RSC Group Report

Task

COMMUNICATION BETWEEN TWO MICROCONTROLLERS USING RF MODULE (433MHZ RF MODULE)

Group Members:

- Kunal Pandurang Pawar
- Utkarsh Pratap Patil
- Krutika Pandya
- Himanshu Kamdi

RSC Group Report

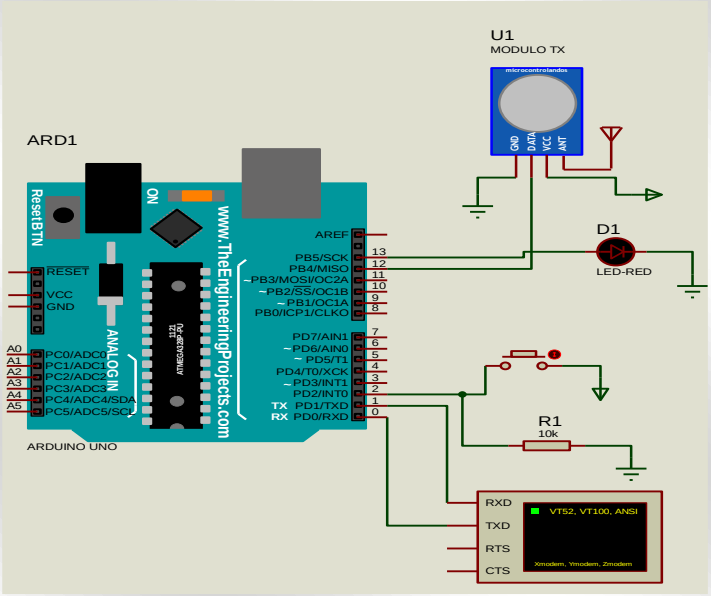
INTRODUCTION

Wireless Communication has become an essential part of human life like T.V. remote or radio communication. It's all about wireless transmission of data for our convenience, so that there is no hassle of wires without any contact with the device.

One of the easiest ways is to implement wireless communication using RF Module (Radio Frequency Module). Arduino on the other hand, is a low cost solution for microcontroller application which uses simple programming and hardware components.

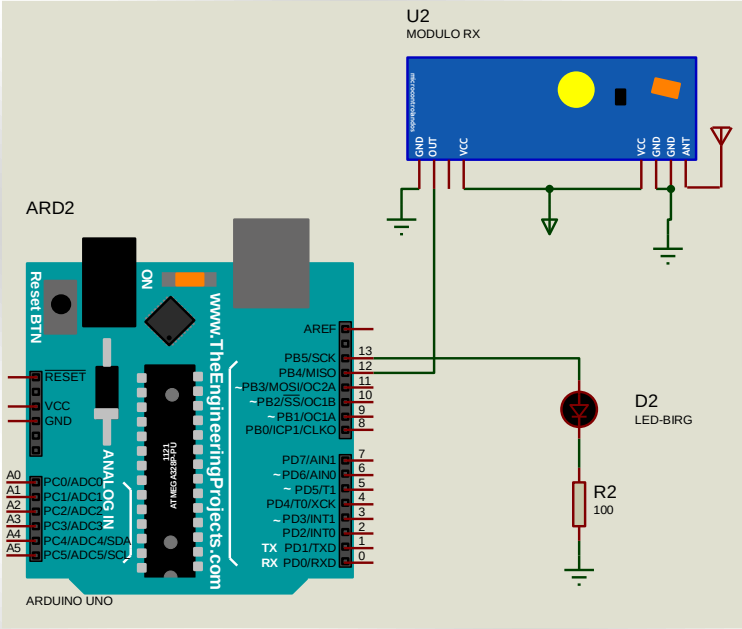
CIRCUIT DESCRIPTION

So, for our task we made use of two Arduino boards to make a system to communicate with each other using RF Module. As this is a wireless communication project, our circuit consists of a **Transmitter** part and a **Receiver** part.



Transmitter Part of Circuit

RSC Group Report



Receiver Part of Circuit

Transmitter part consists of:

- Arduino UNO R3
- 433 MHz Transmitter module
- LED (Red)
- 10 KΩ resistor
- Bread Board
- Power Supply
- Virtual Terminal

Receiver part consists of:

- Arduino UNO R3
- 433 MHz Receiver module
- LED(Green)
- 100 Ω resistor
- Bread board
- Power supply

SOURCE CODE DESCRIPTION

Transmitter_Module | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

Transmitter_Module

```
#include <VirtualWire.h> /*importing VirtualWire library that provides features
                          to send or recieve short messages, without addressing, retransmit or acknowledgment,*/
char *control;  //intialising character variable
const int pinX = 2; // intialising an integer variable who's value is constant throughtout the code
int buttonA = 0; //  intialising an integer variable

void setup() // setup code runs once:
{
Serial.begin(9600);
pinMode(pinX, INPUT);//function telling the Aduino, which pin is used and it is for input (button pressed or not )
pinMode(13, OUTPUT); //function telling the Aduino, which pin is used and it is for output (signal sent is visible)
vw_set_ptt_inverted(true); /*"Configure the push to talk(PPT)" polarity. PPT is a two-way communication method that uses half-duplex.
                            To use PTT, users must press a button on the PTT device while sending signal, then release it when done.*/
                           // Required for DR3100
vw_set_tx_pin(12); //Configuring the transmit pin.(By default 12)
vw_setup(4000); //Speed of data transfer bps

delay(100); // Pauses the program for some amount of time so we can have proper visibility
}

void loop() // loop code runs repeatedly:
{
  buttonA = digitalRead(pinX); //Reads the value from a specified digital pin, either HIGH or LOW.
  Serial.println(control); //Prints data to the serial port as human-readable ASCII text.

 if (buttonA == HIGH) // checking if the input is high
{
 control=(char*)'1' ;
vw_send((uint8_t *)control, strlen(control)); //Transmits a message."message" value [type casting char to integer] and "length" is the bytes occupied by char control
digitalWrite(13,LOW); //The function is used to write a LOW value to a digital pin.[LED OF TRANSMITTER DOES NOT GLOW]
```

Done compiling

Transmitter_Module | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

Transmitter_Module

```
  Serial.println(control); //Prints data to the serial port as human-readable ASCII text.

 if (buttonA == HIGH) // checking if the input is high
{
 control=(char*)'1' ;
vw_send((uint8_t *)control, strlen(control)); //Transmits a message."message" value [type casting char to integer] and "length" is the bytes occupied by char control
digitalWrite(13,LOW); //The function is used to write a LOW value to a digital pin.[LED OF TRANSMITTER DOES NOT GLOW]
}
 else //checking if the input is not high
{
 control=(char*)'0';
vw_send((uint8_t *)control, strlen(control));//Transmits a message. "message" value [type casting char to integer] and "length" is the bytes occupied by char control
vw_wait_tx();  // Wait until the whole message is gone
digitalWrite(13, HIGH); //The function is used to write a HIGH value to a digital pin.[LED OF TRANSMITTER GLOWS]
 }
}
```

Done compiling

Receiver_Module | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

Receiver_Module §

```
#include <VirtualWire.h>  /*importing VirtualWire library that provides features
                          to recieve or  send short messages, without addressing, retransmit or acknowledgment,*/
void setup() // all the code inside the parenthesis will only execute once
{
vw_set_ptt_inverted(true); /*"Configure the push to talk(PPT)" polarity. PPT is a two-way communication method that uses half-duplex.
                           To use PTT, users must press a button on the PTT device while sending signal, then release it when done.*/
                          // Required for DR3100
vw_set_rx_pin(12); // Configure the receiver pin so data can connect to 12 pin
vw_setup(4000); // Begin using all settings and initialize the virtual wire library. All operations will take place at 4000 speed bits per second
Serial.begin(9600); /* sets the baud rate (signifies the data rate in bps)for serial data communication.
                    The default baud rate in Arduino is 9600 bps*/
                    // same as that of transmitter
pinMode(13, OUTPUT); //function telling the Aduino, which pin is used and it is for output (signal recieved is visible)
vw_rx_start(); /* Activate the receiver process and Start the receiver phase lock loop
               (generates an output signal whose phase is related to the phase of an input signal)*/
}
void loop()//all the code inside the parenthesis will execute multiple times
{
uint8_t buf[VW_MAX_MESSAGE_LEN]; //creating "buf", an array where the message is copied.[integer array]
uint8_t buflen = VW_MAX_MESSAGE_LEN; //"buflen"[integer variable]have the array's max size upon input,and upon return the number of bytes actually copied is retured
                                  //VW_MAX_MESSAGE_LEN  Maximum number of bytes in a message, counting the byte count
   Serial.println();  //Prints data to the serial port as human-readable ASCII text.
```

Receiver_Module | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

Receiver_Module

```
if (vw_get_message(buf, &buflen)) /* Reading the last received message.The function itself returns true if the message was verified correct,
                                  or false if a message was received but appears to have been corrupted.*/
{
if(buf[0]=='1')// Checking which signal is recieved
{
digitalWrite(13,HIGH);//The function is used to write a HIGH value to a digital pin.[LED OF RECEIVER GLOWS]
}

else
{
digitalWrite(13,LOW);//The function is used to write a LOW value to a digital pin.[LED OF RECEIVER DOES NOT GLOW]
}
}

}
```

CONSTRUCTION OF CURCUIT

1. We take 1st Arduino and connect the 12th pin to DATA pin of TX Module, while grounding the TX and connecting it to power supply.
2. A red LED is connected to 13th pin of Arduino and the other end is grounded.
3. A button is introduced between Arduino and another power supply, and the button is grounded with 10 KΩ resistor. Here our transmitter circuit is completed.
4. Moving on to the receiver circuit, we take the 2nd Arduino and connect the 12th pin to DATA pin of RX, and similar to TX, we ground the ground pins and connect the power pins to a power source.
5. Again, a red LED is connected to 13th pin of the 2nd Arduino and the other end is grounded with a resistor of 100Ω.
6. The transmitter circuit is now completed and so is the whole circuit.

AIM & WORKING

The aim of our project is to successfully transmit data from RF Transmitter module to Receiver module using two Arduino UNO R3 boards.

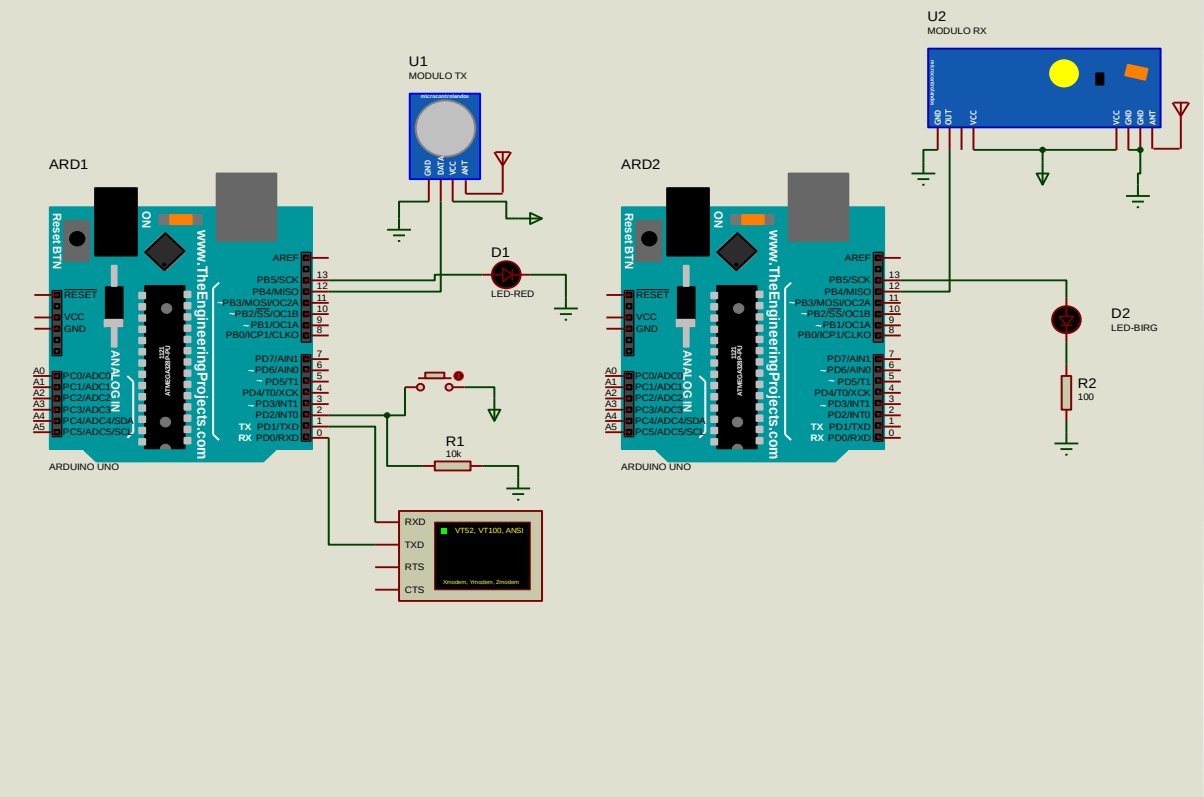
In this project, the transmitter is sending two characters namely “1” and “0” when the button is ON and OFF respectively. Whenever the button is OFF the transmitter LED is ON and sending “0”, showing that the character “0” is sent to receiver module using RF communication.

When the receiver circuit receives “0”, the Arduino on the receiver end of the circuit will keep the LED OFF.

Now when the button is pushed, the chraracter “1” is transmitted by the RF transmitter, the LED on the receiver circuit is turned ON, and simultaneously the transmitter LED goes OFF.

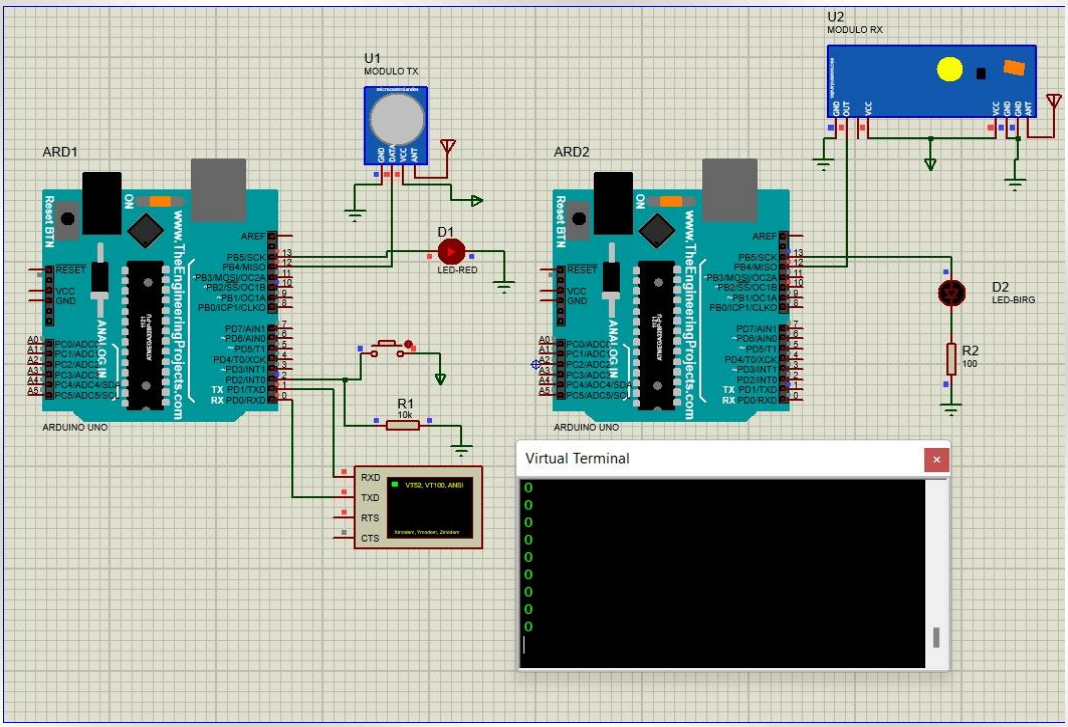
By using the terminal in the circuit, we can see the 0s and 1s being transmitted. The glowing of LED on transmitter circuit in the first case (button OFF) and on the receiver circuit in the second case (button ON) tells us that the “0” and “1” signal is successfully sent and received using RF module communication.

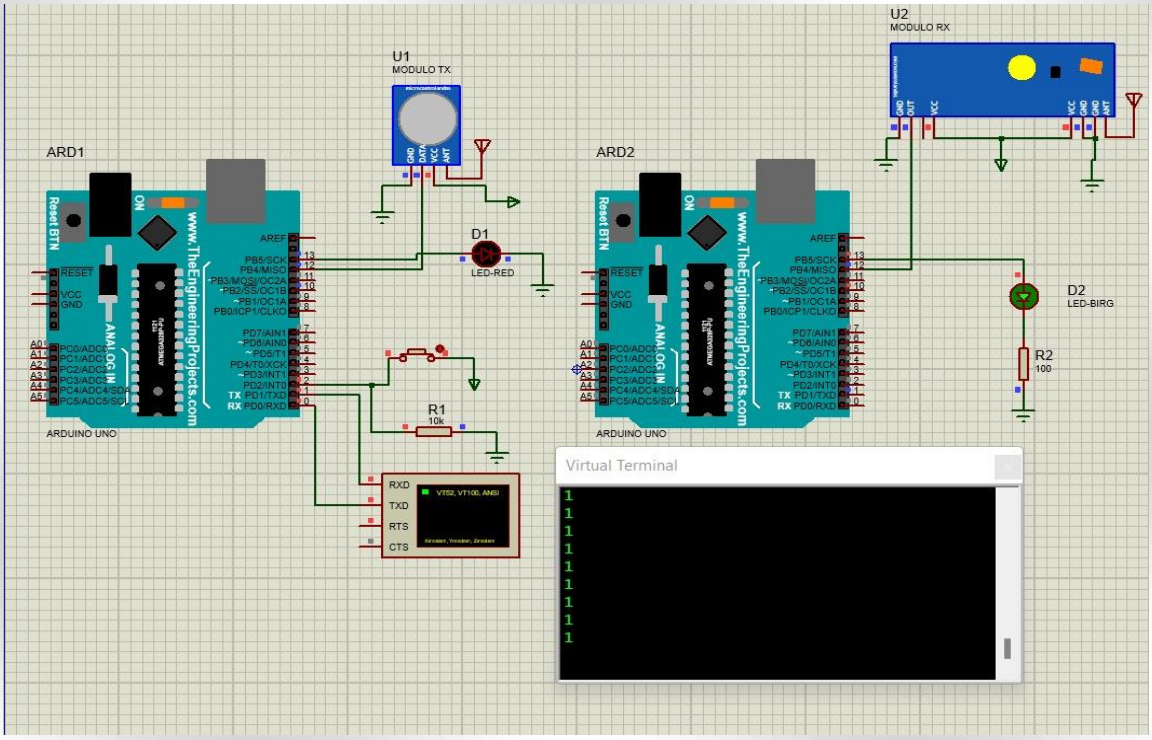
RSC Group Report



Final Circuit Design

SIMULATION DISPLAY





RESOURCES USED

- https://youtu.be/nL34zDTPkcs
- https://youtu.be/PeKZJ-kdcGs
- https://www.circuitbasics.com/basics-uart-communication/
- http://www.pictutorial.net/2015/10/how-to-interface-RF-module-with-microcontroller.html
- https://www.electronicshub.org/arduino-rf-transmitter-receiver-module/#:~:text=The%20RF%20Receiver%20Module%20consists,the%20434%20MHz%20Receiver%20module.
- https://en.wikipedia.org/wiki/RF_module#:~:text=An%20RF%20module%20(short%20for,communicate%20with%20another%20device%20wirelessly.&text=RF%20modules%20are%20typically%20fabricated%20using%20RF%20CMOS%20technology.

KEY POINTS

- RF Modules are popularly used in remote control system. In Quadcopter, Robot remote control, Industrial remote control, telemetry and remote sensing etc.
- By definition, UART is a hardware communication protocol that uses asynchronous serial communication with configurable speed. Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving end UARTs are being used in many DIY electronics projects to connect GPS modules, Bluetooth modules, and RFID card reader modules to your Raspberry Pi, Arduino, or other microcontrollers. One of the best things about UART is that it only uses two wires to transmit data between devices
- Bumpy font means first word is in lower case and then the second letter of the other word is in capital letters so that we don't need to give a space

PROBLEM FACED

- Earlier when we imported the library VirtualWire, and compiled the source code, we were getting errors like library not found. We tried RH_ASK.h instead, still error was occuring.

To overcome this problem, we had to download the zip file of VirtualWire and hence our problem was resolved

THANK YOU