

2023 Arduino Paragliding-Drone

한세대학교 유제원

LoRa module E220-900T30D 원거리 통신 테스트

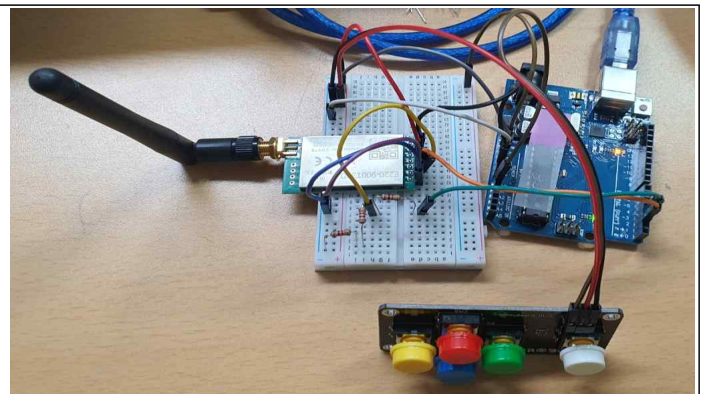
- Arduino와 LoRa module(E220-900T30D)를 사용하여 원거리 데이터 통신 테스트
- 공식 문서를 통한 주파수 규격을 지키며 사용가능한 주파수 주역대를 사용하여 테스트 진행
- 사용되는 pin은 RXD TXD 통신, 전원(5V, 3.3V GND)등이 사용된다.
- E220-900T30D의 pin중 M0, M1은 설정모드와 사용모드에 따라 3.3V와 GND로 연결하여야한다.
- 제품의 통신 가능한 거리로 10km를 확인하였고 그중 교내에서 테스트 가능한 거리인 400m까지 통신에 이상이 없다는 것을 확인하였다. 건물 내부에서 다른 건물 내부만 아니라면 통신에 크게 지장이 없어 보인다.
- 와이파이나 블루투스처럼 통신거리에 제약이 있는 제품보단 확실히 뛰어나다고 보인다.

재료:

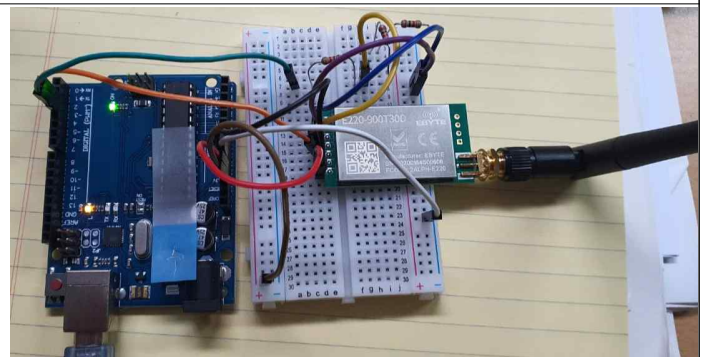
- 아두이노 UNO 보드 2개
- LoRa module : E220-900T30D + 안테나 별도
- 전압 5V~3.3V , RXD 통신용 1k 옴 3개
- 사용 주파수 : 921.3 Mhz
- 최대 통신가능거리 10km, 테스트 거리 400m 이상없음
- 브레드보드2개
- 조이스틱 모듈 [ELB070681], 버튼 모듈 [ELB030639]으로 다양한 값 전송



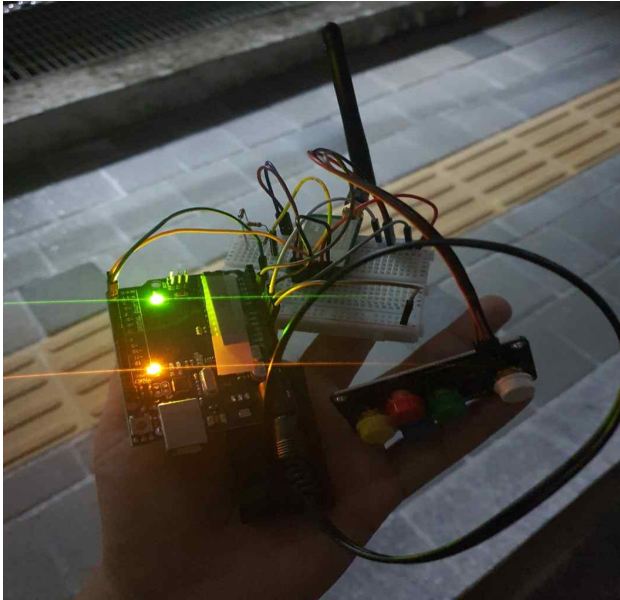
E220-900T30D



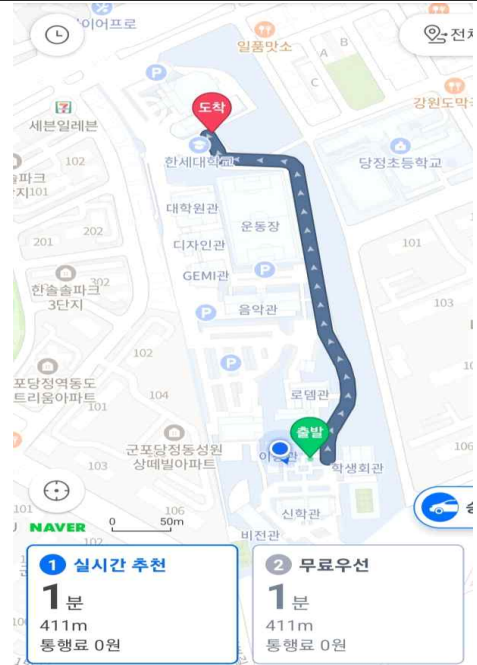
송신기



수신기



송신기 외부 통신 테스트



FD04A 4채널 모터 드라이버 원격제어

- Arduino와 4채널 모터드라이버(FD04A)를 사용하여 모터제어 테스트
- 위의 LoRa 모듈의 통신으로 수신한 데이터의 값에 따라 모터를 제어
- 위 수신기 모듈에 추가로 모터드라이버를 추가하여 진행
- 버튼 4개를 사용하여 모터1개씩 별도 제어, 조이스틱 클릭으로 정지

재료:

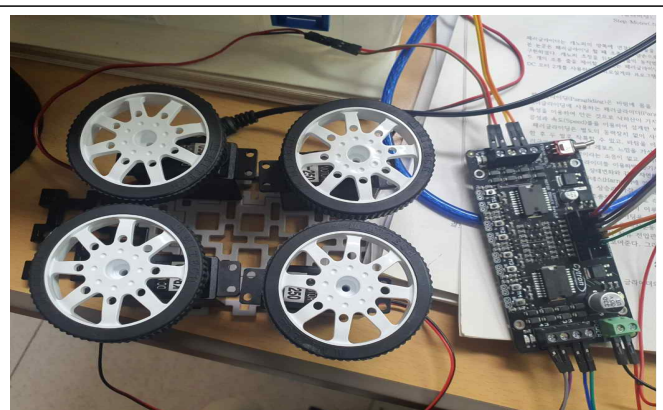
아두이노 LoRa수신기 1개

4채널 모터드라이버(FD04A) 1개

DC모터 4개



송신기 조이스틱 버튼 모듈[ELB070681]



모터드라이버[FD04A] 모터 제어

사용한 코드

```
// LoRa systemSetting Code
```

```
#include <Arduino.h>
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial lora(2,3);
```

```
byte send_buff[4];
```

```
byte recv_buff[4];
```

```
void setup() {
```

```
    // put your setup code here, to run once:
```

```
    Serial.begin(9600); //PC하고 아두이노하코의 통신라인!
```

```
    lora.begin(9600); //아두이노하코 로라모듈하코의 통신라인!
```

```
}
```

```
void loop() {
```

```
    // put your main code here, to run repeatedly:
```

```
    if(Serial.available()){
```

```
        char c = Serial.read();
```

```
        if(c == '0'){
```

```
            //ADDH부터 request해보자!
```

```
            send_buff[0] = 0xC1;
```

```
            send_buff[1] = 0x00;
```

```
            send_buff[2] = 0x01;
```

```
            lora.write(send_buff,3); //전송!
```

```
            lora.readBytes(recv_buff,4);
```

```
            if(recv_buff[0] == send_buff[0] && recv_buff[1] == send_buff[1] && recv_buff[2] ==  
send_buff[2]){
```

```
                Serial.print("ADDH=");
```

```
                Serial.println(recv_buff[3],HEX);
```

```
            }else{
```

```
                Serial.println("에러");
```

```
            }
```

```
        }else if(c == '1'){
```

```
            send_buff[0] = 0xC1;
```

```
            send_buff[1] = 0x01;
```

```
            send_buff[2] = 0x01;
```

```
            lora.write(send_buff,3); //전송!
```

```
            lora.readBytes(recv_buff,4);
```

```
            if(recv_buff[0] == send_buff[0] && recv_buff[1] == send_buff[1] && recv_buff[2] ==
```

```
send_buff[2]){  
    Serial.print("ADDL=");  
    Serial.println(recv_buff[3],HEX);  
}else{  
    Serial.println("에러");  
}  
}else if(c == '2'){  
    send_buff[0] = 0xC1;  
    send_buff[1] = 0x04;  
    send_buff[2] = 0x01;  
    lora.write(send_buff,3); //전송!  
    lora.readBytes(recv_buff,4);  
  
    if(recv_buff[0] == send_buff[0] && recv_buff[1] == send_buff[1] && recv_buff[2] ==  
send_buff[2]){  
        Serial.print("CH=");  
        Serial.print(850.125 + recv_buff[3]);  
        Serial.println("Mhz");  
    }else{  
        Serial.println("에러");  
    }  
}else if(c == '3'){  
    send_buff[0] = 0xC1;  
    send_buff[1] = 0x05;  
    send_buff[2] = 0x01;  
    lora.write(send_buff,3); //전송!  
    lora.readBytes(recv_buff,4);  
  
    if(recv_buff[0] == send_buff[0] && recv_buff[1] == send_buff[1] && recv_buff[2] ==  
send_buff[2]){  
        Serial.println(recv_buff[3],BIN);  
    }else{  
        Serial.println("에러");  
    }  
}else if(c == '4'){  
    //주파수를 설정하는 모드!  
    send_buff[0] = 0xC0;  
    send_buff[1] = 0x04;  
    send_buff[2] = 0x01;  
    send_buff[3] = 0x47; //내가 원하는 주파수를 여기다가 넣기!  
    lora.write(send_buff,4); //전송!  
    lora.readBytes(recv_buff,4);  
  
    if(recv_buff[1] == send_buff[1] && recv_buff[2] == send_buff[2]){  
        Serial.println(recv_buff[3],HEX);
```

```

    }else{
        Serial.println("에러");
    }
}
else if(c == '5'){
    //T모드로 설정하기!
    send_buff[0] = 0xC0;
    send_buff[1] = 0x05;
    send_buff[2] = 0x01;
    send_buff[3] = 0b00000011;
    lora.write(send_buff,4); //전송!
    lora.readBytes(recv_buff,4);

    if(recv_buff[1] == send_buff[1] && recv_buff[2] == send_buff[2]){
        Serial.println(recv_buff[3],BIN);
    }else{
        Serial.println("에러");
    }
}
else if(c == '6'){
    //F모드로 설정하기!
    send_buff[0] = 0xC0;
    send_buff[1] = 0x05;
    send_buff[2] = 0x01;
    send_buff[3] = 0b01000011;
    lora.write(send_buff,4); //전송!
    lora.readBytes(recv_buff,4);

    if(recv_buff[1] == send_buff[1] && recv_buff[2] == send_buff[2]){
        Serial.println(recv_buff[3],BIN);
    }else{
        Serial.println("에러");
    }
}
else if(c == '7'){
    //ADDH를 설정하기
    send_buff[0] = 0xC0;
    send_buff[1] = 0x00;
    send_buff[2] = 0x01;
    send_buff[3] = 0x00; //주소
    lora.write(send_buff,4); //전송!
    lora.readBytes(recv_buff,4);

    if(recv_buff[1] == send_buff[1] && recv_buff[2] == send_buff[2]){
        Serial.println(recv_buff[3],BIN);
    }else{
        Serial.println("에러");
    }
}

```

```

}else if(c == '8'){
  //ADDL를 설정하기
  send_buff[0] = 0xC0;
  send_buff[1] = 0x01;
  send_buff[2] = 0x01;
  send_buff[3] = 0x02; //주소
  lora.write(send_buff,4); //전송!
  lora.readBytes(recv_buff,4);

  if(recv_buff[1] == send_buff[1] && recv_buff[2] == send_buff[2]){
    Serial.println(recv_buff[3],BIN);
  }else{
    Serial.println("에러");
  }
}
}
}
}

```

송신기 코드

```

#include <Arduino.h>
#include <SoftwareSerial.h>

SoftwareSerial lora(2, 3);

const int joyXPin = A1; // X 좌표 핀
const int joyYPin = A0; // Y 좌표 핀
const int joyKPin = 8; // S-K 핀

const int buttonPin1 = 4; // 버튼 1 핀
const int buttonPin2 = 5; // 버튼 2 핀
const int buttonPin3 = 6; // 버튼 3 핀
const int buttonPin4 = 7; // 버튼 4 핀

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600); // PC와의 시리얼 통신
  lora.begin(9600); // LoRa 모듈과의 시리얼 통신

  pinMode(joyKPin,INPUT_PULLUP);
  pinMode(buttonPin1, INPUT_PULLUP);
  pinMode(buttonPin2, INPUT_PULLUP);
  pinMode(buttonPin3, INPUT_PULLUP);
  pinMode(buttonPin4, INPUT_PULLUP);
}

```

```
}
```

```
void loop()
```

```
{
```

```
  int xValue = analogRead(joyXPin);
```

```
  int yValue = analogRead(joyYPin);
```

```
  float mappedX = map(xValue, 0, 1023, 0, 100);
```

```
  float mappedY = map(yValue, 0, 1023, 0, 100);
```

```
  // 버튼 상태 읽기 (HIGH는 버튼이 눌리지 않은 상태)
```

```
  bool button1 = digitalRead(buttonPin1) == HIGH;
```

```
  bool button2 = digitalRead(buttonPin2) == HIGH;
```

```
  bool button3 = digitalRead(buttonPin3) == HIGH;
```

```
  bool button4 = digitalRead(buttonPin4) == HIGH;
```

```
  bool joyK = digitalRead(joyKPin) == HIGH;
```

```
  // 값이 변화했을 때만 출력
```

```
  Serial.print("X: ");
```

```
  Serial.print(mappedX);
```

```
  Serial.print("\tY: ");
```

```
  Serial.print(mappedY);
```

```
  Serial.print("\tButton1: ");
```

```
  Serial.print(button1);
```

```
  Serial.print("\tButton2: ");
```

```
  Serial.print(button2);
```

```
  Serial.print("\tButton3: ");
```

```
  Serial.print(button3);
```

```
  Serial.print("\tButton4: ");
```

```
  Serial.println(button4);
```

```
  Serial.print("\tJoy-K: ");
```

```
  Serial.println(joyK);
```

```
  if(button1 == 0)
```

```
  {
```

```
    lora.println("1");
```

```
    delay(100);
```

```
  }
```

```
  else if(button2 == 0)
```

```
  {
```

```
    lora.println("2");
```

```
    delay(100);
```

```
  }
```

```
  else if(button3 == 0)
```

```
  {
```

```
    lora.println("3");
    delay(100);
}
else if(button4 == 0)
{
    lora.println("4");
    delay(100);
}
else if(joyK == 0)
{
    lora.println("5");
    delay(100);
}
    delay(100);
}
```

수신기 코드

```
#include <Arduino.h>
#include <SoftwareSerial.h>
```

```
SoftwareSerial lora(2, 3);
```

```
const int motorA = 4;
const int motorB = 7;
const int motorC = 8;
const int motorD = 11;
```

```
const int motorAPWM = 5;
const int motorBPWM = 6;
const int motorCPWM = 9;
const int motorDPWM = 10;
```

```
void StopAllMotors();
```

```
void setup()
{
    Serial.begin(9600);
    lora.begin(9600);

    pinMode(motorA, OUTPUT);
    pinMode(motorB, OUTPUT);
    pinMode(motorC, OUTPUT);
```



```
pinMode(motorD, OUTPUT);

// 초기에 모든 모터 정지
StopAllMotors();
}

void Motor1()
{
    digitalWrite(motorA, HIGH);
    analogWrite(motorAPWM, 100);
}

void Motor2()
{
    digitalWrite(motorB, HIGH);
    analogWrite(motorBPWM, 100);
}

void Motor3()
{
    digitalWrite(motorC, HIGH);
    analogWrite(motorCPWM, 100);
}

void Motor4()
{
    digitalWrite(motorD, HIGH);
    analogWrite(motorDPWM, 100);
}

void StopAllMotors()
{
    digitalWrite(motorA, LOW);
    digitalWrite(motorB, LOW);
    digitalWrite(motorC, LOW);
    digitalWrite(motorD, LOW);

    analogWrite(motorAPWM, 0);
    analogWrite(motorBPWM, 0);
    analogWrite(motorCPWM, 0);
    analogWrite(motorDPWM, 0);
}

void loop()
{

```

```

if (lora.available())
{
    String data = lora.readStringUntil('\n');
    int intData = data.toInt();
    Serial.println(data);
    if (intData == 1)
    {
        StopAllMotors(); // 모든 모터 정지
        Motor1(); // 모터 1 작동
    }
    else if (intData == 2)
    {
        StopAllMotors();
        Motor2();
    }
    else if (intData == 3)
    {
        StopAllMotors();
        Motor3();
    }
    else if (intData == 4)
    {
        StopAllMotors();
        Motor4();
    }
    else
    {
        StopAllMotors(); // 알 수 없는 데이터가 오면 모든 모터 정지
    }
}
}

```

코드 참고 <https://github.com/youjewon/embedded-study/tree/main/TestLoRa>

설정코드 : exLoRa.txt

송신기 코드 : exjoystick.txt

수신기 코드 : exMotor.txt

를 참고해 주시면 됩니다.