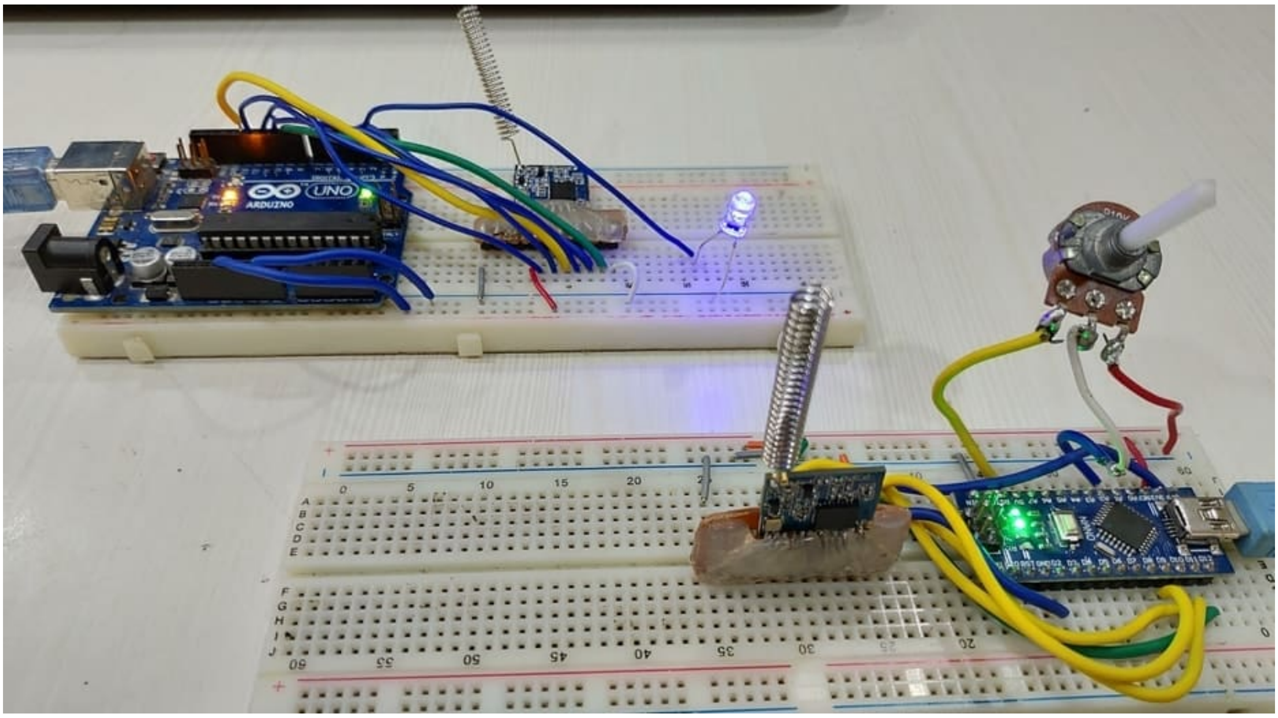
Interfacing SX1278 (Ra-02) LORA Module with Arduino

how2electronics.com/interfacing-sx1278-lora-module-with-arduino

December 5, 2019



In this tutorial, we will learn about **Interfacing SX1278 (Ra-02) LORA Module with Arduino**. We will see **SX1278 LORA & Arduino example** and make a **transmitter** & **receiver** circuit. We will control the brightness of LED wirelessly using potentiometer.

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 and access points**. The **power consumption** of Wi-Fi and Bluetooth technology is high which drains the battery quickly if you go for battery-powered mobile device.

The Same thing is with the **Cellular networks** and LAN which also have problems of high power consumption. The LAN and Cellular network both are quite expensive to cover a wide area. In the last couple of years, IoT industries introduced lots of technologies but none of them proved to be ideal. The IoT devices need to transmit information to a long distance without using much power. So there was a revolution in the IoT field when the LoRa technology was introduced. LoRa Technology can perform very-long range transmission with low power consumption. Lora applications includes Smart water monitoring, Remote Control of Appliances, Smart Parking, Autonomous irrigation, Smart Agriculture & Soil Health Monitoring.

In this tutorial, we will learn about Interfacing SX1278 (Ra-02) LORA Module with Arduino. We will see **SX1278 LORA & Arduino example** and make a **transmitter & receiver circuit**. We will control the brightness of LED wirelessly using a potentiometer. The LoRaLib library can be used with any LoRa module if it's based on one of the supported chips: **SX1272 or SX1273, SX1276, SX1277, SX1278 or SX1279, RFM95, RFM96, RFM97, or RFM98**.

Before getting started, you can visit the following posts as well:
1. *ESP8266 & LoRa SX1278 Transmitter Receiver with DHT11*: Check Here
2. *ESP32 LoRa Sensor Data Monitoring on Web Server*: Check Here
3. *ESP32 & LoRa SX1278/76 Transmitter Receiver with OLED*: Check Here
4. *Sending Sensor Data Wirelessly with LoRa SX1278 & Arduino*: Check Here

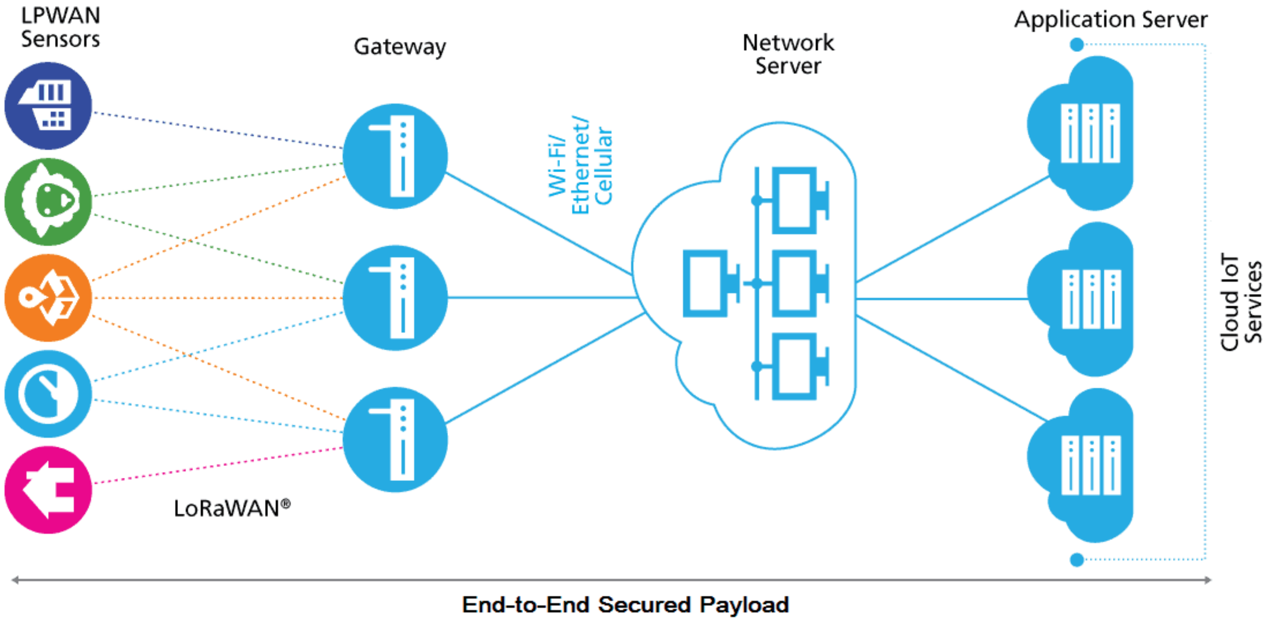
What is LoRa Technology?

The term **LoRa** stands for **Long Range**. It is a **long-range, low power** wireless platform that has become the de-facto technology for **Internet of Things (IoT)** networks worldwide. LoRa is a spread spectrum modulation technique derived from c**hirp spread spectrum (CSS) technology**. LoRa was introduced by a company called **Semtech**.

LoRa modules do come in different frequency ranges, the most common being the **433MHz, 915MHz and 868MHz**. This LoRa technology can be used to transmit bi-directional information to long-distance **(15-20km)** without consuming much power. The technology can be utilized by public, private or hybrid networks and provides greater range than Cellular networks. LoRa Technology can easily plug into existing infrastructure and enables low-cost battery-operated IoT applications.

Understanding the LoRa technology & its working

The basic principle is that information is encoded using chirp (a gradual increase or decrease in the frequency of the carrier wave over time). Before sending a message, the LoRa transmitter will send out a chirp signal to check that the band is free to send the message. Once the LoRa receiver has picked up the preamble chirp from the transmitter, the end of the preamble is signalled by the reverse chirp, which tells the LoRa transmitter that is it clear to begin transmission.



The architecture shown in this figure is explained below:

1. Devices:
It consists of LoRa Modulation, Transceivers & End-Nodes and Picocells & Gateways.

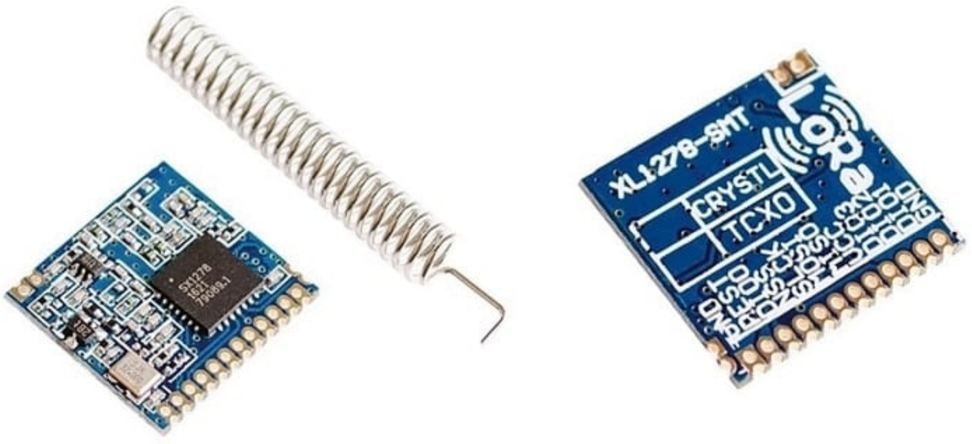
A. LoRa Modulation: LoRa Technology is the physical (PHY) silicon layer, or wireless modulation, used to create the long-range communication link.
B. Transceivers & End-Nodes: Transceivers configured with LoRa Technology are embedded into end-nodes, or sensor devices, designed for a multitude of industry applications.
C. Picocells & Gateways: Sensors capture and transmit data to gateways over distances near and far, indoor and outdoor, with minimal power requirement.

2. Network Server:
Gateways send information via Wi-Fi, Ethernet or Cellular to the network server, which is responsible for network management functions like over-the-air activation, data de-duplication, dynamic frame routing, adaptive rate control, traffic management, and administration.

3. Application Servers & Cloud IoT Services:
Applications interpret the data collected by LoRa-enabled devices, applying techniques like machine learning and artificial intelligence to solve business problems for a Smarter Planet.

Semtech SX1278 LoRa 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.



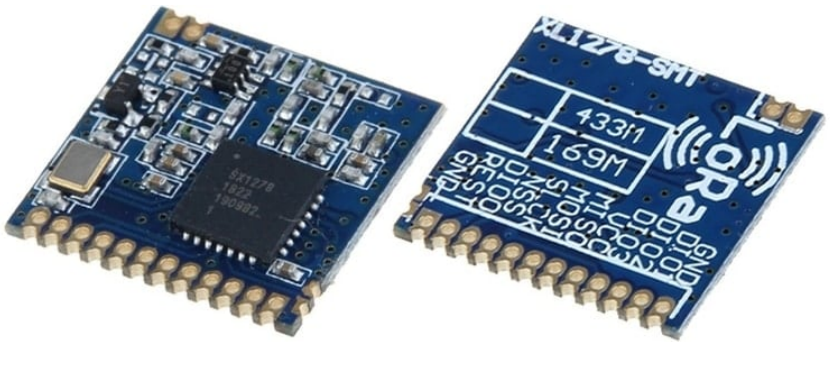
SX1278 can achieve a sensitivity of over **-148dBm** using a low-cost crystal. The high sensitivity combined with the integrated **+20dBm power amplifier** yields industry leading link budget making it optimal for any application requiring range or robustness. Lora SX1278 also provides significant advantages 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 Applications

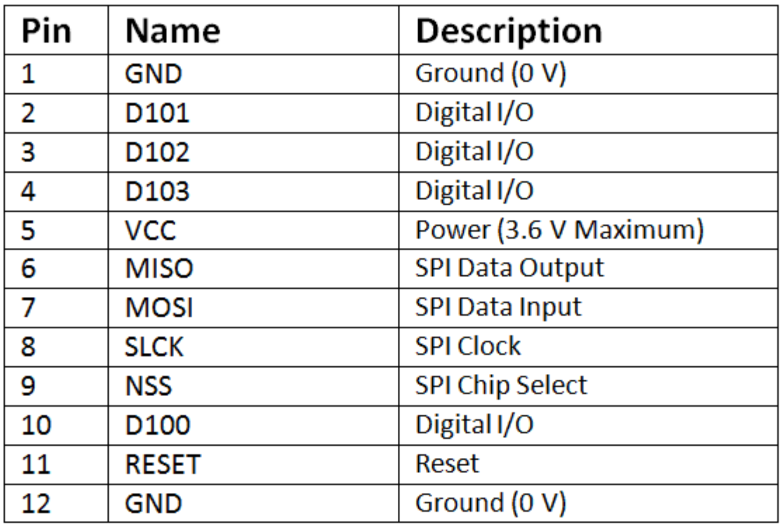
1. Automated Meter Reading
2. Home and Building Automation
3. Wireless Alarm and Security Systems
4. Industrial Monitoring and Control
5. Long range Irrigation Systems

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.



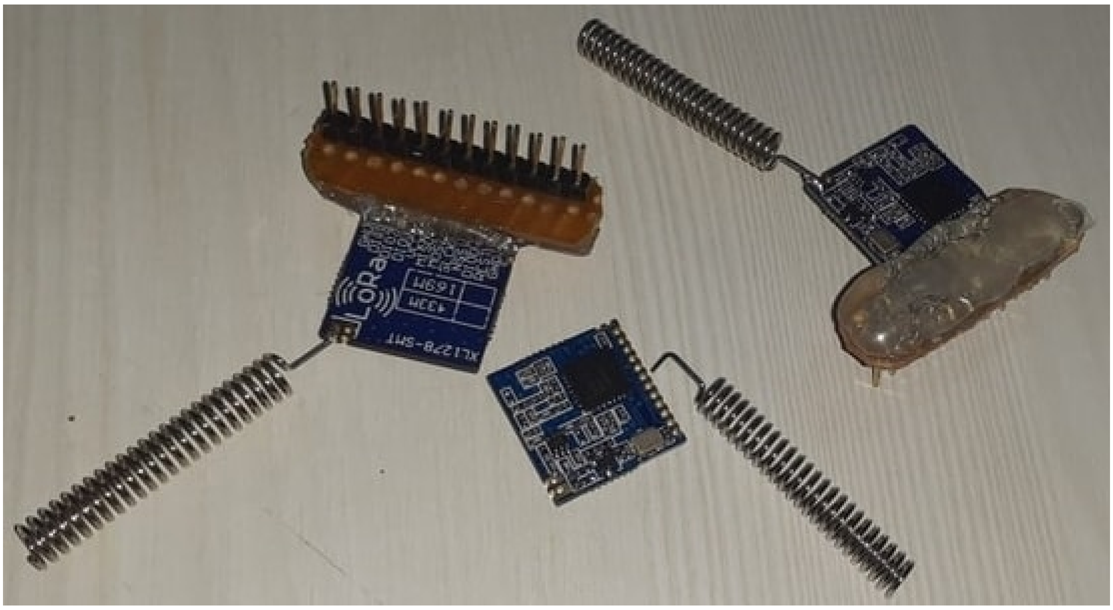
This module version of SX1278 has 12 pins for interfacing with microcontroller & additional two pins for antenna.



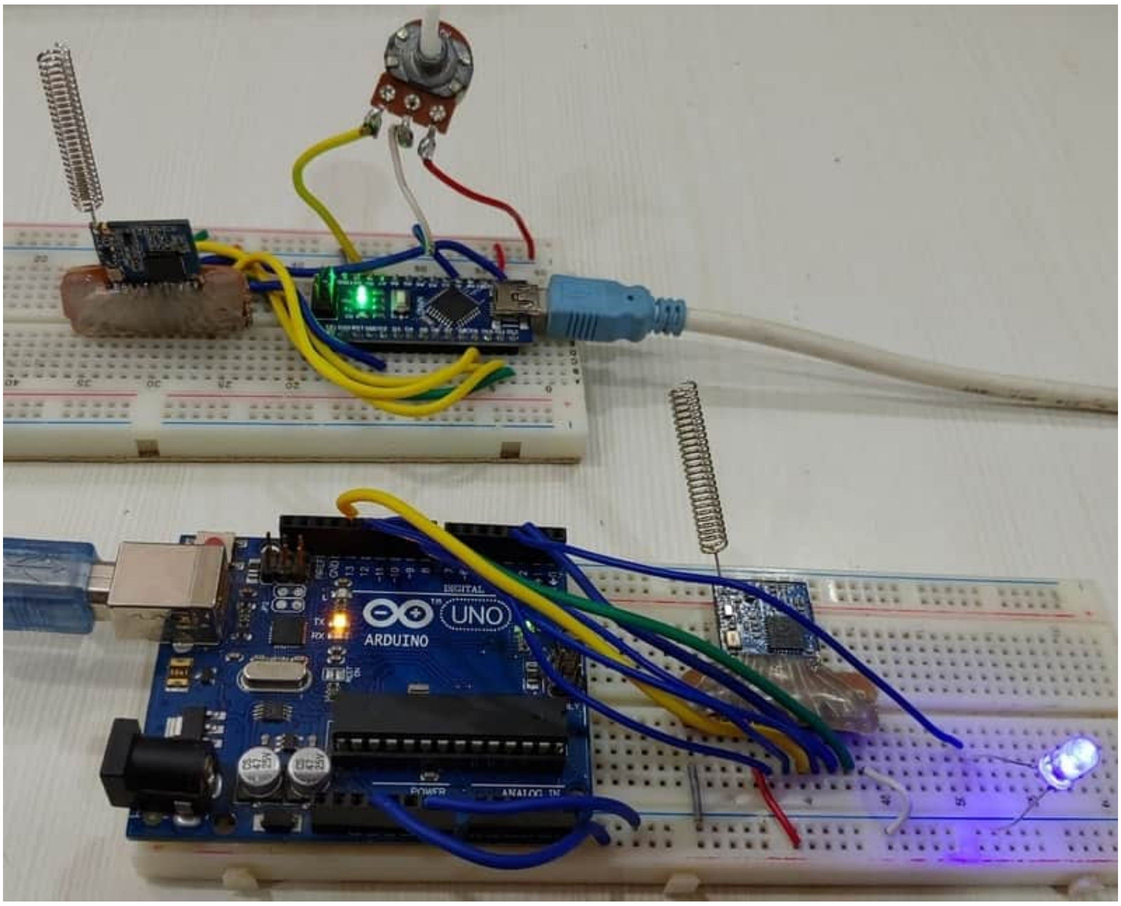
Pin	Name	Description
1	GND	Ground (0 V)
2	D101	Digital I/O
3	D102	Digital I/O
4	D103	Digital I/O
5	VCC	Power (3.6 V Maximum)
6	MISO	SPI Data Output
7	MOSI	SPI Data Input
8	SLCK	SPI Clock
9	NSS	SPI Chip Select
10	D100	Digital I/O
11	RESET	Reset
12	GND	Ground (0 V)

Interfacing SX1278 LoRa Module with Arduino

The LoRa module that I am using here is the SX1278 Ra-02 which operates on 433MHz. But this module is not breadboarded friendly neither it has a soldered antenna. So simply I soldered few 2.54″ male connector to make it breadboard friendly. I also soldered the antenna.



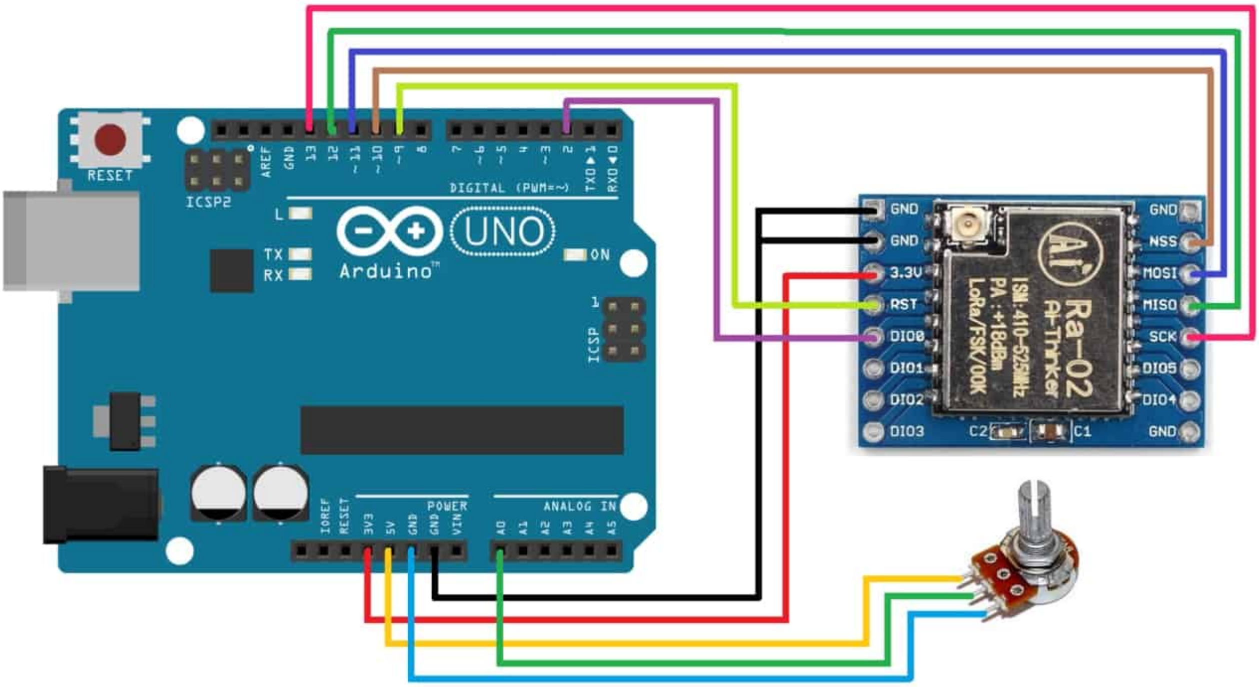
Now let us learn **SX1278 Arduino Interfacing**. For this, we will use two LoRa modules and two **Arduino Boards** to **send data** from one board and receive it on the other. We will use Arduino Nano at the **transmitter** side and Arduino Uno at the **receiver** side.



Let us see the circuit of transmitter and receiver part seperately.

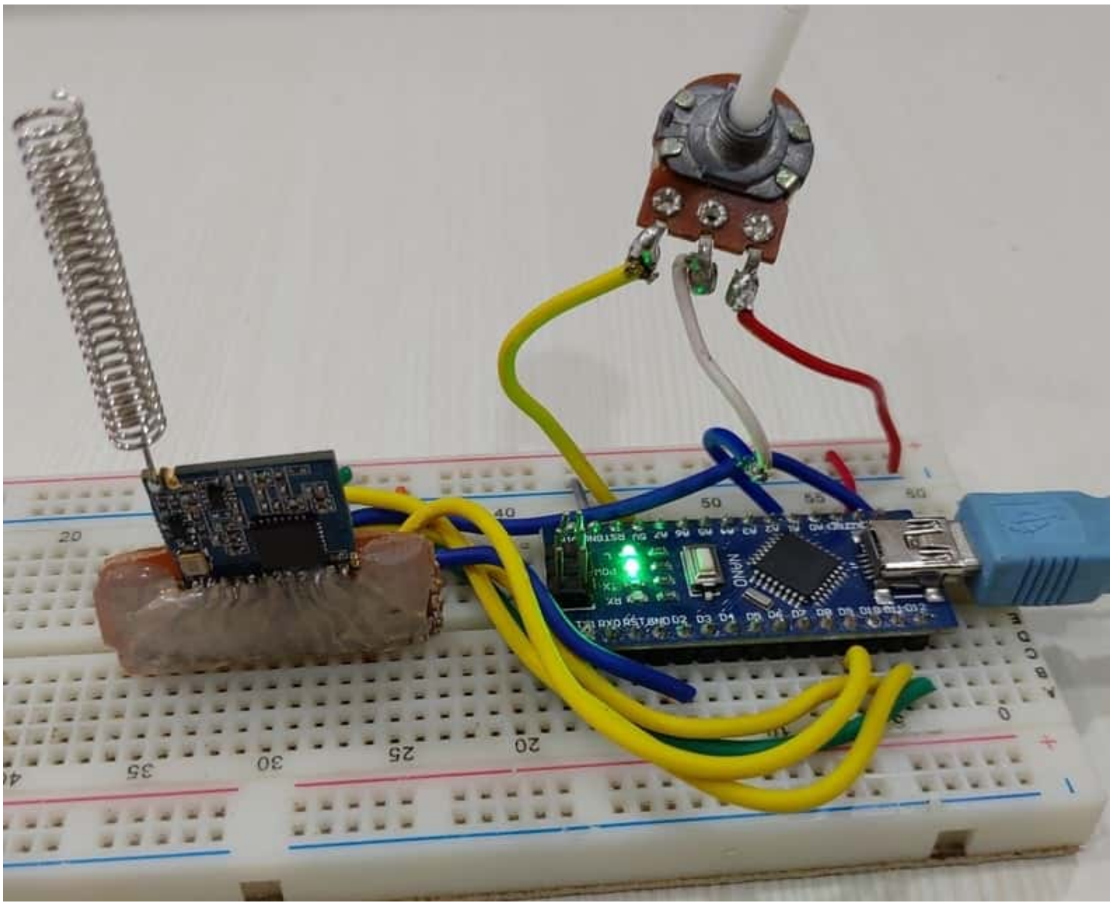
Arduino LoRa SX1278 Transmitter

The circuit diagram for **Arduino LoRa SX1278 Transmitter** is given below. You can either make a pcb for this circuit or simply assemble it on the breadboard.



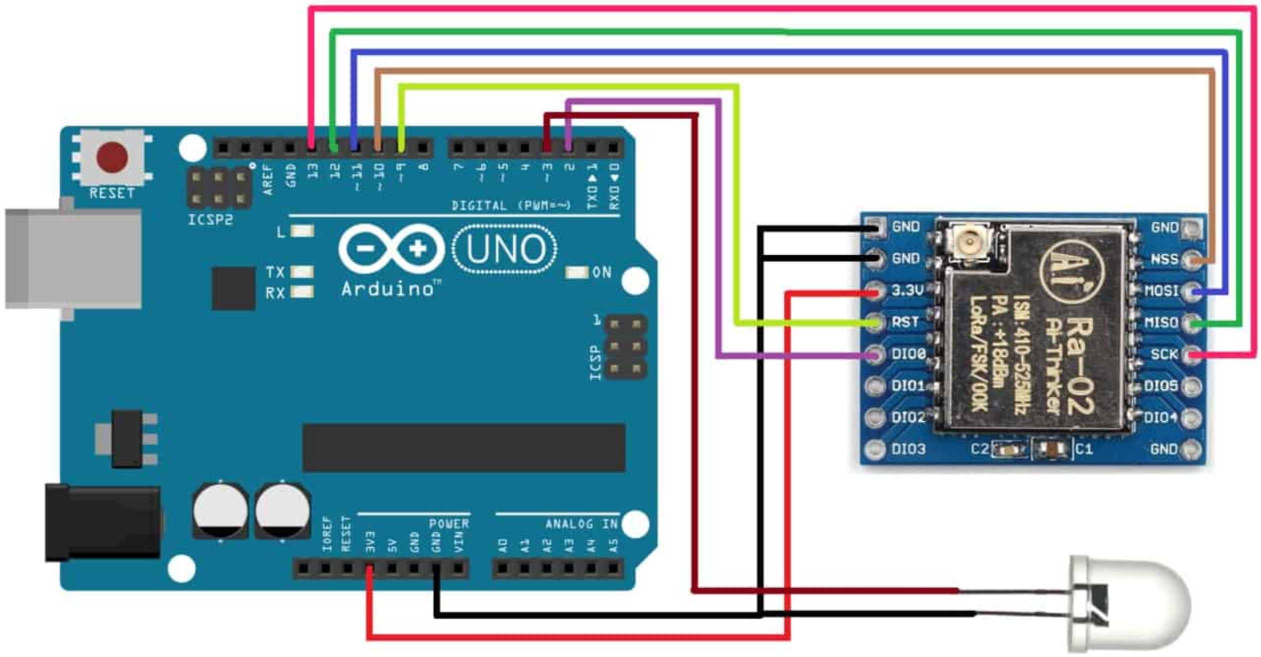
The LoRa SX1278 is not 5V friendly so do not supply 5V to it else the board will get damaged. Use 3.3V of Arduino to connect it to VCC pin. Connect all the GND pins to GND. Connect the RST pin to D9 and DIO0 to D2 of Arduino. Connect the SPI Pins NSS, MOSI, MISO, SCK to Arduino D10, D11, D12, D13 of Arduino respectively as shown in circuit diagram above.

Use any potentiometer like 10K and connect its middle pin to A0 of Arduino and remaining two pins to GND and 5V.



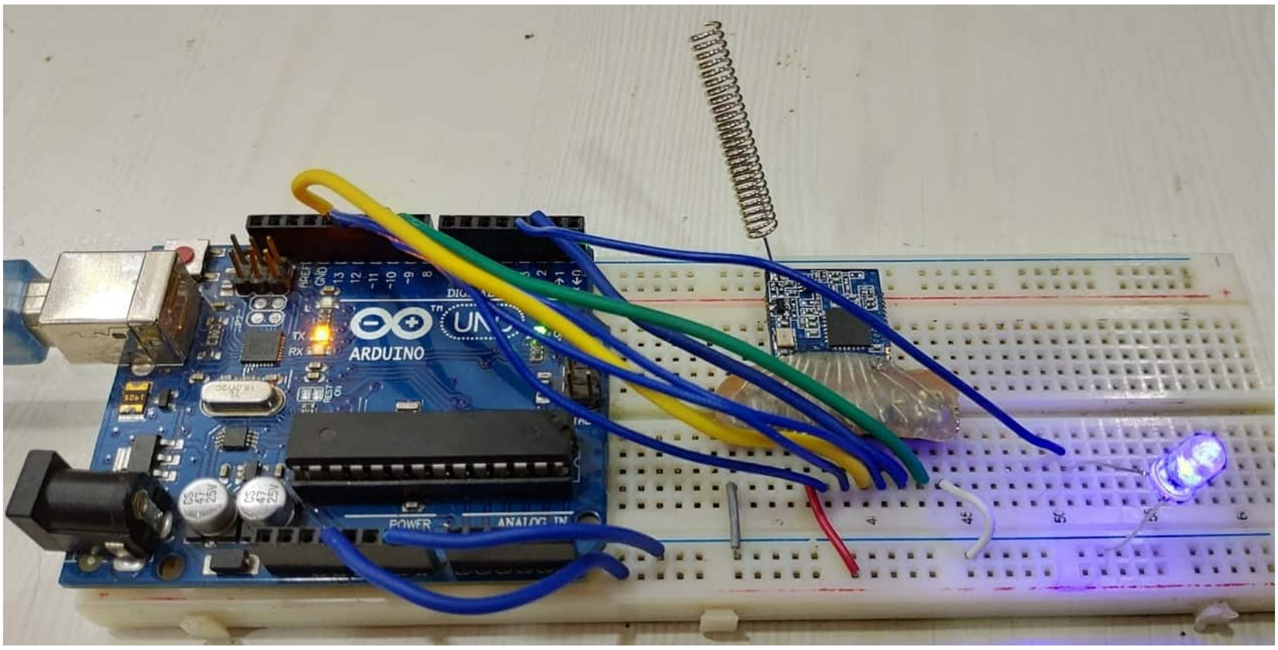
Arduino LoRa SX1278 Receiver

Similarly the circuit diagram for **Arduino LoRa SX1278 Receiver** is given below. You can either make a pcb for this circuit or simply assemble it on the breadboard.



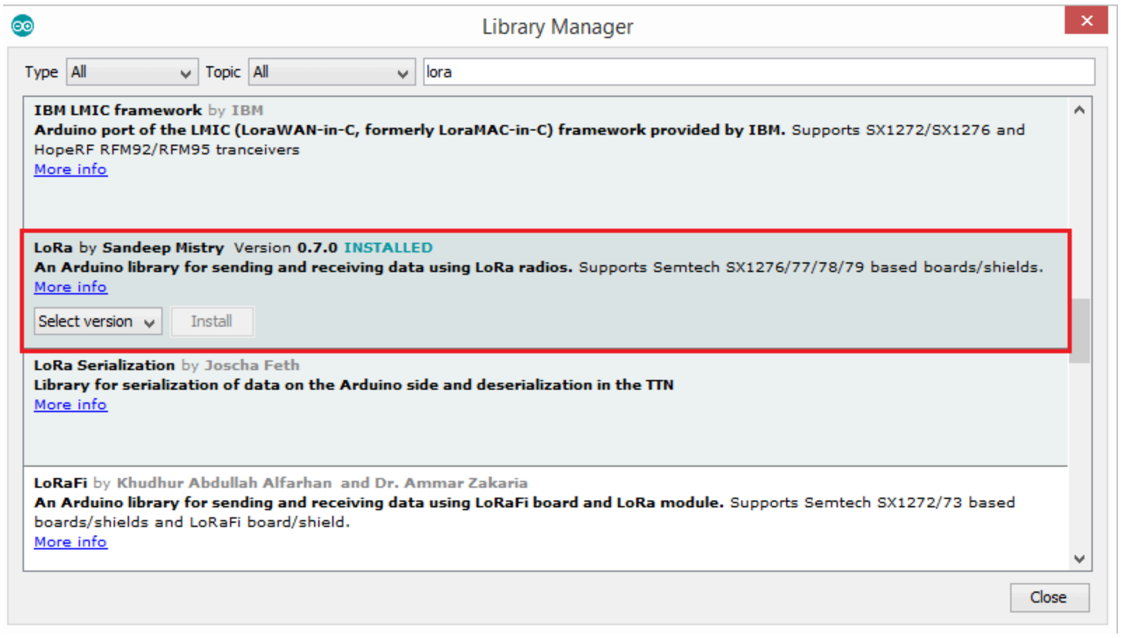
The LoRa SX1278 is not 5V friendly so do not supply 5V to it else the board will get damaged. Use 3.3V of Arduino to connect it to VCC pin. Connect all the GND pins to GND. Connect the RST pin to D9 and DIO0 to D2 of Arduino. Connect the SPI Pins NSS, MOSI, MISO, SCK to Arduino D10, D11, D12, D13 of Arduino respectively as shown in circuit diagram above.

Use any LED and connect it to D3 of Arduino as shown in the photos above.



Source Code/Program

For communication between two **LoRa Module**, you need to have LoRa library. To get the **LoRa Library** go to library manager and search for LoRa and install it.



Type All Topic All lora

IBM LMIC framework by IBM
Arduino port of the LMIC (LoraWAN-in-C, formerly LoraMAC-in-C) framework provided by IBM. Supports SX1272/SX1276 and HopeRF RFM92/RFM95 tranceivers
More info

LoRa by **Sandeep Mistry** Version **0.7.0** INSTALLED
An Arduino library for sending and receiving data using LoRa radios. Supports Semtech SX1276/77/78/79 based boards/shields.
More info

LoRa Serialization by Joscha Feth
Library for serialization of data on the Arduino side and deserialization in the TTN
More info

LoRaFi by Khudhur Abdullah Alfarhan and Dr. Ammar Zakaria
An Arduino library for sending and receiving data using LoRaFi board and LoRa module. Supports Semtech SX1272/73 based boards/shields and LoRaFi board/shield.
More info

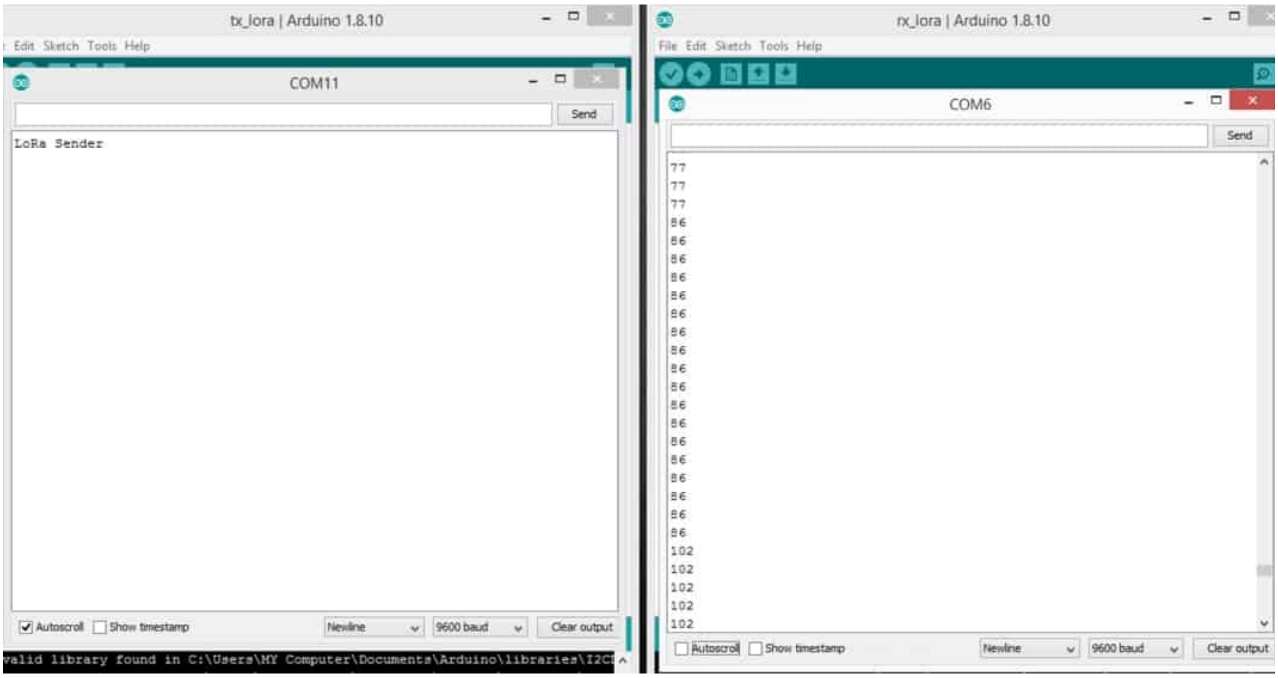
Transmitter Code

```
#include <SPI.h>
#include <LoRa.h>
int pot = A0;

void setup() {
  Serial.begin(9600);
  pinMode(pot,INPUT);

  while (!Serial);
  Serial.println("LoRa Sender");
  if (!LoRa.begin(433E6)) { // or 915E6, the MHz speed of yout module
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {
  int val = map(analogRead(pot),0,1024,0,255);
  LoRa.beginPacket();
  LoRa.print(val);
  LoRa.endPacket();
  delay(50);
}

```

Receiver Code

```
#include <SPI.h>
#include <LoRa.h>
int LED = 3;
String inString = "";    // string to hold input
int val = 0;

void setup() {
  Serial.begin(9600);
  pinMode(LED,OUTPUT);

  while (!Serial);
  Serial.println("LoRa Receiver");
  if (!LoRa.begin(433E6)) { // or 915E6
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {

  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // read packet
    while (LoRa.available())
    {
      int inChar = LoRa.read();
      inString += (char)inChar;
      val = inString.toInt();
    }
    inString = "";
    LoRa.packetRssi();
  }

  Serial.println(val);
  analogWrite(LED, val);
}
```

Once the code is uploaded you can start testing by rotating the knob of the **potentiometer** and check the **brightness of the LED**. You can check the value in serial monitor as well.



```
LoRa Sender
```

```
valid library found in C:\Users\MY Computer\Documents\Arduino\libraries\I2C
```

```
77
77
77
86
86
86
86
86
86
86
86
86
86
86
86
86
86
86
86
86
86
102
102
102
102
102
```

Video Tutorial & Guide

LoRa SX1278/76 Arduino Interfacing Tutorial | Sending Sensor Data Wirelessly with LoRa
Watch this video on YouTube.
You can check this link as well: LoRa SX1278 & ESP8266 Transmitter Receiver with DHT11 Sensor