A project report on

PEER-TO-PEER COMMUNICATION

Submitted by

BATCH A GROUP 01

Aman Sirohi	CB.EN.U4AIE21003
R Sriviswa	CB.EN.U4AIE21046
Rakhil ML	CB.EN.U4AIE21048
Vikhyat Bansal	CB.EN.U4AIE21076

As part of course completion for

21AIE211 – Introduction to Computer Networks

June - 2023

ACKNOWLEDGEMENT

We thank our Professor **Ms. Ganga Gowri** of the **21AIE211 – Introduction to Computer Networks** course who has shared her knowledge in the classroom. It helped us in doing a lot of Research and we came to know about a lot of things related to this topic. We are thankful for the opportunity provided to us. We are grateful for the cooperation and her valuable suggestions.

We also like to express our gratitude to our group members for their ongoing support and hard work on the project. With the support and assistance of one another, we could learn about the numerous concepts in the project.

DECLARATION

We the undersigned solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of **Ms. Ganga Gowri.,** Department of CEN. We assert the statements made and conclusions drawn are outcomes of our research work.

Aman Sirohi	Rakhil ML
CB.EN.U4AIE21003	CB.EN.U4AIE21048
R Sriviswa	Vikhyat Bansal
CB.EN.U4AIE21046	CB.EN.U4AIE21076

Table of contents

ABSTRACT

With the rise of the Internet of Things, connected cars, M2M, Industry 4.0, etc., LoRa is becoming more and more popular. Designers want to utilize it to send and receive data from a battery-powered thing due to its capacity to communicate over large distances with very little power.

So, in our project we have intended to use a LoRa module with Raspberry Pi to communicate with another SX1278 connected to a microcontroller like Arduino. This is nothing but Peer-to-Peer Communication that we learnt in our classes. Our main agenda was to implement the core concepts of ICN therefore it leads us to having this selected as our project.

This method can come in handy at many places since the Arduino could act as a Server to fetch data from sensors and send it to Raspberry Pi over a long distance through LoRa and then the Raspberry Pi acting as a client can receive this information and upload it to the cloud since it has access to internet.

INTRODUCTION

Modern LPWAN networks (Low Power Wide Area Network) extend over a wide territorial area, allowing the implementation of various applications of the so-called Internet of Things (IoT), previously possible only in cellular networks. These networks require communication infrastructure spread over the coverage area connected to the Internet, which entails high costs of managing the operation. However, most applications do not need a very wide coverage area.

LoRa® technology allows communication directly between devices, eliminating the need for a gateway.

With a little development effort, it is possible to implement a communication network. This architecture is commonly referred in the market as peer-to-peer or P2P.

As an example, we can mention a building automation system, with sensors installed dispersed on several floors. Sensors (connected to devices) must send some information to a central. In this way, it is possible to generate commands to the system in a minimum time interval and, after these actions, send the data to the central.

IMPLEMENTATIONS

1. Serial Communication with Arduino-Raspberry

Material:

- Arduino UNO- or any other version – 1
- Raspberry Pi – 1
- USB Cable - 1

2, P2P Communication with Arduino-Arduino

Material:

- SX1278 433MHz LoRa Module – 2 Nos
- 433MHz LoRa antenna – 2Nos
- Arduino UNO- or any other version – 2 Nos

3. P2P Communication with Arduino-Raspberry

Material:

- SX1278 433MHz LoRa Module – 2 Nos
- 433MHz LoRa antenna – 2Nos
- Arduino UNO- or any other version
- Raspberry Pi 3

4. Network Scanner

Serial Communication with Arduino-Raspberry

Introduction:

Serial communication is simply **a way to transfer data**. The data will be sent sequentially, one bit at a time (1 byte = 8 bits), contrary to parallel communication, where many bits are sent at the same time.
UART protocol
More specifically, when you use Serial with Arduino and Raspberry Pi, you're using the UART protocol. UART means "Universal Asynchronous Reception and Transmission".

Basically, it's an **asynchronous multi-master protocol** based on the Serial communication, which will allow you to communicate between the 2 boards. Be reassured, there are libraries that will handle all the low layers for you.
Multi-master means that all connected devices will be free to send data when they want. This is one of the main difference with master-slaves protocols, where only the master device can initiate a communication. Usually, you'll use other protocols such as I2C and SPI when you need master-slaves configurations: for example when you have one Arduino board and multiple sensors or actuators.

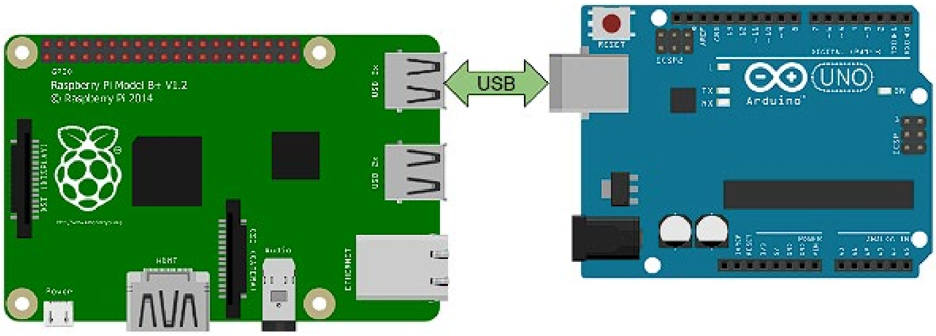
The Arduino Uno board has one UART that you can use either with a USB cable or from the RX/TX pins (don't use it with both at the same time). Some boards have more available UARTs. For example, the Arduino Mega has different Serials (Serial, Serial1, Serial2, Serial3) and the Arduino Zero has a native USB port only (use SerialUSB instead of Serial).

On the Raspberry Pi, you can connect many Serial devices on the USB ports. Each will have a different device name (we'll see how to find them later in this tutorial). You can also use the GPIOs (RX0/TX0) for an additional UART.

CONNECTION:

On the Raspberry Pi side, a simple USB connector is all you need. You can choose any of the 4 USB ports available on the board.

For Arduino, you will use the USB port that you use to upload code from your computer (with the Arduino IDE) to your board. Here the USB connector will depend on which version you have. For boards like Arduino Uno and Mega, the connector will be different from Arduino Nano, and from Arduino Zero.



As you can see, it’s the simplest hardware connection you can make between Raspberry Pi and Arduino.

Note: you will first need to connect your Arduino to your computer, so you can upload the code into the board. After that, connect the USB cable to the Raspberry Pi. The Raspberry Pi will power the Arduino via this cable.

P2P Communication between Arduino-Arduino

Introduction:

In LoRa we can achieve high distance communication without using much power, thus overcoming the drawback of Wi-Fi and BLE communication. But how is it possible? If that is the case why do BLE and RF still exist?

That is because **LoRa comes with its own drawbacks**. In order to achieve high distance with Low power LoRa compromises on Bandwidth, **it operates on very low bandwidth**. The maximum bandwidth for Lora is around 5.5 kbps, this means that you will be able to send only small amount of data through LoRa. So, you cannot send Audio or Video through this technology, it works great only for transmitting less information like sensor values. The below chart shows where **LoRa lies compared with Wi-Fi, Bluetooth and Cellular devices.**

You can think of LoRa to be more like cellular communication. Signal from one **LoRa Node** reaches another.

A LoRa Node usually operates on a Battery and consists of a Radio Module and Microprocessor. The Microprocessor is used to read the data from the senor and send it in the air through the Radio module which will then be picked up by a LoRa Gateway. The LoRa Gateway also has a Radio Module and a Microprocessor but is normally operated over AC mains since they require more power. **A single LoRa Gateway could listen to multiple LoRa nodes, while a single LoRa node could also send information to multiple gateways**, this way the information from the node will be picked up gateway without it being lost. **When information id is sent from the node to the gateway it is called as uplink and when it is sent from gateway to node it is called as down link.**

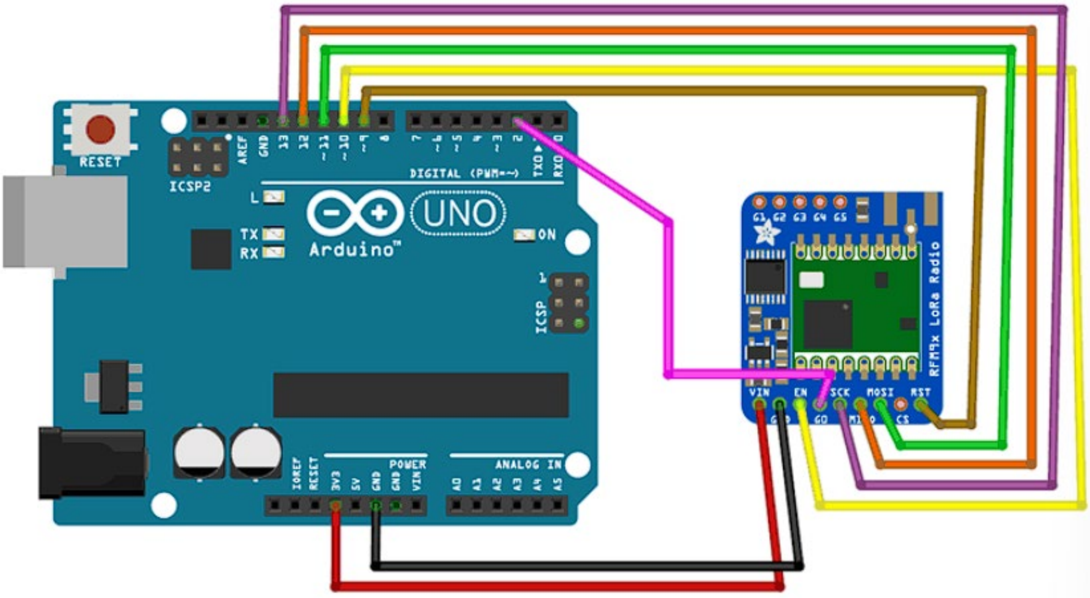
The LoRaWAN Protocol does not allow communication between two LoRa modules, but there is technique called the Radio Head Packet Method which does follow the LoRaWAN protocol but allows us to communicate with two LoRa modules.

LoRa modules do come in different frequency ranges, the most common being the 433MHz, 915MHz and 868MHz. The frequency at which your module works will be mentioned at the back of the module.

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

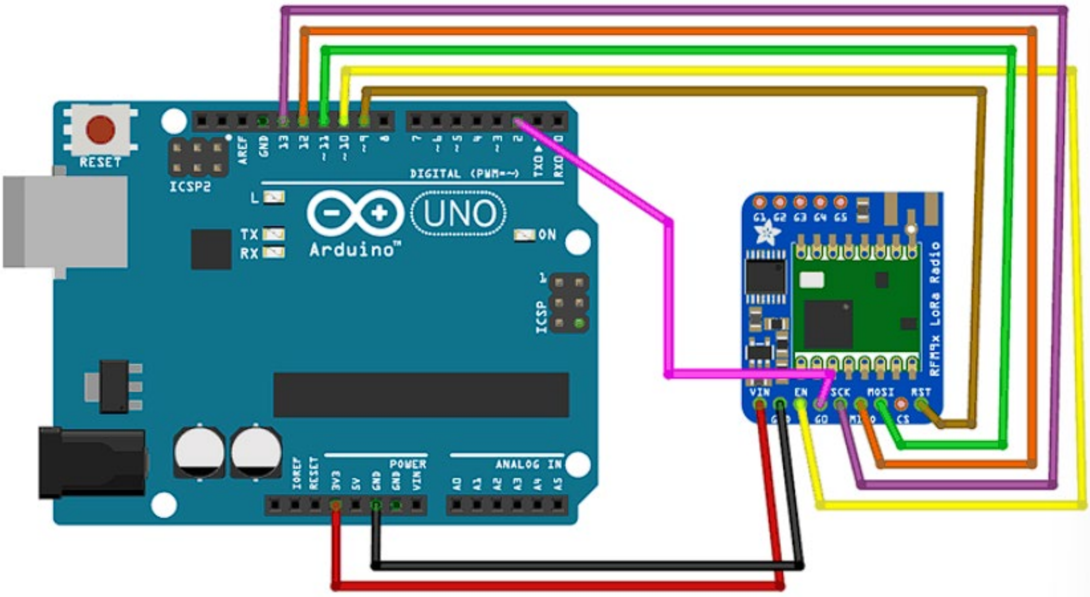
Connections:

Arduino - 1



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

Arduino - 2



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

P2P Communication between Arduino-Raspberry

Introduction:

LoRa is getting increasingly popular with the advent of IoT, Connected Cars, M2M, Industry 4.0 etc. Because of its ability to communicate to long distances with very less power it is preferably used by designers to send/receive data from a battery powered Thing. We have already discussed the basics of LoRa and how to use LoRa with arduino. Although the technology is originally intended for a LoRa Node to communicate with a LoRa gateway, there are many scenarios in which a LoRa Node has to communicate with another LoRa Node to exchange information over long distance.

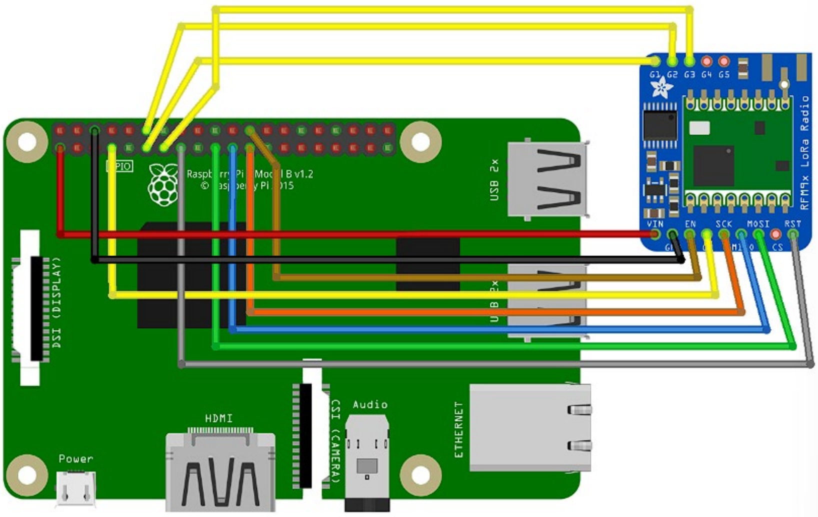
So, in this report we will learn **how to use a LoRa module SX1278 with Raspberry pi** to communicate with another SX1278 connected to a microcontroller like Arduino. This method can come in handy at many places since the Arduino could act as a Server to fetch data from sensors and send it to Pi over a long distance through LoRa and then the Pi acting as a client can receive these information and upload it to the could since it has access to internet.

Connections:

Raspberry-LoRa

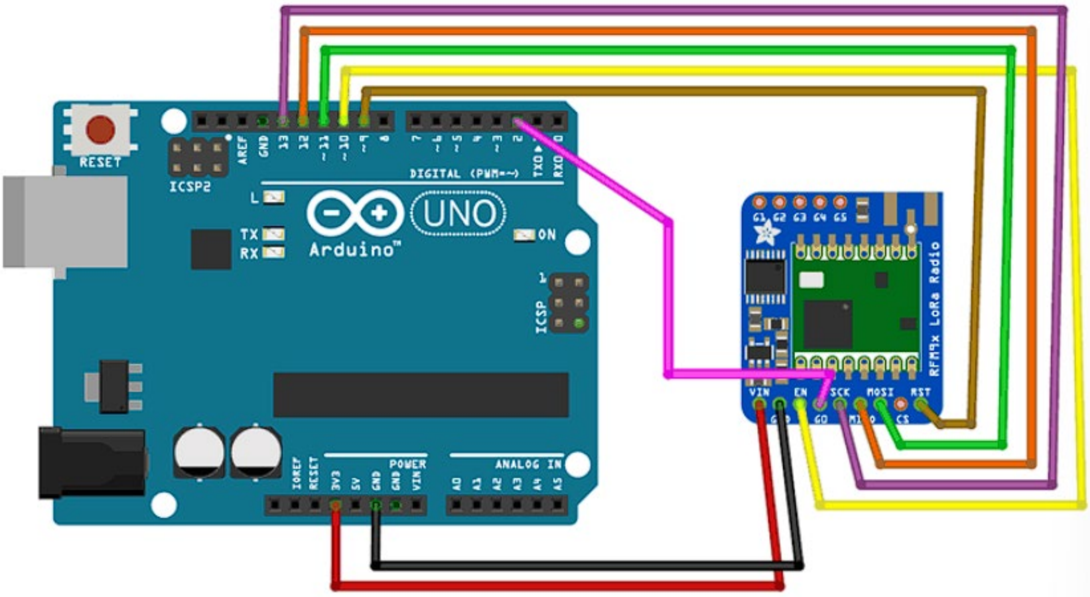
The **SX1278** is a 16-pin **Lora module** that communicates using SPI on 3.3V Logic. The Raspberry pi also operates in 3.3V logic level and also has in-built SPI port and 3.3V regulator. So we can directly connect the LoRa module with the Raspberry Pi.

The connections are pretty straight forward, only problem you might face is that the SX1278 is not breadboard compatible hence you have to use connecting wires directly to make the connections or use two small breadboards.



Raspberry Pi	Lora – SX1278 Module
3.3V	3.3V
Ground	Ground
GPIO 10	MOSI
GPIO 9	MISO
GPIO 11	SCK
GPIO 8	Nss / Enable
GPIO 4	DIO 0
GPIO 17	DIO 1
GPIO 18	DIO 2
GPIO 27	DIO 3
GPIO 22	RST

Arduino-LoRa



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

pyLoRa for Raspberry Pi

There are many python packages that you can use with LoRa. Also, commonly the Raspberry Pi is used as a LoWAN to get data from multiple LoRa nodes. But, in this project our aim to do Peer to Peer communication between two Raspberry Pi modules or between a Raspberry Pi and an Arduino.

It has a LoRa Arduino and LoRa Raspberry-Pi modules which can be used on the Arduino and the Raspberry Pi environment.

Code and Output:

Displayed in the PPT itself

Network Scanner

Introduction:

In this implementation, we have written a python script that gives us the MAC Address and the IP Address of all the devices(except the PI) connected to the network to which our Raspberry Pi is connected.

It returns nothing but the ARP (Address Resolution Protocol) Table.

There is no TTL (Time to Live) data entry because there is no 'ttl' attribute which can be accessible from the received ARP Response object in Scapy (library used).

Scapy is a Python program that enables the user to send, sniff and dissect and forge network packets. This capability allows construction of tools that can probe, scan or attack networks.

In other words, Scapy is a powerful interactive packet manipulation program. It is able to forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more. Scapy can easily handle most classical tasks like scanning, tracerouting, probing, unit tests, attacks or network discovery. It can replace hping, arpspoof, arp-sk, arping, p0f and even some parts of Nmap, tcpdump, and tshark.

Connection (Software Implementation)

References

https://books.google.co.in/books?hl=en&lr=&id=BAGiEAAAQBAJ&oi=fnd&pg=PA271&dq=peer+to+peer+communication+using+lora+circuit+digest&ots=k4gZEtGFdz&sig=FuJJhEBNjGOy3-FNIZokdQX7BJU

https://rb.gy/tx9d7

https://ieeexplore.ieee.org/abstract/document/10060496/

https://rb.gy/mn7xh

https://scapy.net/

https://rb.gy/gisqp

THANK YOU!!