



**BACHELOR OF COMPUTER SCIENCE
(BSc) PROGRAMME**

PROJECT TITLE

RF SECURE CODED COMMUNICATION SYSTEM

SUBMITTED BY

Isabella Mercy Abuor (P15/136964/2019)

SUPERVISOR

Dr. Kahonge

Declaration

I here by declare that the project report entitled “RF Secure Coded Communication System” written and submitted to University of Nairobi, School of Computing is an outcome of my own efforts and a record of an original work done by me under the guidance of Dr. Kahonge. And has not been previously submitted to any other University. For the partial fulfilment of Bachelor of Science, Computer Science.

Signature of the Candidate: _____ Date: _____

Abstract

This project was design to ensure maximum level of secrecy when transmitting data over RF from one device to another Securely using AES (Advanced Encryption Standard) algorithm. The User will enter a message through the GUI from the PC through a computer keyboard and the Message will be Encrypted using AES algorithm and transmitted using RF to the Receiver. On the receiver side, The message will be decrypted and they will be alerted that there is a New message displayed on the LCD, then they will be prompt to put a password and if and only if the password is correct then the transmitted Message will be displayed on the screen.

Keywords: *RF, Encryption, Decryption, AES, Secure*

Acknowledgements

First and foremost, praises and thanks to God, the Almighty, for His showers of blessings throughout my project to complete the project successfully on time.

I would like to express my special thanks of gratitude to my supervisor Dr. Kahonge who guided me through the whole process of creating this project and who gave me this golden opportunity to do this wonderful project. I am extending my heartfelt thanks to my family for their love, prayers, caring sacrifices and for supporting me financially and morally to be able to achieve the making of this project. I would also like to thank my friends for moral support for times where I felt I was stuck. Special shout out to my Classmates and group 6 members who have also kept me on my toes when where there when I needed them.

TABLE OF CONTENTS

1. Chapter 1: Introduction	1
1.1 Background	1
1.2 Problem Definition.....	2
1.3 Objectives.....	2
1.4 Justifications.....	3
2. Chapter 2: Literature Review	4
2.1 RF Communications	4
2.1.1 The History of Radio Frequency (RF).....	4
2.1.2 How RF Communication Works	4
2.2 CRYPTOGRAPHY	5
2.2.1 The History of Cryptography	5
2.2.2 AES Algorithm	7
3. Chapter 3: Methodology.....	10
4. Chapter 4: Systems Design.....	11
4.1 Block Diagrams.....	11
4.2 Flowcharts.....	13
5. Chapter 5: Systems Implementation	16
6. Chapter 6: Conclusion	23
7. Appendix A: References and Bibliography.....	24
8. Appendix B: User and Technical Manual	25
9. Appendix C: Sample Programs.....	28
9.1 The Results	30
9.2 Source Code	36

1. Chapter 1: Introduction

1.1 Background

There are times where two people are communicating and they do not want a third party to eavesdrop in a face-to-face conversation the same situation can be applied when communicating through devices. Many people fear that their data is been “Stolen” or any of their private chat conversation isn’t really private. Currently there has been an increase of (Internet of things) IoT attacks in the world. Possible attack on such a data transfer is trivial; data can be easily captured, modified, and resent. The use of RF secure coded based communication system is particularly advantageous when it comes to such situations. Implementation of the AES algorithm is what will make the system more secure than ever.

A radio frequency (RF) signal refers to a wireless electromagnetic signal used as a form of communication, if one is discussing wireless electronics. This is the frequency band at which wireless telecommunications signals are being transmitted and broadcasted. They are many other types of signals with different frequency band that are responsible for other things, For example FM which is used for radio broadcasting.

Cryptography is the science of keeping information secure by transforming it into form that unintended recipients cannot understand. In cryptography, an original human readable message, referred to as plaintext, is changed by means of an algorithm, or series of mathematical operations, into something that to an uninformed observer would look like gibberish; this gibberish is called ciphertext.

Encryption is what converts the data sent by the sender into Ciphertext and decryption is the opposite of Encryption, Where the data is converted back to plaintext for the receiver to interpret the data with the help of encryption and decryption keys. There are types of encryption which are Symmetric Encryption, Asymmetric Encryption, Hashing. The Symmetric Encryption the plaintext is scrambled using a key and once the data has been encrypted it is transmitted to the receiver. At the receiver the same key that was used for encryption is used to decrypt the data.

Under the Symmetric Encryption, the scrambling of data is achieved using a symmetric encryption algorithm which are Data Encryption Standard (DES), the Advanced Encryption Standard (AES), or the International Data Encryption Algorithm (IDEA). I decided to go with the Advanced Encryption Standard (AES) also known as the Rijndael algorithm because it is extremely efficient and it provides various options for key lengths (128-bit, 192-bit, and 256-bit key). The algorithm is a symmetrical block cipher algorithm that takes plain text in blocks of 128 bits and converts them to ciphertexts using keys of 128, 192 and 256 bits. Since the AES algorithm is considered secure, it is in the worldwide standard. (*Aes Algorithm Steps - Google Search*, n.d.)

1.2 Problem Definition

As the possibilities of the Internet continue to grow so has the Cybercrimes. Currently in the world, Cyber-crime is one of the leading problems in the world with an Estimated global loss from cybercrime is projected to hit just under a record \$1 trillion for 2020 as the coronavirus pandemic provided new opportunities for malicious actors. One of these Cyber-crimes is eavesdropping. For example, in military operations, government or other sensitive communications, secrecy is of paramount importance. So, when there is a need for sending any secret message, one can type the message through a computer keyboard interfaced with the system comprising of Arduino and a RF transmitting module and the message would be Encrypted using AES algorithm and on the receiver side the implementation of the keypad to ensure maximum security so that the message should be read by the person the message was intended to.

1.3 Objectives

The main aim for this project is to ensure a message is transmitted successfully and securely through the Rf module

1. Create two hardware platforms that send and receive messages to each other using RF
2. To implement the AES algorithm to secure the message in transit through Symmetric Encryption.
3. To test the above system.

1.4 Justifications

The system would help in the outside world in that it would

- To protect customer data
- To protect your intellectual property
- To shield internal communication

Why I decided to choose AES algorithm Standard:

- This robust security algorithm may be implemented in both hardware and software.
- It is resilient against hacking attempts, thanks to its higher-length key sizes (128, 192, and 256 bits).
- It is an open-source solution. Since AES is royalty-free, it remains highly accessible for both the private and public sectors.
- AES is the most commonly used security protocol today, used for everything from encrypted data storage to wireless communications

2. Chapter 2: Literature Review

2.1 RF Communications

2.1.1 The History of Radio Frequency (RF)

What is Radio Frequency (Rf), RF refers to the frequencies that fall within the electromagnetic spectrum associated with radio wave propagation. Frequency is the number of cycles per unit time(seconds) RF waves that have been modulated to contain information are called RF signals. These RF signals have some behaviors that can be predicted and detected and they can interface with other signals. There are some free bands available that are used for wireless communication. The most attractive frequency band is 433MHz. The reason we use the 433Mhz is because they are extremely cheap and can be found in any local hardware store, They are also easy to use because they only have four pins to connect plus it is legal as per The Communications Authority of Kenya which is in charge of controlling how communications are used within the county.(*RF Wireless Technology / Mouser, n.d.*)

2.1.2 How RF Communication Works

The primary motivation of this project was the design of a wireless communications system using 433Mhz Rf module but in order to send data wirelessly we must convert analog data into digital data and send it over radio. There are many methods of sending digital data over radio which are Frequency Shift Keying or FSK. In this method the digital signal modifies the frequency of the carrier wave. This is similar to Frequency Modulation or FM radio. Phase Shift Keying or PSK. This works by modifying the phase of the carrier in response to the input signal. Many modems work this way. Amplitude Shift Keying or ASK. This is a simpler method, similar to Amplitude Modulation or AM radio.

For the purpose of this Project, we used Amplitude Shift Keying (ASK) In Amplitude Shift Modulation the amplitude of the carrier wave is changed in response to the incoming data signal. Digital 1 – This drives the carrier at full strength. Digital 0 – This cuts the carrier off completely. In ASK, it requires two input signals, First input is binary sequence signal and the second input is carrier signal. Here the most important point we need to always consider the second input which is the carrier signal has the more

amplitude/voltage range than the input binary sequence signal. (*Amplitude Shift Keying*, 2019)

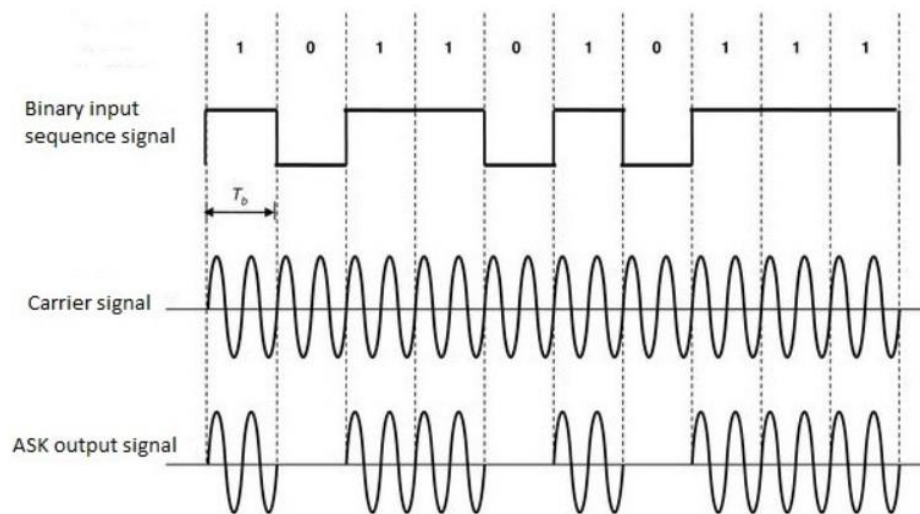


Figure 1 Showing the ASK modulation

Advantages of RF secure coded communication systems

- I. It has different penetration through the walls of the buildings or houses based on the frequency. Hence used for radio and television transmission and for cellular mobile phone service.
- II. Used in various medical applications. It is used in Diathermy instrument for surgery. It is used in MRI for taking images of human body. It is also used for skin tightening.
- III. It is used in radar for object detection.
- IV. It is used for satellite communication.
- V. It is used in microwave line of sight communication system.

2.2 CRYPTOGRAPHY

Cryptography is the science of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it. (*What Is Cryptography? How Algorithms Keep Information Secret and Safe / CSO Online*, n.d.)

2.2.1 The History of Cryptography

The art of cryptography is considered to be born along with the art of writing. As civilizations evolved, human beings got organized in tribes, groups, and kingdoms. This led to the emergence of ideas such as power, battles, supremacy, and politics. These ideas

further fueled the natural need of people to communicate secretly with selective recipient which in turn ensured the continuous evolution of cryptography as well.

The roots of cryptography are found in Roman and Egyptian civilizations.

Hieroglyph – The Oldest Cryptographic Technique

The first known evidence of cryptography can be traced to the use of 'hieroglyph'. Some 4000 years ago, the Egyptians used to communicate by messages written in hieroglyph. This code was the secret known only to the scribes who used to transmit messages on behalf of the kings.

The earlier Roman method of cryptography, popularly known as the **Caesar Shift Cipher**, relies on shifting the letters of a message by an agreed number (three was a common choice), the recipient of this message would then shift the letters back by the same number and obtain the original message.



Figure 2 Showing how Caesar Shift Cipher works

At the start of the 19th century when everything became electric, Hebern designed an electro-mechanical contraption which was called the Hebern rotor machine. It uses a single rotor, in which the secret key is embedded in a rotating disc. The key encoded a substitution table and each key press from the keyboard resulted in the output of cipher text. This also rotated the disc by one notch and a different table would then be used for the next plain text character. This was again broken by using letter frequencies.

The Enigma machine was invented by German engineer Arthur Scherbius at the end of World War I, and was heavily used by the German forces during the Second World War. The Enigma machine used 3 or 4 or even more rotors. The rotors rotate at different rates as you type on the keyboard and output appropriate letters of cipher text. In this case the key was the initial setting of the rotors. The Enigma machine's cipher was eventually broken by Poland and the technology was later transferred to the British cryptographers who designed a means for obtaining the daily key.

In the early 1970's, IBM realized that their customers were demanding some form of encryption, so they formed a "crypto group" headed by Horst-Feistel. They designed a cipher called Lucifer. In 1973, the Nation Bureau of Standards (now called NIST) in the US put out a request for proposals for a block cipher which would become a national standard. They had obviously realized that they were buying a lot of commercial products without any good crypto support. Lucifer was eventually accepted and was called DES or the Data Encryption Standard. In 1997, and in the following years, DES was broken by an exhaustive search attack. The main problem with DES was the small size of the encryption key. As computing power increased it became easy to brute force all different combinations of the key to obtain a possible plain text message.

In 1997, NIST again put out a request for proposal for a new block cipher. It received 50 submissions. In 2000, it accepted Rijndael, and christened it as AES or the Advanced Encryption Standard. (*Origin of Cryptography - Tutorialspoint*, n.d.)

2.2.2 AES Algorithm

The AES algorithm (also known as the Rijndael algorithm) is a symmetrical block cipher algorithm that takes plain text in blocks of 128 bits and converts them to ciphertext using keys of 128, 192, and 256 bits. Since the AES algorithm is considered secure, it is in the worldwide standard.

How it Works?

For Example, let's take a 128-bit block

1. The plaintext is XORed (which means a key is added) and the Key is expounded.
2. Byte Substitution (Sub-Bytes)

Instead of having a long string of bits, AES likes to arrange them in grid (4x4 Grid for 128-bits)

b0	b4	b8	b12
b1	b5	b9	b13
b2	b6	b10	b14
b3	b7	b11	b15

3. Shift rows

Each of the four rows is shifted to the left. The first Row remains the same, The second is shifted by one position, The third rows is shifted by two positions and the fourth is shifted three positions. Hence the grid would look like this

b0	b4	b8	b12
b5	b9	b13	b1
b10	b14	b2	b6
b15	b3	b7	b11

4. Mix columns

Each columns of fours bytes are now transformed using a matrix

5. Add Round Key

XOR is now used on the 128 bits.

This is now repeated until the last round where It just outputs the ciphertexts. The number of rounds depends if its 128-bit,192-bit or 256-bit key. For 128-bits its 10 rounds, For 192-bits its 12 rounds, For 256-bits its 14 rounds

For the Decryption Part is just the inverse of the encryption:

1. Inverse add round key
2. Inverse Shift rows
3. Inverse byte substitution

4. Inverse add round key
5. Inverse mix columns
6. Inverse shift rows
7. Inverse byte substitution
8. Inverse add round Key (x 9 or x 11 or x 13, depending on whether the key is 128,192 Or 256-bits)

(Aes Algorithm Steps - Google Search, n.d.)

A summary of the AES process represented below

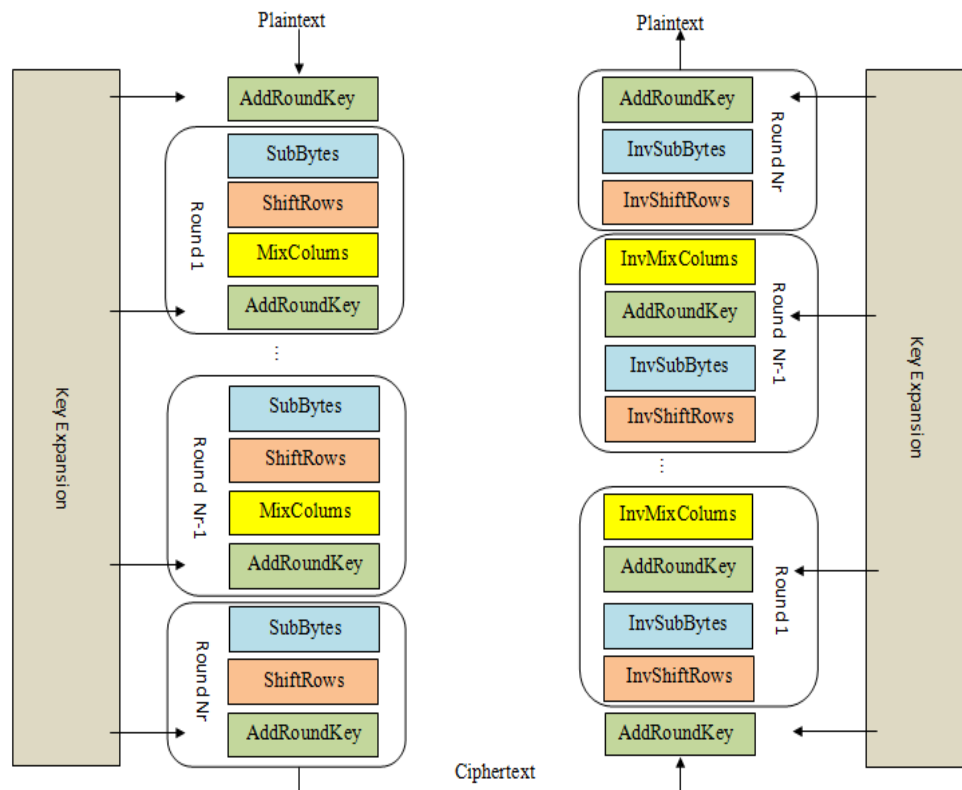


Figure 3 Showing the summary of AES Algorithm

3. Chapter 3: Methodology

The approach used in solving the insecurity in transmitting messages was the Waterfall Model.

The waterfall model describes a highly structured process for creating new project that flows linearly from gathering requirements to design, implementation, and testing — a process that is rooted in best practices from industries where design changes even early-on can be cost-prohibitive, like manufacturing and construction.

This model values completeness and quality over speed hence why it is chosen for this project.

The Project Requirements.

- The User should be able to input a message
- The Users message should be displayed with its ciphertext
- The transmitters message should be transmitted using Radio frequency
- The receiver should be able to key in the password
- The receiver should be able to see the message transmitted (plaintext) if the password is correct

Analysis

The requirements are analyzed and the problem fully researched and understood. It is seen that the cyber-crime rates are on a high and sensitive information is compromised due to hackers. The best way to combat this issue especially in a narrow range area, for example, the military bases is to use the free RF band (433MHz) and develop a strong encryption algorithm when transmitting messages.

The necessary hardware needed was researched and the encryption algorithm that suited this project best was noted to be Advanced Encryption Standard (AES) algorithm.

The microcontroller in use is the Arduino Uno.

Design

The circuit design layouts to ensure maximum secure communication is created for both the transmitter and receiver side. The GUI appearance is also designed.

Implement

The technical implementation of the circuits is realized and coding in the Arduino IDE is done.

Testing

The whole system is tested to ensure functionality.

4. Chapter 4: Systems Design

4.1 Block Diagrams

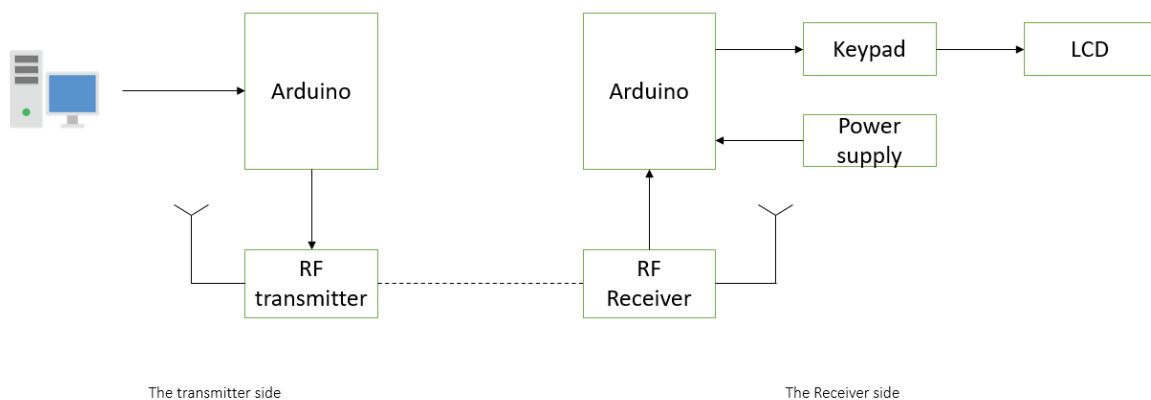


Figure 4 Showing the block diagram for the overall system

The above diagram shows the block diagram of the whole system with the receiver side and the transmitter side and their respective components.

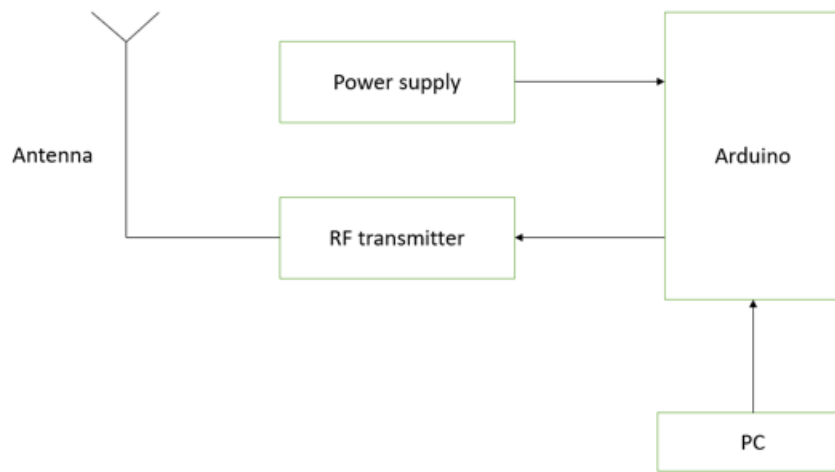


Figure 5 Showing the Transmitter block diagram

The above diagram shows the block diagram of the transmitter side and its respective components: the computer that the user will interact with, the micro-controller (Arduino), Power supply (battery), and the RF receiver, and how they are connected to each other.

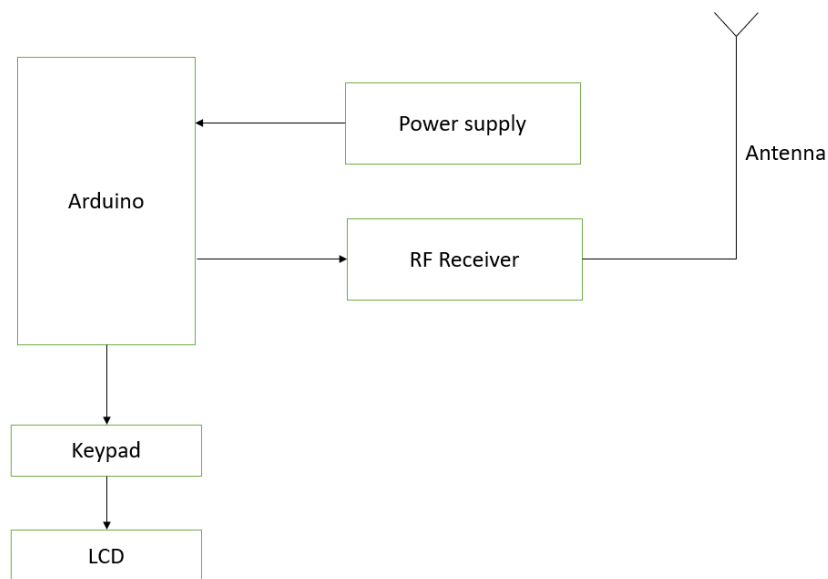


Figure 6 Showing the Receiver Block Diagram

The above diagram shows the block diagram of the receiver side and its respective components: the micro-controller (Arduino), Power supply (battery) and the Rf receiver and how they are connected to each other.

4.2 Flowcharts

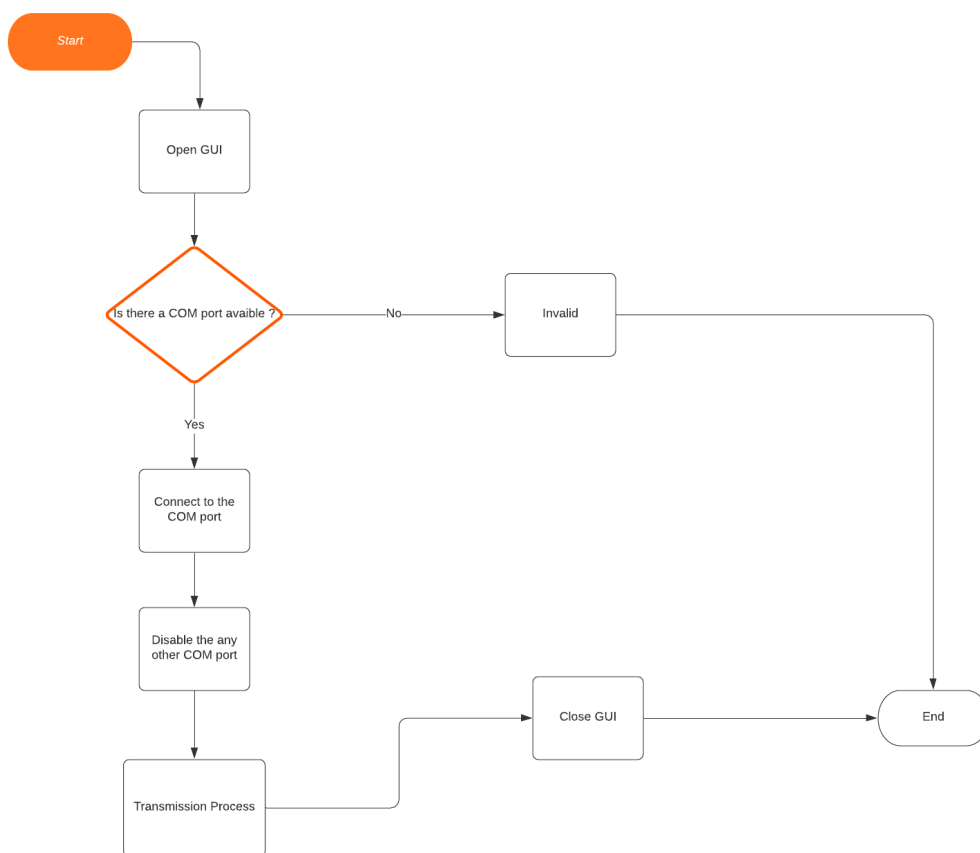


Figure 7 Showing the flowchart of the GUI

The diagram above shows the flow of events when the user interacts with the GUI.

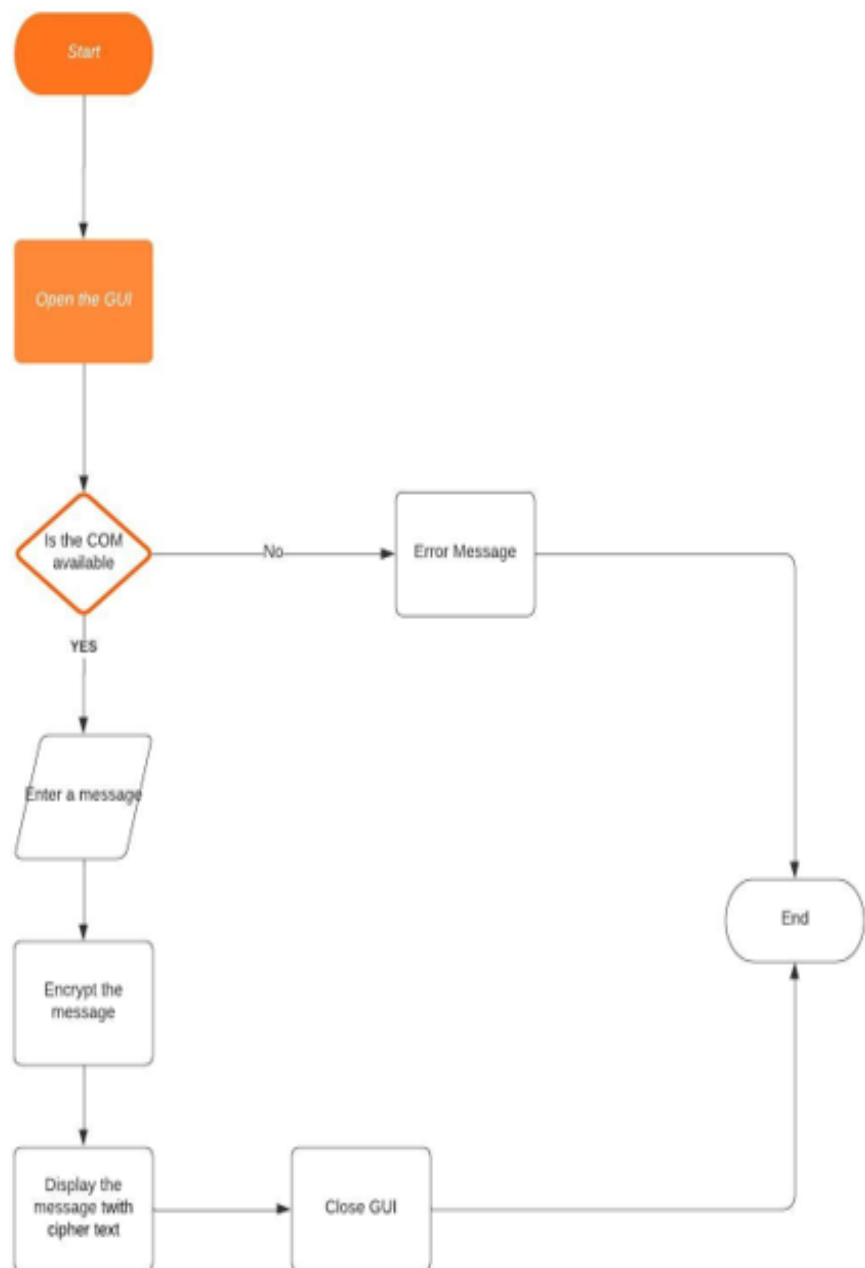


Figure 8 Showing the flowchart of the Transmitter

The diagram above shows the flow of events when the user interacts with the transmitter side and shows the process of sending a message with the whole Encryption process.

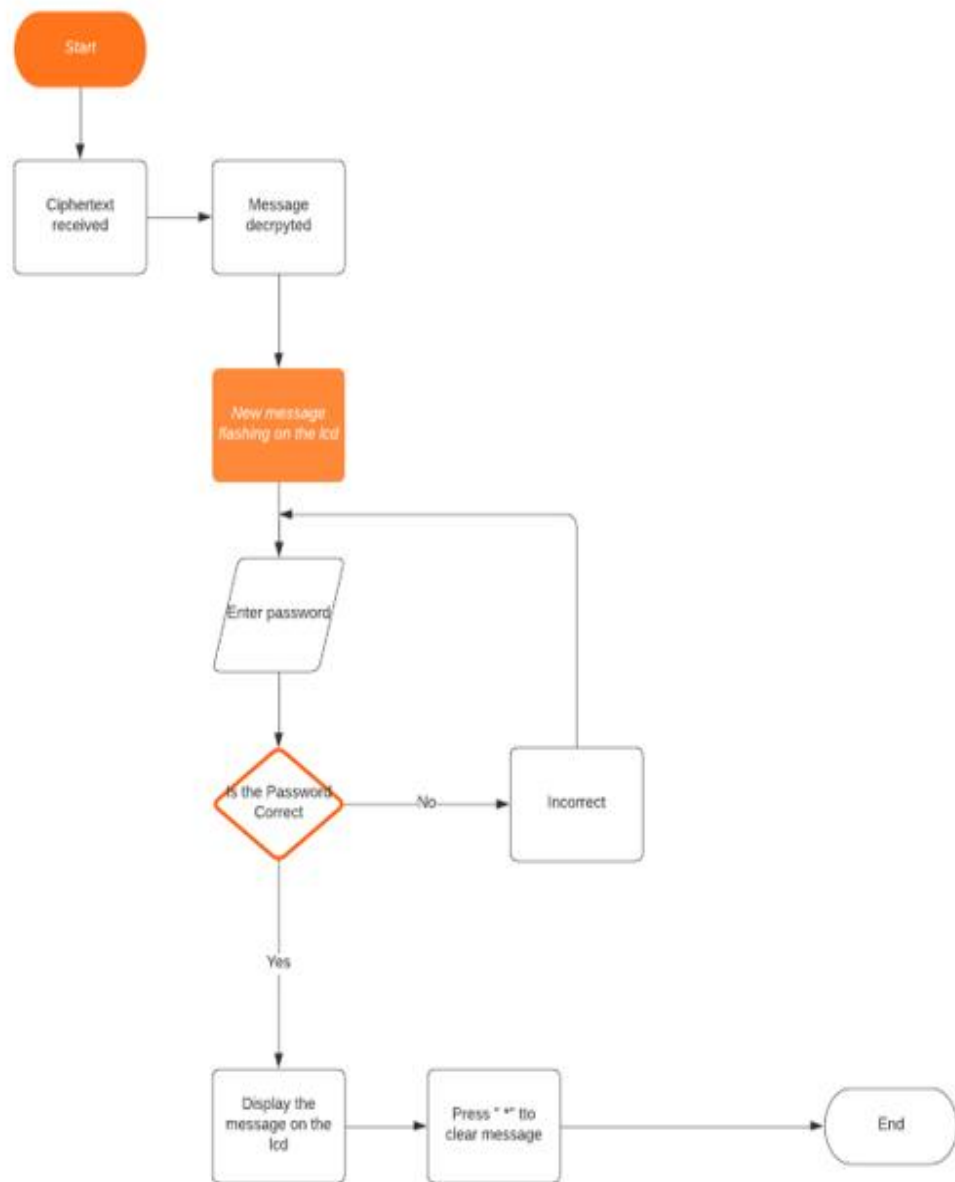


Figure 9 Showing the flowchart of Receiver side

The diagram above shows the flow of events when the user interacts with the Receiver side and shows the whole process of receiving a message with the whole decryption process and how the decrypted message is displayed on the LCD.

5. Chapter 5: Systems Implementation

In this section, will be a discussion in detail how the project was built from the start to the end.

After all the equipment was bought, The project was divided how into smaller sections, One must download the Radiohead library needed for the transmission process. The first part was to connect 433 MHz RF transmitter and receiver and test it if the message was sent through the two devices. The challenge that was faced in this section was that the message was not been transmitted to the RF receiver because the Rf module bought was not compatible where the receiver was not 433MHz while the Rf transmitter was 433 MHz.

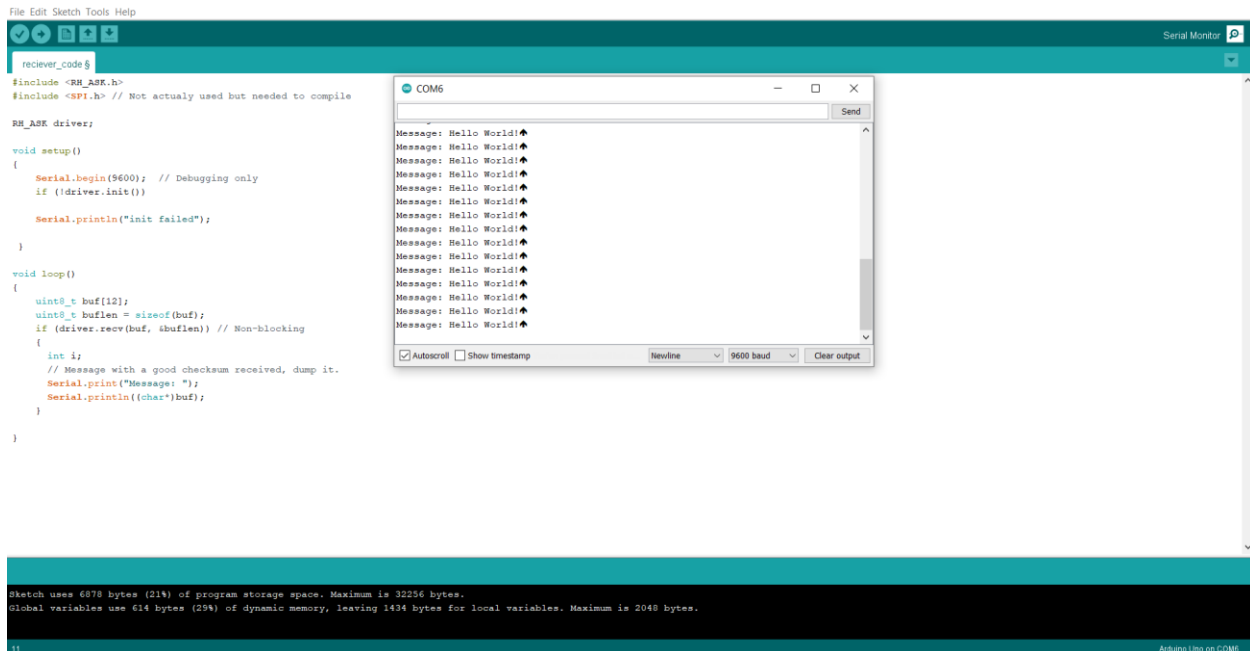


Figure 10 Showing the test for transmission

In the Next section was to connect the LCD to the receiver side, One should connect the I2C pins that the LCD came with. Connect the VCC to the 5V pin on the Arduino, the ground to the ground to the Arduino and the SDA and SCL on the LCD to the A5 AND A5 pins in the Arduino. One should download the wire library and liquid crystal I2C needed. Then find the I2C address

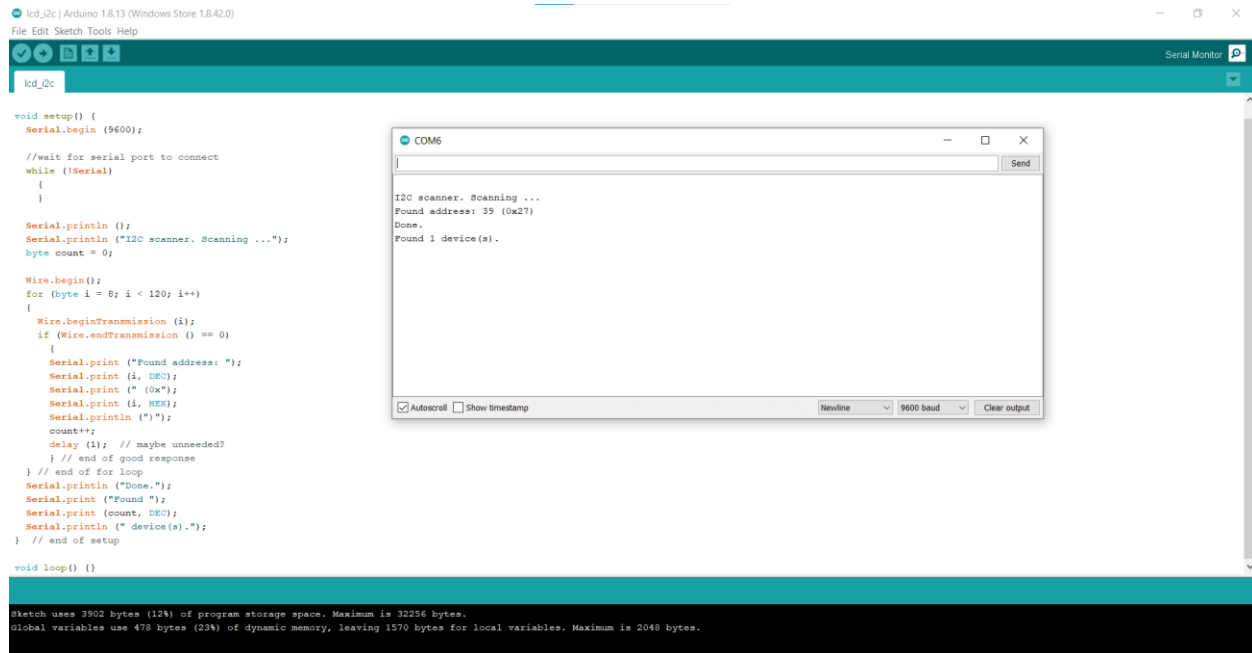


Figure 11 finding the address of the I2C

Then setup up the LCD using the I2C address one found above and tested if message on the serial monitor is displayed on the LCD. A problem that encountered during this section was to remove the null character from displaying on the LCD together with the message.

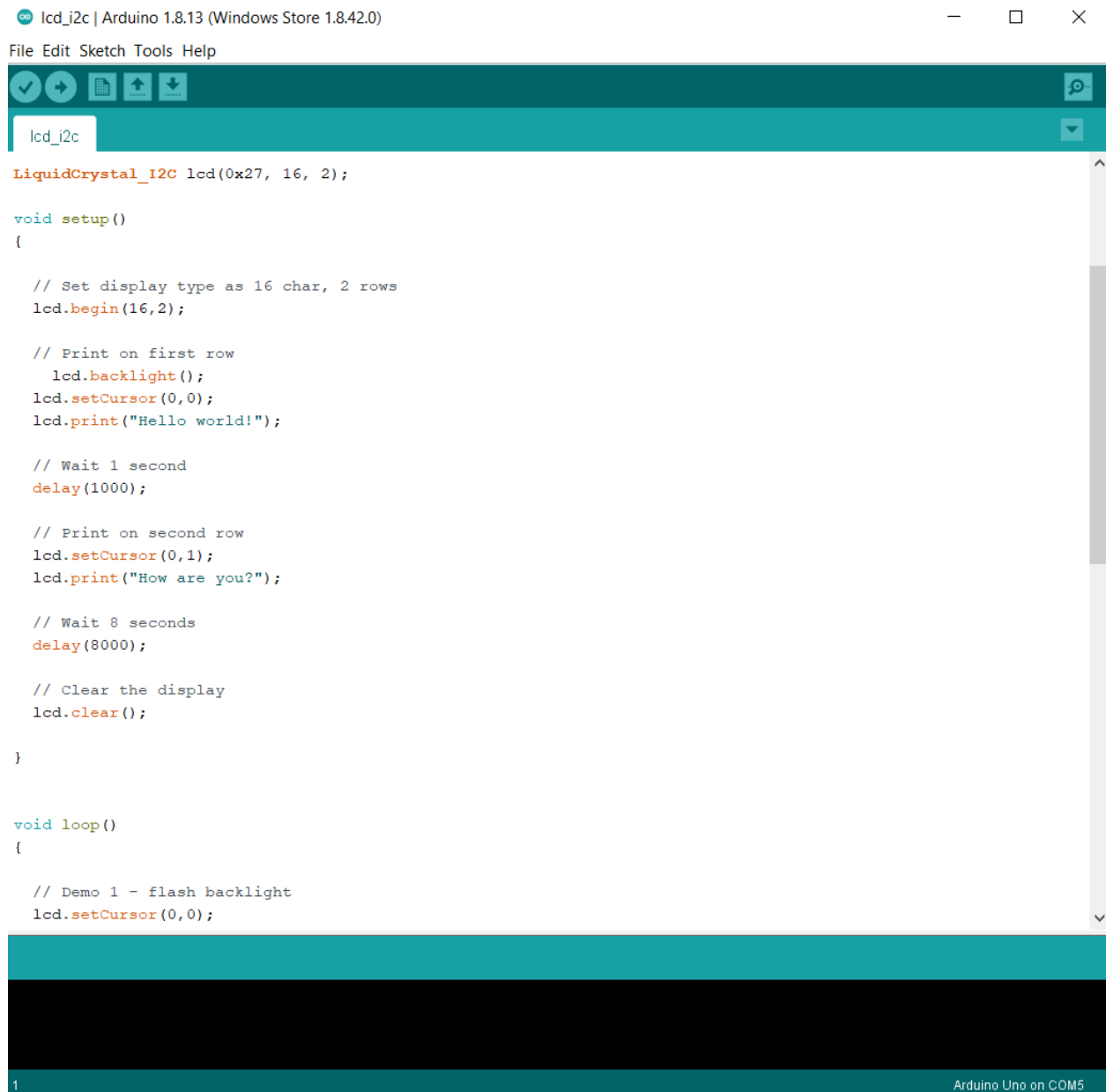


Figure 12 testing the functionality of the LCD

In the next section connect the keypad to the receiver side of the RF module and connect all the pins from the (*) symbol to the (#) symbol to the pin 2-8 on the Arduino respectively. Use a 4x3 keypad. Download the keypad library needed for implementing a keypad. Test the keypad to see if it works, Prompt the user to input a password and the password to be displayed on the LCD and instead of displaying the password for every character keyed it displays a (*) character on the LCD. The challenge was whenever one was keying for example the char 1 is it was displaying a 49 instead of character 1

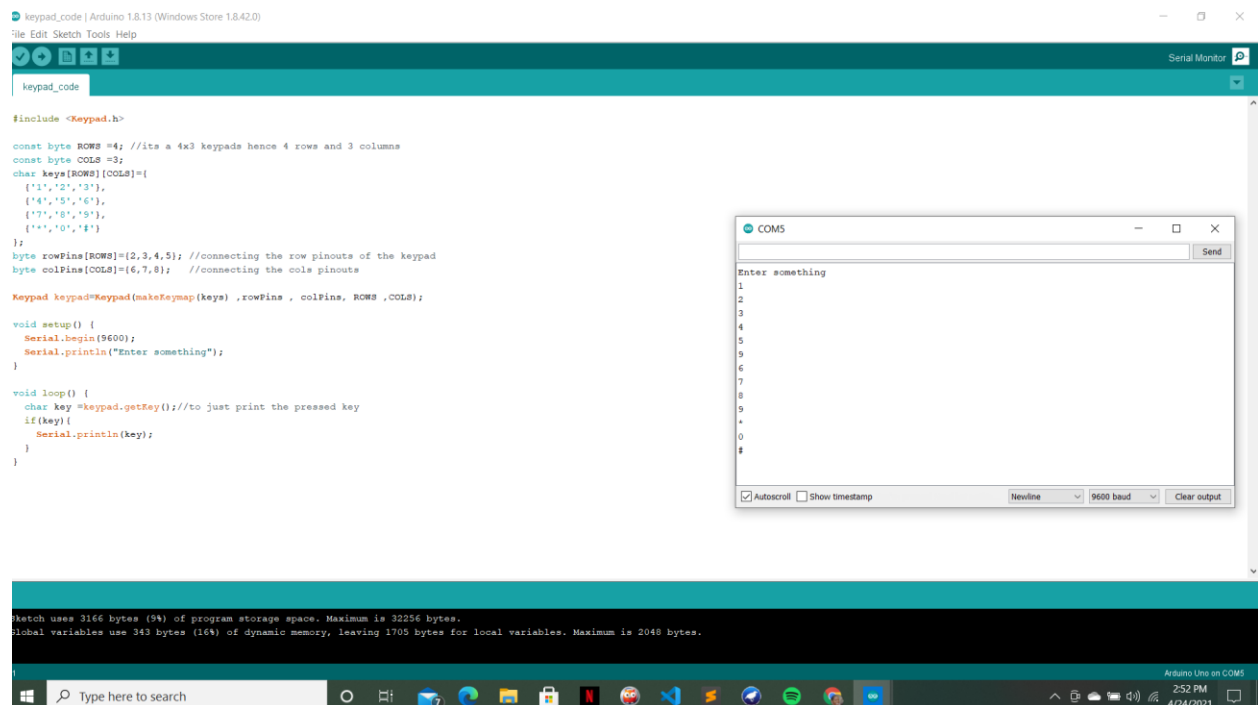


Figure 13 Showing the keypad functionality

In the next section, Implement the Encryption process using the AES algorithm and download the AES library needed for that.


```

void encryptNow(char*data,byte bits,byte length)
{
    byte paddedLength =length +N_BLOCK -length% N_BLOCK;
    aes.iv_inc();
    byte iv[N_BLOCK];
    byte plain_paded[paddedLength];
    byte cipher[paddedLength];
    byte check[paddedLength];
    unsigned long mirco_sec =micros();
    aes.set_IV(my_iv);
    aes.get_IV(iv);
    aes.do_aes_encrypt(data,length,cipher,key,bits,iv);

    //to check if encryption and decryption
    aes.set_IV(my_iv);
    aes.get_IV(iv);
    aes.do_aes_decrypt(cipher,paddedLength,check,key,bits,iv);
    printf("\nciphertext: ");
    aes.printArray(cipher, (bool) false);
}

```



Figure 14 Showing line of code that Encrypts

```

buffer = printable[i];
}
Serial.println();

aes.set_IV(my_iv); //setting initial vector
aes.get_IV(iv); //geeting the initial vector

aes.do_aes_decrypt(data,paddedLength,final_result,key,bits,iv);
}

```




Figure 15 Showing the line of code for decryption

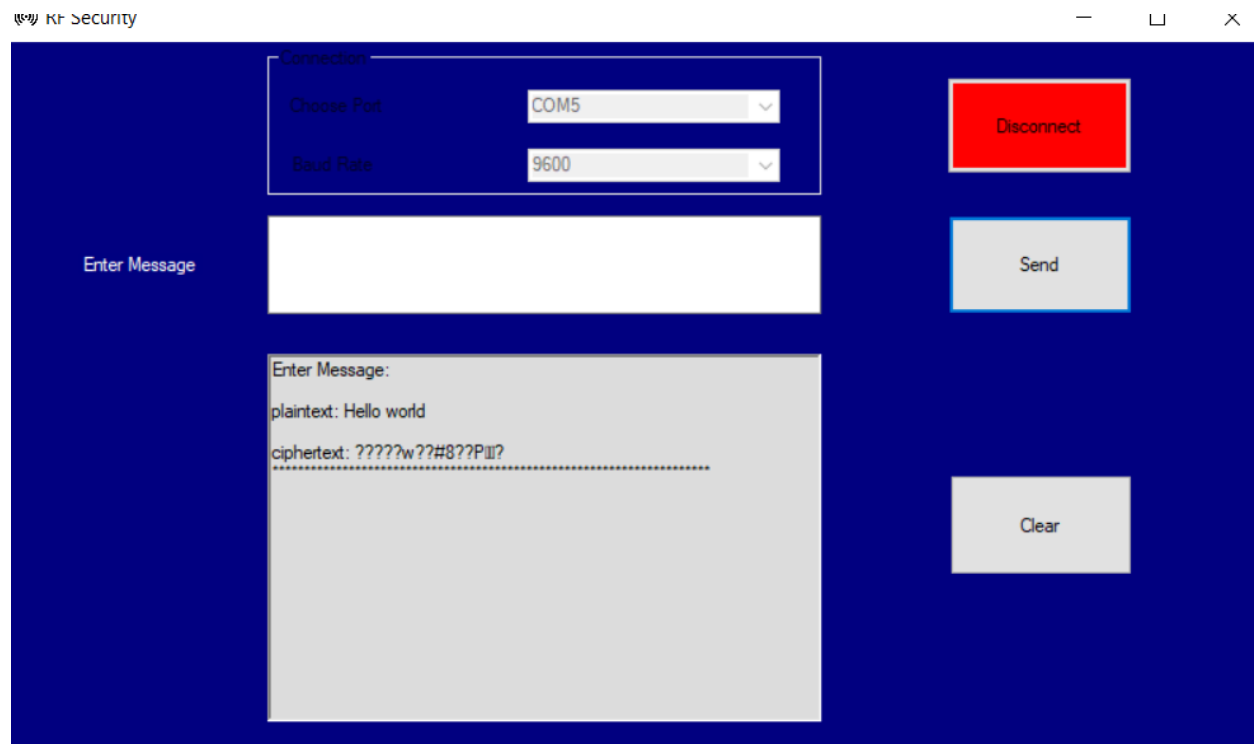


Figure 17 testing the GUI functionality

6. Chapter 6: Conclusion

The Internet is growing rapidly and so is the Cyber-crime rate especially during these times of the pandemic which has provided people with new opportunities for malicious actors. One of these cybercrimes is eavesdropping and with this project we can curb that problem by implementing Encryption (AES algorithm) when one wants to send a private message. The main aim for this project to ensure a message is transmitted successfully and securely through the Rf module and which it was achieved by the AES algorithm and on the receiver side a password system was implemented to enhance the security even further. However there is no communication between two or more people that can be 100% secure or 100% guaranteed that there is no unwanted third party but it, for sure makes it close to impossible to be hacked/eavesdrop.

The RF is easily affected by the environment hence sometimes if the two devices are far apart connection can be lost or message can be interfered with. Implementing an antenna would curb this problem but due to lack of equipment I left it out when design the project.

The system can be implemented in organizations where information is sensitive for example Governments, Military, Online banking and many more.

7. Appendix A: References and Bibliography

aes algorithm steps—Google Search. (n.d.). Retrieved May 9, 2021, from

https://www.google.com/imgres?imgurl=https%3A%2F%2Fstatic.commonlounge.com%2Ffp%2F600w%2Fy6UQ3zYSQRlWMrW537E7ooK1m1520492304_kc&imgrefurl=https%3A%2F%2Fwww.commonlounge.com%2Fdiscussion%2F632fdd267aaa4240a4464723bc74d0a5&tbid=yeciZMvROcwYnM&vet=12ahUKEwiW68P_j73wAhVVlp4KHZEtc4YQMygCegUIARC3AQ..i&docid=zUplisjIQuMj6M&w=600&h=377&q=aes%20algorithm%20steps&ved=2ahUKEwiW68P_j73wAhVVlp4KHZEtc4YQMygCegUIARC3AQ

Amplitude Shift Keying: Circuit Diagram, Working and Its Applications. (2019, October 6). ElProCus -

Electronic Projects for Engineering Students. <https://www.elprocus.com/amplitude-shift-keying-ask-working-and-applications/>

Origin of Cryptography—Tutorialspoint. (n.d.). Retrieved May 9, 2021, from

https://www.tutorialspoint.com/cryptography/origin_of_cryptography.htm

RF Wireless Technology | Mouser. (n.d.). Retrieved May 9, 2021, from

<https://www.mouser.com/applications/rf-wireless-technology/>

What is cryptography? How algorithms keep information secret and safe | CSO Online. (n.d.). Retrieved

May 9, 2021, from <https://www.csoononline.com/article/3583976/what-is-cryptography-how-algorithms-keep-information-secret-and-safe.html>

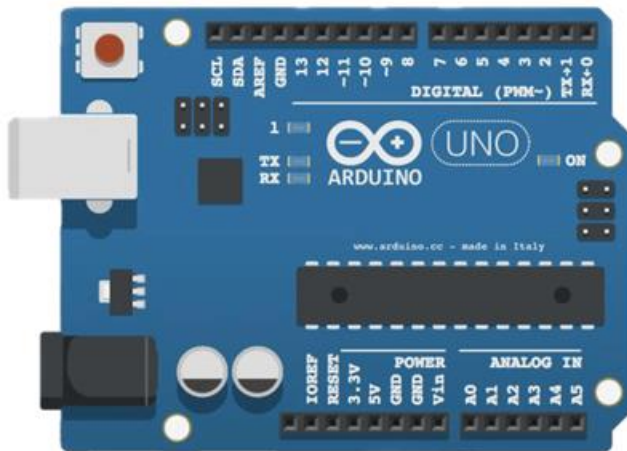
8. Appendix B: User and Technical Manual

433 MHz RF Module



- Wireless (RF) Simplex Transmitter and Receiver
- Receiver Operating Voltage: 3V to 12V
- Receiver Operating current: 5.5mA
- Operating frequency: 433 MHz
- Transmission Distance: 3 meters (without antenna) to 100 meters (maximum)
- Modulating Technique: ASK (Amplitude shift keying)
- Data Transmission speed: 10Kbps
- Circuit type: Saw resonator
- Low cost and small package

Arduino UNO



- The operating voltage is 5V
- The recommended input voltage will range from 7v to 12V
- The input voltage ranges from 6v to 20V
- Digital input/output pins are 14
- Analog i/p pins are 6

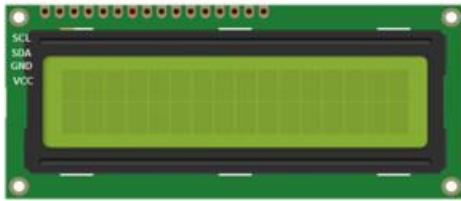
- DC Current for each input/output pin is 40 mA
- DC Current for 3.3V Pin is 50 mA
- Flash Memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK Speed is 16 MHz

4x3 KEYPAD



- Length of the keypad 93mm
- Weight 7.6grams
- Keypad dimensions 68.5mm x 76.5mm x 1mm
- 7-pin pitch connector

16x2 LCD with a I2C



- LCD screen size 16x2
- Includes an I2C with four pins VCC, GROUND, SCL, SDA
- +5V power supply (Also available in +3V)
- LED can driver by PIN1, PIN2, PIN15, PIN16 or A and K

9. Appendix C: Sample Programs

On the transmitter side it consists of 4 pins in which we are going to connect the VCC pin to the 5V on the Arduino, The ground on the transmitter will be connected to the ground on the Arduino ground terminals, The Data pin will be connected on the pin 12 on the Arduino and antenna on the antenna hole.

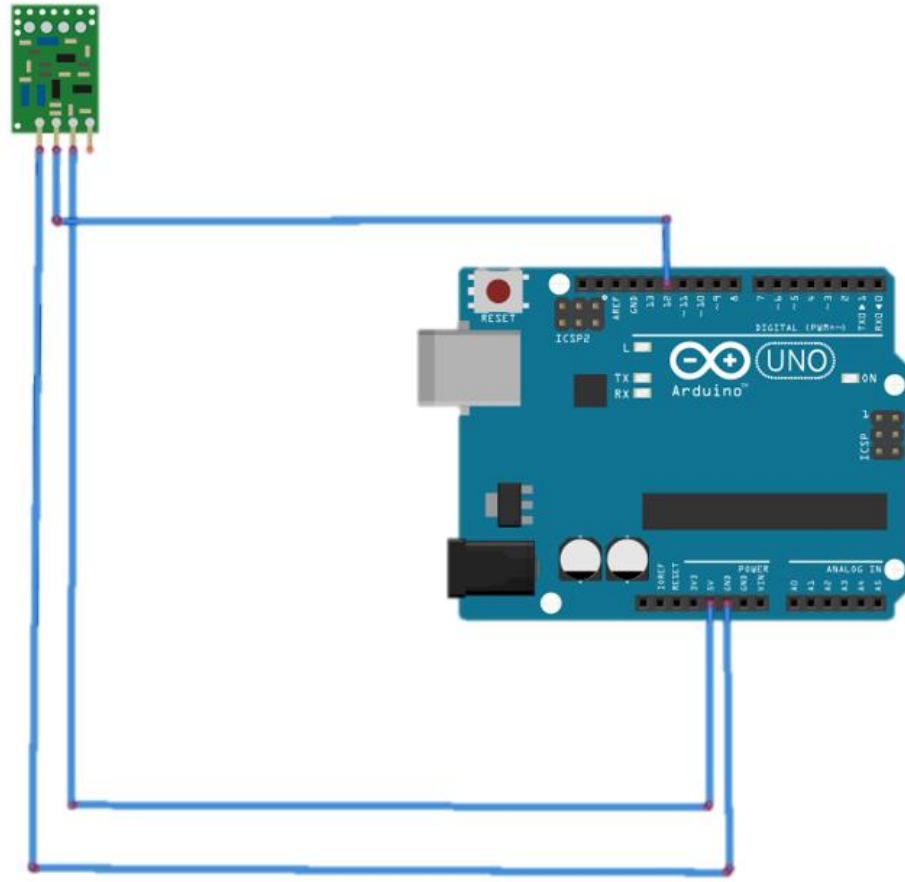


Figure 18 Showing the circuit diagram for the transmitter

On the Receiver side it consists of 4 pins in which we are going to connect the VCC pin to the 5V on the Arduino, The ground on the Receiver will be connected to the ground on the Arduino ground terminals, The Data pin will be connected on the pin 11 on the Arduino and antenna on the antenna hole.

Then connect the lcd through the I2C which has 4 pins in which I connected the power to the 5v on the Arduino, The ground to the ground on the Arduino.

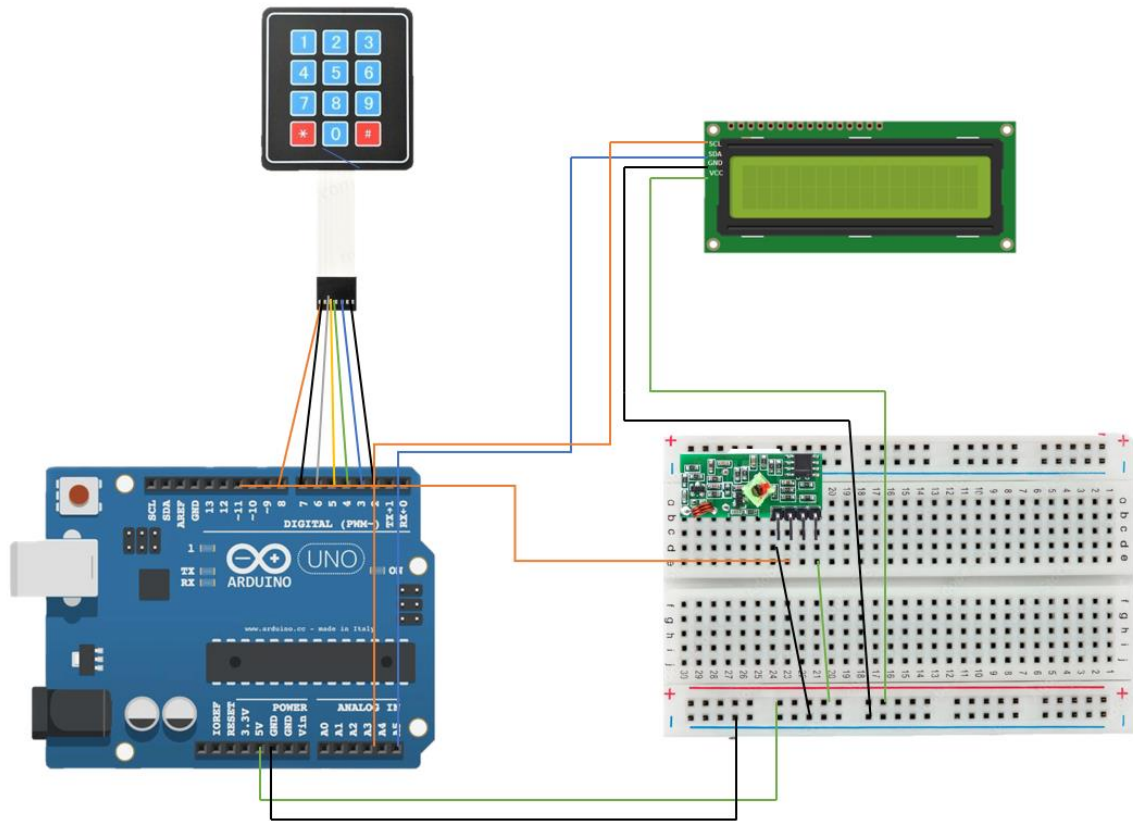


Figure 19 Showing The circuit diagram for the receiver side

9.1The Results

The following shows the Results of the whole Project

On the receiver

Side The user inputs a message and its shows them the plaintext and ciphertext they sent.

On the GUI

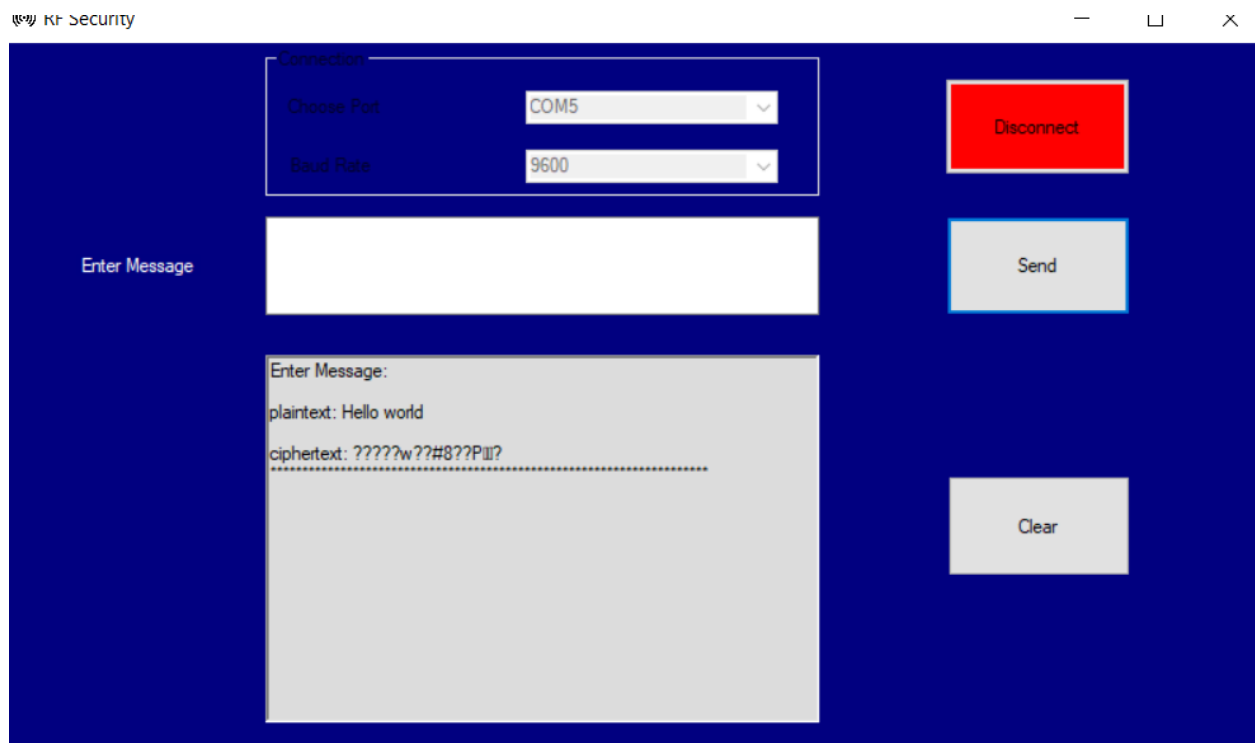


Figure 20 showing the results of the user sending a message

On the Receiving End

The user is alerted that there is a new message



Figure 21 Showing alert message in the lcd

The user is prompt to enter a password

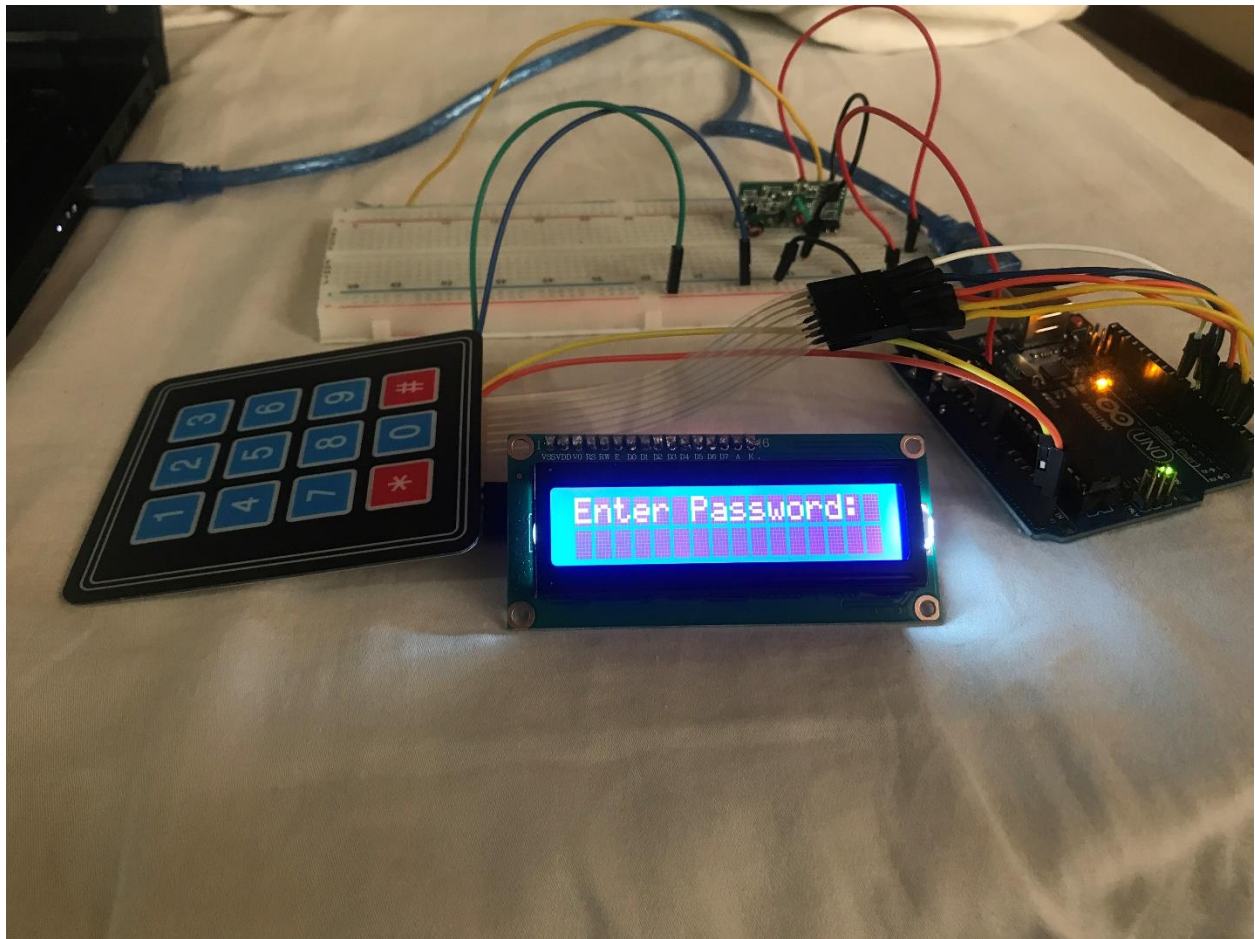


Figure 22 Showing password prompt

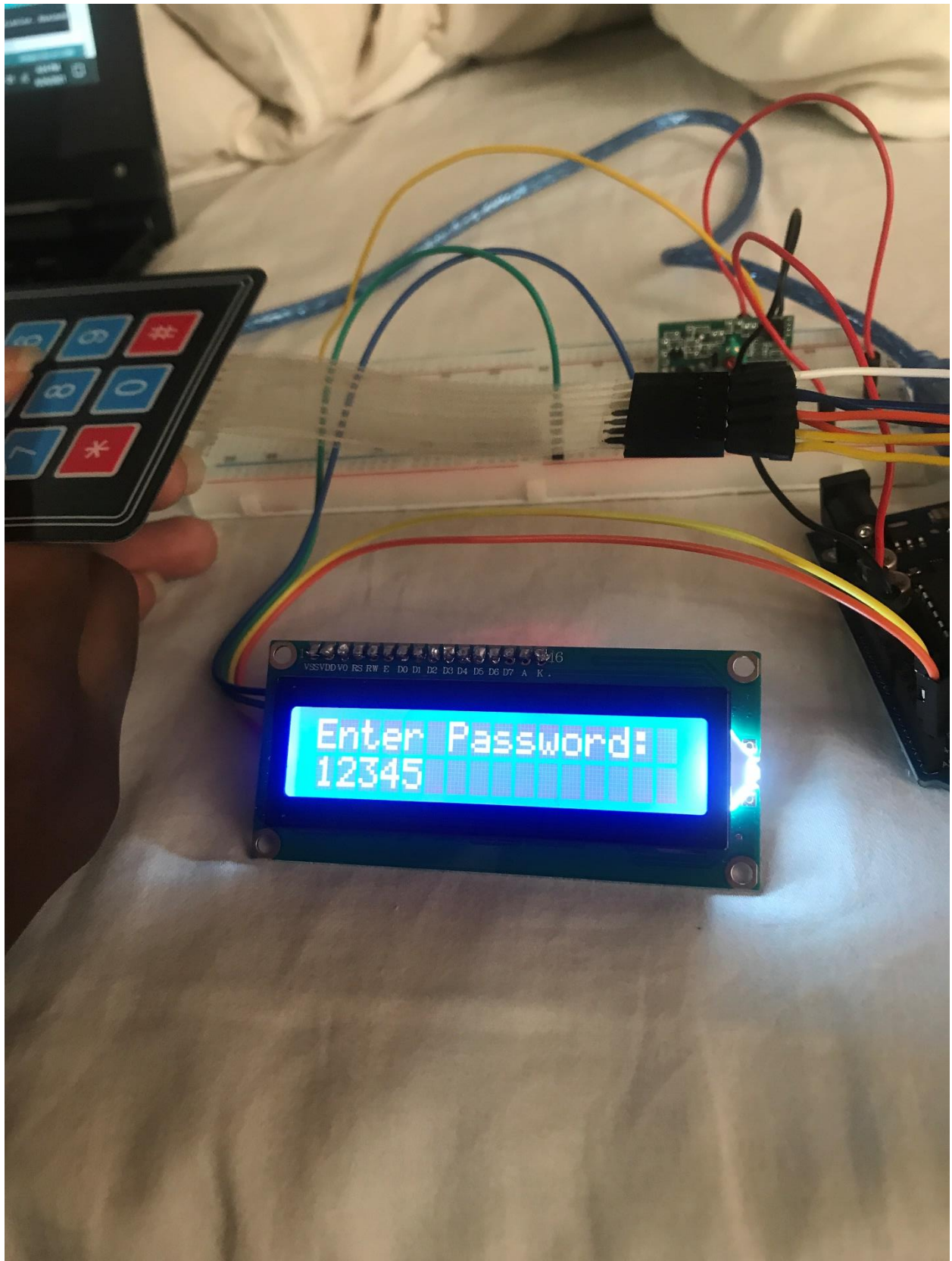


Figure 23 Showing user inputing the password

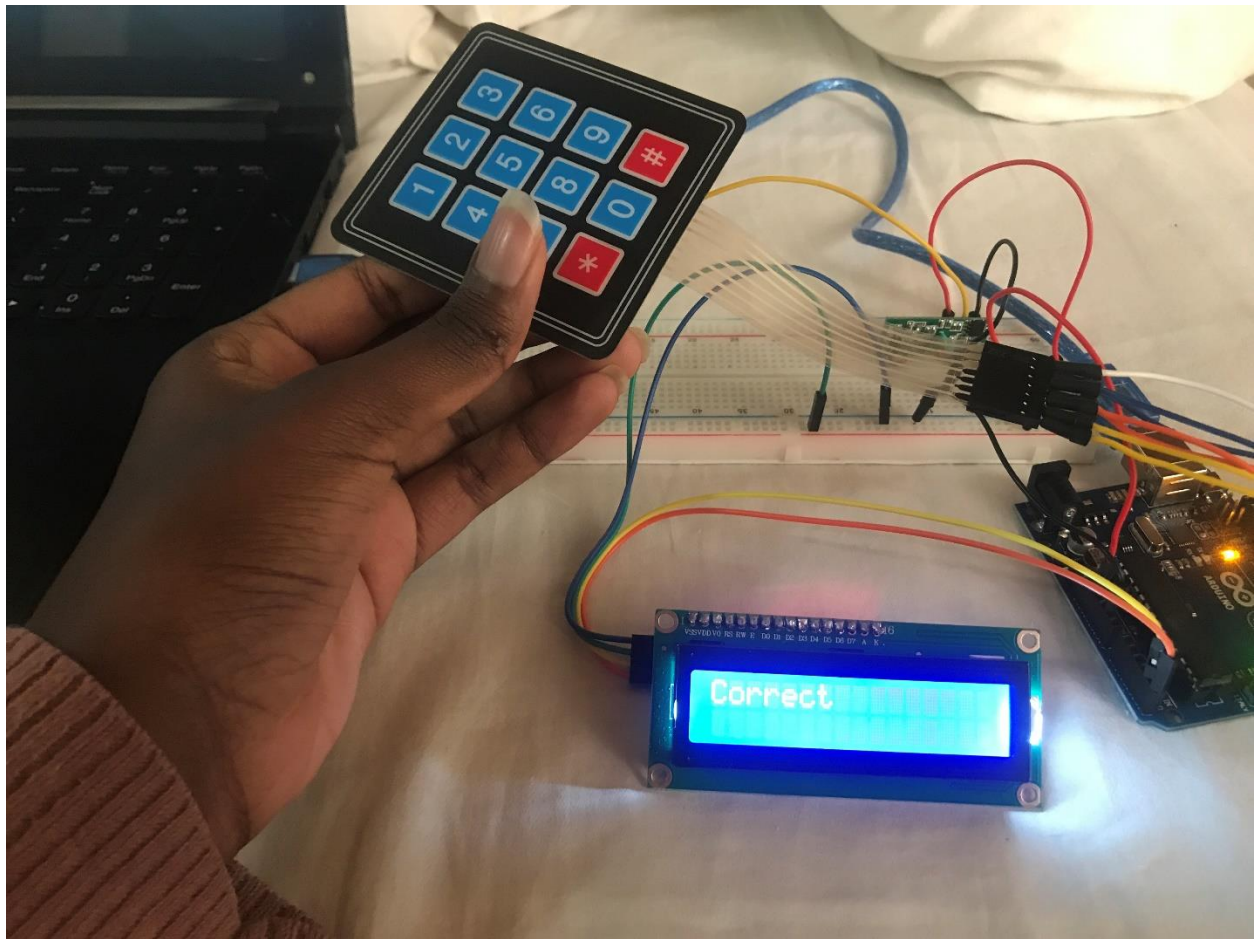


Figure 24 showing password validation

The message is display if and only if the password entered is correct



Figure 25 Showing the transmitted message

9.2 Source Code

THE TRANSMITTER CODE

```
#include<AES.h>
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile
#include " ./printf.h"

//aes encryption part
AES aes;

unsigned long long int my_iv = 01234567;
//byte cipher;
//byte confirm;
//int plainLength =sizeof(plain);

byte *key = (unsigned char*)"0123456789101112";

byte max_payload =0;

RH_ASK driver;
String serialdata; //Holds data coming in
//char msg[50]; //Char array to send data out

void encryptNow(char*data,byte bits,byte length)
{
    byte paddedLength =length +N_BLOCK -length% N_BLOCK;
    aes.iv_inc();
    byte iv[N_BLOCK];
    byte plain_paded[padedLength];
    byte cipher[padedLength];
    byte check[padedLength];
    unsigned long mirco_sec =micros();
    aes.set_IV(my_iv);
    aes.get_IV(iv);
    aes.do_aes_encrypt(data,length,cipher,key,bits,iv);

    //to check if encryption and decryption
    aes.set_IV(my_iv);
    aes.get_IV(iv);
    aes.do_aes_decrypt(cipher,padedLength,check,key,bits,iv);
    printf("\nciphertext: ");
    aes.printArray(cipher,(bool>false);
```

```

//sending the ciphertext to the reciever
driver.send((uint8_t *)cipher,sizeof(cipher));//Sending encrypted message
driver.waitPacketSent();

//printing the ciphered result
char result;
#define printable check
for (byte i = 0; i < sizeof(printable); ++i){

    result = printable[i];
}
Serial.println("*****");
}

byte max_payload=0;
void setup()
{
    Serial.begin(9600);    // Debugging only
    if (!driver.init()){

        Serial.println(F("init failed"));
    }

    printf_begin();
    max_payload =(driver.maxMessageLength()/N_BLOCK)*N_BLOCK;
    Serial.println(F("Enter Message: "));
    Serial.println("");
}

void loop()
{
    char data[128]="";
    byte bytes =0;
    if (Serial.available()){
        while(Serial.available()>0)
        {
            char buff;
            buff=Serial.read();
            if(buff!=0x0A && buff!=0x0D)
            {
                data[bytes]=buff;
                ++bytes;
            }
        }
    }
}

```

```

    delay(10);
}
data[bytes]=0x0;
Serial.print("plaintext: ");
Serial.println(data);
byte data_len=strlen(data)+1;
{
    if (data_len<max_payload)
    {
        encryptNow(data,128,data_len);
    }
    else
    {
        Serial.println(F("Exceeded the radiohead library maximum payload"));
    }
}

//driver.send((uint8_t *)msg, strlen(msg));
//driver.waitPacketSent();
//Serial.println ("Data Sent");

delay(1000);
}

```

THE RECEIVER CODE

```
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile
#include <Wire.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <AES.h>
#define Password_length 8

//for the encryption part
AES aes;
unsigned long long int my_iv = 01234567;
byte *key = (unsigned char*)"0123456789101112";

RH_ASK driver;
const byte coded_max_payload = 60;
String text = "";
const byte coded_max_char = coded_max_payload/N_BLOCK*N_BLOCK;
//encrypted;

//For the keypad password

char Data[Password_length];
int Master[Password_length] = {1,2,3,4,5,6,7};
byte data_count = 0 ,master_count = 0 ;
bool Pass_is_good;
char custom_key;

const byte ROWS =4; //its a 4x3 keypads hence 4 rows and 3 columns
const byte COLS =3;

char keys[ROWS][COLS]={
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte rowPins[ROWS]={2,3,4,5}; //connecting the row pinouts of the keypad
byte colPins[COLS]={6,7,8}; //connecting the cols pinouts

Keypad keypad=Keypad(makeKeymap(keys) ,rowPins , colPins, ROWS ,COLS);

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```

void clearData();

//Function for the password
int password()
{
    lcd.setCursor(0,0);
    lcd.print("New Message");
    delay(2000);

    for(int i=0; i<4; i++)
    {
        lcd.backlight();
        delay(250);
        lcd.noBacklight();
        delay(250);
    }
    //Turn backlight back on
    lcd.backlight();
    lcd.clear();
    delay(1000);

    lcd.setCursor(0,0);
    lcd.print("Enter Password:");
    delay(2000);

    int pass_word[7];
    int index=0;
    while(index<7)
    {
        custom_key = keypad.getKey();
        if(custom_key)
        {
            lcd.setCursor(index,1);
            pass_word[index]=custom_key;
            lcd.print("*");
            index++;
            Serial.print(custom_key);
        }
    }
    lcd.clear();
    int check=1;
    for(int i=0; i<=5; i++)
    {

```

```

        if(pass_word[i]-48==Master[i]){
            check*1;}

        else{
            check=check*0;
            Serial.print(pass_word[i]-48);
            Serial.print(" ");
            Serial.print(Master[i]);
            Serial.print(" ");
            Serial.println("Wrong");}
    }

    if(check == 1)
    {
        lcd.print("Correct");
        delay(2000);
        lcd.clear();
        clearData();
        return (1);
    }
    else{
        lcd.print("Incorrect");
        delay(2000);
        lcd.clear();
        clearData();
        //return 0;
        password();
    }
}

void clearData(){
    while(data_count !=0){
        Data[data_count--]=0;
    }
    return;
}

//function for decryption

void decryptNow(char*data,byte bits,byte*final_result)
{
    int plainLength =coded_max_char;
    int paddedLength = plainLength + N_BLOCK - plainLength % N_BLOCK;

```

```

aes.iv_inc();
byte iv [N_BLOCK] = "";

Serial.print(F("MaxSizevar: "));
Serial.print(paddedLength);
Serial.println("");
Serial.print(F("MaxSizemsg: "));
Serial.print(plainLength);
Serial.println("");
Serial.print(F("Ciphertext padded: "));
Serial.print(data);
char buffer;
#define printable data
for (byte i = 0; i < paddedLength; i++){
  buffer = printable[i];
}
Serial.println();

aes.set_IV(my_iv); //setting initial vector
aes.get_IV(iv); //geeting the initial vector

aes.do_aes_decrypt(data,paddedLength,final_result,key,bits,iv);
}

void setup()
{
  lcd.begin(16,2);
  Serial.begin(9600); // Debugging only
  if (!driver.init())
    Serial.println("init failed");
  byte max_payload =0;
  max_payload = (driver.maxMessageLength()/N_BLOCK)*N_BLOCK;
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0,0);

}

void loop()
{
  uint8_t buf[coded_max_payload];

```



```

uint8_t buflen = sizeof(buf);
if (driver.recv(buf, &buflen)) // Non-blocking
{
    void decryptNow(char*data ,byte bits ,byte*final_result);
    decryptNow(buf,128,buf);
    Serial.print(F("Message Decoded:"));
    Serial.print(F(""));

    byte i=0;
    char buffer;
    #define printable buf
    buffer = printable[i];

    if(buffer !=0){

        text=(char*)buf;
        Serial.println(text);
        lcd.clear();
        //lcd.print(text);

        char result = password();
        if( result == 1)
        {
            lcd.print(text);
            while (true)
            {
                custom_key = keypad.getKey();
                Serial.println(custom_key);
                if (custom_key == '*')
                {
                    break;
                }
            }
            lcd.clear();
        }
    }
}

delay(8000);
lcd.clear();
//lcd.write((char*)buf);
}

```