A secured communication system between two sides

Team Members:

اشرف عبدالخالق بسيوني عبدالعزيز	سكشن 1
ابانوب فهمى عطا الله	سكشن 1
احمد السيد محمد	سكشن 1
احمد مصطفى رجب مصطفى	سكشن 1
سيف الدين عماد الشامي	سكشن 2
فوزى ابراهيم عبدالرازق ابراهيم	سكشن 3
كريم الصاوي احمد	سكشن 3
عبدالسلام عثمان عبدالسلام الخضر	سكشن 3

Components

Name	Quantity	Price	Total
Arduino nano	2	175	350
Arduino nano cable	2	25	50
NRF24L01 Module	2	65	130
LCD 2x16	1	45	45
10k potentiometer	1	3.5	3.5
20cm male-male wires	20	35	35
20 cm male-female wires	20		
Breadboard	1	30	30
Total			643.5

Code (transmitter)

```
// include header files
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

// initialise global varaibles
const int CE = 9,
        CSN = 10;

const byte len = 16;
const char key[] = "IEEE802";

const uint64_t address = 0xA1B2C3D4;

// initialise rNF Module
RF24 radio(CE, CSN);

// loop function
void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  radio.begin();
  radio.enableDynamicPayloads();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}
```

```
// loop function
void loop() {
  String text;
  while (Serial.available()) {
    text = Serial.readString();
  }

  if(text.length() != 0){
    String string_cipher = Encode(text, text.length());
    char char_cipher[len];

    for(int i = 0; i < text.length(); i++){
      char_cipher[i] = string_cipher[i];
    }

    radio.write(char_cipher, text.length() - 1);

    digitalWrite(2, HIGH);
    delay(500);
    digitalWrite(2, LOW);

    Serial.print("text: ");
    Serial.println(text);
    Serial.print("cipher: ");
    Serial.println(string_cipher);
    Serial.print("");
  }
}

// get string from user function
String get_string() {
```

```
  String input;
  while (Serial.available() != 0) {
     input = Serial.readString();
  }
  return input;
}

// encryption function
String Encode(String text, int string_length) {
  int alpha,
     keyword,
     output,
     i = 0,
     j = 0;
  String cipher;

  while (i < string_length) {
    alpha = text[i];
    keyword = key[j];
    output = ((alpha - 65) + (keyword - 65));
    cipher += char(output);
    if(sizeof(key) - 1 >= i){
      j = 0;
    }
    i++;
  }
  return cipher;
}
```

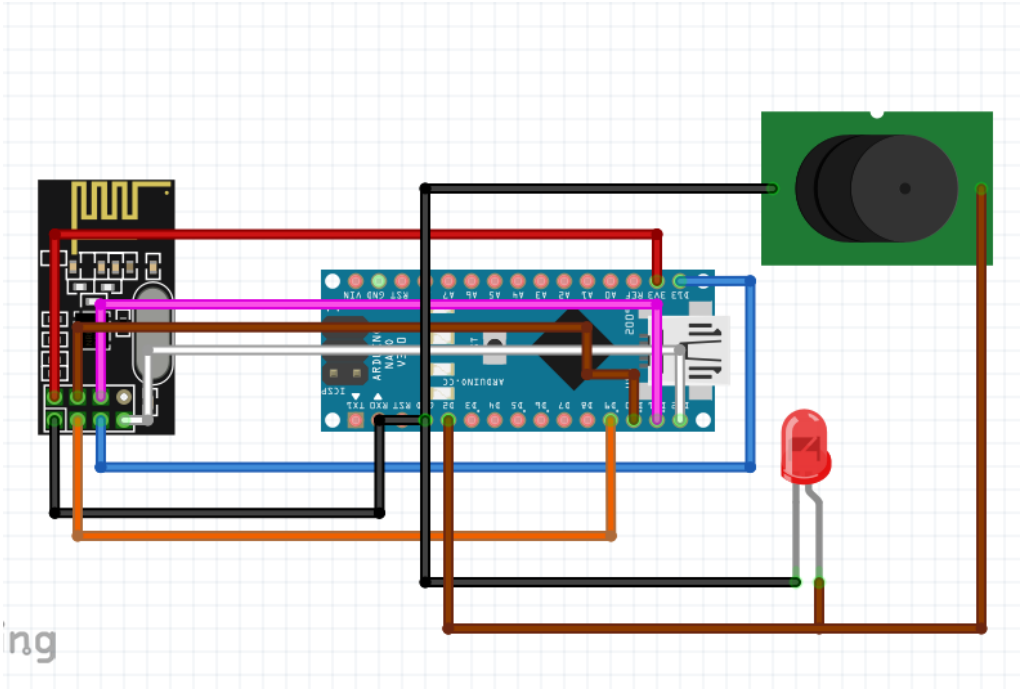
Code (receiver)

```
// include header files
#include <LiquidCrystal.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

// initialise global varaibles
const int rs = 3,
        en = 4,
        d4 = 5,
        d5 = 6,
        d6 = 7,
        d7 = 8,
        CE = 9,
        CSN = 10;

const char key[] = "IEEE802";

const uint64_t address = 0xA1B2C3D4;

// initialise rNF Module
RF24 radio(CE, CSN);

// initialise LCD screen
LiquidCrystal LCD(rs, en, d4, d5, d6, d7);

// setup function
void setup() {
```

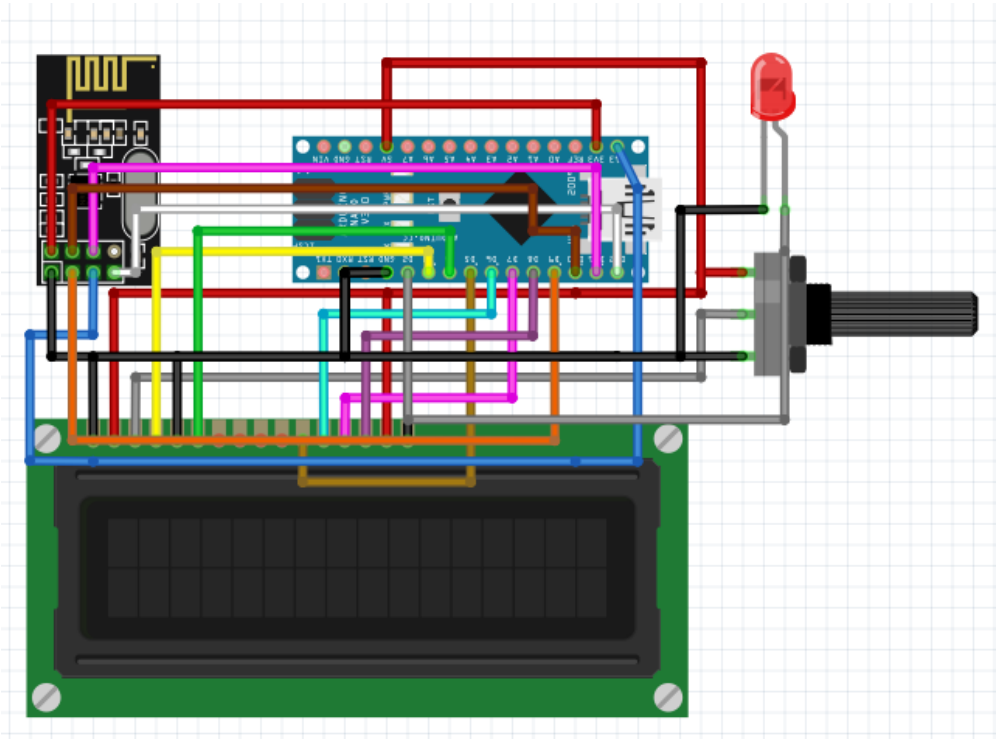
```
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  radio.begin();
  radio.enableDynamicPayloads();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();

  LCD.begin(16, 2);
  LCD.print("   System On");
  delay(2500);
  LCD.clear();
}

// loop function
void loop() {
  if (radio.available()) {
    uint8_t len = radio.getDynamicPayloadSize();
    char cipher[len];

    radio.read(&cipher, len);
    String text = Decode(cipher, len);

    digitalWrite(2, HIGH);
    delay(500);
    digitalWrite(2, LOW);

    LCD.clear();
    LCD.print("Dx: ");
    LCD.setCursor(0, 1);
    LCD.print("Ex: ");
```

```
    LCD.setCursor(3, 0);
    LCD.print(cipher);
    LCD.setCursor(3, 1);
    LCD.print(text);
  }
}

// decryption function
String Decode(char cipher[], int string_length) {
  int alpha,
      keyword,
      output,
      i = 0,
      j = 0;
  String text;

  while (i < string_length) {
    alpha = cipher[i];
    keyword = key[j];
    output = ((alpha + 65) - (keyword - 65));
    text += char(output);
    if(sizeof(key) - 1 >= i){
      j = 0;
    }
    i++;
  }
  return text;
}
```

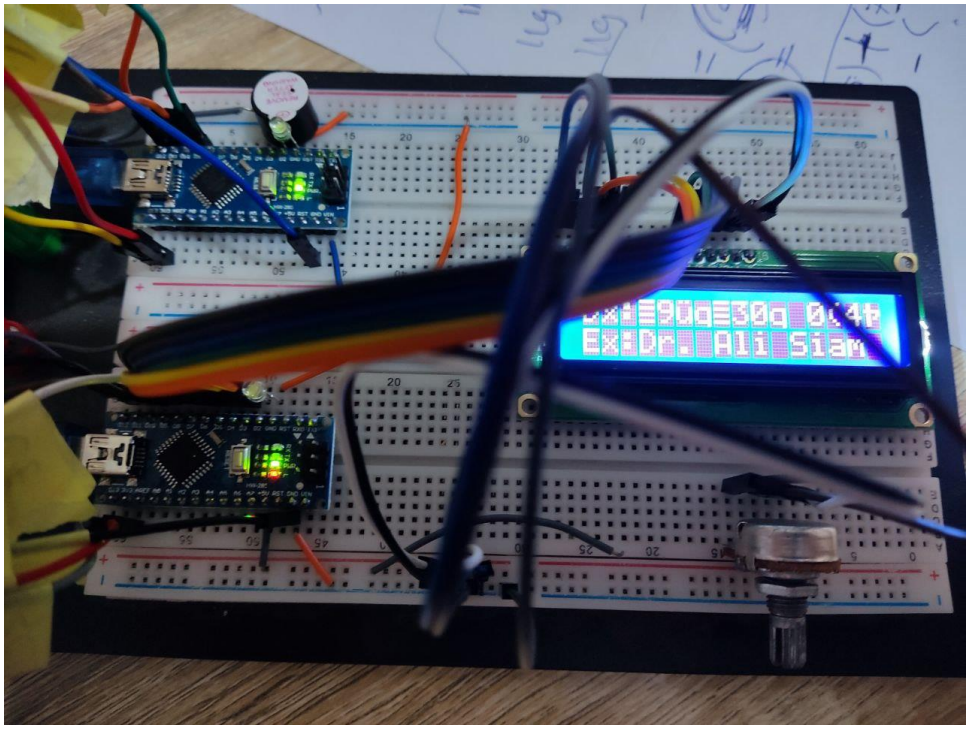
Circuit (transmitter)



Circuit (receiver)



Output



Note:

Dx: is the cipher text

Ex: is the original text after decryption