

// IoT 7일차,

// 아두이노 블루투스

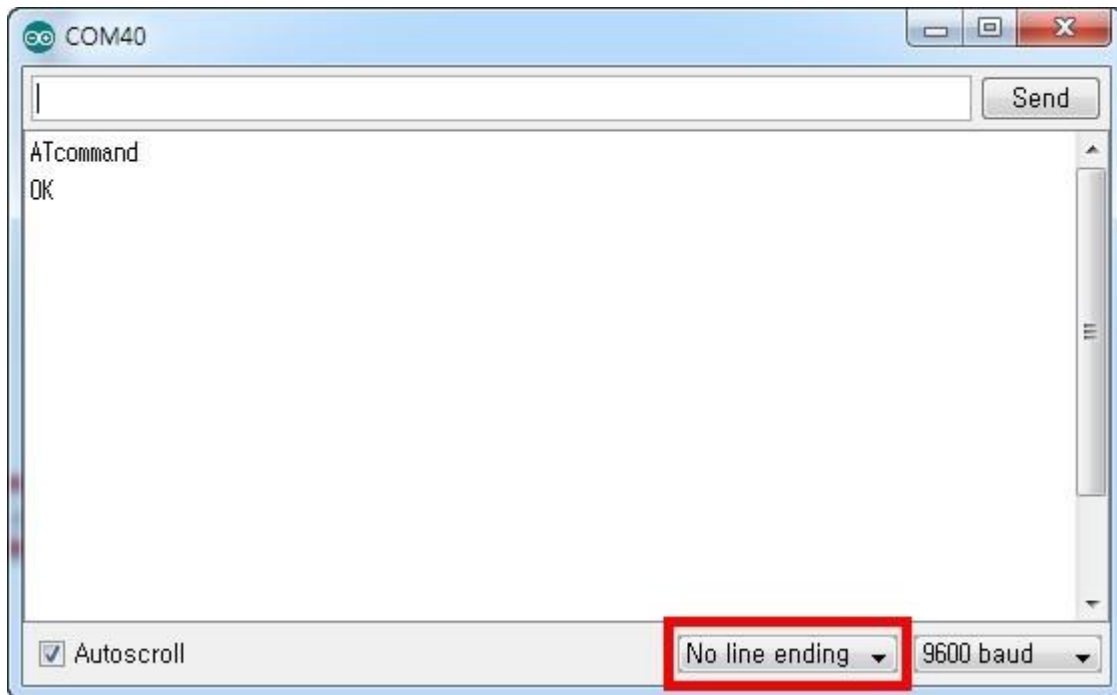
RXD ----- D3
TXD ----- D2
GND ----- GND
VCC ----- 5V



아두이노 펌웨어 업로드 시에는 핀이 중복되므로,
블루투스 점퍼선을 제거하고 프로그램 업로드 함.

아래 코드를 실행, No Line Ending, 9600 baud

```
// 시리얼모니터 열고
// AT 엔터 -> 응답 OK
// AT+NAMEXXXXXX -> 응답 OKsetname : 이름변경
// AT+PIN1234 -> 응답 OKsetPIN : 비밀번호변경
// AT+BAUD4 -> 응답 OK9600 : 통신속도변경
//          1 -> 1200  2 -> 2400  3 -> 4800  4 -> 9600  5 ->19200  6 -> 38400  7 ->
57600  8 -> 115200
// AT+ROLE=S -> 응답 OK+ROLE:S => M: Master S: Slave (기본 SLAVE 변경안해줘도 됨)
```



연결후 60초쯤 기다렸다가 AT

AT+NAME 하고 띄어쓰기 없이 설정할 이름.

```
#include <SoftwareSerial.h> //시리얼통신 라이브러리 호출
```

```
int blueTx = 2; //Tx (보내는핀 설정)at
```

```
int blueRx = 3; //Rx (받는핀 설정)
```

```
SoftwareSerial mySerial(blueTx, blueRx); //시리얼 통신을 위한 객체선언
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600); //시리얼모니터
```

```
    mySerial.begin(9600); //블루투스 시리얼
```

```
}
```

```
void loop()
```

```
{
```

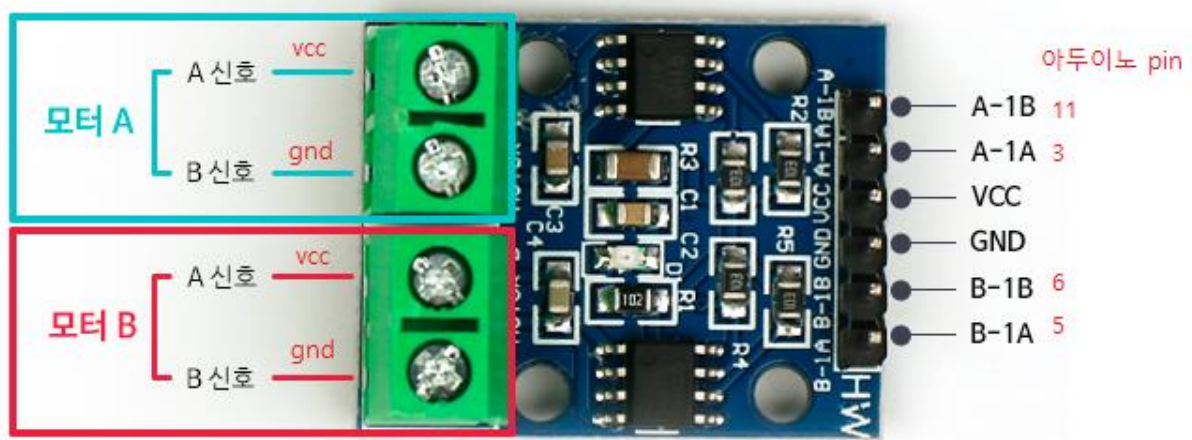
```
    if (mySerial.available()) {
```

```

    Serial.write(mySerial.read()); //블루투스측 내용을 시리얼모니터에 출력
}

if (Serial.available()) {
    mySerial.write(Serial.read()); //시리얼 모니터 내용을 블루투스 측에 WRITE
}
}

```



L9110s 소형 모터 드라이버 DM141, 단가 1020원

동작 전압 : 2.5 ~ 12v

전류 : 최대 0.8A

동작 원리 : <https://3dplife.tistory.com/199>

// 건전지별 전류 <https://blog.naver.com/lheebok/221976924026>

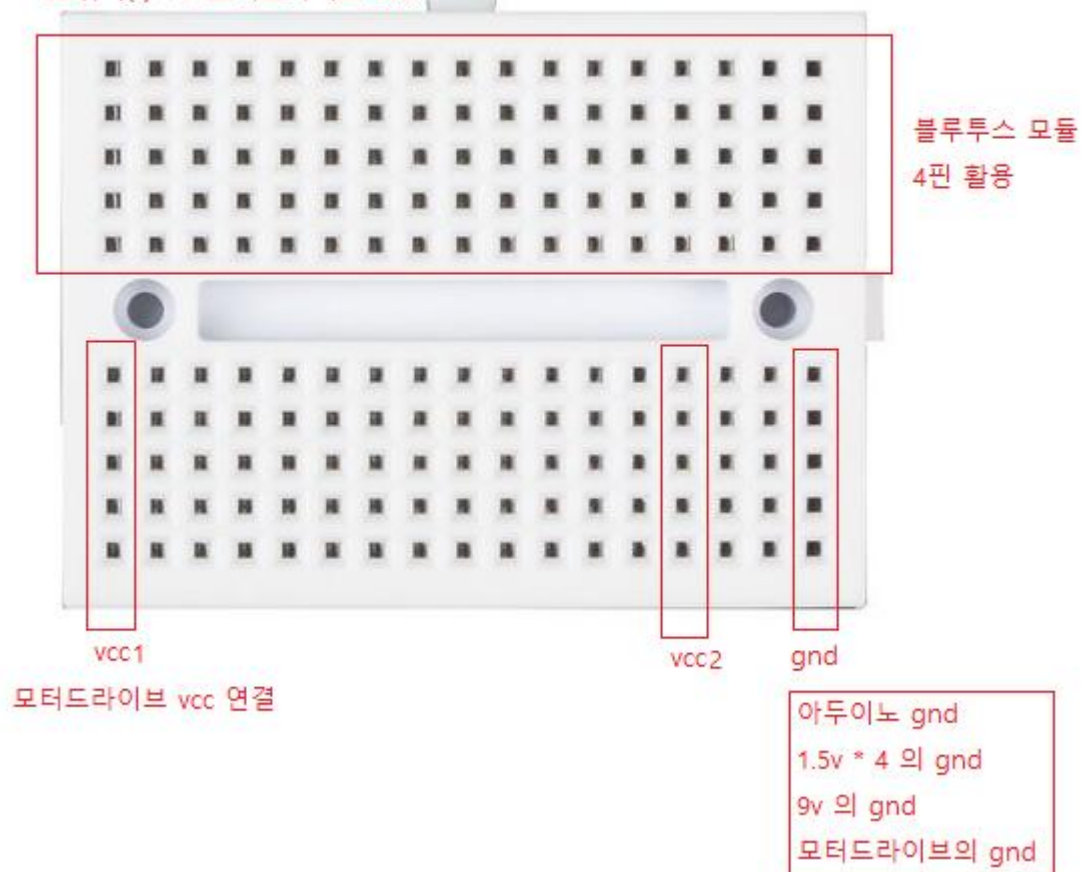
	AAA타입	AA타입	C타입	D타입	9V 사각타입
용량	1150mAh	2850mAh	7800mAh	15000mAh	550mAh
이미지					

건전지 사이즈별 용량

1.5v * 4 의 vcc 는 스위치를 거쳐서 드라이버 모듈의 vcc 를 공급하도록 설정

즉, 1.5v * 4 의 vcc >> 스위치(+)

스위치(-) >> 모터드라이브 vcc



// AT 명령으로 이름 지정후, 아래 프로그램 설치하고, 휴대폰 어플에서 메세지 확인.

// 폰에서 보내는 메세지는 터미널에서 확인 가능하지만,

// 터미널에서 보낸 메세지는 어플의 한계로 폰에서 보이지 않지만, 정상.



Simple Bluetooth Chat

Pelagic
광고 포함

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

```
SoftwareSerial mySerial(2, 3); //블루투스의 Tx, Rx 핀을 2 번 3 번핀으로 설정
```

```
void setup() {  
  // 시리얼 통신의 속도를 9600 으로 설정  
  Serial.begin(9600);  
  while (!Serial) {  
    ; //시리얼통신이 연결되지 않았다면 코드 실행을 멈추고 무한 반복  
  }  
  Serial.println("Hello World!");  
  
  //블루투스 와 아두이노의 통신속도를 9600 으로 설정  
  mySerial.begin(9600);  
  while (!mySerial) {  
    ; //블루투스 통신이 연결되지 않았다면 코드 실행을 멈추고 무한 반복  
  }  
  Serial.println("hi, ble");  
}  
  
void loop() { //코드를 무한반복합니다.  
  if (mySerial.available()) { //블루투스에서 넘어온 데이터가 있다면  
    Serial.write(mySerial.read()); //시리얼모니터에 데이터를 출력  
  }  
  if (Serial.available()) { //시리얼모니터에 입력된 데이터가 있다면  
    mySerial.write(Serial.read()); //블루투스를 통해 입력된 데이터 전달  
  }  
}
```

```
// 블루투스 모듈 제거후 프로그램 업로드
```

```
#define MOTOR_A_a 3 //모터 A 의 +출력핀은 3 번핀입니다
```

```

#define MOTOR_A_b 11    //모터 A의 -출력핀은 11번핀입니다

#define MOTOR_B_a 5     //모터 B의 +출력핀은 5번핀입니다

#define MOTOR_B_b 6     //모터 B의 -출력핀은 6번핀입니다

#define MOTOR_SPEED 150 //모터의 기준속력입니다(0~255)

unsigned char m_a_spd = 0, m_b_spd = 0; //모터의 속력을 저장하는 전역변수

boolean m_a_dir = 0, m_b_dir = 0;      //모터의 방향을 결정하는 전역변수

void setup() {
    //모터 제어 핀들을 출력으로 설정합니다.

    pinMode(MOTOR_A_a, OUTPUT);
    pinMode(MOTOR_A_b, OUTPUT);
    pinMode(MOTOR_B_a, OUTPUT);
    pinMode(MOTOR_B_b, OUTPUT);

    Serial.begin(9600);           //시리얼 통신 초기화

    Serial.println("Hello!");    //터미널 작동 확인용 문자열
}

void loop() {
    unsigned char bt_cmd = 0; //명령어 저장용 문자형 변수

    if (Serial.available()) //데이터가 입력되었을 때
    {
        bt_cmd = Serial.read(); //변수에 입력된 데이터 저장

        rc_ctrl_val(bt_cmd);     //입력된 데이터에 따라 모터에 입력될 변수를 조정하는 함수
    }

    motor_drive(); //모터를 구동하는 함수
}

```

```

void rc_ctrl_val(unsigned char cmd) //입력된 데이터에 따라 모터에 입력될 변수를 조정하는 함수
{
    if (cmd == 'w') // 'w'가 입력되었을 때, 전진
    {
        m_a_dir = 0;           //모터 A 정방향
        m_b_dir = 0;           //모터 B 정방향
        m_a_spd = MOTOR_SPEED; //모터 A의 속도값 조정
        m_b_spd = MOTOR_SPEED; //모터 B의 속도값 조정
    } else if (cmd == 'a') // 'a'가 입력되었을 때, 제자리 좌회전
    {
        m_a_dir = 1;           //모터 A 역방향
        m_b_dir = 0;           //모터 B 정방향
        m_a_spd = MOTOR_SPEED; //모터 A의 속도값 조정
        m_b_spd = MOTOR_SPEED; //모터 B의 속도값 조정
    } else if (cmd == 'd') // 'd'가 입력되었을 때, 제자리 우회전
    {
        m_a_dir = 0;           //모터 A 정방향
        m_b_dir = 1;           //모터 B 역방향
        m_a_spd = MOTOR_SPEED; //모터 A의 속도값 조정
        m_b_spd = MOTOR_SPEED; //모터 B의 속도값 조정
    } else if (cmd == 's') // 's'가 입력되었을 때, 후진
    {
        m_a_dir = 1;           //모터 A 역방향
        m_b_dir = 1;           //모터 B 역방향
        m_a_spd = MOTOR_SPEED; //모터 A의 속도값 조정
        m_b_spd = MOTOR_SPEED; //모터 B의 속도값 조정
    } else if (cmd == 'x') {
        m_a_dir = 0; //모터 A 정방향
        m_b_dir = 0; //모터 B 정방향
        m_a_spd = 0; //모터 A의 정지
    }
}

```

```

    m_b_spd = 0; //모터 B 의 정지
}
}

void motor_drive() //모터를 구동하는 함수
{
    if (m_a_dir == 0) {
        digitalWrite(MOTOR_A_a, LOW); //모터 A+ LOW
        analogWrite(MOTOR_A_b, m_a_spd); //모터 A-의 속력을 PWM 출력
    } else {
        analogWrite(MOTOR_A_a, m_a_spd); //모터 A+의 속력을 PWM 출력
        digitalWrite(MOTOR_A_b, LOW); //모터 A- LOW
    }
    if (m_b_dir == 1) {
        digitalWrite(MOTOR_B_a, LOW); //모터 B+ LOW
        analogWrite(MOTOR_B_b, m_b_spd); //모터 B-의 속력을 PWM 출력
    } else {
        analogWrite(MOTOR_B_a, m_b_spd); //모터 B+의 속력을 PWM 출력
        digitalWrite(MOTOR_B_b, LOW); //모터 B- LOW
    }
}
}

```

// 내일은 nodeMCU 를 이용하여 wifi 제어를 해보겠습니다.

// wifi 실습 후에는 3일 정도의 시간을 드리고, 각자 미니 프로젝트 진행 바랍니다.

// 오늘도 수고 많이 하셨습니다.