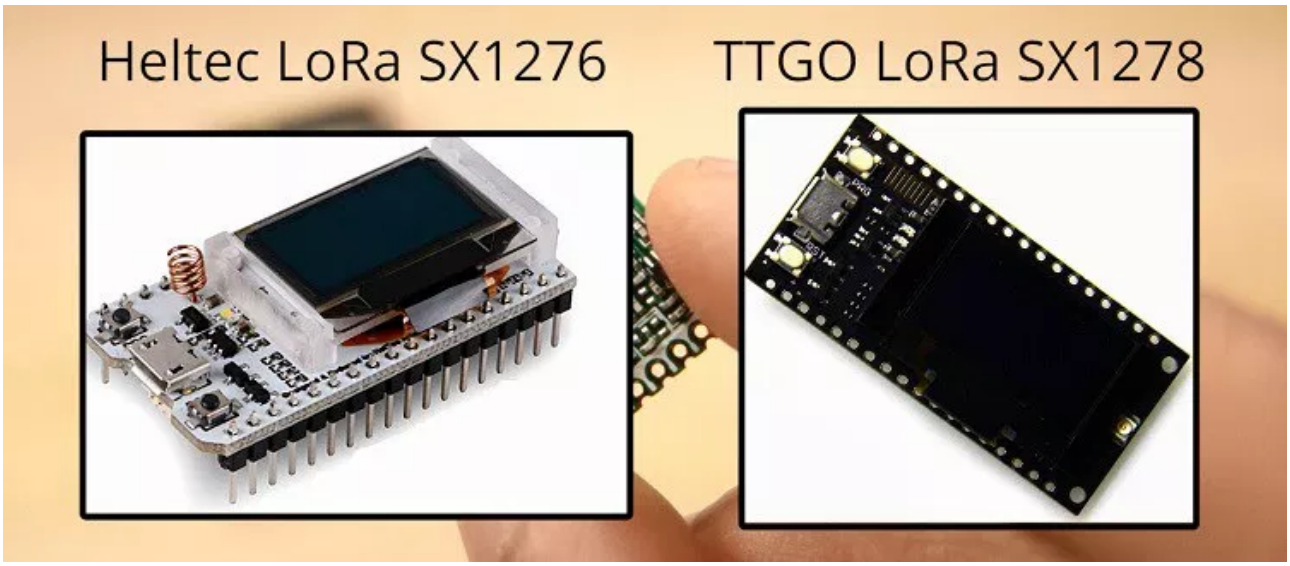
Getting LoRa Tranceiver Modules

To send and receive LoRa messages with the ESP32 we'll be using the RFM95 transceiver module. All LoRa modules are transceivers, which means they can send and receive information. You'll need 2 of them.



You can also use other compatible modules like Semtech SX1276/77/78/79 based boards including: RFM96W, RFM98W, etc...

Alternatively, there are ESP32 boards with LoRa and OLED display built-in like the ESP32 Heltec Wifi Module, or the TTGO board.

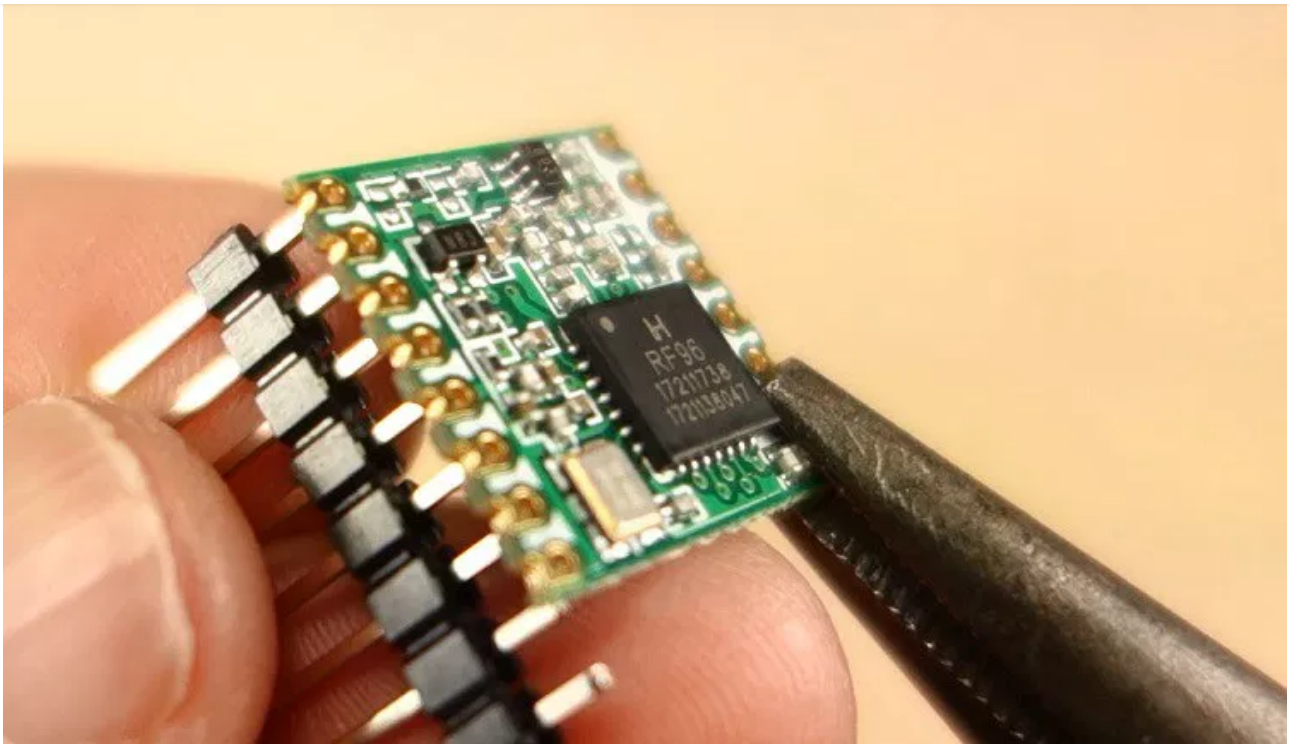


Before getting your LoRa transceiver module, make sure you check the correct frequency for your location. You can visit the following web page to learn more about RF signals and regulations according to each country. For example, in Portugal we can use a frequency between 863 and 870 MHz or we can use 433MHz. For this project, we'll be using an RFM95 that operates at 868 MHz.

Preparing the RFM95 Transceiver Module

If you have an ESP32 development board with LoRa built-in, you can skip this step.

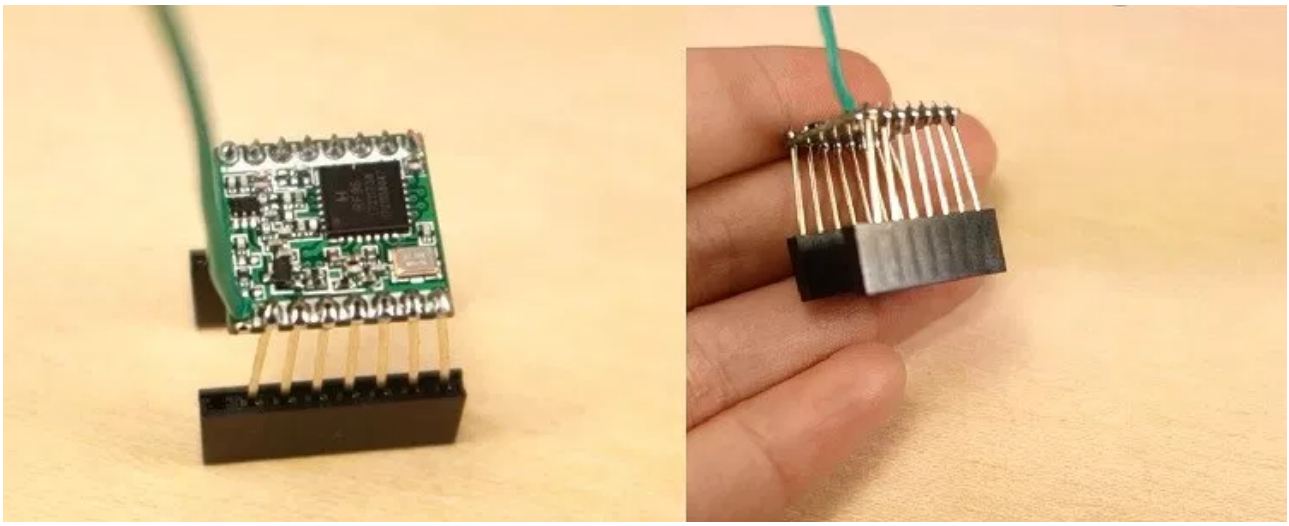
The RFM95 transceiver isn't breadboard friendly. A common row of 2.54mm header pins won't fit on the transceiver pins. The spaces between the connections are shorter than usual.



There are a few options that you can use to access the transceiver pins.

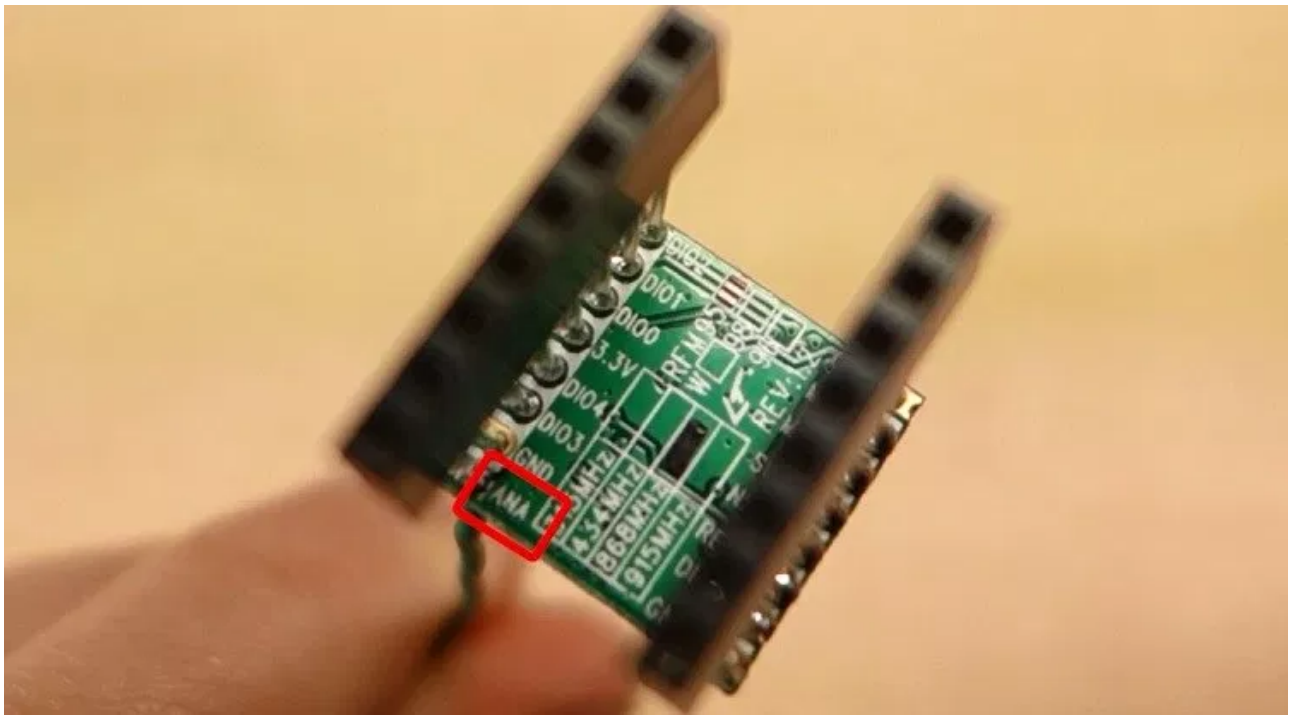
- You may solder some wires directly to the transceiver;
- Break header pins and solder each one separately;
- Or you can buy a breakout board that makes the pins breadboard friendly.

We've soldered a header to the module as shown in the figure below.

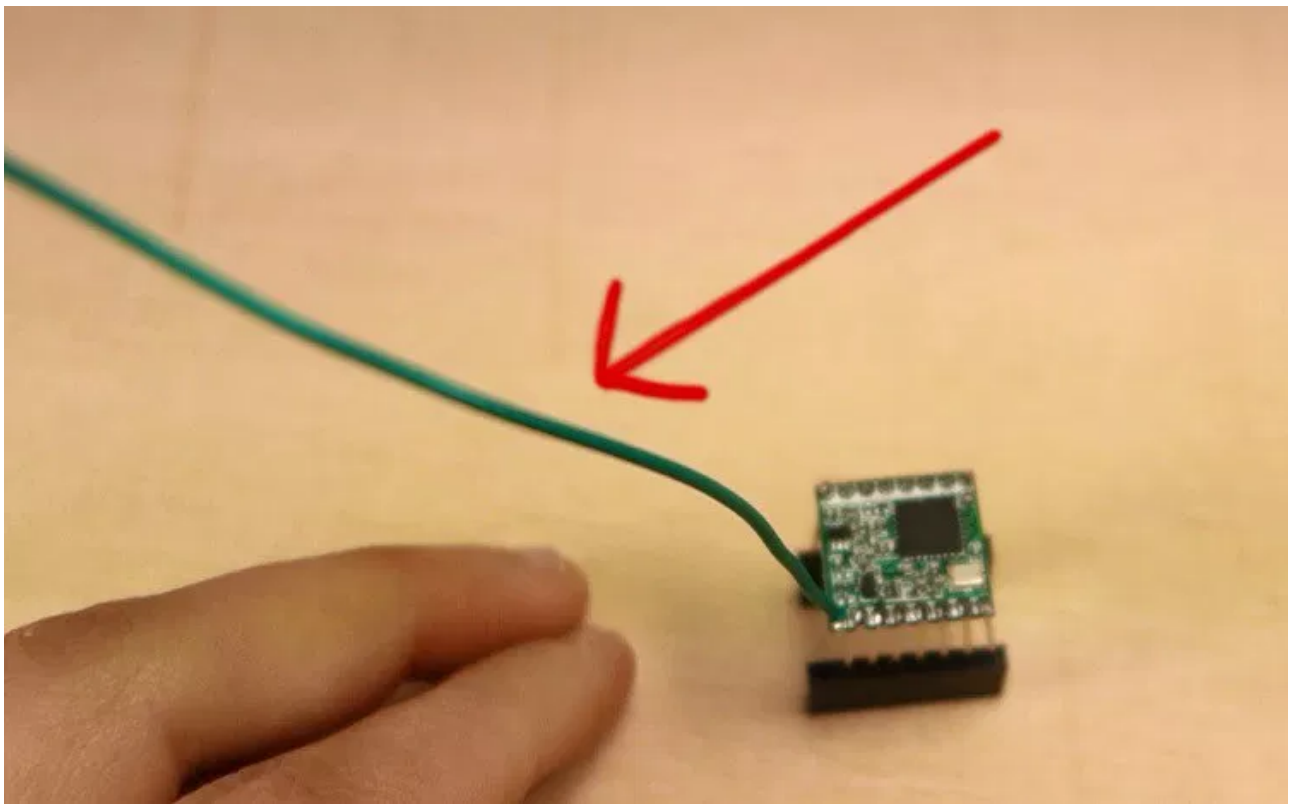


This way you can access the module's pins with regular jumper wires, or even put some header pins to connect them directly to a stripboard or breadboard.

The RFM95 transceiver chip requires an external antenna connected to the ANA pin.



You can connect a “real” antenna, or you can make one yourself by using a conductive wire as shown in the figure below. Some breakout boards come with a special connector to add a proper antenna.



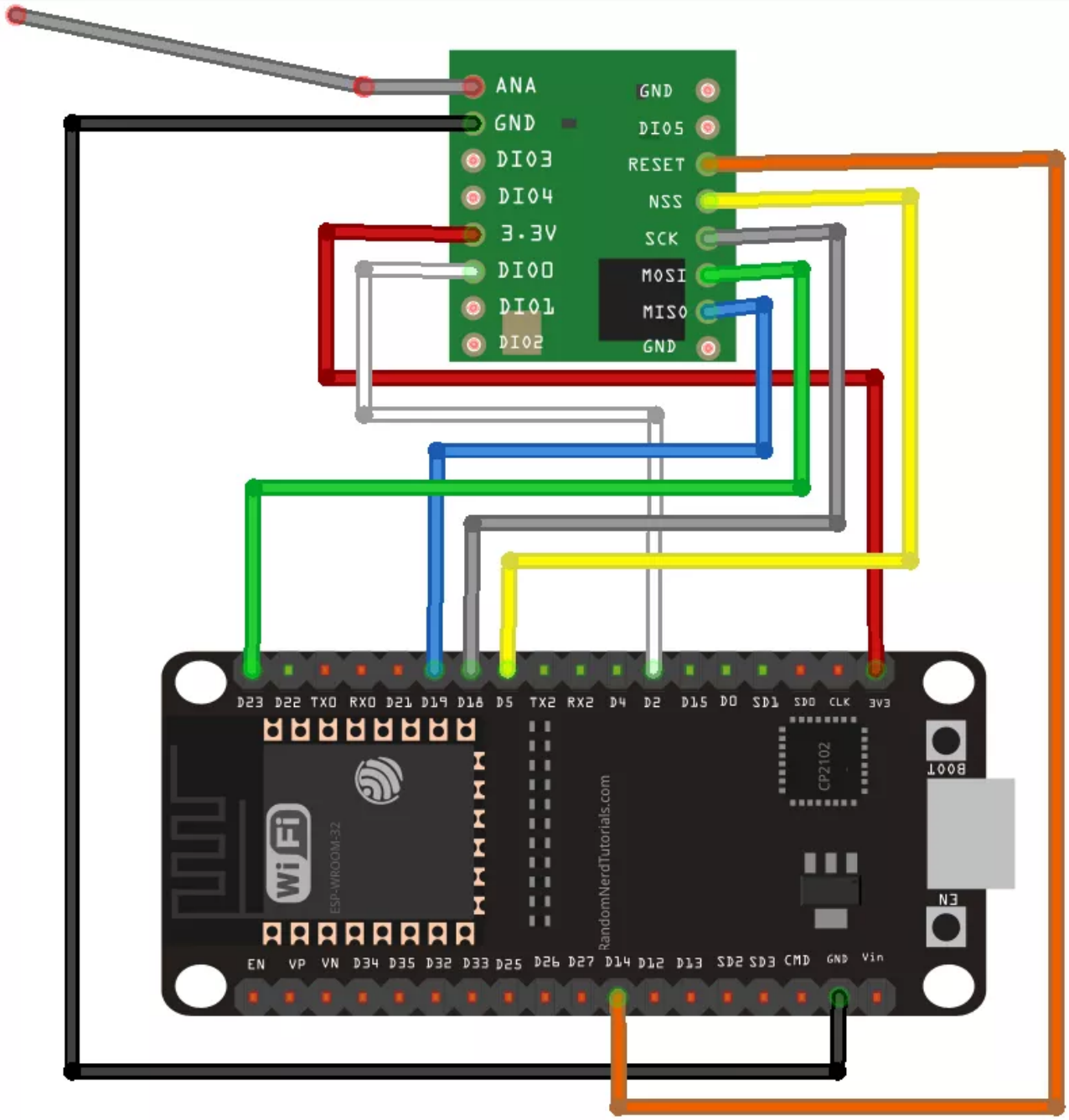
- 868 MHz: 86,3 mm (3.4 inch)
- 915 MHz: 81,9 mm (3.22 inch)
- 433 MHz: 173,1 mm (6.8 inch)

For our module we need to use a 86,3 mm wire soldered directly to the transceiver's ANA pin. Note that using a proper antenna will extend the communication range.

Important: you MUST attach an antenna to the module.

Wiring the RFM95 LoRa Transceiver Module

The RFM95 LoRa transceiver module communicates with the ESP32 using SPI communication protocol. So, we'll use the ESP32 default SPI pins. Wire both ESP32 boards to the corresponding transceiver modules as shown in the next schematic diagram:



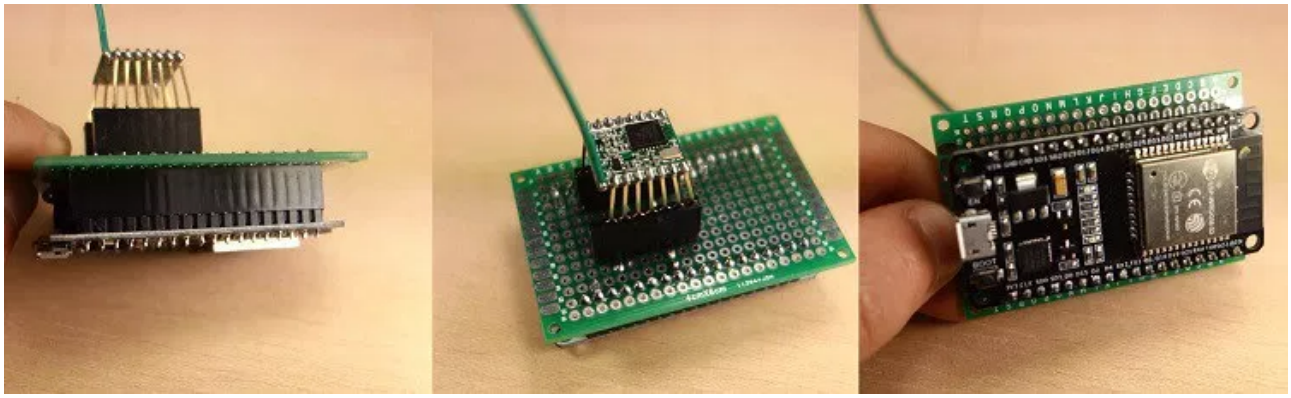
Here's the connections between the RFM95 LoRa transceiver module and the ESP32:

- ANA: Antenna
- GND: GND
- DIO3: don't connect
- DIO4: don't connect
- 3.3V: 3.3V
- DIO0: GPIO 2
- DIO1: don't connect
- DIO2: don't connect
- GND: don't connect

- RESET: GPIO 14
- NSS: GPIO 5
- SCK: GPIO 18
- MOSI: GPIO 23
- MISO: GPIO 19
- GND: don't connect

Note: the RFM95 transceiver module has 3 GND pins. It doesn't matter which one you use, but you need to connect at least one.

For practical reasons we've made this circuit on a stripboard. It's easier to handle, and the wires don't disconnect. You may use a breadboard if you prefer.



The LoRa Sender Sketch

Open your Arduino IDE and copy the following code. This sketch is based on an example from the LoRa library. It transmits messages every 10 seconds using LoRa. It sends a "hello" followed by a number that is incremented in every message.

```
//433E6 for Asia
//866E6 for Europe
//915E6 for North America
while (!LoRa.begin(866E6)) {
  Serial.println(".");
  delay(500);
}
 // Change sync word (0xF3) to match the receiver
// The sync word assures you don't get LoRa messages from other LoRa trans
```

```
    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++;

    delay(10000);
```

View raw code

Let's take a quick look at the code.

It starts by including the needed libraries.

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

Then, define the pins used by your LoRa module. If you've followed the previous schematic, you can use the pin definition used in the code. If you're using an ESP32 board with LoRa built-in, check the pins used by the LoRa module in your board and make the right pin assignment.

```
#define ss 5
#define rst 14
#define dio0 2
```

```
int counter = 0;
```

In the `setup()`, you initialize a serial communication.

```
Serial.begin(115200);
while (!Serial);
```

Set the pins for the LoRa module.

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

And initialize the transceiver module with a specified frequency.

```
while (!LoRa.begin(866E6)) {
  Serial.println(".");
  delay(500);
}
```

You might need to change the frequency to match the frequency used in your location. Choose one of the following options:

- 433E6
- 866E6
- 915E6

LoRa transceiver modules listen to packets within its range. It doesn't matter where the packets come from. To ensure you only receive packets from your sender, you can set a sync word (ranges from 0 to 0xFF).

```
LoRa.setSyncWord(0xF3);
```

Next, in the `loop()` you send the LoRa packets. You initialize a packet with the `beginPacket()` method.

```
LoRa.beginPacket();
```

You write data into the packet using the `print()` method. As you can see in the following two lines, we're sending a hello message followed by the counter.

```
LoRa.print("hello ");
LoRa.print(counter);
```

Then, close the packet with the `endPacket()` method.

```
LoRa.endPacket();
```

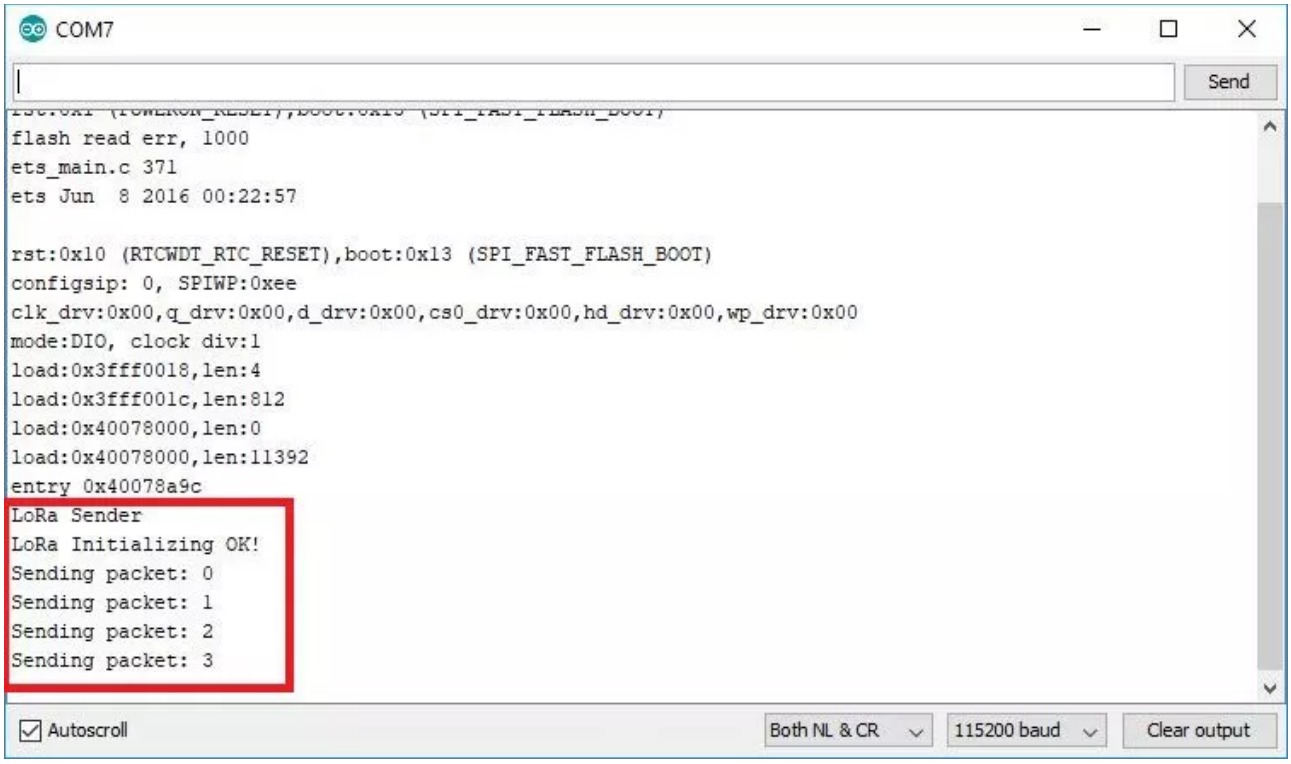
After this, the counter message is incremented by one in every loop, which happens every 10 seconds.

```
counter++;
delay(10000);
```

Testing the Sender Sketch

Upload the code to your ESP32 board. Make sure you have the right board and COM port selected.

After that, open the Serial Monitor, and press the ESP32 enable button. You should see a success message as shown in the figure below. The counter should be incremented every 10 seconds.



The LoRa Receiver Sketch

Now, grab another ESP32 and upload the following sketch (the LoRa receiver sketch). This sketch listens for LoRa packets with the sync word you've defined and prints the content of the packets on the Serial Monitor, as well as the RSSI. The RSSI measures the relative received signal strength.

```
/*********
  Modified from the examples of the Arduino LoRa library
  More resources: https://randomnerdtutorials.com
*********/

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

//define the pins used by the transceiver module
#define ss 5
#define rst 14
#define dio0 2

void setup() {
```

```
  while (!Serial);
  Serial.println("LoRa Receiver");

  //setup LoRa transceiver module
  LoRa.setPins(ss, rst, dio0);

  //replace the LoRa.begin(---E-) argument with your location's frequency
  //433E6 for Asia
  //866E6 for Europe
  //915E6 for North America
```

View raw code

This sketch is very similar to the previous one. Only the `loop()` is different.

You might need to change the frequency and the sycnword to match the one used in the sender sketch.

In the `loop()` the code checks if a new packet has been received using the `parsePacket()` method.

```
int packetSize = LoRa.parsePacket();
```

If there's a new packet, we'll read its content while it is available.

To read the incoming data you use the `readString()` method.

```
while (LoRa.available()) {
  String LoRaData = LoRa.readString();
  Serial.print(LoRaData);
}
```

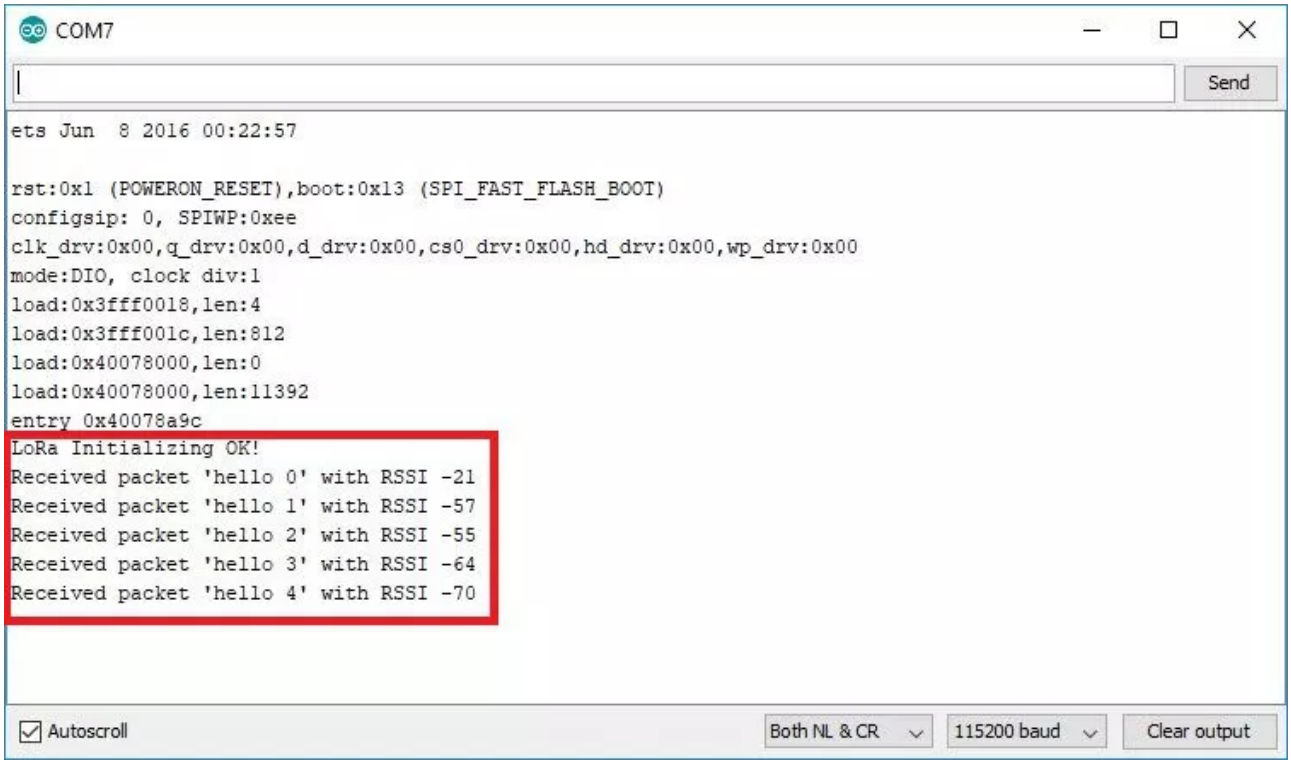
The incoming data is saved on the `LoRaData` variable and printed in the Serial Monitor.

```
  Serial.print("' with RSSI ");
  Serial.println(LoRa.packetRssi());
```

Testing the LoRa Receiver Sketch

Upload this code to your ESP32. At this point you should have two ESP32 boards with different sketches: the sender and the receiver.

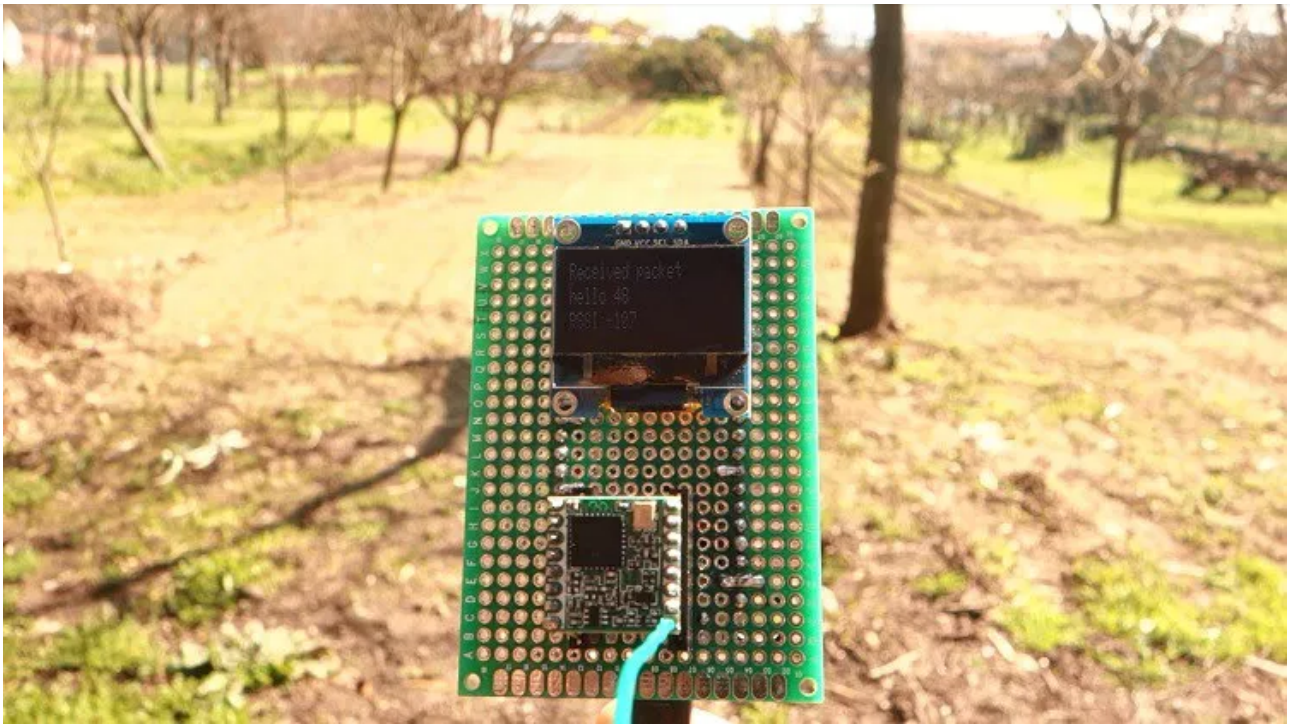
Open the Serial Monitor for the LoRa Receiver, and press the LoRa Sender enable button. You should start getting the LoRa packets on the receiver.



Congratulations! You've built a LoRa Sender and a LoRa Receiver using the ESP32.

Taking It Further

Now, you should test the communication range between the Sender and the Receiver on your area. The communication range greatly varies depending on your environment (if you live in a rural or urban area with a lot of tall buildings). To test the communication range you can add an OLED display to the LoRa receiver and go for a walk to see how far you can get a



In this example we're just sending an hello message, but the idea is to replace that text with useful information.

Wrapping Up

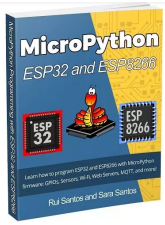
In summary, in this tutorial we've shown you the basics of LoRa technology:

- LoRa is a radio modulation technique;
- LoRa allows long-distance communication of small amounts of data and requires low power;
- You can use LoRa in point to point communication or in a network;
- LoRa can be especially useful if you want to monitor sensors that are not covered by your Wi-Fi network and that are several meters apart.

We've also shown you how to build a simple LoRa sender and LoRa receiver. These are just simple examples to get you started with LoRa. We'll be adding more projects about this subject soon, so stay tuned!

This is an excerpt from our course: Learn ESP32 with Arduino IDE. If you like ESP32 and you want to learn more, we recommend enrolling in Learn ESP32 with Arduino IDE course.

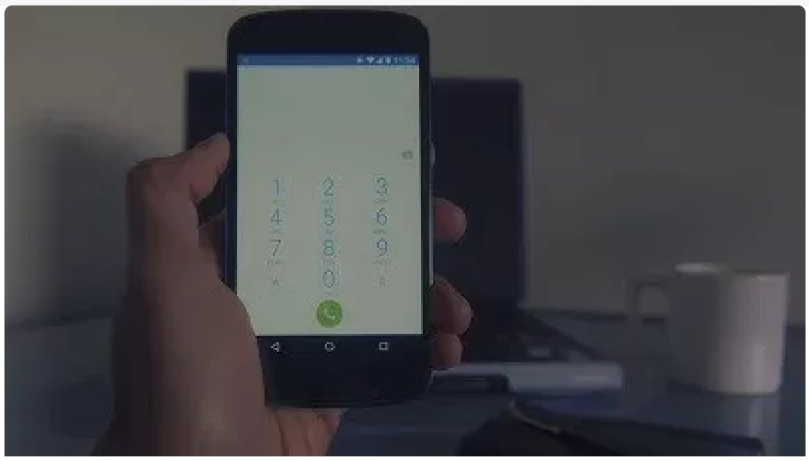
Thanks for reading.



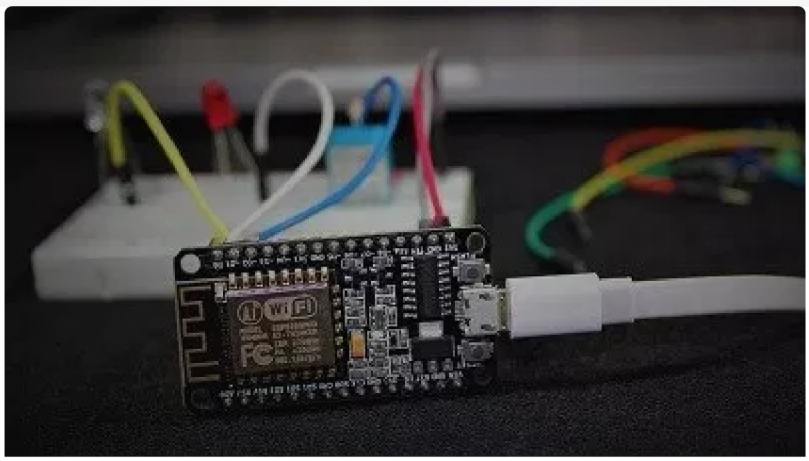
with ESP32 and ESP8266

Learn how to program and build projects with the ESP32 and ESP8266 using MicroPython firmware **DOWNLOAD »**

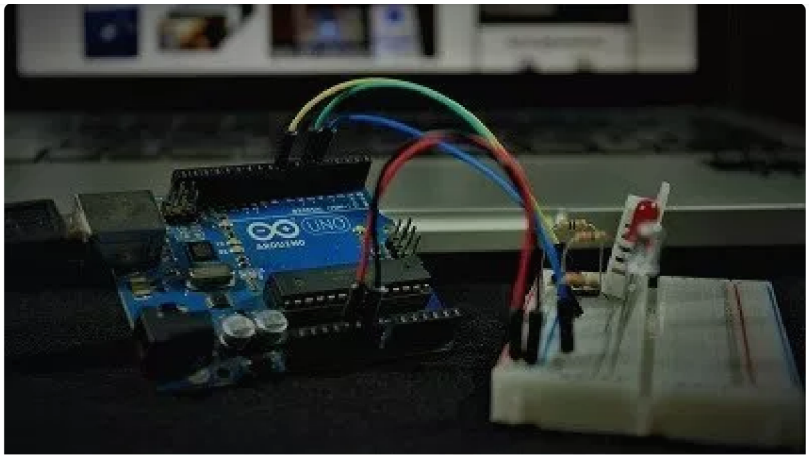
Recommended Resources



Build a Home Automation System from Scratch » With Raspberry Pi, ESP8266, Arduino, and Node-RED.



Home Automation using ESP8266 eBook and video course » Build IoT and home automation projects.



What to Read Next...

Raspberry Pi Publishing MQTT Messages to ESP8266

ESP32/ESP8266 RGB LED Strip with Color Picker Web Server

Power ESP32/ESP8266 with Solar Panels (includes battery level monitoring)

ESP32-CAM Video Streaming Web Server (works with Home Assistant)

Flashing MicroPython Firmware with esptool.py on ESP32 and ESP8266

ESP8266 OTA Updates with Arduino IDE | Over the Air

Enjoyed this project? Stay updated by subscribing our weekly newsletter!

Your Email Address

SUBSCRIBE

46 thoughts on “ESP32 with LoRa using Arduino IDE –

Bruno Hoegger
June 23, 2018 at 11:04 am | Reply

LoRa article is super, I love it

Sara Santos
June 25, 2018 at 11:04 am | Reply

Thanks 🙂

Sheldon
June 23, 2018 at 2:14 pm | Reply

Another great article, Rui. Thank you!

Umar Muhammad
June 23, 2018 at 6:43 pm | Reply

Great tutorial, thank you so much.

Umar Muhammad
June 23, 2018 at 6:58 pm | Reply

Sara Santos
June 25, 2018 at 11:27 am | Reply

Hi.
We have just tested the examples with the ESP32 (but I think it should work with the ESP8266).
If it doesn't work you can use the RadioHead library with a few modifications in the code.

Jack
June 23, 2018 at 9:08 pm | Reply

Thanks Rui, for a very well documented project :-). I will need to jig the code to use RS232 LoRa radios!

Sara Santos
June 25, 2018 at 11:07 am | Reply

Thanks 🙂

jony
June 24, 2018 at 5:46 am | Reply

Nice.

Kevin Hogan
June 26, 2018 at 1:26 am | Reply

Working through your course. Love it. Thanks. Do you have a Fritzing object for the ESP32? The ones on the forum are for a different board then the one you recommended (Geekcreit® ESP32 Development Board from Banggood). I have the DEV1 board with 18 pins on each side. Also, this board does not fit a standard breadboard (to wide). I am going to solder to a PCB, but was wondering if you had any other suggestions that were less permanent.

Sara Santos
June 26, 2018 at 8:50 am | Reply

Hi Kevin.
I have a suggestion for a less permanent solution to make the ESP32 breadboard friendly.
Usually breadboards come with two power rails on each side. Get two breadboards.
Remove one power rail per breadboard.
Then you can fit your ESP32 between the two breadboards and have plenty of space to connect your components.
I hope this helps.
(We have a part for the ESP32. I've designed it myself. But it is really bad designed to use in Fritzing. I don't recommend using it, as you need to make some hacks later to design the schematics.)
Regards,
Sara 🙂

cezabaltazar
July 6, 2018 at 7:43 am | Reply

Menu

thanks for the article. But how to crate a point to point communication to 3 Esp32? the first esp32 has a sensor attached to it and its capable of sending data to the second esp32, then the second esp32 must receive the data from the first esp32 then send it again to the third esp32. but I'm stuck on my second esp32 because i can't figure it out how can i send the data to the third one. can anyone help me about this? please.

Sara Santos
July 7, 2018 at 3:57 pm | Reply

Hi.
We don't have any tutorial about that subject.
But, I think you may find useful taking a look at the "LoRa Duplex communication with callback" example on the LoRa library.
In your Arduino IDE, go to File > Examples > LoRa > LoRaDuplexCallback.
That example sets both a receiver and a transmitter on the same esp32 (you probably need to do this for your second esp32). This code uses a callback to detect incoming messages.
I hope you find this tips useful.
Good luck with your project.
Regards,
Sara 🙂

John B
July 11, 2018 at 12:24 am | Reply

How would you compare this to other wireless options like the nrf24l01 ?
I would appreciate your thoughts. Some points of comparison are

Thanks in advance
John

Rui Santos
July 11, 2018 at 10:37 am | Reply

Hi John,
Both options are good and have different applications.
I recommend using LoRa for projects that need to cover a long range.
In terms of battery both boards should be fairly similar, but only testing in your specific scenario and code will give you accurate results.
Regards,
Rui

Ralph Hulslander
July 12, 2018 at 6:51 pm | Reply

Why ESP32?

Couldn't any micro be used such as UNO or even Raspberry Pi or ESP8266?

Ralph

Rui Santos
July 13, 2018 at 2:54 pm | Reply

You're right. You could use any other board with these LoRa modules. In fact, I'll be publishing tutorials using LoRa with Raspberry

price, but the ESP8266 would also be a good alternative.
Regards,
Rui

Scott
August 17, 2018 at 7:54 pm | Reply

I am looking into using these, but lets say I have 3 devices, could I set the sync word all to be the same and then just use something in the packets to say its from x or y like LoRa.print("10: hello"); LoRa.print("10: world"); where 10 is the device sending. I am basically trying to make it so if i have a LoRa come in reach of another one they talk to each other, and if i had let's say 5, they could all blast each other. I haven't tried these yet but if you have any suggestions that would be awesome to get me going on something.

Sara Santos
August 20, 2018 at 3:34 pm | Reply

Hi Scott.
I haven't tried it, but I think you can use the same syncword to all devices and then, change the message accordingly to the device that sent the message.
For example, device 1 sends something like this: "1/hello".
Then, you can split that message to save which device sent the message.
Something as follows to split the message:

```
int pos1 = LoRaData.indexOf('/');
device = LoRaData.substring(0, pos1);
message = LoRaData.substring(pos1 +1, LoRaData.length());
```

I hope this helps in your future project.
Regards

Naz
October 11, 2018 at 7:06 am | Reply

Hi, I'm using Heltec Wifi Lora kit with your code to test the communication between them. It managed to send and received the data but only within the short range where LoRa supposely can transmit within a long range. Can I know how far that your ESP32 can transmitted the data?

Sara Santos
October 13, 2018 at 9:57 am | Reply

Hi Naz.
With a very rudimentar antenna, we were able to transmit data between two esp32 boards 250 meters apart.
Regards,
Sara 🙂

Luis Canossa
November 23, 2018 at 5:19 pm | Reply

More examples with MQTT integration... 😉

Sara Santos
November 23, 2018 at 5:47 pm | Reply

Hi Luis.
In the following example we show how to integrate ESP32 with Node-RED using MQTT:
https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/
Regards,
Sara 🙂

hussainb
November 28, 2018 at 3:18 pm | Reply

can something custom other than 0 to 0xFF be set as syncword?
I was just wondering that 0 to 255 is not a huge range and in future when LoRa becomes common this range will not be enough to isolate devices.

Sara Santos
November 29, 2018 at 10:48 am | Reply

Hi.
That's how the library is configured at the moment.
However, other libraries may have a bigger range of sync words, or maybe you can modify the library yourself to include a bigger range.
Regards,
Sara

jak
May 25, 2019 at 12:23 am | Reply

Sync word size can be set from 1 to 8 bytes (i.e. 8 to 64 bits) via SyncSize in RegSyncConfig

surender mohan
February 20, 2019 at 6:29 am | Reply

dear sir/ma'am
i am just beginner
i want to know that we can make some node with sensor and one gateway which connect through internet using wifi
sensor retrieve data using esp 32 and send to lora network and gateway we receive and send internet server
i want use multiple node
sorry bed english
please give me some advice

Sara Santos
February 20, 2019 at 5:23 pm | Reply

Hi.
We don't have any free tutorial about that subject.
If you want to build an ESP32 web server that publishes sensor readings received via LoRa from other ESP32 boards, I recommend that you take a look at how to build an ESP32 web server to publish sensor readings. Then, take a look at the examples on the LoRa library. They provide simple examples on how to send and receive data.
Here are some resources:
– Arduino LoRa library: github.com/sandeepmistry/arduino-LoRa
– Publish sensor readings using ESP32 (instead of publishing DHT readings, publish your own readings received via LoRa):

I hope this helps.
Regards,
Sara

Guillermo amorin
February 27, 2019 at 2:28 am | Reply

I try to use the examples but i have a problen woth the library when it checks some version.

Because of that the sketch stay infinitely in the while

while (!LoRa.begin(866E6)) { Serial.println(“.”);
delay(500);
}

Sara Santos
February 28, 2019 at 12:16 pm | Reply

Hi Guillermo.
That means that it can’t initialize the LoRa module. It is probably not properly connected.
Make sure you double check all your connections.
Regards,
Sara

Soap
March 22, 2019 at 1:08 pm | Reply

Sara Santos
March 26, 2019 at 11:18 am | Reply

Hi.
I'm sorry, but we don't have any tutorial about that subject.
Regards,
Sara

Nuttaphat Juprempree
March 25, 2019 at 5:04 pm | Reply

Hi, Blog's Owner
I want to use LoRa ESP 32 to get data from sensor and send data to Receiver and I want to use Receiver to control relay to work.

Sara Santos
March 26, 2019 at 10:17 am | Reply

Hi.
This example exchanges information between the sender and receiver.
You just need to modify the code to include the sensor readings and sen them in the packet to the receiver.
Then, on the receiver you just need to control the relay. You can learn more about how to control a relay here: https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/ This guide is for the arduino, but it will work in the ESP32.
Regards,
Sara

Salvatore Fusto
March 26, 2019 at 10:27 am | Reply

hi,
i'm interested il LORA tech: my goal is to transmit from a lora tx to a number of different receivers so the question is how many receivers can i address?
the transmission is a simple of/off command.
thanks

Sara Santos
March 27, 2019 at 10:30 pm | Reply

Hi.
I think it depends on many factors. But I don't really know the maximum number of receivers. You'll have to try that in your specific example.
I'm sorry that I can't help much.
Regards,
Sara

Salvatore Fusto
March 28, 2019 at 9:02 am | Reply

Hi Sara,
than ks for youe replay. i've read some docs on LORA and rfm chips, and it seems it is possible to set sync words, to couple transmitter and receiver chips, from 1 to 8 bytes long: do you know this?
regards

Sara Santos
March 28, 2019 at 10:27 am | Reply

Hi Salvatore.
Yes, I talk about that in the tutorial. That way you only receive packets from your devices and ignore other devices in your surroundings.
Also, you can check the library we're using to find more information, there are a couple of examples that may be useful for your projects: github.com/sandeepmistry/arduino-LoRa/tree/master/examples
Regards,
Sara

salvatore fusto
March 28, 2019 at 4:03 pm | Reply

Thank you andvregards
Salvatore

Amira
April 17, 2019 at 1:11 pm | Reply

Hi sarah,
how can I measure the consumption of ESP32 based HUZZAH32 board. is possible with a card to add by USB !!
is it possible to replace RFM95 with NRF24L01 ???

Sara Santos
April 18, 2019 at 11:07 am | Reply

Hi Amira.
NRF24L01 uses a different communication protocol.
You need the RFM95 module for this tutorial.

I'm not sure what you mean by a "card to add by USB".
Regards,
Sara

Dimas
May 16, 2019 at 8:03 am | Reply

Hi sara,
i using RFM98 in shield dragino and ESP32, its not work. in serial monitor just print "……"
can you help me? i need for final exam. thankyou

Sara Santos
May 19, 2019 at 10:11 am | Reply

Hi.
That is probably a connection problem.
The ESP32 is not able to initialize the LoRa transceiver module.
It is probably a wrong connection or insufficient power supply.
I hope this helps and good luck for your final exam.
Regards,
Sara

Dimas
May 24, 2019 at 2:22 am | Reply

Krzysztof
May 21, 2019 at 1:51 pm | Reply

Hello,
I have TTGO T-Beam ESP32 868 Mhz with Gps Neo-M8N.
How can I send GPS data from TTGO by Wi fi or bluetooth to my smartphone with android ?

Sara Santos
May 22, 2019 at 11:59 am | Reply

Hi.
I don’t have any tutorials about that subject at the moment.
Regards,
Sara

Leave a Comment

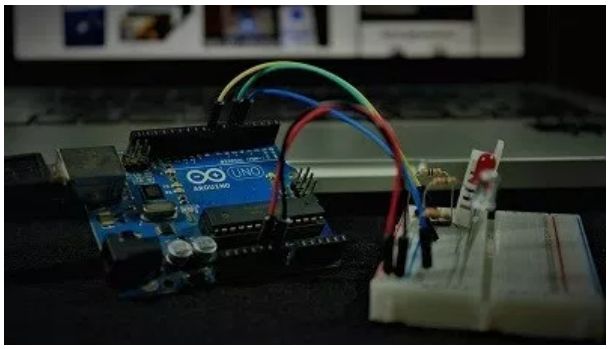
Name *

Website

Post Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.



Arduino Step-by-Step Projects »
Build 25 cool Arduino projects with our course even with no prior experience!

Find Awesome Deals For Makers @ MakerAdvisor.com

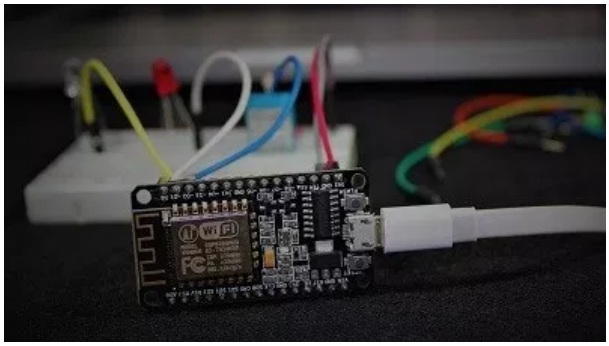
Menu

Best Low Cost Multimeter?

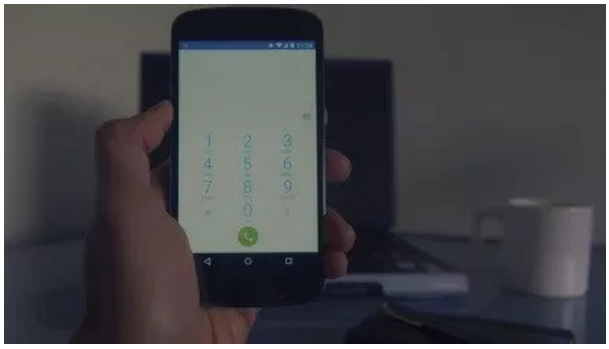
Best Bench Power Supply

Best Raspberry Pi Starter Kits

Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »



Home Automation using ESP8266 eBook and video course » Build IoT and home automation projects.



Build a Home Automation System from Scratch » With Raspberry Pi, ESP8266, Arduino, and Node-RED.