**The LoRa module that I am using here is the SX1278 Ra-02 which operates on 433MHz**.

two LoRa modules cannot communicate with each other

because the LoRaWAN Protocol does not allow communication between two LoRa modules

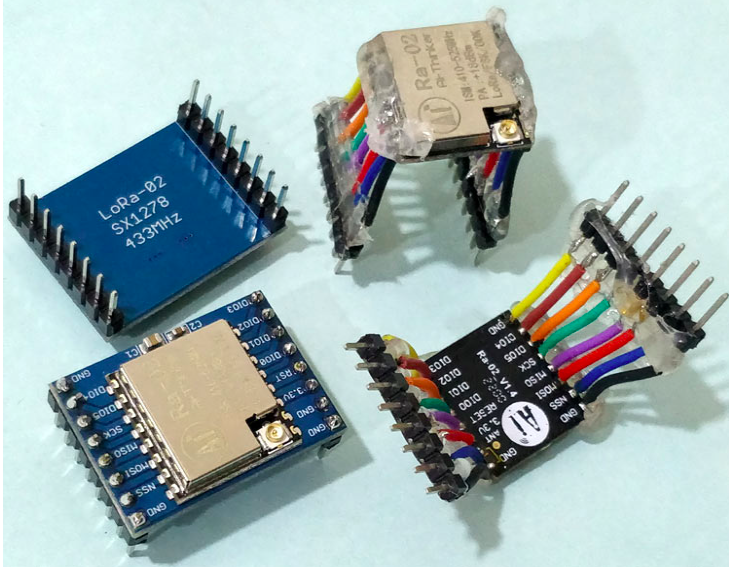
technique called the Radio Head Packet Method which does follow the LoRaWAN protocol but allows us to communicate with two LoRa modules

So let’s use two LoRa modules and two Arduino Boards to send data from one board and receive it on the other. We will use ARDUINO UNO at transmitter side and  ARDUINO UNO at receiving side.

I have to check the frequency range in Malaysia

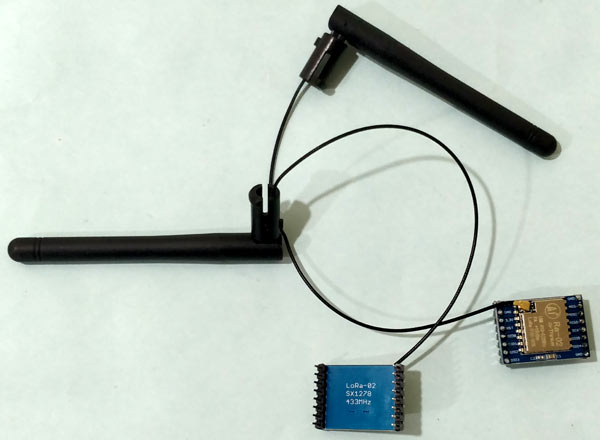
the most common being the 433MHz, 915MHz and 868MHz Frequency

you must buy the Module and Chip version soldered with wires as shown below.



**Next important thing to have with your LoRa module is your Antenna**. Remember that it is mandatory to operate the LoRa module only with an antenna, else the output transmitting power will damage the Module.

***you have to select your antenna accordingly.***



Connection part:

Transmitting Side-Connecting LoRa SX1278 with Arduino UNO

For the transmitting side we will use an **Arduino UNO with our LoRa module**. The circuit diagram to connect the Arduino UNO with LoRa is shown below

A screenshot of a computer

Description automatically generated with medium confidence

The LoRa module consists of 16 pins with 8 pins on each side. Out of these 16 pins, six are used by GPIO pins ranging from DIO0 to DIO5 and four are used by Ground pins. The module operates in 3.3V and hence the 3.3V pin on LoRa is connected to the 3.3v pin on the Arduino UNO board. Then we **connect the SPI pin on the LoRa to the SPI pins on Arduino Board** as shown above. You can also use the table below to make sure the connection is done correctly

|  |  |
| --- | --- |
| **LoRa SX1278 Module** | **Arduino UNO Board** |
| 3.3V | 3.3V |
| Gnd | Gnd |
| En/Nss | D10 |
| G0/DIO0 | D2 |
| SCK | D13 |
| MISO | D12 |
| MOSI | D11 |
| RST | D9 |

I have used connecting wires to make my connection between **Arduino UNO and LoRa Module**. The setup looks something like this shown below. The **whole set-up can is powered by a power bank** to make it portable to test the range.

A picture containing electronics

Description automatically generated

### **Receiving Side-** **Connecting LoRa SX1278 with Arduino Nano**

For the Receiving side we will use an **Arduino Nano with LoRa module**. You can use any Arduino board that you have for transmitter and receiver, but make sure you connect them accordingly. The circuit diagram to connect the **Arduino Nano with LoRa** is shown below

Graphical user interface, diagram

Description automatically generated

The connections almost remain the same except for one subtle change. **The 3.3V pin of the LoRa module is not powered by the Arduino Nano but with an external 3.3V regulator**. This is because the on-board regulator on Arduino Nano cannot provide enough current for the LoRa module to operate. Other than this the connections remain the same.

|  |  |
| --- | --- |
| **LoRa SX1278 Module** | **Arduino Nano Board** |
| 3.3V | - |
| Gnd | Gnd |
| EN/Nss | D10 |
| G0/DIO0 | D2 |
| SCK | D13 |
| MISO | D12 |
| MOSI | D11 |
| RST | D9 |

We connect the breadboard power supply

Also note that the LoRa modules that we have to use will be two small breadboard to make the connections as shown below.

A picture containing electronics

Description automatically generated

### **Preparing the Arduino IDE for LoRa Wireless Communication**

1. We add lora library

To add the library, open you Arduino IDE and follow

**Sketch -> Include Library -> Manage Libraries**.

Then search for **“LoRa Radio”** and look for the library that was made by Sandeep Mistry and click on install. Wait for the installation to complete and you should see something like this in the end.

1. restart the Arduino IDE and open the LoRa Example program by using **File -> Example -> LoRa** and then open both LoRa Receiver and LoRa Sender Program
2. The Sender program sends the data with a value of counter being incremented. The receiver then receives this and prints on the Serial monitor with the RSSI value.
3. Don’t forget to change the function LoRa.begin() inside the () you write the module that you will use 🡪 LoRa.begin(915MHz) that’s mean the frequency is 915Mhz.

The code

This is an example to send hello for 5 seconds