LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

TOPIC-Communication Between Devices Using RF Modules

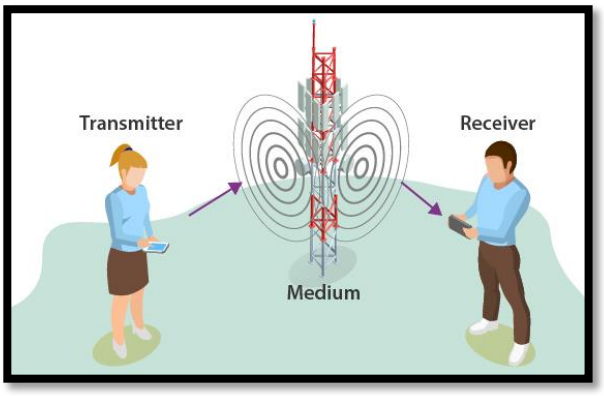

Figure 1-Communication System

SUBMITTED BY:

Name: Siddharth Mehrotra **Registration Number**:12006050

Name: Palvai Manoj Kumar **Registration Number:** 12009417

Name: Reetesh Kumar **Registration Number:** 12010632

Name: Shivam Rai **Registration Number:** 12018182

Section: E2001

School: School of Electronics and Electrical Engineering

Course Code & Name: ECE321, Communication System Laboratory

SUBMITTED TO:

Dr. Manwinder Singh

Professor at School of Electronics and Electrical Engineering

TABLE OF CONTENTS

List Of Figures

1. Introduction

1.1 RF Communication-

RF (Radio Frequency) communication refers to the wireless transmission of information through the use of electromagnetic signals at radio frequencies. This technology is used in a variety of applications, including television and radio broadcasting, wireless networking, cellular communication, and satellite communication.

In RF communication, information is encoded onto a high-frequency carrier wave, which is then transmitted through the air via antennas. The receiver then decodes the information from the received signal.

RF communication can be used over short distances, such as in Bluetooth and Wi-Fi, or over long distances, such as in satellite communication. It is also used in many IoT (Internet of Things) devices to enable wireless communication between sensors, controllers, and other devices.

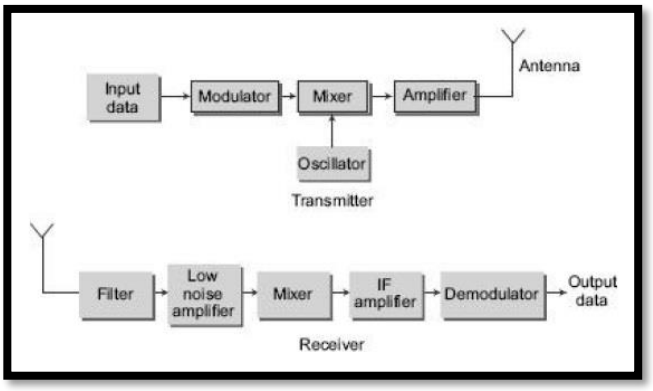

Figure 2-RF Communication Flow Chart

LOVELY PROFESSIONAL UNIVERSITY

2. Circuit Diagram

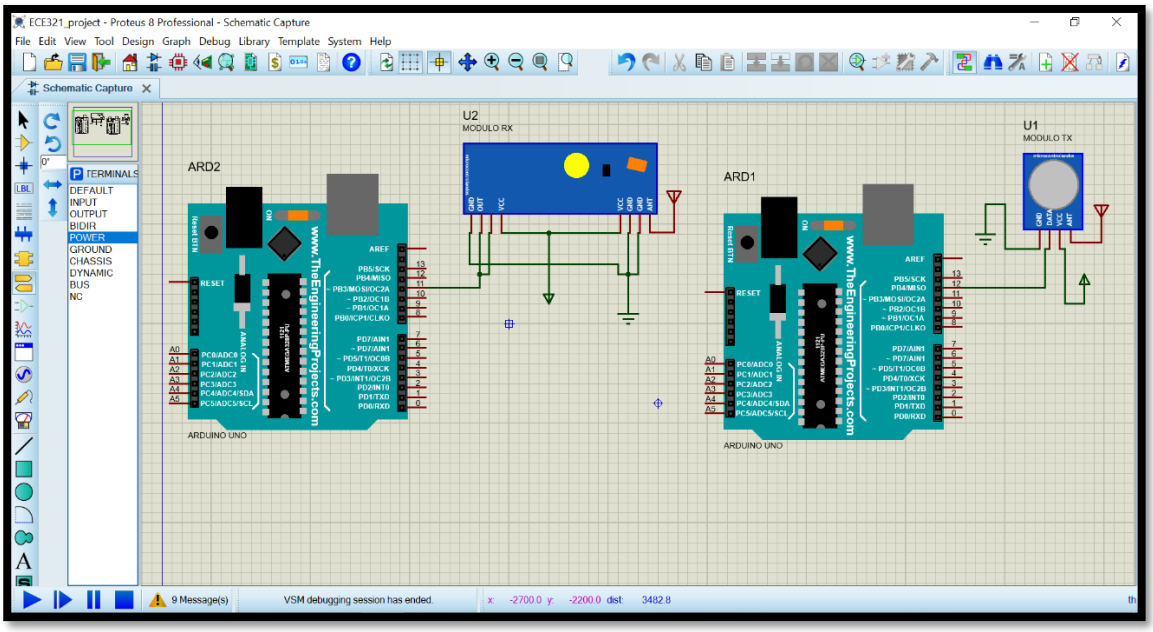

Figure 3-Circuit Diagram

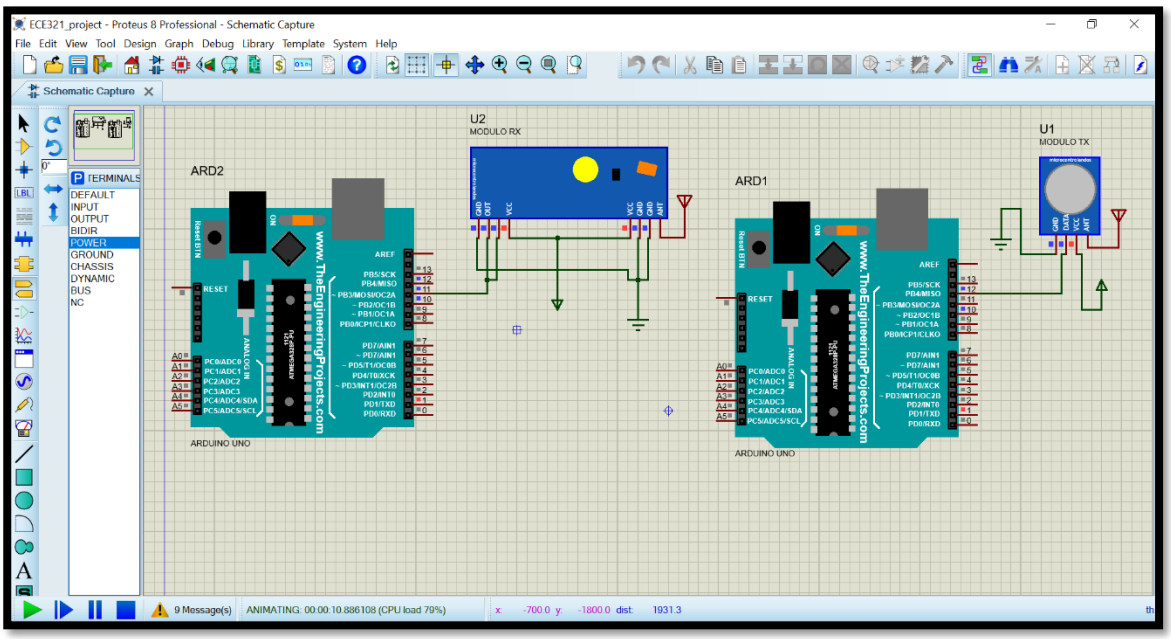

Figure 4-Circuit in Simulation

3. List Of Components

3.1 Arduino Uno-

Arduino Uno is a popular microcontroller board based on the ATmega328P microcontroller, which is designed to make it easy for hobbyists and professionals to create interactive projects. It has a simple and easy-to-use interface, and a wide range of input and output pins, which can be used to connect to a variety of sensors, actuators, and other components.

Arduino Uno boards are typically programmed using the Arduino Integrated Development Environment (IDE), which provides a user-friendly environment for writing, compiling, and uploading code to the board. With the Arduino Uno, you can build a variety of projects such as robots, smart home devices, interactive art installations, and more.

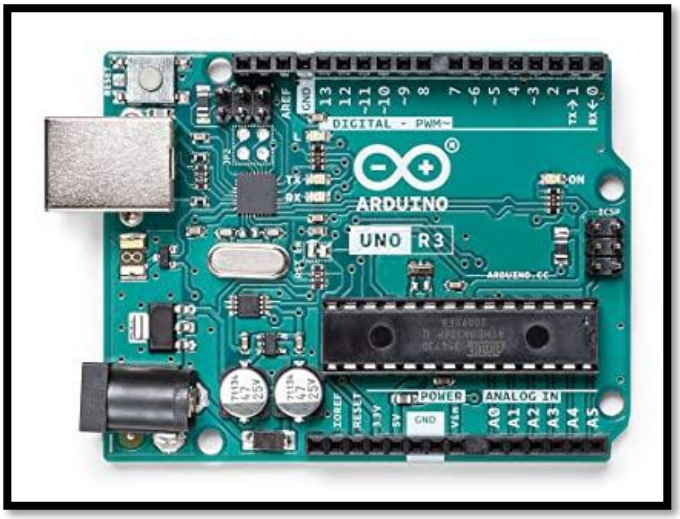

Figure 5-Arduino Uno

Pin Description-

Vin: This is the input voltage pin of the Arduino board used to provide input supply from an external power source.

5V: This pin of the Arduino board is used as a regulated power supply voltage and it is used to give supply to the board as well as onboard components.

3.3V: This pin of the board is used to provide a supply of 3.3V which is generated from a voltage regulator on the board

GND: This pin of the board is used to ground the Arduino board.

Reset: This pin of the board is used to reset the microcontroller. It is used to Resets the microcontroller.

Analog Pins: The pins A0 to A5 are used as an analog input and it is in the range of 0-5V.

Digital Pins: The pins 0 to 13 are used as a digital input or output for the Arduino board.

Serial Pins: These pins are also known as a UART pin. It is used for communication between the Arduino board and a computer or other devices. The transmitter pin number 1 and receiver pin number 0 is used to transmit and receive the data resp.

External Interrupt Pins: This pin of the Arduino board is used to produce the External interrupt and it is done by pin numbers 2 and 3.

PWM Pins: This pin of the board is used to convert the digital signal into an analog by varying the width of the Pulse. The pin numbers 3,5,6,9,10 and 11 are used as a PWM pin.

SPI Pins: This is the Serial Peripheral Interface pin, it is used to maintain SPI communication with the help of the SPI library.

LED Pin: The board has an inbuilt LED using digital pin-13. The LED glows only when the digital pin becomes high.

AREF Pin: This is an analog reference pin of the Arduino board. It is used to provide a reference voltage from an external power supply.

3.2 RF Transmitter-

RF (Radio Frequency) transmitter is an electronic device that is designed to transmit radio signals wirelessly from one location to another. It typically consists of a radio frequency oscillator, an amplifier, and an antenna.

The RF transmitter generates high-frequency electromagnetic waves, which are then picked up by the antenna and transmitted into the air. These radio waves can travel through the air, and can be picked up by a receiver tuned to the same frequency.

RF transmitters are commonly used in wireless communication systems, such as remote control systems, wireless sensor networks, and mobile phones. They are also used in various other applications, including wireless audio and video transmission, telemetry, and satellite communication.

433MHz RF Transmitter is one of the best and most efficient RF modules. It is simple to use and easy to implement. It is the first part of 433MHz RF communication. This module uses a serial data input and transmits it to the 3 Meter range but the range is extendable up to 100

meters. The transmitter doesn't operate on its own. It uses ASK (Amplitude Shift Keying) which makes it easy to interface with any microcontroller.

433MHz RF Transmitter Module consists of 4-pins which makes it operatable. Pins are:

- **VCC:** This is the power input pin of the RF module to make it functional.
- **GND:** Ground pin helps the module to make the common ground with the controllers and encoder/decoders.
- **ATAD:** This is the data pin of the transmitter. It takes the data from the input device in encoded form and broadcast it.
- **ANT:** The antenna pin is not necessary to use. The module can operate max 3 meters without antenna but its range is also extendable up to 100 meters by using a small wire as an Antena.

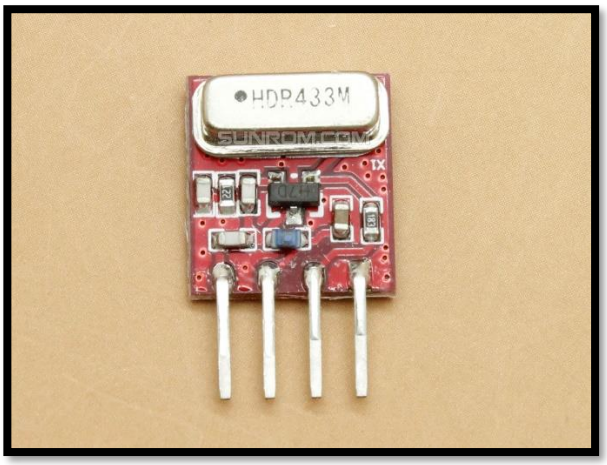

Figure 6-RF Transmitter

3.3 RF Receiver-

433MHz RF receiver module is the most common and cheapest RF receivers for commercial use. This module helps to receiver the 433MHz frequency commonly but it also has a node on it which is changeable by users for different frequencies. The variable node can adjust the frequency from 315MHz to 433MHz. The module can receive the data but it is unable to decode and view the data from the receiver without using any third device. Most of the time smart boards and microcontrollers interface with the receiver. The output of the module comes in an encoded form that is decodable through programming and encoders. On the other hand, it works well with the 433MHz RF transmitter module.

The RF receiver picks up the radio waves transmitted by the RF transmitter using its antenna, and amplifies the signal to a usable level. The mixer and local oscillator are then used to convert the high frequency RF signal to a lower frequency signal, which can be processed by the demodulator to extract the original information signal. The audio amplifier then amplifies the audio signal for output to a speaker or other audio device.

RF receivers are used in a wide range of applications, including wireless communication systems, such as remote control systems, wireless sensor networks, and mobile phones. They are also used in various other applications, such as wireless audio and video transmission, telemetry, and satellite communication.

33MHz RF receiver has 6 pins, which offers 4 types of functions. The Pins are:

- **VCC Pin:** VCC is the power input pin for the RF module. The power will active the internal circuit to make it functional.
- **GND Pin:** For common ground, the RF module has only one ground pin. The module needs to use with other devices and the common ground will help the RF module to interface with external devices.
- **Data Pin:** 433MHz RF Module has two data input pins which are internally common with each other. Only data should receive from one pin at a time.
- **Antenna Pin:** This module has an antenna pin which helps to connect the external wire to extend the range up to 100 meters. The size of the antenna will depend on the operating frequency.

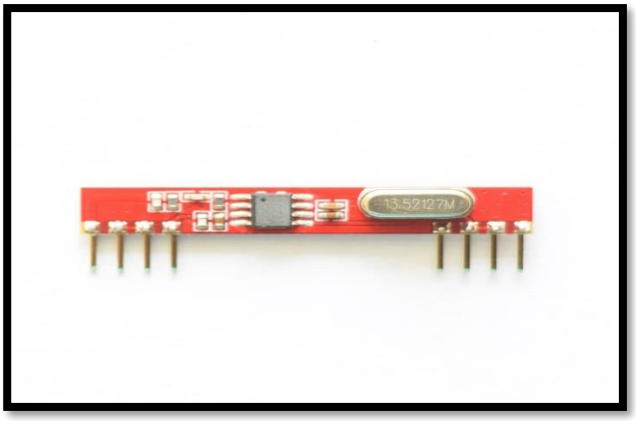

Figure 7-RF Receiver

3.4 Jumper Wires-

Jumper wires are a type of electrical wire that are used to make connections between different components in electronic circuits. They are typically made of stranded or solid core wire, with connectors at each end that can be inserted into the pins or sockets of electronic components such as sensors, actuators, and microcontrollers.

Jumper wires come in a variety of lengths, colors, and connectors, and can be easily bent and manipulated to fit into tight spaces on a breadboard or circuit board. They are commonly used in prototyping and experimenting with electronic circuits, and are an essential tool for anyone working in the field of electronics.

Jumper wires can be male-to-male, female-to-female, or male-to-female, depending on the types of connectors at each end. They can also be insulated or uninsulated, and can be made of different materials, such as copper, aluminum, or steel.

Figure 8-Jumper Wires

3.5 Breadboard-

A breadboard is a reusable device used to build and test electronic circuits without the need for soldering. It is a flat board made of plastic or other insulating materials, with a grid of holes or contacts arranged in a pattern that allows for easy insertion and connection of electronic components such as resistors, capacitors, transistors, and integrated circuits.

Breadboards are designed to allow quick and easy prototyping of circuits, as components can be inserted into the board and connected with jumper wires without the need for any additional tools or equipment. They are also reusable, so components can be easily removed and replaced, allowing for rapid iteration and experimentation.

Breadboards come in various sizes and configurations, with different numbers of holes and contacts, and may include additional features such as power rails for supplying voltage to the circuit. They are widely used in education, research, and hobbyist electronics, and are an essential tool for anyone working in the field of electronics.

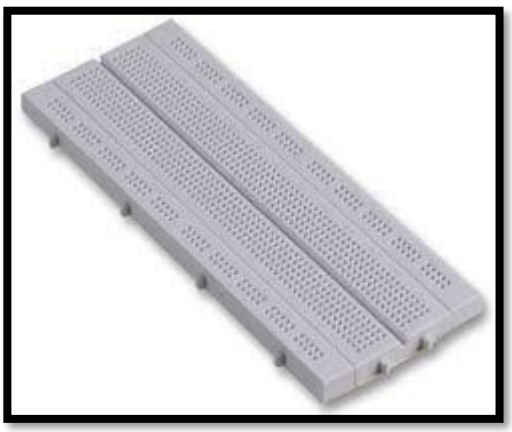

Figure 9-Breadboard

3.6 Software Used-

Arduino IDE (Integrated Development Environment) is a software application used to program and upload code to Arduino microcontroller boards. It is a user-friendly interface that allows developers to write, edit, and compile code for their Arduino boards, as well as upload the code to the board.

The Arduino IDE includes a text editor with syntax highlighting and code completion features, a compiler, and a serial monitor for debugging and testing. It also includes a library manager for easy installation and management of libraries, which are pre-written code modules that can be used to extend the functionality of the Arduino board.

The Arduino IDE supports a wide range of programming languages, including C and C++, and is compatible with various operating systems such as Windows, Mac OS, and Linux. It is open source software, meaning that anyone can modify and contribute to the code.

The Arduino IDE is an essential tool for anyone working with Arduino boards, and makes it easy to create a wide range of projects, from simple LED blinkers to complex robotics projects.

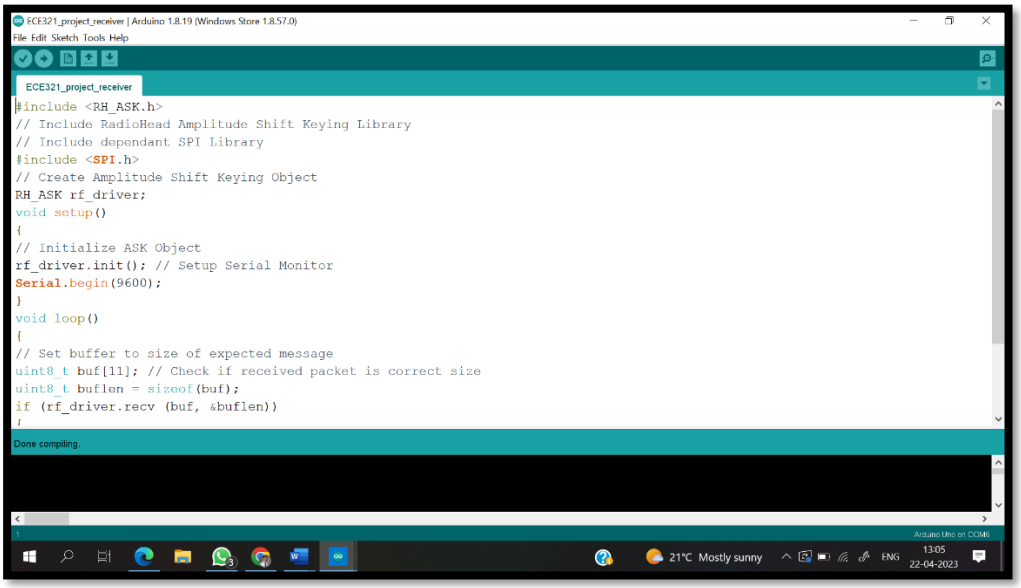

Figure 10-Receiver Code in Arduino IDE

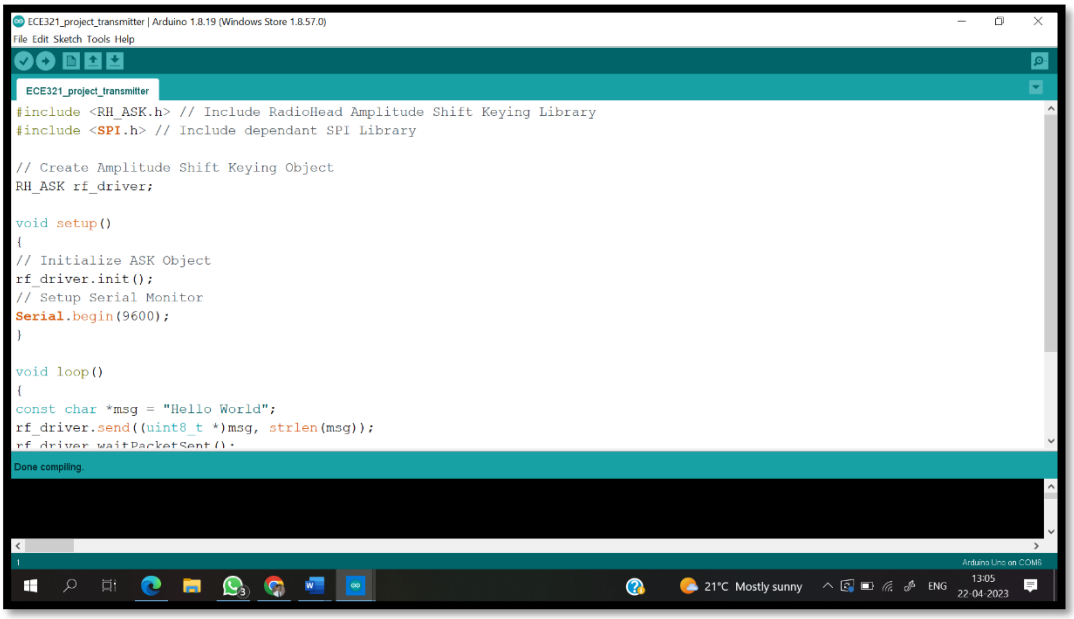

Figure 11-Transmitter Code in Arduino IDE

4. Arduino Code

4.1 Code for RF transmitter-

```
#include <RH_ASK.h> // Include RadioHead Amplitude Shift Keying Library
#include <SPI.h> // Include dependant SPI Library
// Create Amplitude Shift Keying Object
RH_ASK rf_driver;
void setup()
{
        // Initialize ASK Object
        rf_driver.init();
        // Setup Serial Monitor
        Serial.begin(9600);
}

void loop()
{
        const char *msg = "Hello World";
        rf_driver.send((uint8_t *)msg, strlen(msg));
        rf_driver.waitPacketSent();
        {
                // Message Transmitted
                Serial.println("Message Transmitted: ");
                delay(1000);
        }
}
```

4.2 Code for RF Receiver-

```
#include <RH_ASK.h>
// Include RadioHead Amplitude Shift Keying Library
// Include dependant SPI Library
#include <SPI.h>
// Create Amplitude Shift Keying Object
RH_ASK rf_driver;
void setup()
{
        // Initialize ASK Object
        rf_driver.init(); // Setup Serial Monitor
        Serial.begin(9600);
}
void loop()
{
        // Set buffer to size of expected message
        uint8_t buf[11]; // Check if received packet is correct size
        uint8_t buflen = sizeof(buf);
        if (rf_driver.recv (buf, &buflen))
        {
                // Message received with valid checksum
                Serial.print("Message Received: ");
                Serial.println((char*) buf);
        }
}
```

5. Working

5.1 General Working-

1. Connect the 433 MHz RF transmitter module to the Arduino board. Connect the data pin of the transmitter module to a digital pin on the Arduino board.
2. Connect the 433 MHz RF receiver module to the Arduino board. Connect the data pin of the receiver module to another digital pin on the Arduino board.
3. Write the code to transmit data using the RF transmitter module. The code should send a signal with a unique code to identify the transmitter and the data to be transmitted.
4. Write the code to receive data using the RF receiver module. The code should check for the unique code and extract the data from the received signal.
5. Test the transmitter and receiver by sending and receiving data between them.
6. Optionally, connect LEDs and resistors to the Arduino board and use them to indicate the transmission and reception of data.

The project uses an Arduino board along with a 433 MHz RF transmitter module and a 433 MHz RF receiver module to wirelessly transmit and receive data.

The RF transmitter module is connected to the Arduino board by connecting its VCC pin to the 5V pin on the Arduino board, its GND pin to the GND pin on the Arduino board, and its data pin to a digital pin on the Arduino board (for example, pin 12).

Similarly, the RF receiver module is connected to the Arduino board by connecting its VCC pin to the 5V pin on the Arduino board, its GND pin to the GND pin on the Arduino board, and its data pin to another digital pin on the Arduino board (for example, pin 11).

The transmitter code is written to transmit data wirelessly using the RF transmitter module. First, the digital pin used to connect the transmitter module is defined as an output pin using the pinMode() function. Then, a unique code for the transmitter is defined using an array of characters or integers (for example, "A1B2C3"). The data to be transmitted is defined using a variable or an array of values. The RF transmitter library is used to send the data using the send() function. The unique code and data are passed as arguments to the send() function.

The receiver code is written to receive the data transmitted by the transmitter using the RF receiver module. First, the digital pin used to connect the receiver module is defined as an input pin using the pinMode() function. Then, the unique code for the transmitter is defined as an array of characters or integers (for example, "A1B2C3"). The RF receiver library is used to receive the data using the recv() function. The recv() function returns the number of bytes received. If the number of bytes received is greater than zero, the received data is checked to see if it matches the unique code of the transmitter. If it does, the data is extracted from the received signal.

Finally, the transmitter and receiver are tested by sending and receiving data wirelessly between them. Optionally, LEDs and resistors can be connected to the Arduino board to indicate the transmission and reception of data.

5.2 ASK modulation in the RF modules-

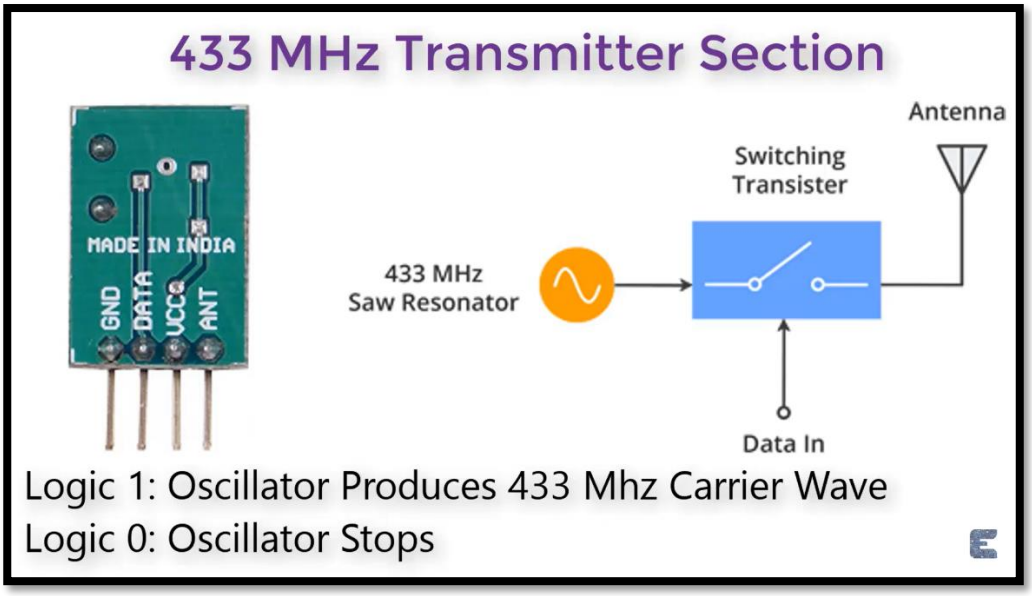

Figure 12-RF Transmitter Signal Generation

The 433 MHz RF transmitter has a RF saw resonator (oscillator) which transmits 433 Mhz frequency at logic 1 and the oscillator stops at logic 0. It also has a switching transistor which has one input as data in and the output of this goes to an antenna.

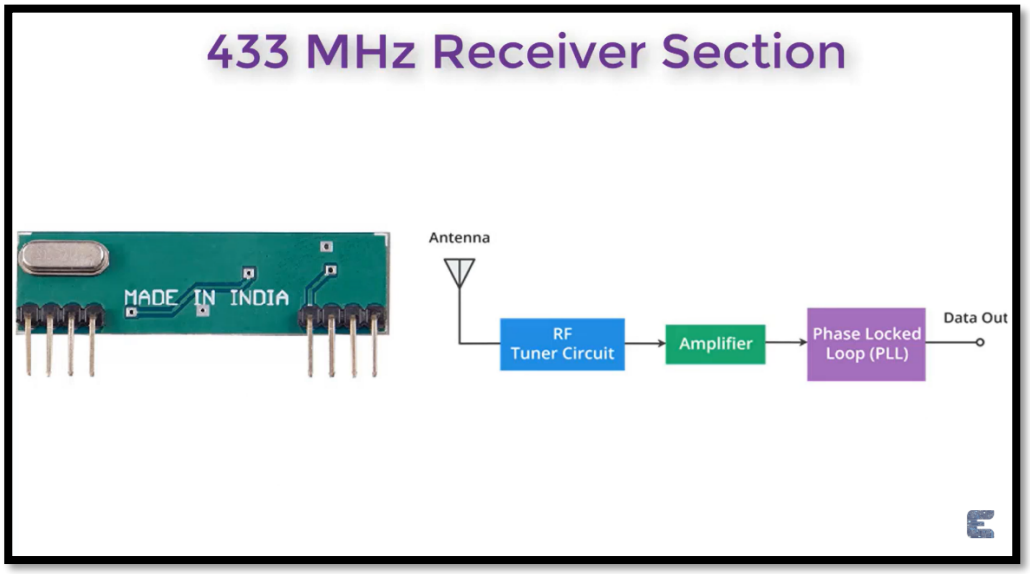

Figure 13-Signal Receiving in the RF Receiver

The receiver module has an antenna which receives the signa and is transferred to RF tuner circuit and a couple of op amps to amplify the signal. the amplified signal is then transferred to a phase locked loop which lock onto a digital stream of bits which gives better decoded output and noise immunity. These modules ASK to send the data over.

5.3 ASK Modulation-

ASK modulation is a method of transmitting digital signals by varying the amplitude of the radio frequency carrier wave. In ASK modulation, a digital signal with two discrete levels (0 and 1) is used to modulate the amplitude of the carrier wave.

A logic level 1 corresponds to a high amplitude carrier wave, and a logic level 0 corresponds to a low amplitude carrier wave. The receiver detects the changes in amplitude and converts them back into the original digital signal. ASK modulation is relatively simple and inexpensive to implement, making it a popular choice for low-cost RF applications.

However, ASK modulation is less efficient in terms of power and bandwidth compared to other modulation techniques, such as FSK (Frequency Shift Keying) or PSK (Phase Shift Keying). To use ASK modulation with a 433 MHz RF transmitter and receiver, the transmitter needs to modulate the carrier wave using the digital signal, while the receiver needs to demodulate the carrier wave to recover the original digital signal.

The Arduino can be used to generate the digital signal and control the transmitter and receiver accordingly.

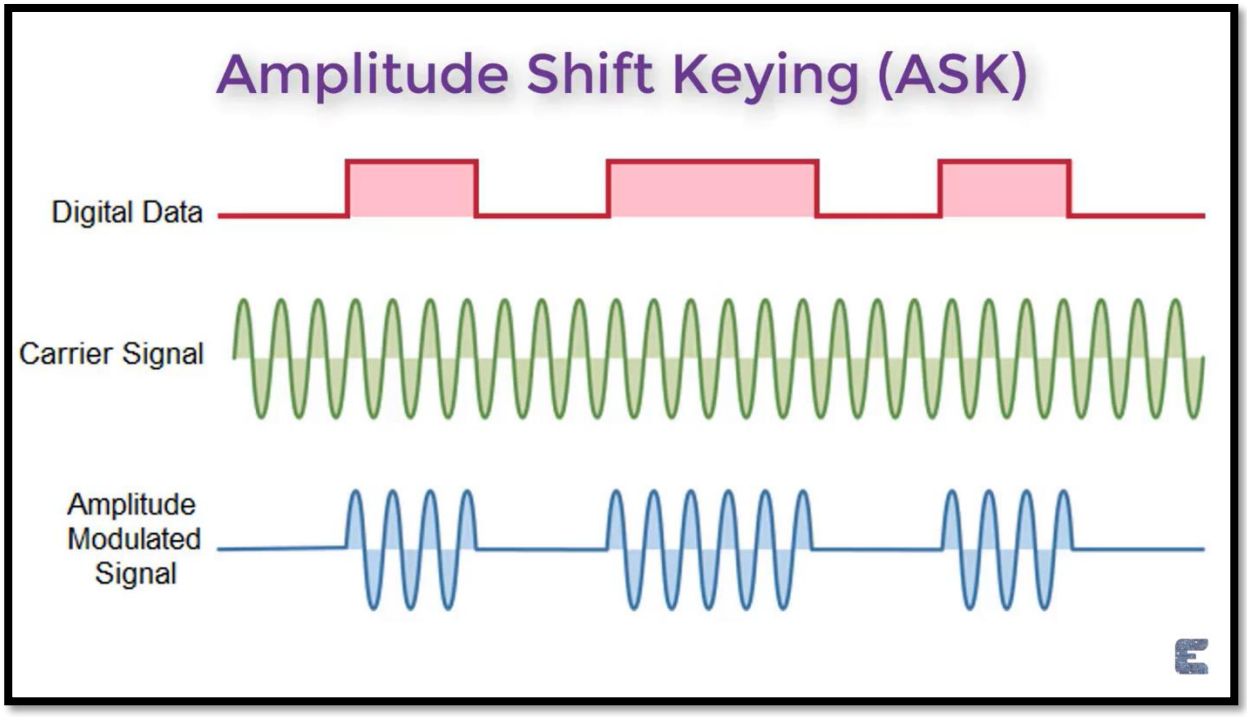

Figure 14-ASK Modulation

6. Result

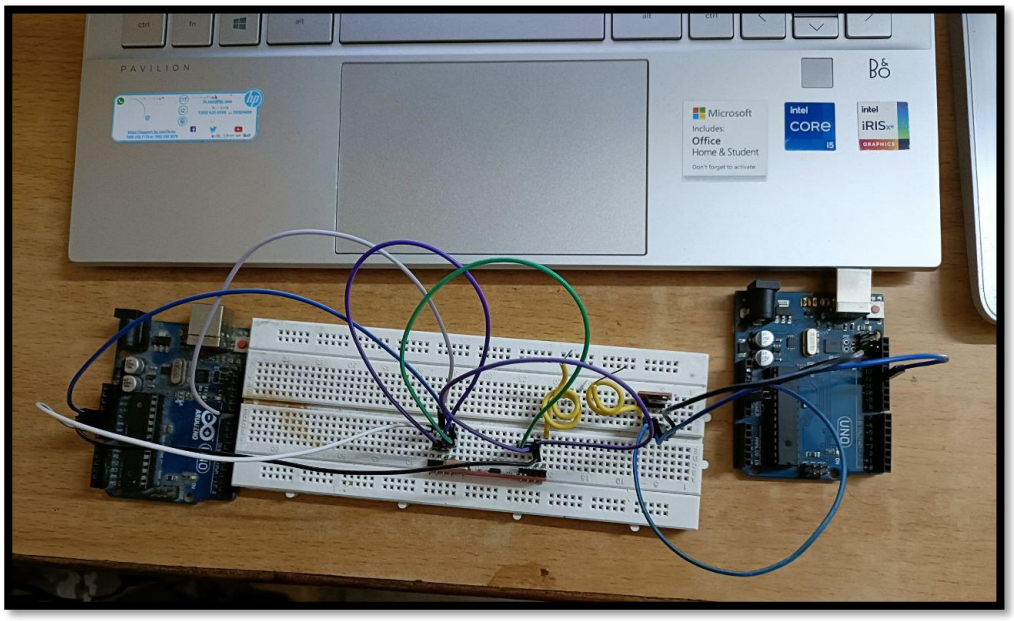

Figure 15-Hardware Circuit

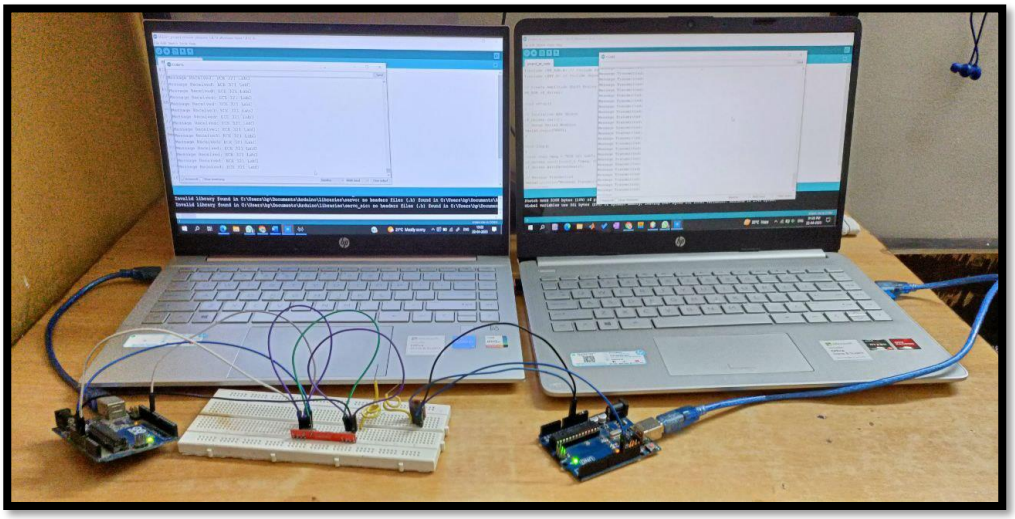

Figure 16-Communication between two devices

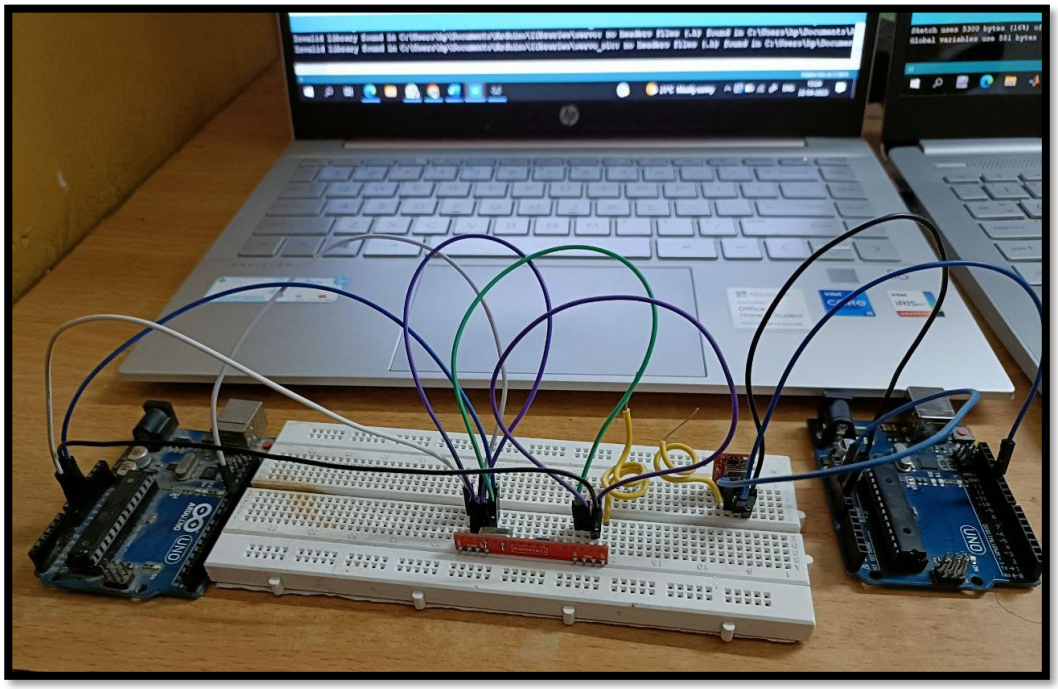

Figure 17-Hardware Circuit (side view)

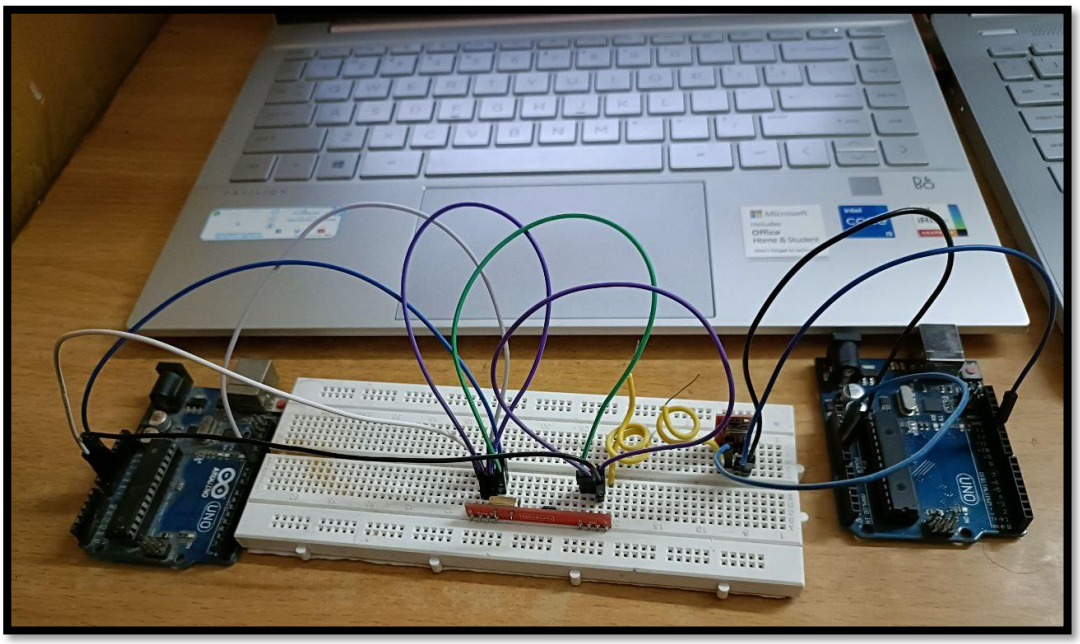

Figure 18-Hardware Circuit (top view)

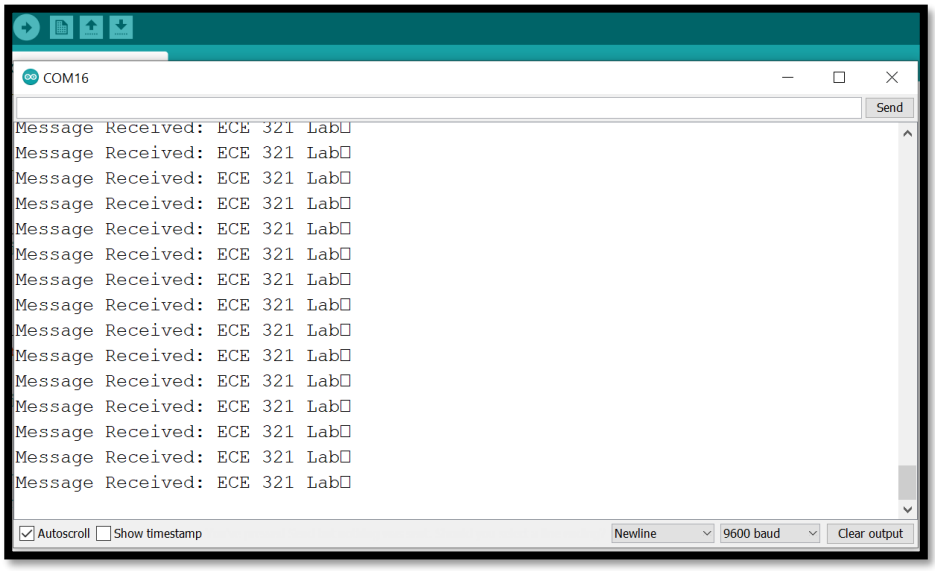

Figure 19-Serial Moniter for receiver

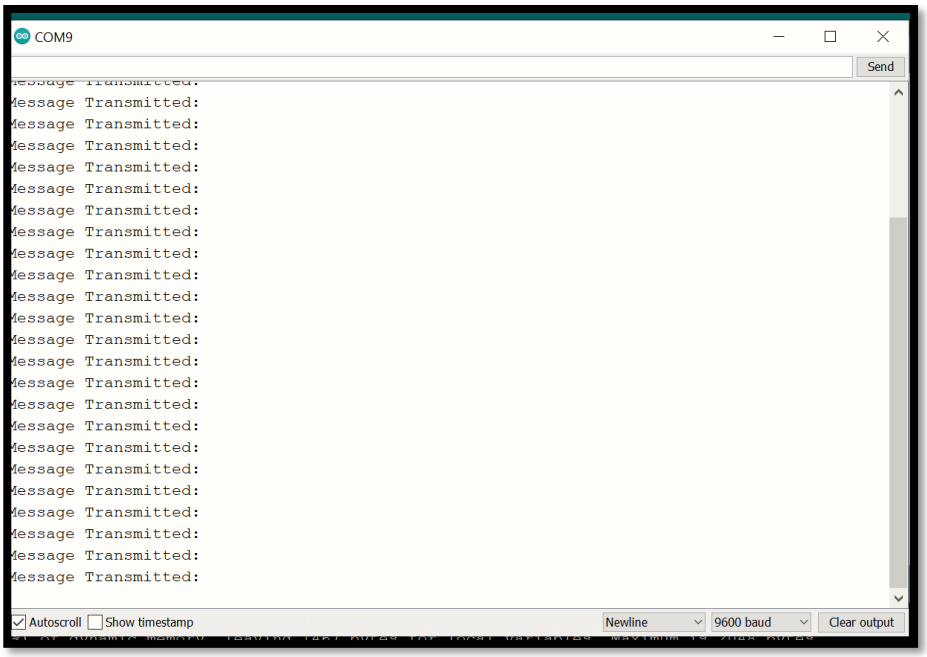

Figure 20-Serial Monitor for Transmitter

- By connecting the Arduinos to a computer via USB, we can view the data transmitted and received wirelessly using the Arduinos' serial port.
- The serial monitor in the Arduino IDE allows us to visualize and analyze the data transmitted and received, making it easier to debug and troubleshoot any issues in the communication.

7. Conclusion

The project has demonstrated the use of 433 MHz RF receiver and transmitter modules with Arduino to wirelessly transmit and receive data. The project can be used in various applications where wired communication is not feasible or convenient, such as remote sensing, home automation, and robotics.

The project has also shown how to use the VirtualWire library with Arduino to implement wireless communication. The library provides a simple and easy-to-use interface for transmitting and receiving data using RF modules.

In conclusion, the project has successfully achieved its goal of demonstrating wireless communication using 433 MHz RF receiver and transmitter modules with Arduino. The project can be further extended by adding more sensors and actuators, improving the transmission range, and implementing encryption and security features for data transmission. Overall, this project is a great starting point for anyone interested in wireless communication with Arduino and RF modules.