How-To-Guide

Wireless Communication b/w two Arduino boards using nRF24L01

By Group: Protocol Phantom (Rohit Bhardwaj, Riyan Roy, Husain Kanthawala)

ENSC 351

FALL 2023

This document guides the user through:

1. Understanding the nRF24L01 module
2. Bi-directional Wireless Connection with two nRF24L01 and Arduino

Table of Contents

1. Understanding the nRF24L01 module……..…….……..2-3
    - 1.1 Introduction……..2
    - 1.2 nRF24L01 transceiver module…….2
    - 1.3 nRF24L01 Pin Information……..3
    - 1.4 Connecting nrF24L01 to Arduino……..……..3

2. Understanding the nRF24L01 module……..…….……..4-5
    - 2.1 Installing RF24 Library…….....4
    - 2.2 Transmitter Code……...4
    - 2.3 Receiver Code……... 5
    - 2.4 Troubleshooting…….5

3. References…….…….……..6

1.1 Introduction

Wireless communication between Arduino boards using the nRF24L01 module opens up a realm of possibilities for remote sensing, home automation, and more. This step-by-step guide will walk you through the process of establishing a reliable wireless link between two Arduino boards using the nRF24L01 transceiver modules. By the end of this guide, you will have a working wireless communication system.

Materials Required

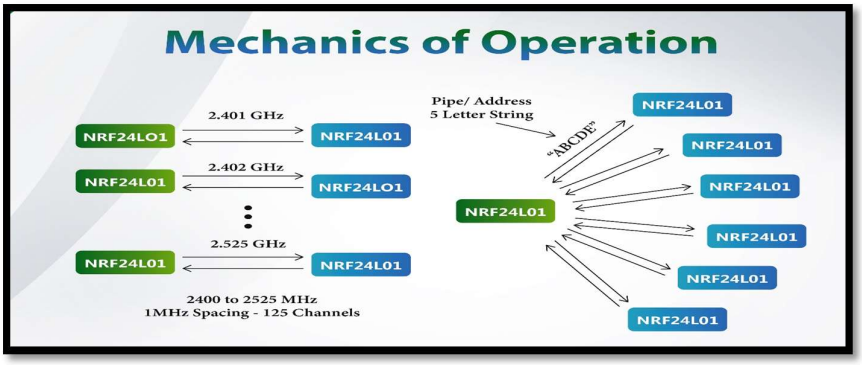
- Two Arduino Boards
- Two nRF24L01 transceiver modules
- Breadboard
- Jump wires
- Power supply (If needed)

1.2 Understanding nRF24L01 transceiver module

The nRF24L01 is a single-chip 2.4 GHz transceiver module with an embedded protocol engine. It can operate with band rates from 250 kbps up to 2 Mbps.

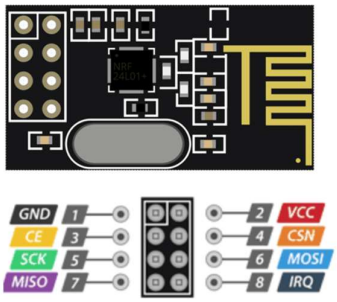
Working:

The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time. This module uses even less power than a single LED while transmitting. The module has an operational voltage range of 1.9V to 3.6V, thus connecting it to an Arduino is simple and doesn't require the use of logic-level converters.



1.3 nRF24L01 Pin Information

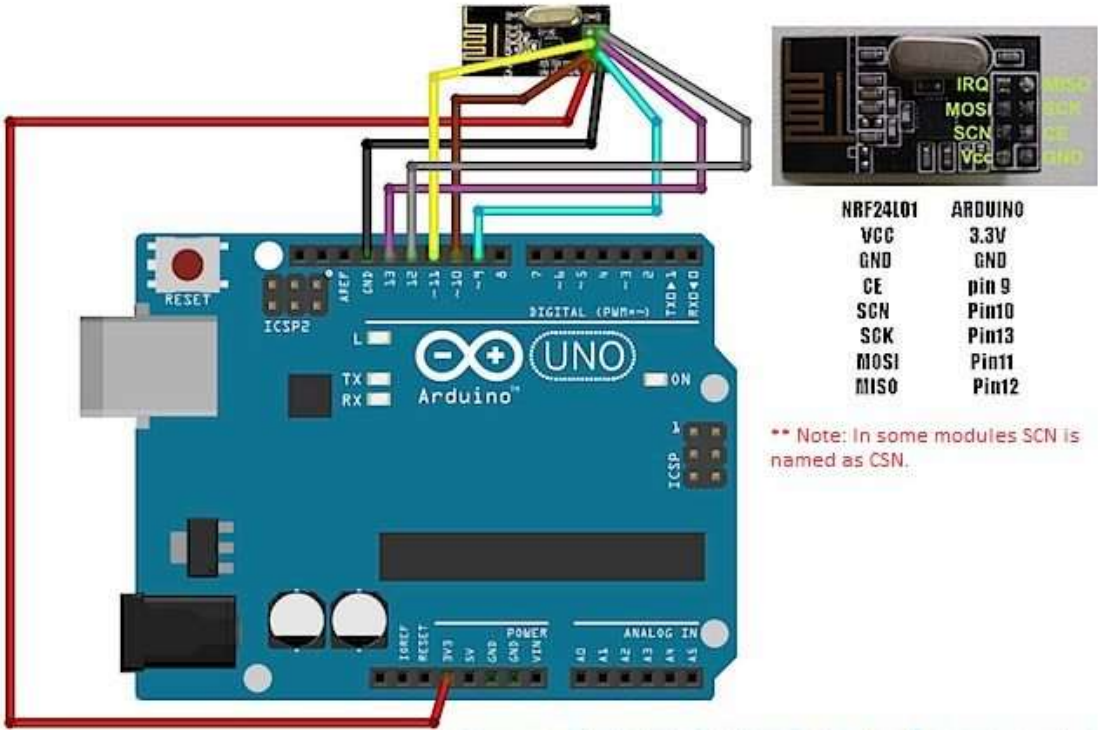
It should be noted that each Arduino board has distinct SPI pins. Three of these pins are for SPI communication, and they must be linked to the Arduino's SPI pins. The pins CSN and CE are used to place the module in standby or active mode and to switch between transmit and command mode. They may be linked to any digital pin on the Arduino board.



1.4 Connecting nRF24L01 to Arduino

- Connect the VCC pin of the nRF24L01 to the 3.3V output on the Arduino.
- Connect the GND pin to the ground (GND) on the Arduino.
- Connect CE, CSN, MOSI, MISO, and SCK to the respective digital pins on the Arduino.

Make sure the connections are secure and double-check the wiring.



2.1 Installing the RF24 Library

- Open the Arduino IDE.
- Navigate to Sketch > Include Library > Manage Libraries.
- In the Library Manager, search for "RF24" and install the "RF24" library by TMRh20.

This library is essential for handling communication between the Arduino boards and the nRF24L01 modules.

Establishing Bi-Directional Wireless Connection

2.2 Transmitter Code

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#define led 12

RF24 radio(7, 8); // CE, CSN
const byte addresses[][6] = {"00001", "00002"};
boolean buttonState = 0;

void setup() {
  pinMode(12, OUTPUT);
  radio.begin();
  radio.openWritingPipe(addresses[1]); // 00002
  radio.openReadingPipe(1, addresses[0]); // 00001
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);

  radio.stopListening();
  int potValue = analogRead(A0);
  int angleValue = map(potValue, 0, 1023, 0, 180);
  radio.write(&angleValue, sizeof(angleValue));

  delay(5);
  radio.startListening();
  while (!radio.available());
  radio.read(&buttonState, sizeof(buttonState));
  if (buttonState == HIGH) {
    digitalWrite(led, HIGH);
  }
  else {
    digitalWrite(led, LOW);
  }
}
```

2.3 Receiver Code

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>

#define button 4

RF24 radio(7, 8); // CE, CSN
const byte addresses[][6] = {"00001", "00002"};
Servo myServo;
boolean buttonState = 0;

void setup() {
  pinMode(button, INPUT);
  myServo.attach(5);
  radio.begin();
  radio.openWritingPipe(addresses[0]); // 00001
  radio.openReadingPipe(1, addresses[1]); // 00002
  radio.setPALevel(RF24_PA_MIN);
}

void loop() {
  delay(5);
  radio.startListening();
  if ( radio.available()) {
    while (radio.available()) {
      int angleV = 0;
      radio.read(&angleV, sizeof(angleV));
      myServo.write(angleV);
    }
    delay(5);
    radio.stopListening();
    buttonState = digitalRead(button);
    radio.write(&buttonState, sizeof(buttonState));
  }
}
```

2.4 Troubleshooting

1. **Address Mismatch**
   Ensure that the addresses in the code match on both the transmitter and receiver.
   Update the addresses and upload the code again.

2. **Power Supply Issues**
   Double-check the power supply to the nRF24L01 modules. Consider using an external power supply for both Arduino boards.

3. **Range Issues**
   Increase the distance between the transmitter and receiver gradually.
   Check if obstacles are affecting the signal.

3 References

- https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/
- https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf