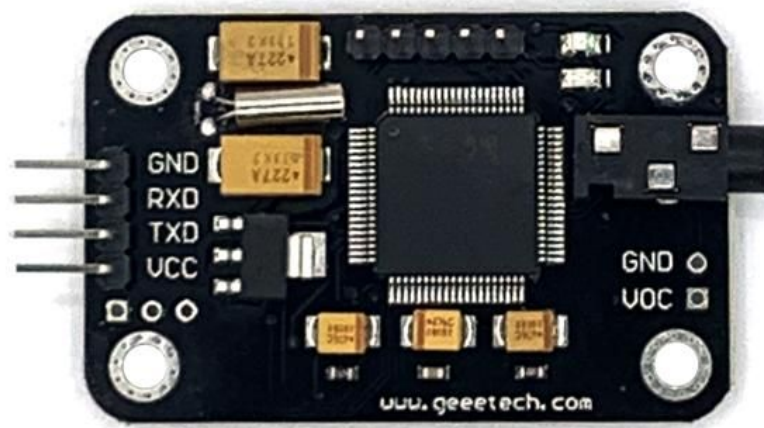


# 음성 인식 모듈 매뉴얼

## Voice Recognition Module manual

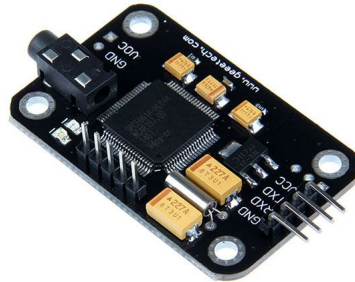


# 목차

<b>음성 인식 모듈 사용법</b>	<b>3</b>
1.1 음성 인식 모듈 스펙	3
1.2 음성 인식 명령어 저장	4
<b>음성 인식 모듈 - 아두이노 사용 예제</b>	<b>8</b>
2.1 음성 인식 모듈로 RGB LED 와 서보모터 제어	8
2.1.1 준비물	8
2.1.2 회로도	9
2.1.3 소스코드	10
2.2 음성 인식 모듈로 도트 매트릭스 제어	12
2.2.1 준비물	12
2.2.2 회로도	13
2.2.3 소스코드	14

# 1. 음성 인식 모듈 사용법

## 1.1 음성 인식 모듈 스펙



- 전압 : 4.5~5VDC
- 전류 : < 40mA
- 디지털 인터페이스 : 5V TTL Level UART interface
- 아날로그 인터페이스 : 3.5mm mono-channel microphone connector
- 사이즈 : 30mm x 47.4mm

음성 인식 모듈은 최대 15개의 음성 명령을 저장할 수 있습니다.

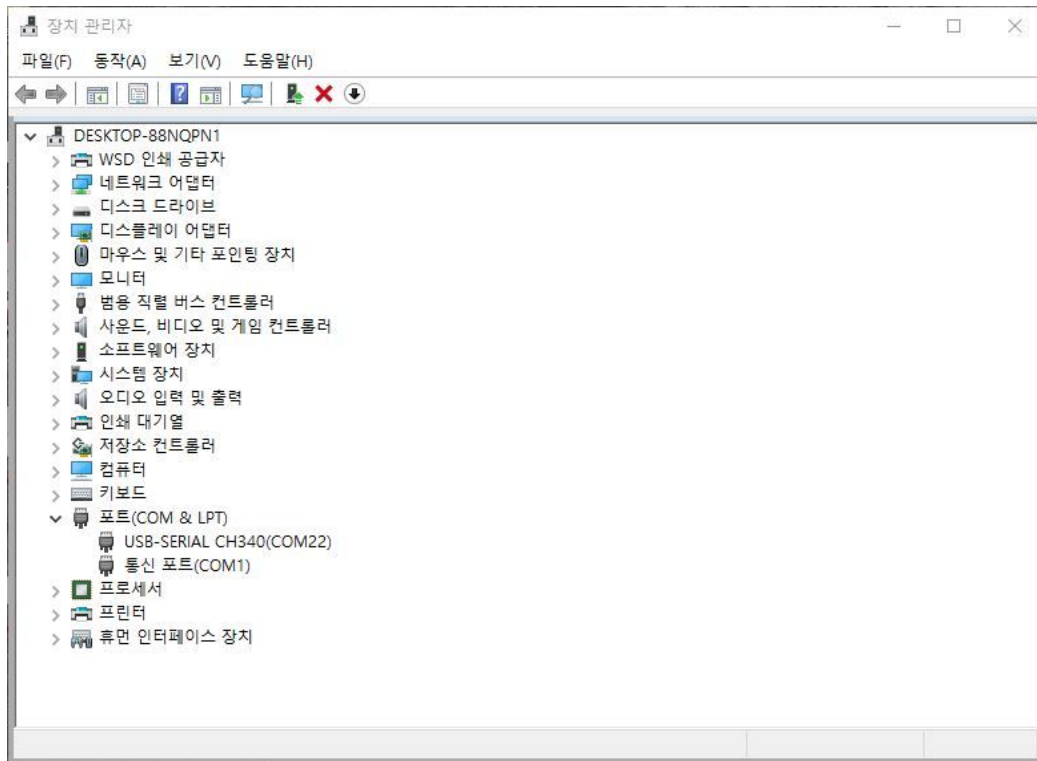
음성 명령은 3개의 그룹으로 나뉘며, 각 5개의 음성 명령이 저장됩니다.

먼저 그룹 별로 음성 명령어를 모듈에 저장 한 후, 사용할 그룹을 선택해야 합니다.

모듈을 PC와 연결하기 위해서 USB to TTL 컨버터와 모듈을 연결해야 합니다.

음성 인식 모듈	USB to TTL 컨버터
VCC	VCC(+5V)
GND	GND
TxD	RxD
RxD	TxD

모듈과 USB to TTL 컨버터 모듈 연결할 때는 Rx와 Tx가 교차되어야 합니다.



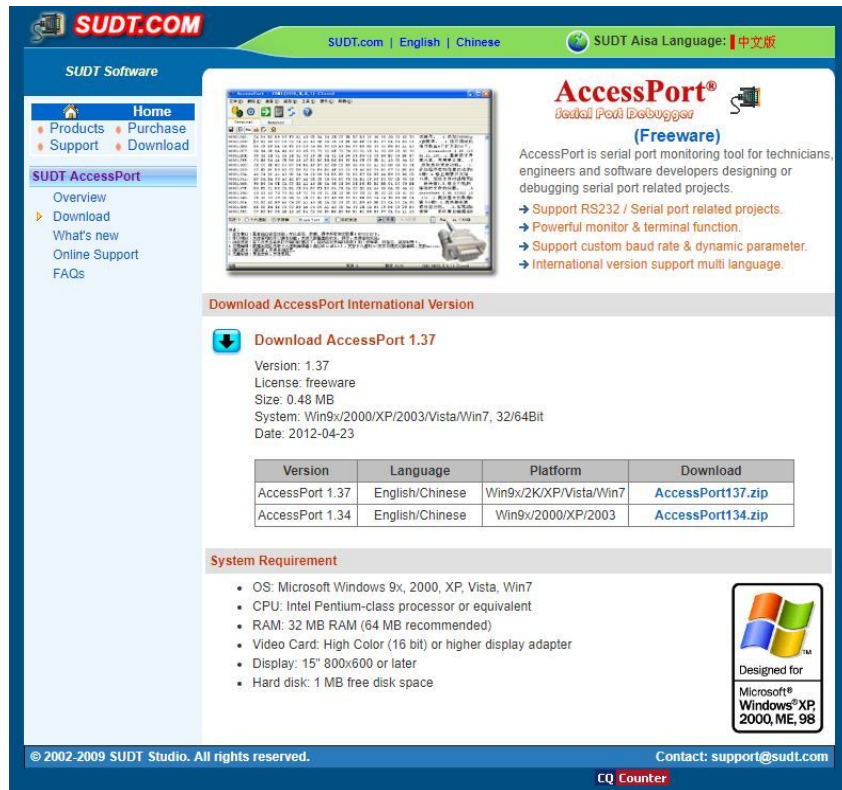
음성 인식 모듈과 USB to TTL 컨버터를 연결한 후 PC의 장치관리자에서 포트 번호를 확인 할 수 있습니다. 저의 경우에는 포트 번호가 COM22입니다.

## 1.2 음성 인식 명령어 저장

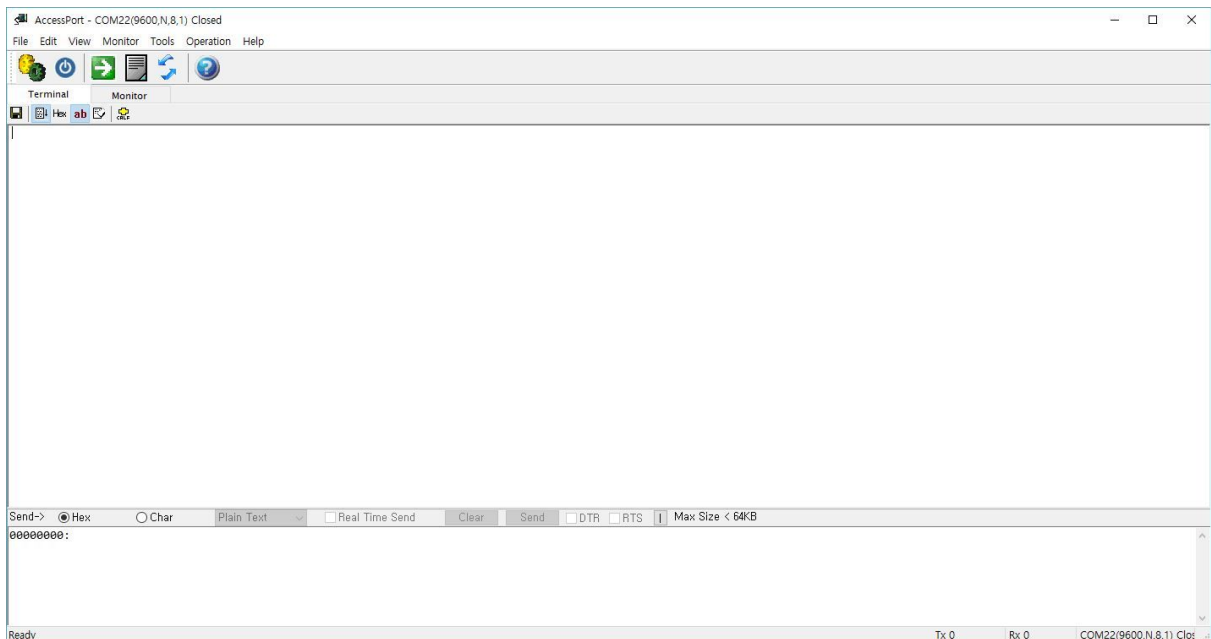
음성 인식 모듈의 [데이터 시트](#)에서는 PC와의 연결때, [AccessPort](#)라는 시리얼 통신 프로그램 사용을 권하고 있지만 다른 시리얼 통신 프로그램을 사용해도 가능합니다.

이 매뉴얼에서는 음성 인식 모듈의 데이터 시트에 권장하는 AceesPort라는 프로그램을 사용하며 음성 인식 명령어를 녹음하는 방법에 대해 알려드리겠습니다.

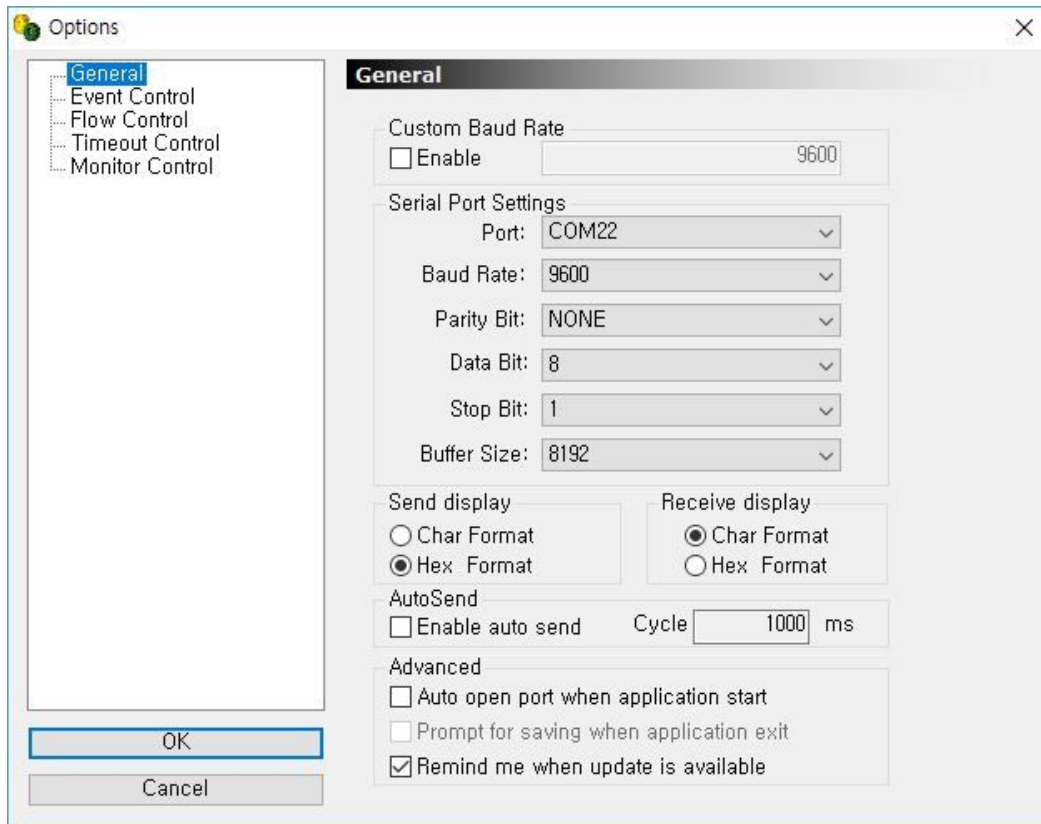
우선 링크를 클릭해주시면 프로그램을 다운 받을 수 있는 페이지로 이동하게 됩니다.



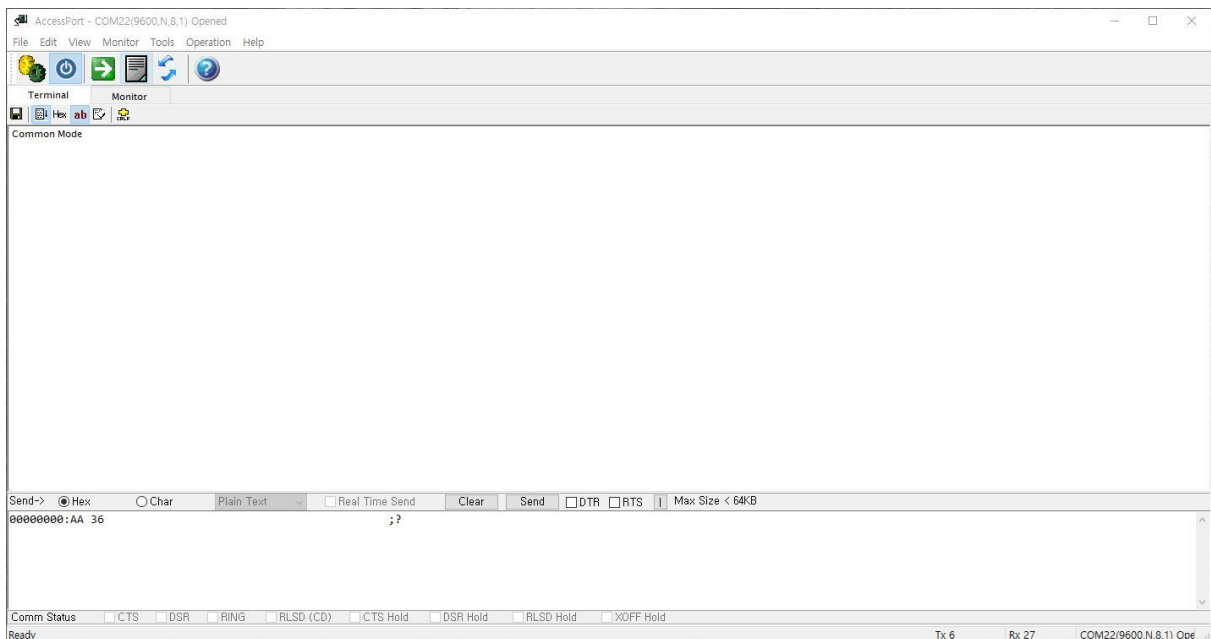
오른쪽 하단에 있는 AccessPort137.zip을 클릭해서 다운받아준 후, 압축을 해제하면 AccessPort.exe파일이 있습니다. 해당 프로그램을 실행시켜줍니다.



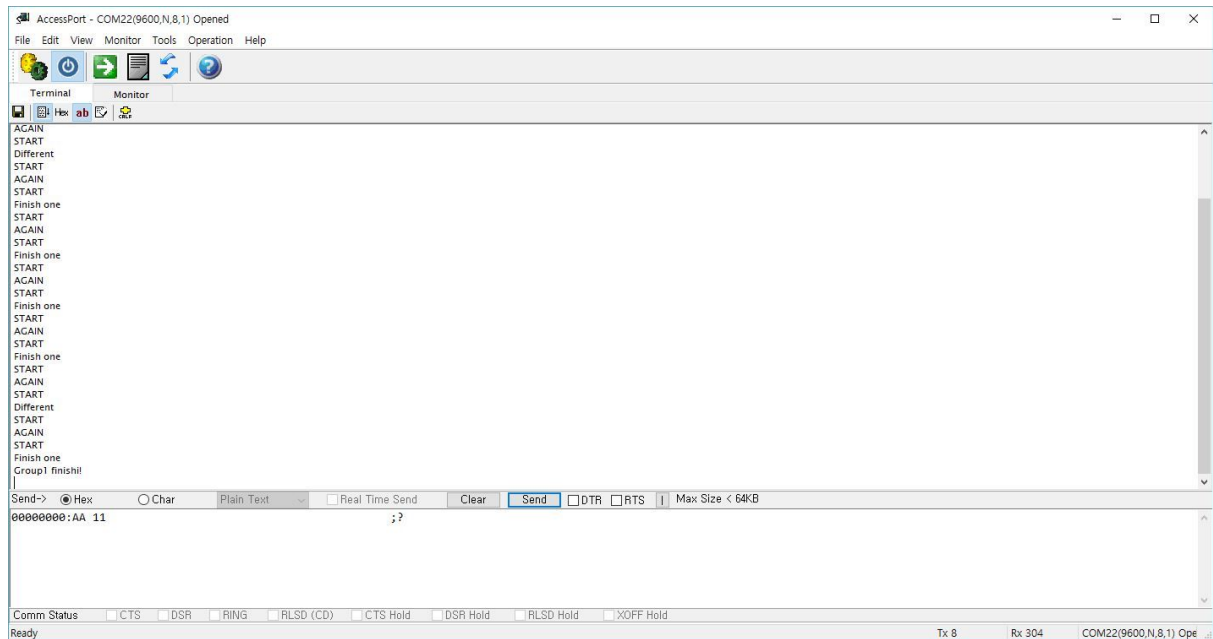
AccessPort 프로그램을 실행 시킨 후, 왼쪽 상단을 보면 노란색과 초록색의 톱니바퀴 모양의 Configuration 아이콘을 눌러줍니다.



그럼 Options 창이 뜹니다. 여기서 USB to TTL 컨버터와 연결된 Port 번호를 입력하고, Baud Rate는 9600으로 맞춰줍니다. Send display는 Hex format으로 Receive display는 Char Format으로 설정해준후 OK 버튼을 눌러줍니다.



이 후 Hex 명령어로 'AA36'을 Send 버튼을 통해 보내주었을 때, Common Mode라는 답변이 온다면, 제대로 연결된 것입니다.

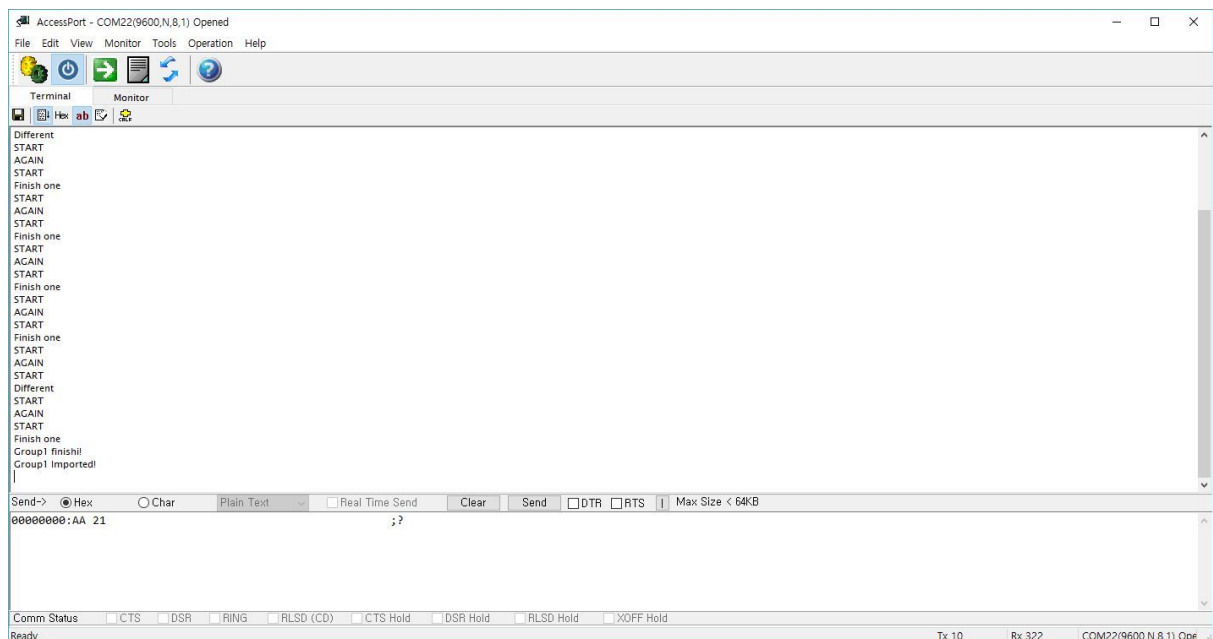


Hex 명령어로 'AA11'을 보내주시면, 음성 명령어 녹음이 시작됩니다. 음성 인식 모듈의 마이크에 대고 명령어를 녹음해주시면 됩니다. 각 명령의 길이는 최대 1300ms(1.3초)이므로, 단어로 명령을 넣는게 좋습니다. 녹음을 시작하면, 한 그룹의 5가지 음성 명령 녹음을 끝내기 전까지 녹음 과정을 멈출 수 없는점 유의해 주시기 바랍니다.

START 라는 답변이 오면 명령어를 녹음해주시면 됩니다.

No Voice, Different등을 통해 이전 녹음과 다른지, 녹음이 되고 있는지를 확인 가능합니다.

Finish one이 나오면 1개의 명령어가 녹음되었다는 뜻입니다. 순차적으로 5개가 모두 녹음 완료되면, Group 1 finish라는 답변이 옵니다.



이 후 Hex 명령어로 'AA21'을 보내주시면, Group1 Imported라는 답변이 옵니다.

첫번째 그룹의 5개의 명령어 입력이 끝났습니다.

사용 가능 명령어 리스트입니다.

이외의 추가적인 명령어는 데이터 시트를 참고하시기 바랍니다.

- Delete Group 1 - 0xAA01
- Delete Group 2 - 0xAA02
- Delete Group 3 - 0xAA03
- Delete All Groups - 0xAA04
- Record Group 1 - 0xAA11
- Record Group 2 - 0xAA12
- Record Group 3 - 0xAA13
- Import Group 1 - 0xAA21
- Import Group 2 - 0xAA22
- Import Group 3 - 0xAA23
- Switch Common Mode - 0xAA36
- Switch Compact Mode - 0xAA37

Common Mode는 음성 명령값을 'Result:11'과 같은 ASCII 코드 값으로 전달해 주지만, Compact Mode는 음성 명령값을 '11'과 같은 Hex값으로 전달해줍니다.

따라서 처음 PC와 음성 인식 모듈을 연결하여, 음성 명령을 녹음 할 때는 Common Mode로 녹음하는게 좋고, 아두이노등의 플랫폼에서 사용 할때는 Compact Mode로 사용해야 합니다.

음성 명령어는 녹음할 때 사용자의 목소리 위주로 인식 하기 때문에, 다른 사람이 명령어를 넣으면 제대로 작동하지 않을 수 있습니다.

## 2. 음성 인식 모듈 - 아두이노 사용 예제

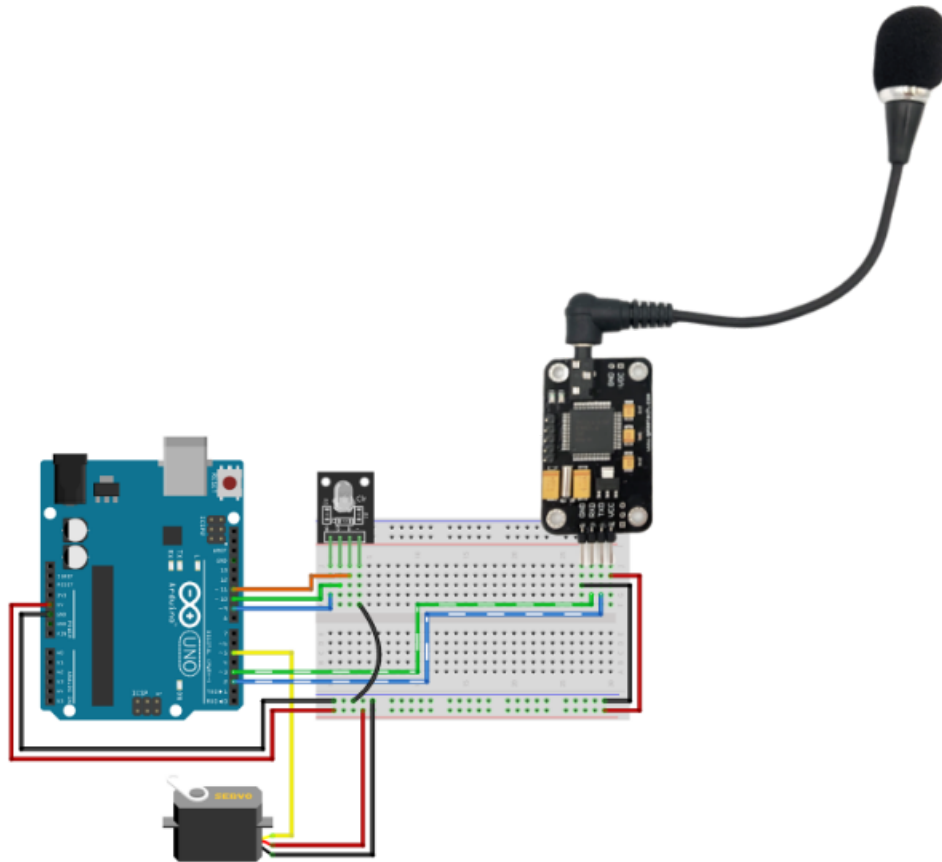
### 2.1 음성 인식 모듈로 RGB LED 와 서보모터 제어

#### 2.1.1 준비물

- 음성 인식 모듈
- 아두이노 우노
- USB to TTL 컨버터 모듈
- 브레드 보드
- 점퍼선
- RGB LED 모듈
- 서보 모터(SG90)



## 2.1.2 회로도



아두이노	음성 인식 모듈	RGB LED 모듈	서보모터
5V	VCC	-	VCC
GND	GND	GND	GND
D2	TxD	-	-
D3	RxD	-	-
D5	-	-	Signal
D9	-	Blue pin	-
D10	-	Green pin	-
D11	-	Red pin	-

### 2.1.3 소스코드

```
/*
Voice Recognition Module test
http://www.devicemart.co.kr/
*/

#include<Servo.h> // 서보 모터 라이브러리 사용
#include <SoftwareSerial.h> // 소프트웨어 Serial 통신 라이브러리 사용

Servo myservo; // 서보모터 객체를 myservo로 생성
Serial 통신핀으로 D2번핀을 Rx로, D3번핀을 Tx로 선언
SoftwareSerial mySerial(2, 3); // 모듈 TX=2번핀 , RX=3번핀 연결

int redled = 11; // RGB led 모듈의 redpin을 D11에 연결
int greenled = 10; // RGB led 모듈의 greenpin을 D10에 연결
int blueled = 9; // RGB led 모듈의 bluepin을 D9에 연결
byte voice_recogn = 0; // 음성 인식 모듈 반응 변수 선언

void setup()
{
  Serial.begin(9600); // 통신 속도 9600bps로 PC와 시리얼 통신 시작
  myservo.attach(5); // 서보모터 signal pin을 D5에 연결
  myservo.write(0);
  mySerial.begin(9600); // 통신 속도 9600bps로 모듈과 시리얼 통신 시작

  Serial.println("wait settings are in progress");

  pinMode(redled, OUTPUT); // RGB모듈 출력 모드 실행
  pinMode(greenled, OUTPUT);
  pinMode(blueled, OUTPUT);
  delay(1000);
  mySerial.write(0xAA); // compact mode 사용
  mySerial.write(0x37);
  delay(1000);
  mySerial.write(0xAA); // 그룹1 음성 명령어 imported
  mySerial.write(0x21);

  Serial.println("The settings are complete");
}

void loop()
{
  while(mySerial.available())
  {
    Serial.println("voice input");
    voice_recogn = mySerial.read();
  }
}
```

```

switch(voice_recogn)
{
    case 0x11:
        color(255, 0, 0); // 사용자 정의 함수 color 사용
        Serial.println("red turn on");
        break;

    case 0x12:
        color(0, 255, 0);
        Serial.println("green turn on");
        break;

    case 0x13:
        color(0, 0, 255);
        Serial.println("blue turn on");
        break;

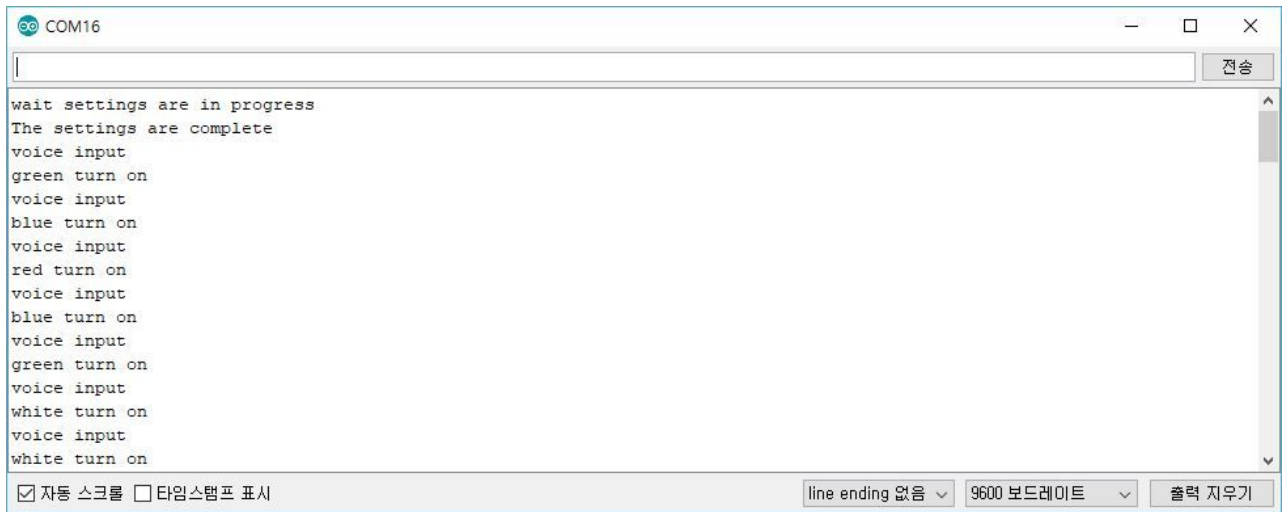
    case 0x14:
        color(255, 255, 255);
        Serial.println("white turn on");
        myservo.write(180);
        break;

    case 0x15:
        color(0, 0, 0);
        Serial.println("led off");
        myservo.write(0);
        break;
}
}

void color (int red, int green, int blue) // 색 변환 함수
{
    analogWrite(redled, red);
    analogWrite(blueled, blue);
    analogWrite(greenled, green);
}

```

아두이노의 시리얼 모니터를 사용하면, 음성 명령이 들어갔는지 확인 가능합니다.



COM16

```
wait settings are in progress
The settings are complete
voice input
green turn on
voice input
blue turn on
voice input
red turn on
voice input
blue turn on
voice input
green turn on
voice input
white turn on
voice input
white turn on
```

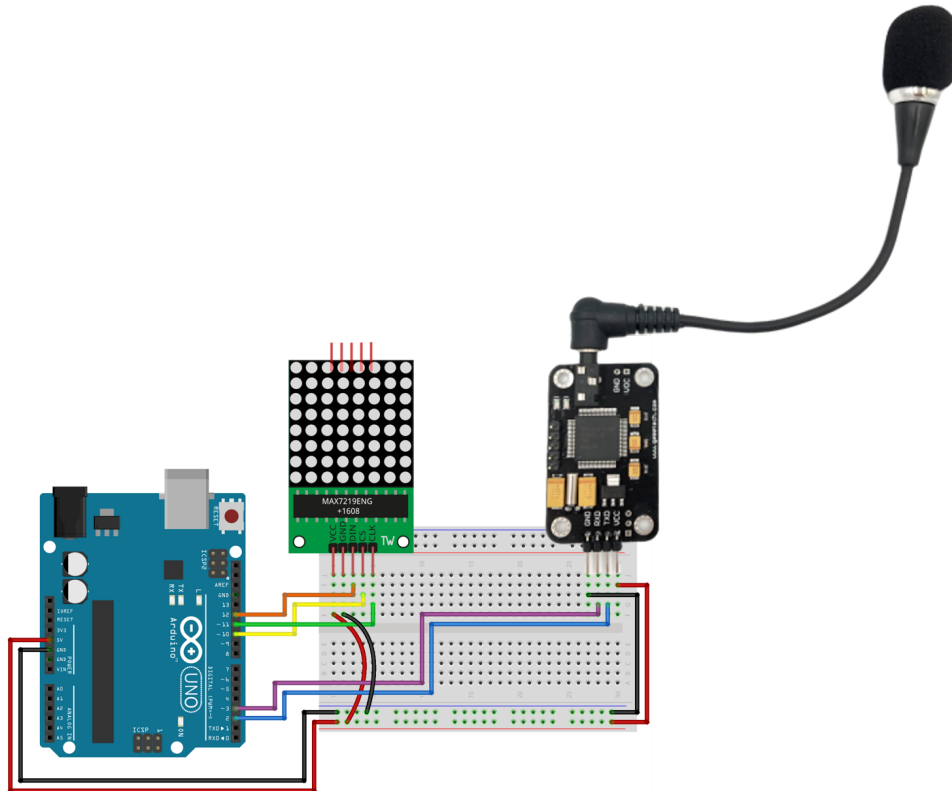
☒ 자동 스크롤 ☐ 타임스탬프 표시 line ending 없음 9600 보드레이트 출력 지우기

## 2.2 음성 인식 모듈로 도트 매트릭스 제어

### 2.2.1 준비물

- 음성 인식 모듈
- 아두이노 우노
- USB to TTL 컨버터 모듈
- 브레드 보드
- 점퍼선
- Max7219 도트 매트릭스 모듈

### 2.2.2 회로도



아두이노	음성 인식 모듈	도트 매트릭스 모듈
5V	VCC	VCC
GND	GND	GND
D2	TxD	-
D3	RxD	-
D5	-	-
D10	-	CS
D11	-	CLK
D12	-	DIN

### 2.2.3 소스코드

```
/*
8x32 dot matrix
아래 링크에서 LedControl 라이브러리를 다운로드
https://github.com/wayoda/LedControl

http://www.devicemart.co.kr/
*/
#include "LedControl.h" // 라이브러리 사용 선언
#include <SoftwareSerial.h> // 소프트웨어 Serial 통신 라이브러리 사용

// Serial 통신핀으로 D11번핀을 Rx로, D10번핀을 Tx로 선언
SoftwareSerial mySerial(2, 3); // 모듈 TX=11번핀 , RX=10번핀 연결

// Din 핀을 12번, Clk핀을 11번 CS핀을 10번에 연결
// 매트릭스는 1개를 사용 선언
LedControl lc=LedControl(12,11,10,1);

int voice_recogn=0;

byte data[5][8]=
{
    // 무표정 이미지
    {
        B11111111,
        B10000001,
        B10100101,
        B10000001,
        B10000001,
        B10111101,
        B10000001,
        B11111111
    },
    // 웃는 표정 이미지
    {
        B11111111,
        B10000001,
        B10100101,
        B10000001,
        B10100101,
        B10111101,
        B10000001,
        B11111111
    },
}
```

```

// 십자가 모양
{
    B00011000,
    B00011000,
    B00011000,
    B11111111,
    B11111111,
    B00011000,
    B00011000,
    B00011000
},

// 하트 모양
{
    B00000000,
    B01100110,
    B11111111,
    B11111111,
    B11111111,
    B01111110,
    B00111100,
    B00011000
},

// 동그란 웃는모양
{
    B00111100,
    B01000010,
    B10100101,
    B10000001,
    B10100101,
    B10011001,
    B01000010,
    B00111100
}

};

void setup()
{
    Serial.begin(9600); // 통신 속도 9600bps로 PC와 시리얼 통신 시작
    mySerial.begin(9600); // 통신 속도 9600bps로 모듈과 시리얼 통신 시작

    lc.shutdown(0, false); // 0~3번까지 매트릭스 절전모드 해제
    lc.setIntensity(0, 8); // 매트릭스의 밝기 선언 0~15의 수
    lc.clearDisplay(0); // 매트릭스 led를 초기화

    Serial.println("wait settings are in progress");
    delay(1000);
    mySerial.write(0xAA); // compact mode 사용
    mySerial.write(0x37);
    delay(1000);
    mySerial.write(0xAA); // 그룹1 음성 명령어 imported

```

```

mySerial.write(0x21);

Serial.println("The settings are complete");
}

void loop()
{
while(mySerial.available())
{
Serial.println("voice input");
voice_recogn=mySerial.read();

switch(voice_recogn)
{
case 0x11:
for(int j=0; j<8; j++)
{
lc.setRow(0,j,data[0][j]); // 무표정 이미지 출력
}
Serial.println("무표정 이미지 출력");
break;

case 0x12:
for(int j=0; j<8; j++)
{
lc.setRow(0,j,data[1][j]); // 웃는 표정 이미지 출력
}
Serial.println("웃는 표정 이미지 출력");
break;

case 0x13:
for(int j=0; j<8; j++)
{
lc.setRow(0,j,data[2][j]); // 십자가 모양 이미지 출력
}
Serial.println("십자가 모양 이미지 출력");
break;

case 0x14:
for(int j=0; j<8; j++)
{
lc.setRow(0,j,data[3][j]); // 하트 모양 이미지 출력
}
Serial.println("하트 모양 이미지 출력");
break;

case 0x15:
for(int j=0; j<8; j++)
{
lc.setRow(0,j,data[4][j]); // 동그란 웃는 표정 이미지출력

```

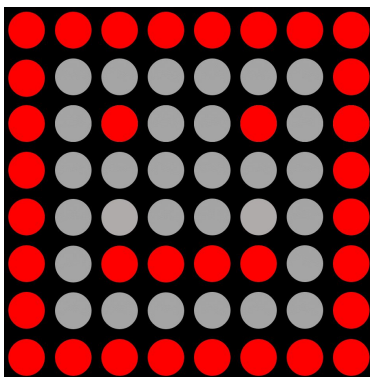


```

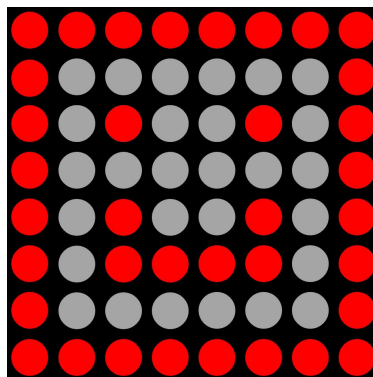
    }
    Serial.println("둥글 웃는 이미지 출력");
    break;
  }
}
}

```

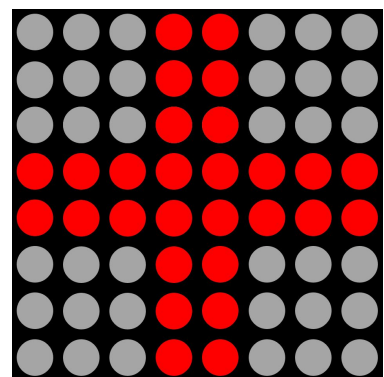
아두이노의 시리얼 모니터를 사용하면, 음성 명령이 들어갔는지 확인 가능합니다.



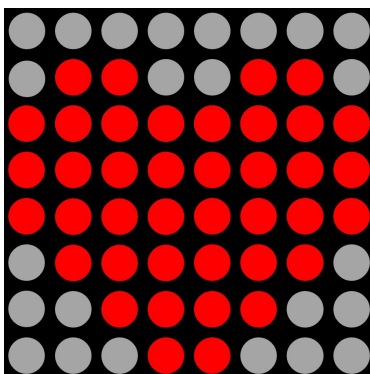
네모 무표정



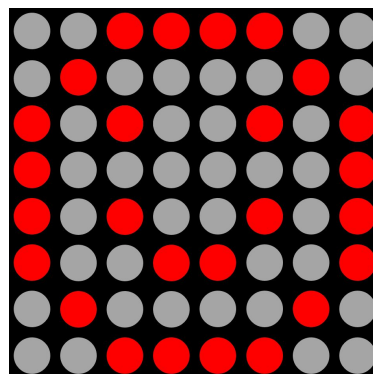
네모 웃는 표정



십자가 모양



하트 모양



둥글 웃는 표정