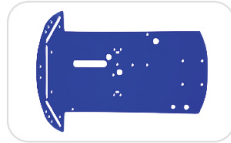




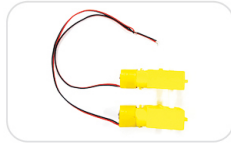
# 블루투스 RC카 사용설명서

## 1. 구성품 확인

### KIT 구성품



아크릴 프레임  
x 1



모터  
x 2



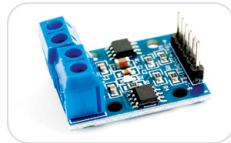
캐스터  
x 1



타이어  
x 2



토글스위치  
x 1



모터드라이버 (L9110S)  
x 1



미니 브레드보드  
x 1



AA배터리 홀더  
x 1



AA 배터리  
x 4



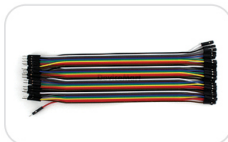
9V 배터리 홀더  
x 1



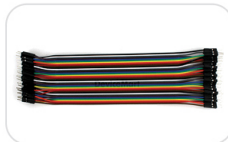
9V 배터리  
x 1



고정용 테이프  
x 1



20cm 점퍼케이블 M-F  
x 1



10cm 점퍼케이블 M-M  
x 1



십자드라이버  
x 1



일자드라이버  
x 1



모터 브라켓  
x 2



볼트 M3\*30  
x 4



너트 M3  
x 6



황동서포트 M3\*12  
x 4



플라스틱서포트 M3\*10  
x 4



볼트 M3\*6  
x 24



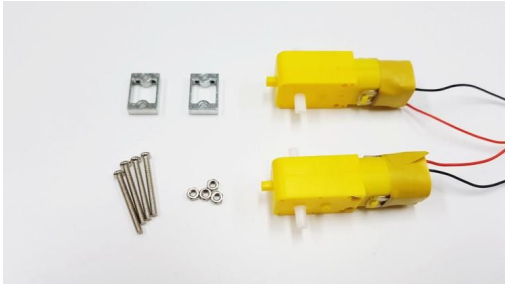

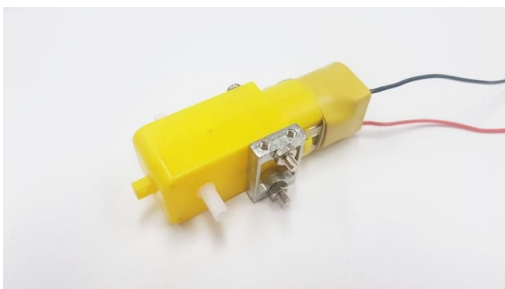
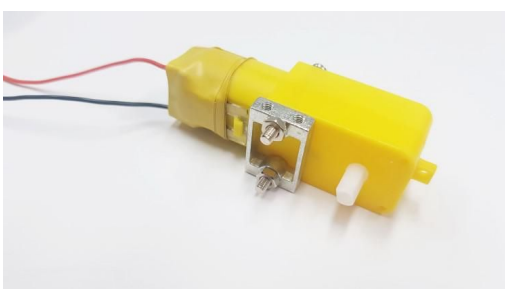
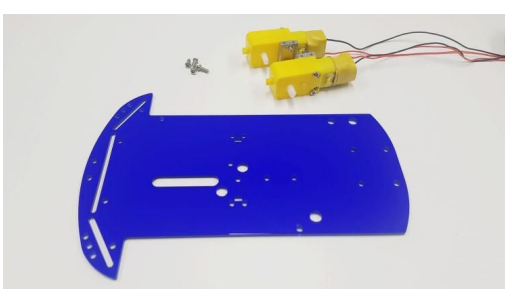
아두이노 호환보드  
x 1



케이블  
x 1

모든 구성품은 제조사 사정에 따라 변경될 수 있습니다.  
본 상품은 반제품으로 조립이 필요한 상품입니다.

## 2. 블루RC카 조립하기

	<p>모터조립에 앞서 이하의 구성품을 준비합니다.</p> <p>모터 2개 브라켓 2개 긴 볼트 4개 너트 4개</p>
	<p>돌출부가 없는 평평한 면에 브라켓이 오도록 조립합니다.</p>
	<p>너트가 브라켓쪽으로 오도록 주의하여 조립합니다.</p>
	<p>반대쪽 모터도 대칭되는 모양으로 동일하게 조립합니다. 위쪽에서 볼때 볼트가 체결될 수 있는 구멍이 위쪽 방향을 향하도록 방향에 유의합니다.</p>
	<p>모터를 아크릴에 체결하기 위해 짧은 M3볼트 4개와 아크릴을 준비합니다. 아크릴에 비닐이 붙어있다면 미리 제거합니다. 아크릴에 세로로 길게 뚫린 홈이 있는데 해당 홈이 좌측을 향하도록 조립하셔야 이미지와 동일하게 조립됩니다.</p>

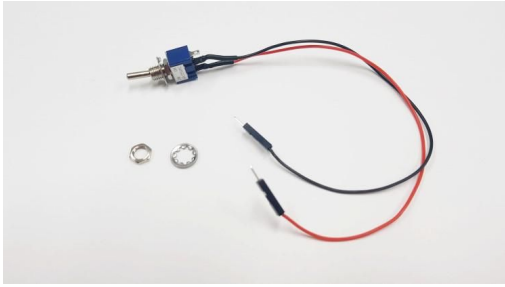
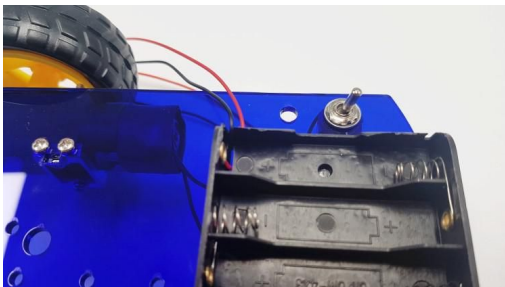

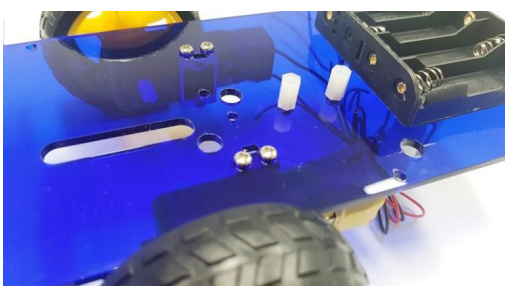
## 2. 블루RC카 조립하기

	<p>위쪽방향을 향하던 볼트 구멍에 맞춰 아크릴 방향에 유의하여 조립해줍니다. 모터의 축은 차량의 전방쪽으로 향하도록 조립합니다.</p>
	<p>캐스터를 부착하기위해 M3 황동 서포트 4개와 짧은 M3볼트 8개를 준비합니다.</p>
	<p>먼저 볼트를 이용해 아크릴 밑면에 서포트를 조립합니다. 캐스터 홈에 맞추기 위해 완전히 조이지 않습니다.</p>
	<p>서포트에 캐스터를 조립합니다. 캐스터의 결합부는 가로로 좀 더 넓으므로 가로방향으로 조립합니다. 4개의 볼트를 모두 조립했다면 서포트에 완전히 조이지 않았던 볼트들을 단단히 조립해줍니다.</p>
	<p>모터 축의 샤프트는 플랫가공이 되어있어 홈에 맞게 끼우시면됩니다.</p>

## 2. 블루RC카 조립하기

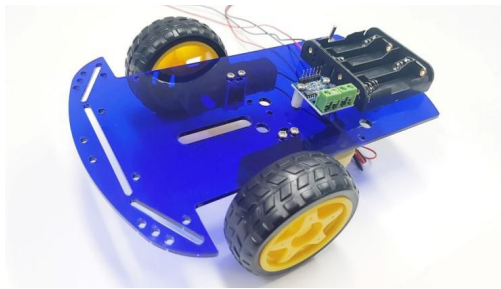
	바퀴를 끼우고 나면 제법 차량의 모습을 갖추게 됩니다.
	모터에 전원을 공급하기 위한 AA배터리 4개입 홀더와 폼테이프를 준비합니다.
	폼테이프는 일부만 절단하여 배터리 홀더에 부착합니다.
	후미 중앙 적당한 위치에 부착합니다. 출력 와이어의 위치는 오른쪽으로 향하는 편이 배선하기 편리합니다.
	온오프 스위치에 해당하는 토글 스위치를 준비합니다.

## 2. 블루RC카 조립하기

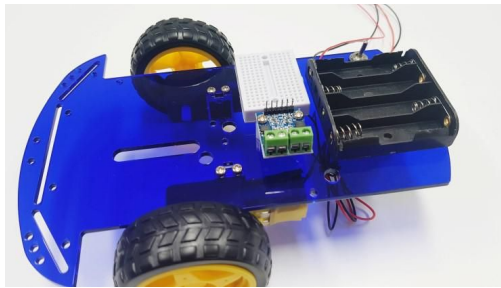
	<p>결합하기 위해 이불이 와셔와 너트를 분리합니다.</p>
	<p>아래와 같은 순서가 되도록 스위치를 고정합니다.</p> <p>너트 이불이 와셔 아크릴 스위치</p>
	<p>스위치의 작동 방향은 배터리 홀더에 걸리지 않도록 블루RC카의 방향과 나란하게 배치합니다. 너트를 사용해 단단히 체결합니다. 나중에 작동 테스트 이후에 온오프 방향을 바꾸고 싶으시다면 스위치를 180도 돌려서 고정하시면 됩니다.</p>
	<p>모터드라이버와 플라스틱 서포트 2개와 짧은 M3볼트 4개를 준비합니다.</p>
	<p>먼저 볼트를 이용해 서포트를 고정합니다. 캐스터와 다르게 이번에는 차체의 윗방향으로 서포트를 고정해줍니다.</p>



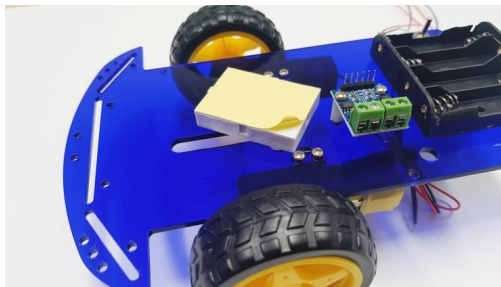
## 2. 블루RC카 조립하기



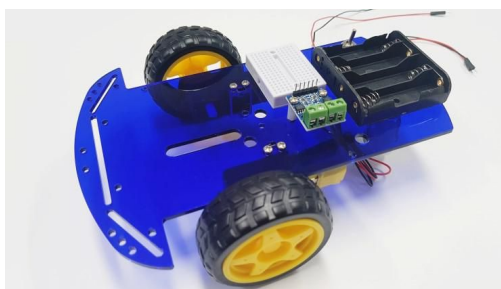
남은볼트로 고정된 서포트에 모터드라이버를 고정합니다.



미니 브레드보드를 준비합니다.



뒷면의 폼테이프의 접착면의 보호지를 제거합니다.

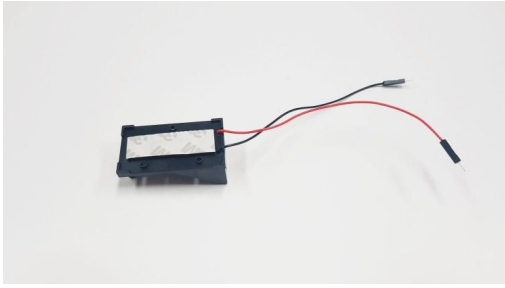
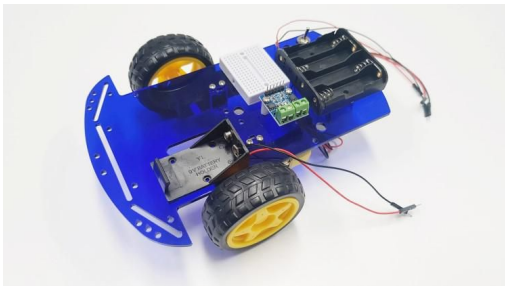
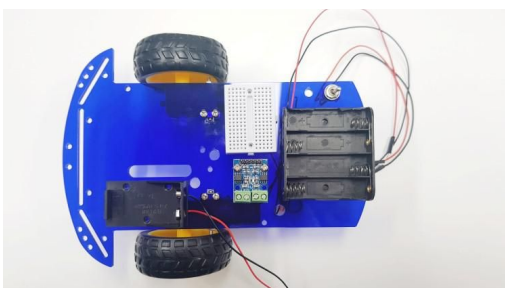


다른 구멍을 가리지 않도록 배터리홀더와 모터드라이버와 인접한 곳에 부착합니다.



아두이노의 전원을 공급하기 위한 9V 배터리 홀더와 부착할 폼테이프를 준비합니다.

## 2. 블루RC카 조립하기

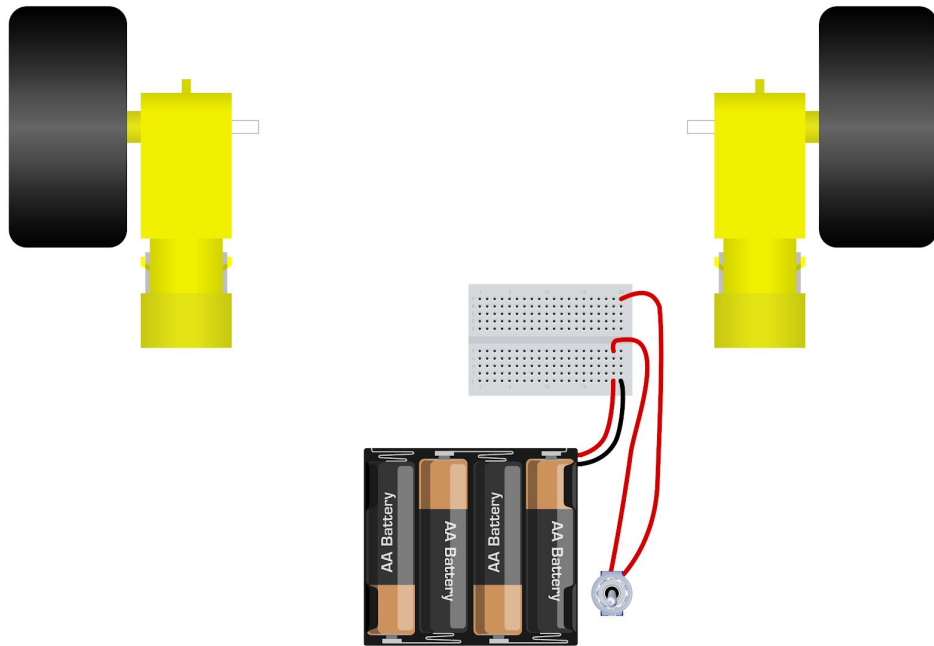
	<p>일부만 절단하여 배터리 홀더에 부착해줍니다.</p>
	<p>차량 전방 좌측의 여백에 배터리 홀더를 부착해줍니다. 출력 와이어의 위치가 브레드보드 쪽을 향하도록 부착해줍니다.</p>
	<p>여기까지 아두이노보드와 센서를 제외한 제품들의 기본적인 배치가 완료되었습니다.</p>



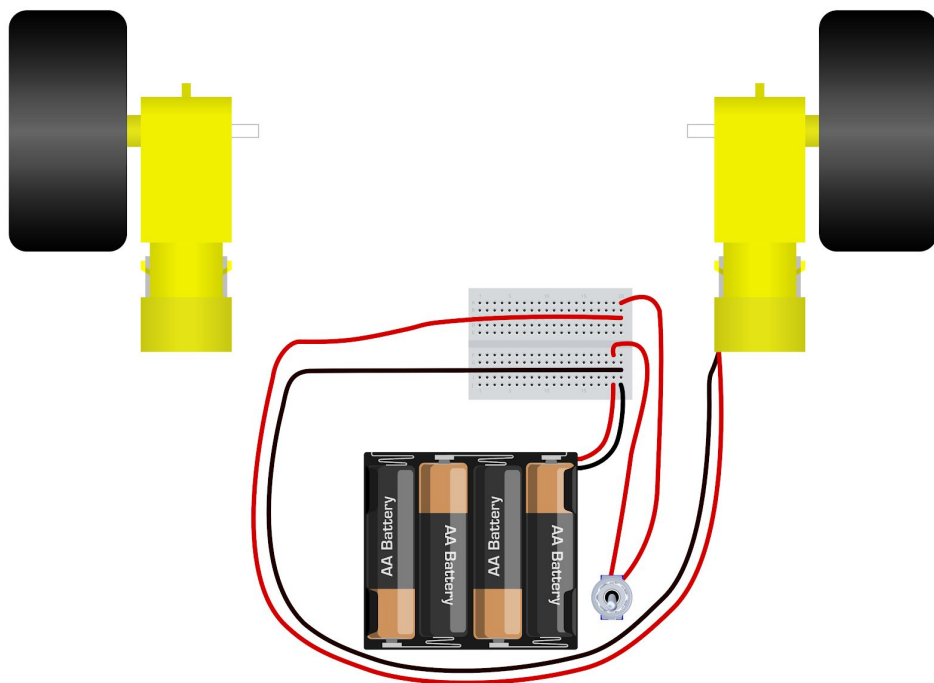
## 2-★. 미니카 모드로 구동하기

추가구성으로 아두이노를 구매하지 않으셨다면 원격 제어는 불가능하지만 간단한 배선을 통해 미니카처럼 전원을 켜올 때 연속으로 전진하도록 구성할 수 있습니다.

블루투스 제어나 라인트레이서, 초음파 제어를 하실 분들은 해당 단계를 건너 뛰어도 됩니다.

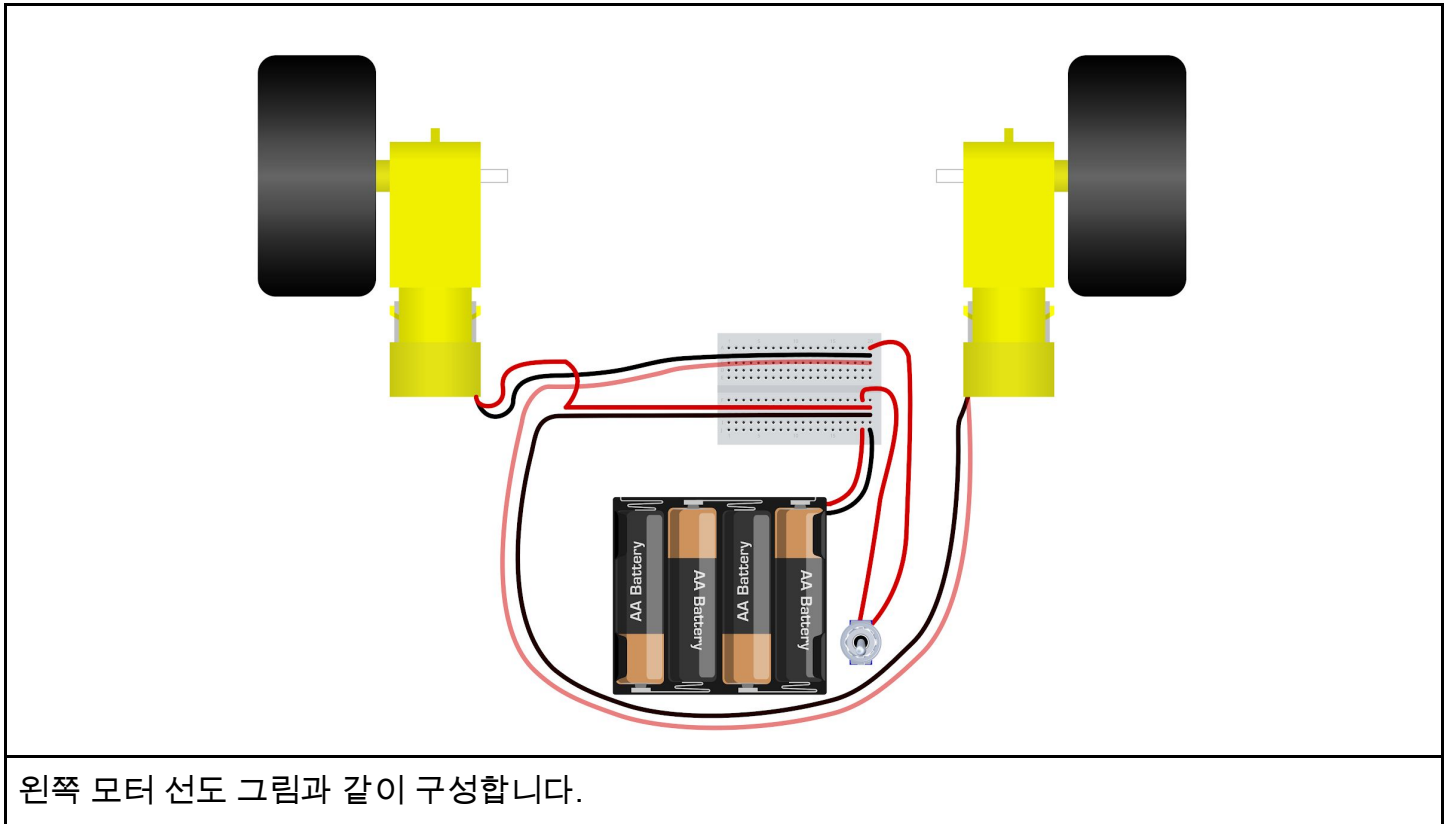


토글 스위치의 선과 배터리 홀더의 선을 미니브레드보드에 그림과 같이 연결합니다.



오른쪽 모터를 그림과 같이 연결합니다.  
만약 연결하다가 모터가 돈다면 토글 스위치를 끄고 연결합니다.

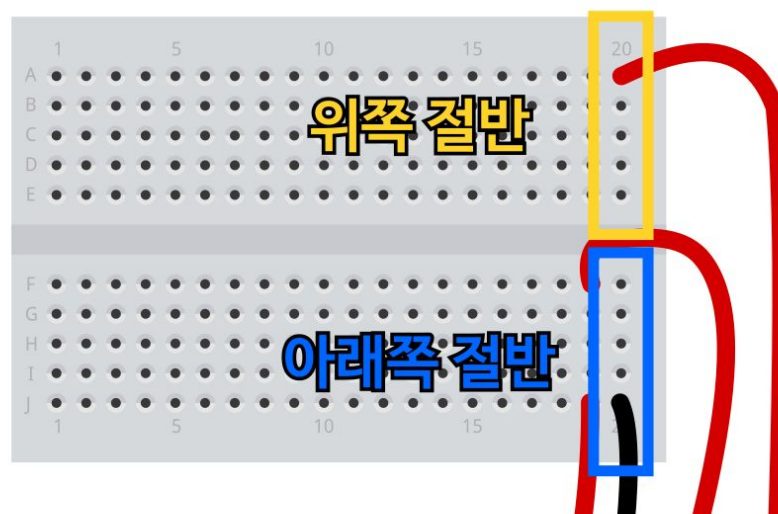
## 2-★. 미니카 모드로 구동하기



스위치를 켜면 전진한다면 정상적으로 배선 된 상태입니다.

만일 후진한다면 이미지상에서의 빨간선과 검정선을 서로 반대로 꽂아주시면 됩니다.  
아래의 그림을 보시면 브레드보드를 가로로 절반으로 나눠서 위쪽에 꽂혀있던 빨간선과 검정선을 아래쪽 절반에 꽂아주시고 반대로 밑에 꽂혀있던 빨간선과 검정선은 위쪽 절반에 꽂아주시면 됩니다.

만일 전진도 후진도 아닌 회전을 한다면 빨간색끼리 위쪽절반, 검정색끼리 아래쪽 절반에 꽂아주시거나 반대로 검정색을 위쪽 절반 빨간색을 아래쪽 절반에 꽂아주시면 됩니다.

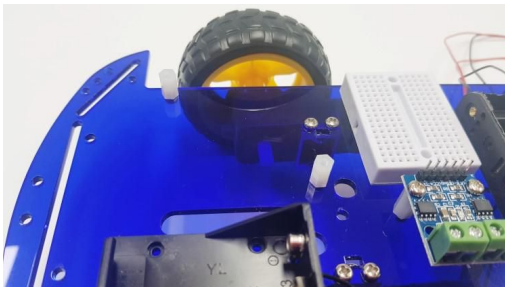


### 3 아두이노 조립하기

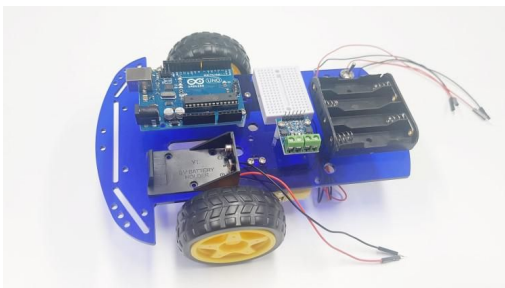


아두이노를 부착하기 위해 이하의 구성품을 준비합니다.

**아두이노 Uno**  
플라스틱 서포트 2개  
짧은 M3볼트 4개



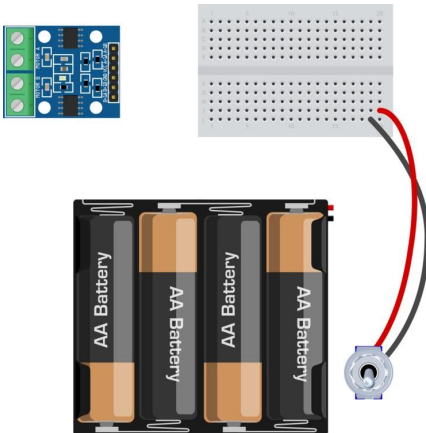
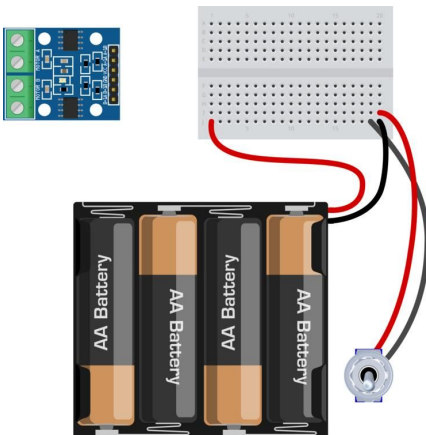
볼트를 이용해 대각선으로 서포트를 조립합니다.

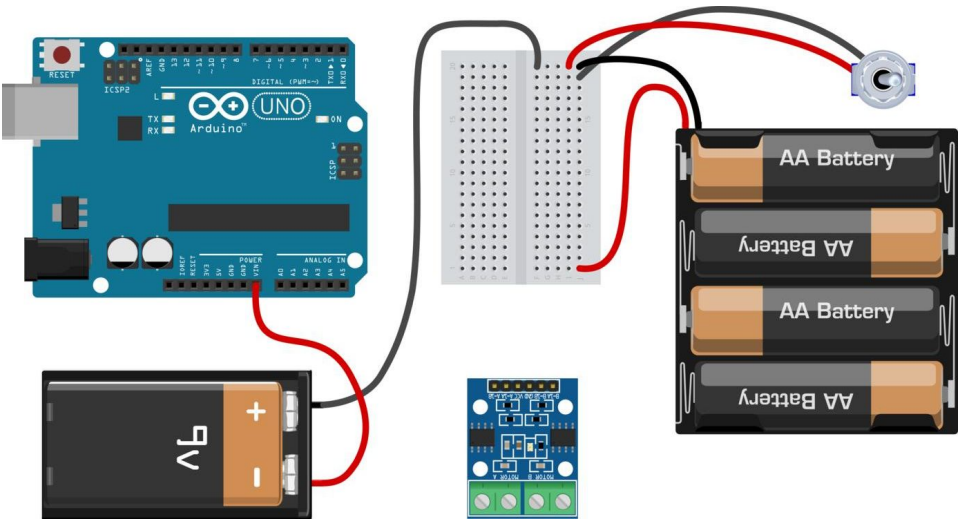


서포트에 아두이노를 올려 볼트로 고정합니다.  
업로드 및 배선을 용이하게 하기 위해 방향은 사진과 같이  
해주시기 바랍니다.

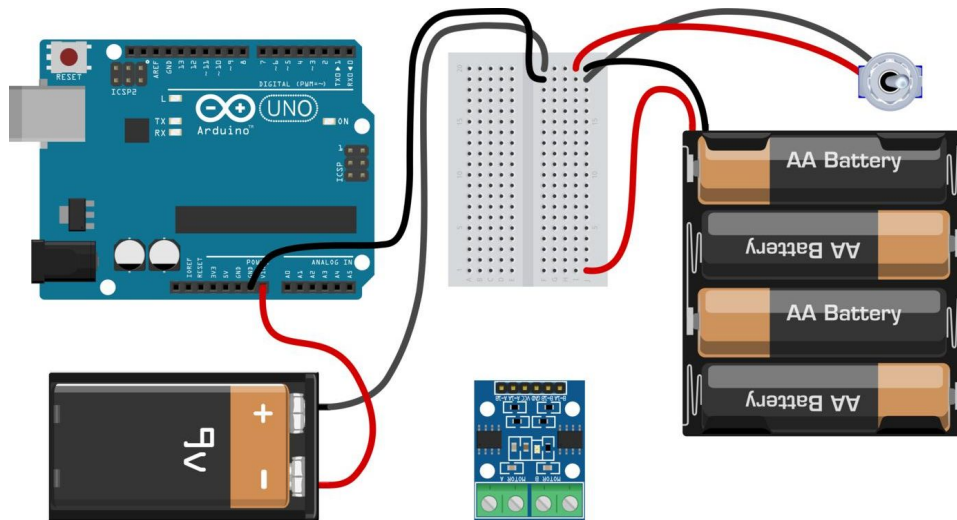
## 4. 블루RC카 배선하기

그림에는 배터리가 삽입되어있지만 배선할때에는 배터리를 뺀 상태로 배선하시기를 권장드립니다.  
전원은 VCC와 GND로 이루어져 있으며 VCC는 빨강선, GND는 검정선을 사용하시면 배선이 용이합니다.

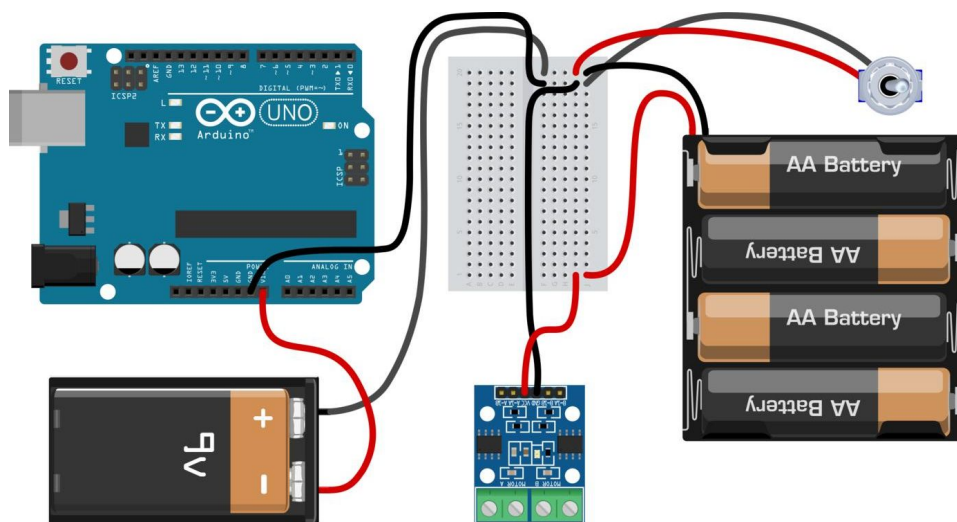
	<p>먼저 스위치의 선을 미니브레드보드에 그림과 같이 연결합니다.</p>
	<p>모터의 전원으로 사용되는 AA배터리홀더의 선도 그림과같이 연결합니다.</p>


<p>보드 전원 공급을 위해 9V 배터리 홀더의 전원을 아두이노 보드의 Vin핀으로 연결합니다. GND라인은 스위치를 거쳐야 하기 때문에 브레드보드에 그림과 같이 연결합니다.</p>

#### 4. 블루RC카 배선하기

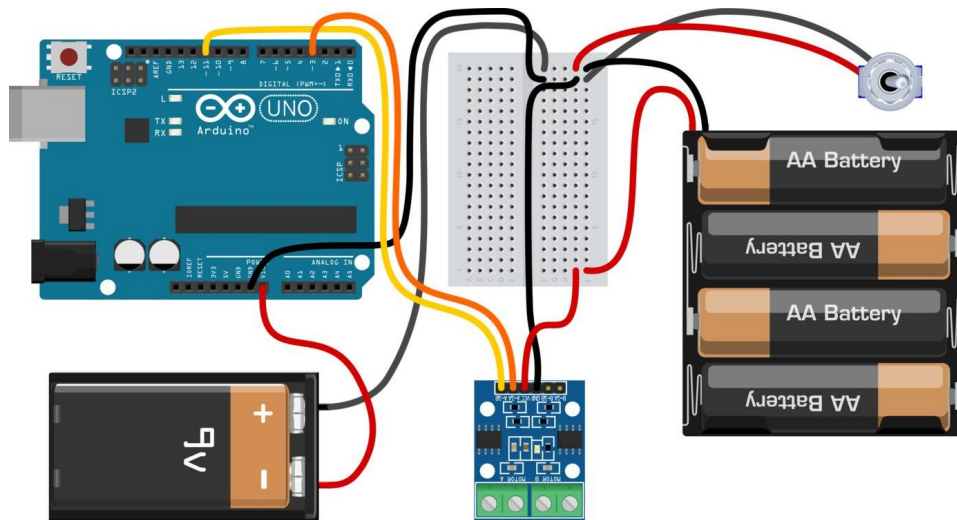


M/M 점퍼와이어 한줄로 아두이노 보드의 GND를 브레드보드에 그림과같이 연결합니다. 스위치를 ON 하면 보드의 GND와 배터리의 GND가 연결되어 보드가 구동되도록 연결됩니다. 9V 배터리를 꽂고 스위치를 올려서 보드가 동작하는지 테스트 해볼 수 있습니다.

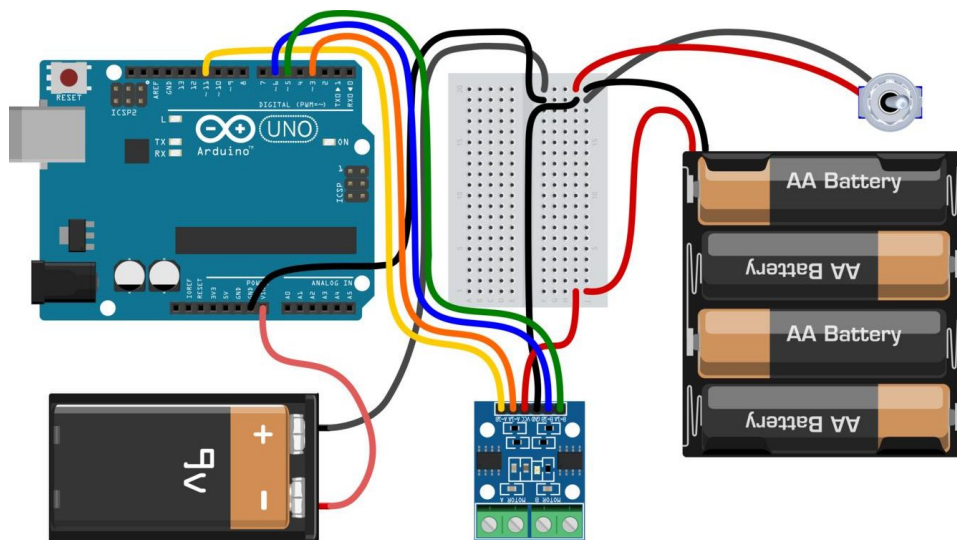


M/F 점퍼와이어를 두줄 나눠서 모터드라이버에 연결합니다. 반대쪽은 그림과 같이 브레드보드에 연결합니다. 모터드라이버 역시 배터리를 꽂고 LED가 들어오는지 스위치테스트를 해볼 수 있습니다.

#### 4. 블루RC카 배선하기



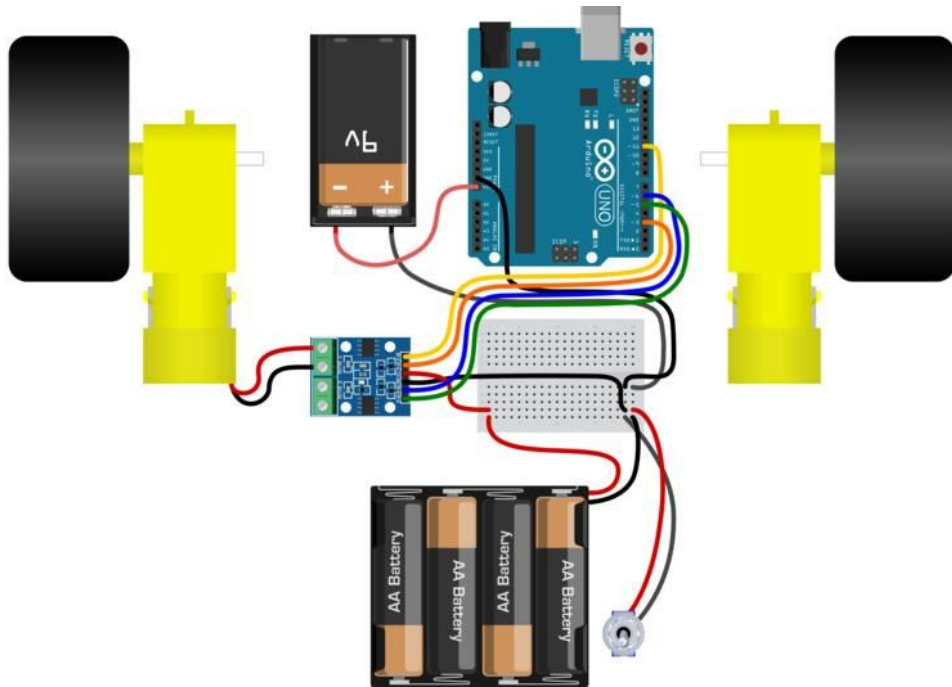
모터A(왼쪽모터)를 제어하기 위해 M/F 점퍼와이어를 두 줄 나눠서 모터드라이버의 모터A 제어핀을 아두이노의 3번핀과 11번핀에 연결합니다.  
그림과 같이 핀의 순서를 맞춰주셔야 이후에 예제코드를 업로드했을때 모터 방향이 일치합니다.



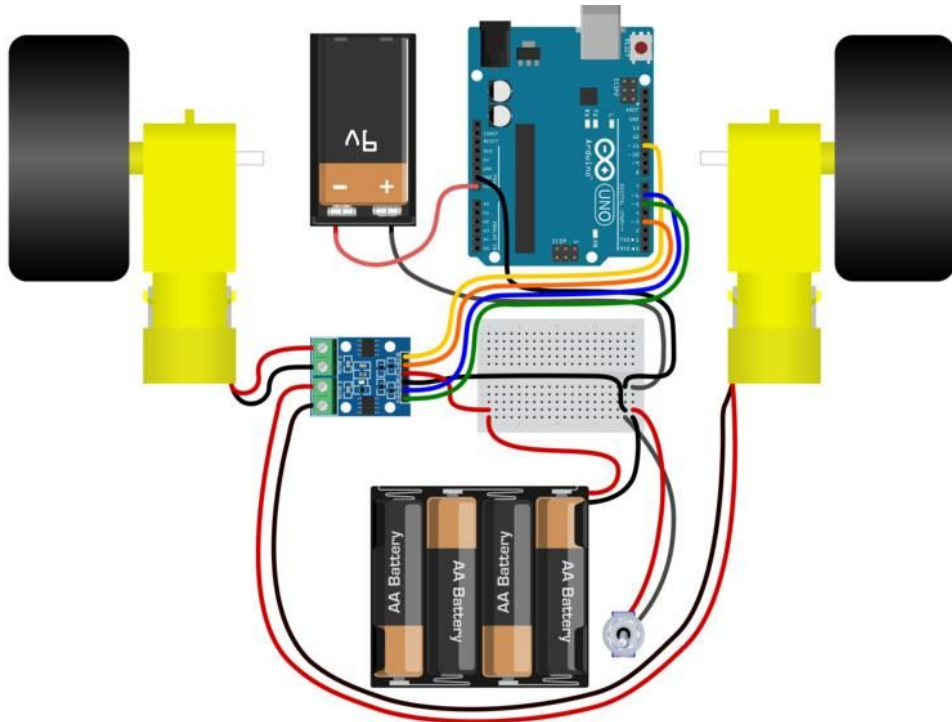
동일한 요령으로 모터드라이버의 모터B(오른쪽모터)제어핀을 아두이노의 5번과 6번핀에 각각 연결합니다.



#### 4. 블루RC카 배선하기



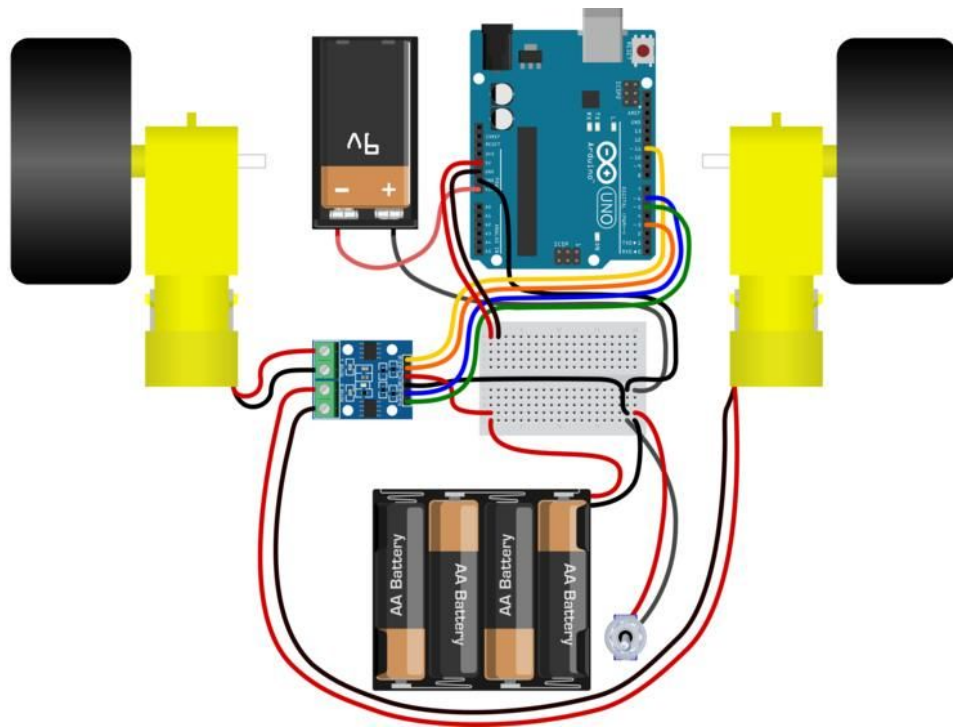
왼쪽 모터의 제어선을 모터드라이버의 모터A에 해당하는 터미널에 연결합니다.  
피복이 벗겨져 있지 않다면 니퍼나 리드커터를 이용해 피복을 벗긴 뒤 스크류드라이버를 이용하여 튼튼하게 고정시켜줍니다.  
피복을 너무 벗기면 합선이 생길 수 있으니 적당한 길이로 피복을 벗깁니다.  
그림과 같은 색상 순서로 연결해야 예제코드 업로드시 모터 방향이 일치합니다.



동일한 요령으로 오른쪽 모터 역시 모터B터미널에 연결해줍니다.



#### 4. 블루RC카 배선하기

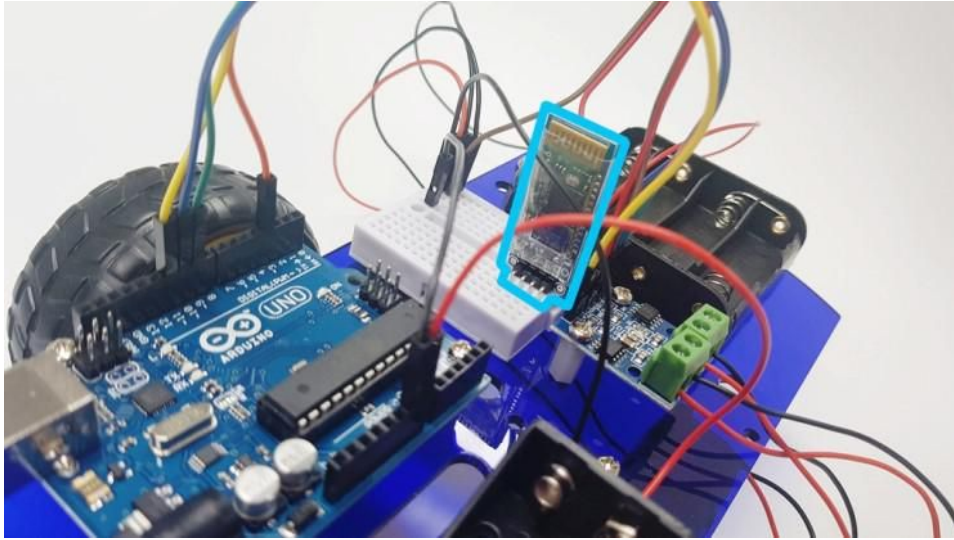


센서 및 모듈에 전원을 공급하기 용이하도록 M/M 점퍼와이어 두줄로 아두이노 보드의 5V와 GND 핀을 그림과같이 브레드보드에 연결합니다.

여기까지 배선하시면 기본적인 구동이 가능한 상태가 됩니다.

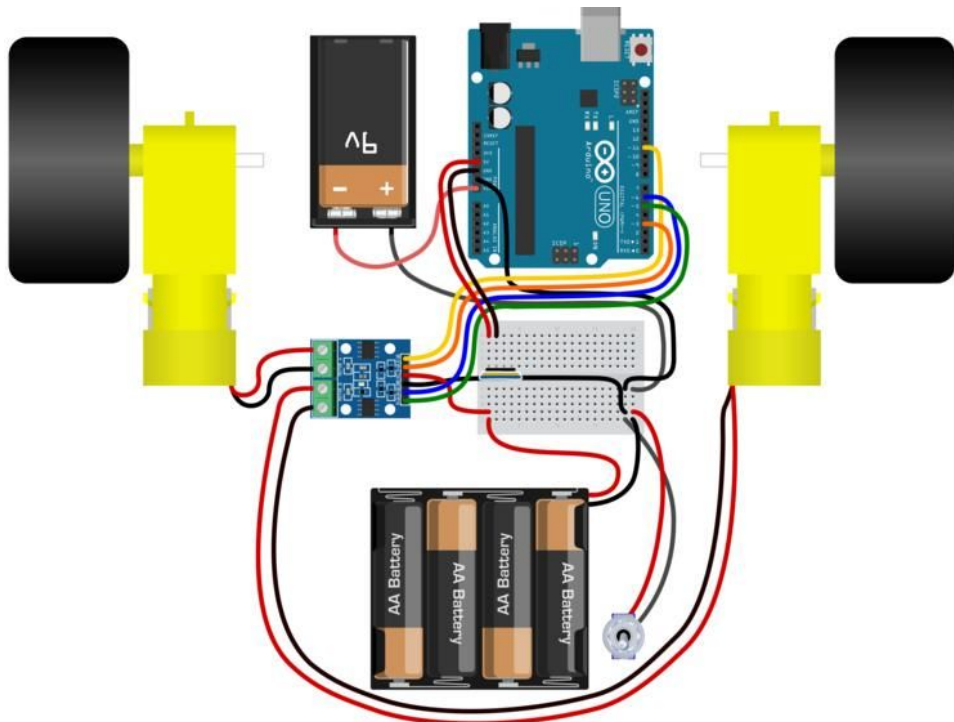
## 5. 블루투스(옵션구매) 조립

추가구성상품으로 블루투스를 구매하셨다면 아래와 같이 진행해주세요.



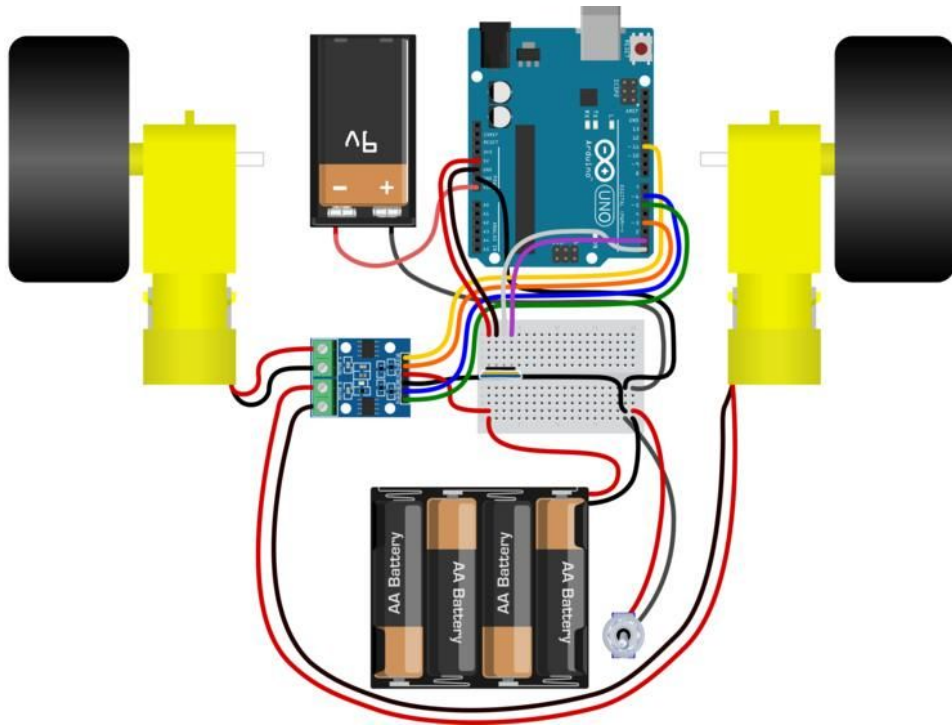
사진과 같은 위치에 블루투스를 삽입합니다.  
배선이 완료되지 않은 사진이므로 블루투스의 위치만 참고해주세요.

## 6. 블루투스 배선



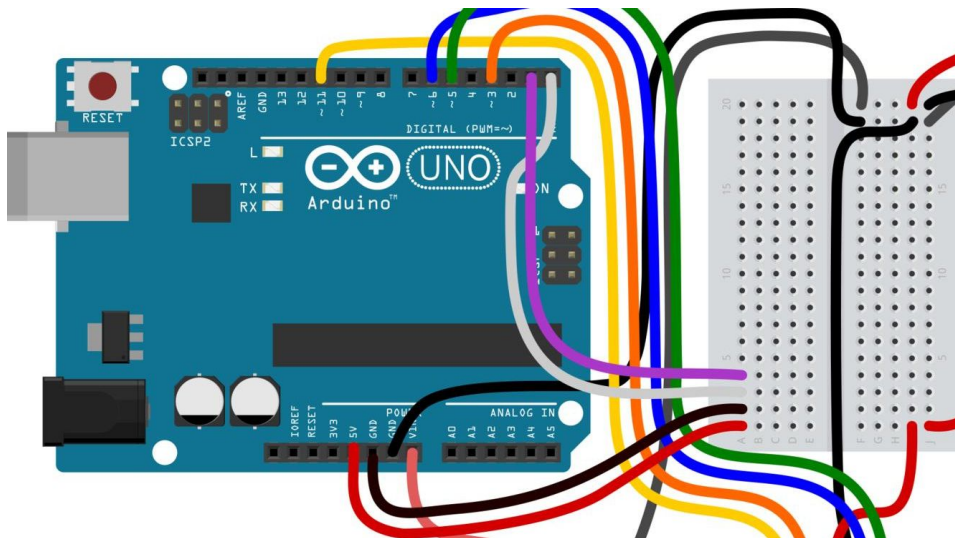
정상적으로 배선하셨다면 블루투스를 켜고나면 그림과 같은 배치가 됩니다.

## 6. 블루투스 배선



정상적으로 배선하셨다면 블루투스를 켜고나면 그림과 같은 배치가 됩니다.

### ※ 프로그램 업로드시 주의사항

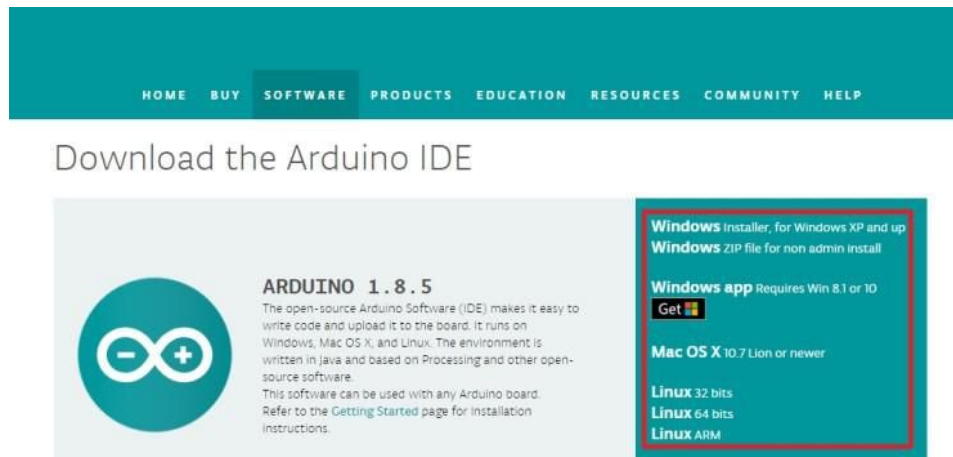


블루투스를 이용하실때 프로그램 업로드시에는 그림과같이 블루투스모듈을 제거하고 프로그램을 업로드하셔야합니다.

## 7. 아두이노 IDE 설치

<https://www.arduino.cc/en/main/software>

[아두이노 IDE]가 설치되어있지 않으시다면 위 URL에서 다운로드 받아 설치합니다.



본인의 OS를 선택합니다.

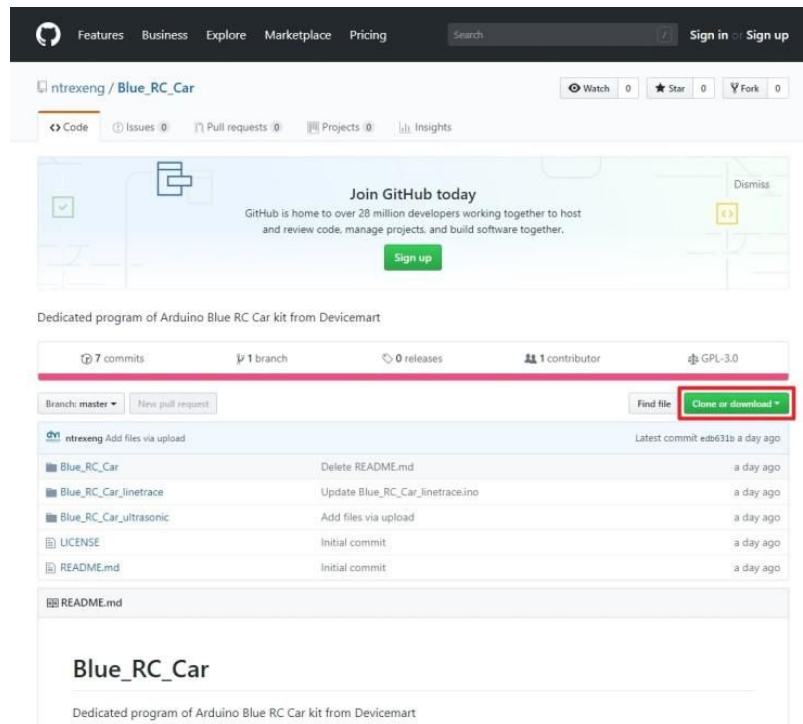


[JUST DOWNLOAD]버튼을 눌러 다운로드를 받아 프로그램을 설치합니다.

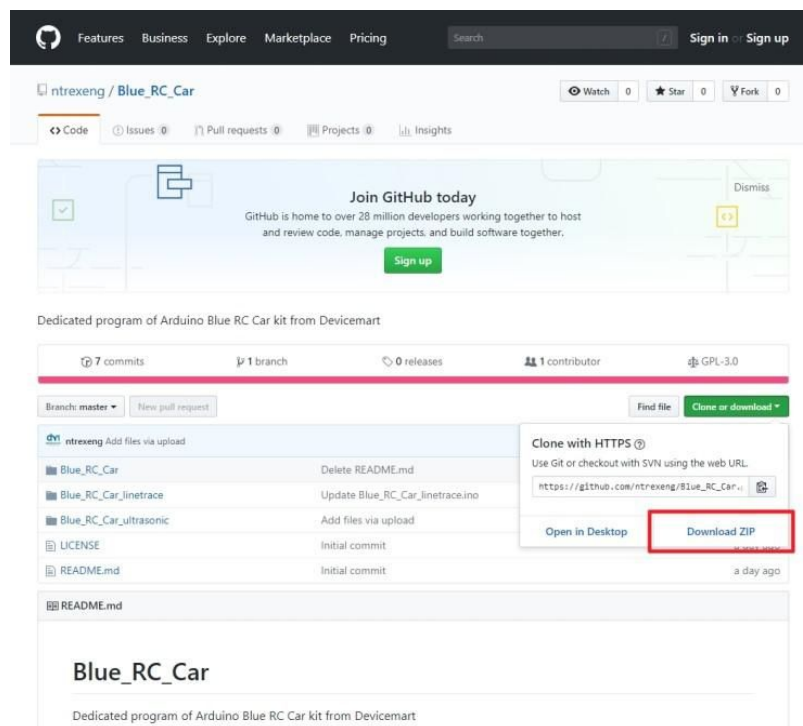
## 8. 예제코드

[https://github.com/ntrexeng/Blue\\_RC\\_Car](https://github.com/ntrexeng/Blue_RC_Car)

위 URL에서 다운받으시거나 아래 기재된 소스코드대로 작성하시면 됩니다.



본인의 OS를 선택합니다.



[JUST DOWNLOAD]버튼을 눌러 다운로드를 받아 프로그램을 설치합니다.

## 8. 예제코드

```
#define MOTOR_A_a 3 //모터A의 +출력핀은 3번핀입니다
#define MOTOR_A_b 11 //모터A의 -출력핀은 11번핀입니다
#define MOTOR_B_a 5 //모터B의 +출력핀은 5번핀입니다
#define MOTOR_B_b 6 //모터B의 -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') //w가 입력되었을 때, 전진
    {
        m_a_dir = 0; //모터A 정방향
        m_b_dir = 0; //모터B 정방향
        m_a_spd = MOTOR_SPEED; //모터A의 속력값 조정
        m_b_spd = MOTOR_SPEED; //모터B의 속력값 조정
    }
    else if(cmd == 'a') //a가 입력되었을 때, 제자리 좌회전
    {
        m_a_dir = 1; //모터A 역방향
        m_b_dir = 0; //모터B 정방향
        m_a_spd = MOTOR_SPEED; //모터A의 속력값 조정
        m_b_spd = MOTOR_SPEED; //모터B의 속력값 조정
    }
    else if(cmd == 'd') //d가 입력되었을 때, 제자리 우회전
    {
        m_a_dir = 0; //모터A 정방향
        m_b_dir = 1; //모터B 역방향
        m_a_spd = MOTOR_SPEED; //모터A의 속력값 조정
        m_b_spd = MOTOR_SPEED; //모터B의 속력값 조정
    }
}
```



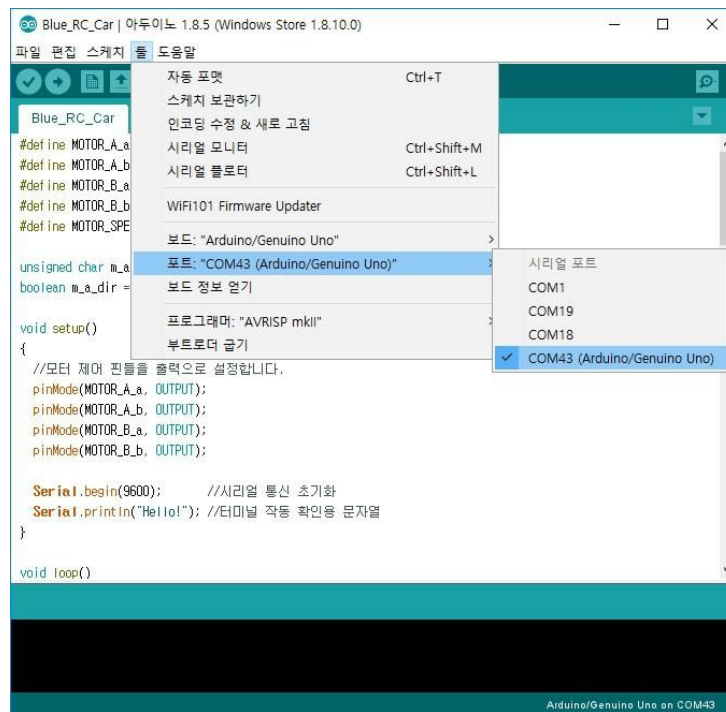
## 8. 예제코드

```
else if(cmd == 's') //s'가 입력되었을 때, 후진
{
    m_a_dir = 1; //모터A 역방향
    m_b_dir = 1; //모터B 역방향
    m_a_spd = MOTOR_SPEED; //모터A의 속도값 조정
    m_b_spd = MOTOR_SPEED; //모터B의 속도값 조정
}
else if(cmd == 'x')
{
    m_a_dir = 0; //모터A 정방향
    m_b_dir = 0; //모터B 정방향
    m_a_spd = 0; //모터A의 정지
    m_b_spd = 0; //모터B의 정지
}
}

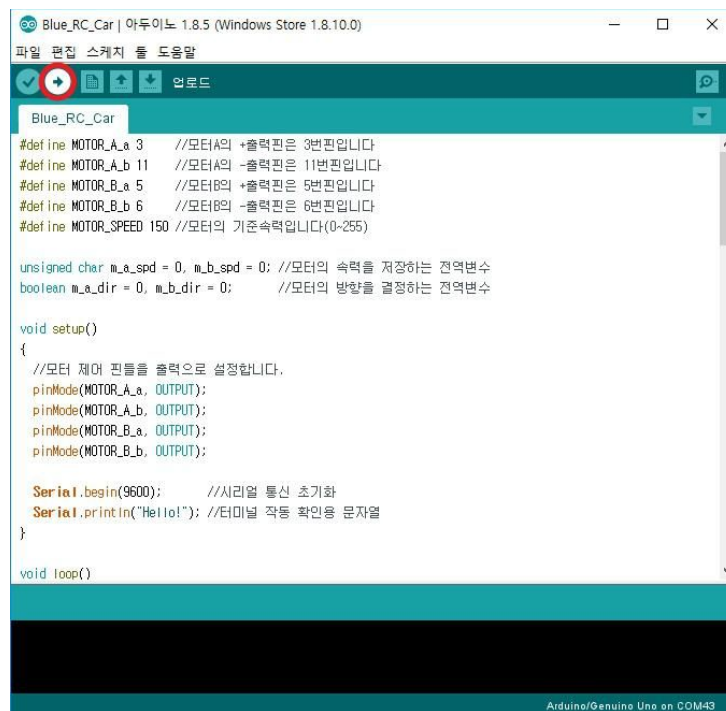
void motor_drive() //모터를 구동하는 함수
{
    if(m_a_dir == 0)
    {
        digitalWrite(MOTOR_A_a, LOW); //모터A+ LOW
        analogWrite(MOTOR_A_b, m_a_spd); //모터A-의 속력을 PWM 출력
    }
    else
    {
        analogWrite(MOTOR_A_a, m_a_spd); //모터A+의 속력을 PWM 출력
        digitalWrite(MOTOR_A_b, LOW); //모터A- LOW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



## 9. 코드 업로드

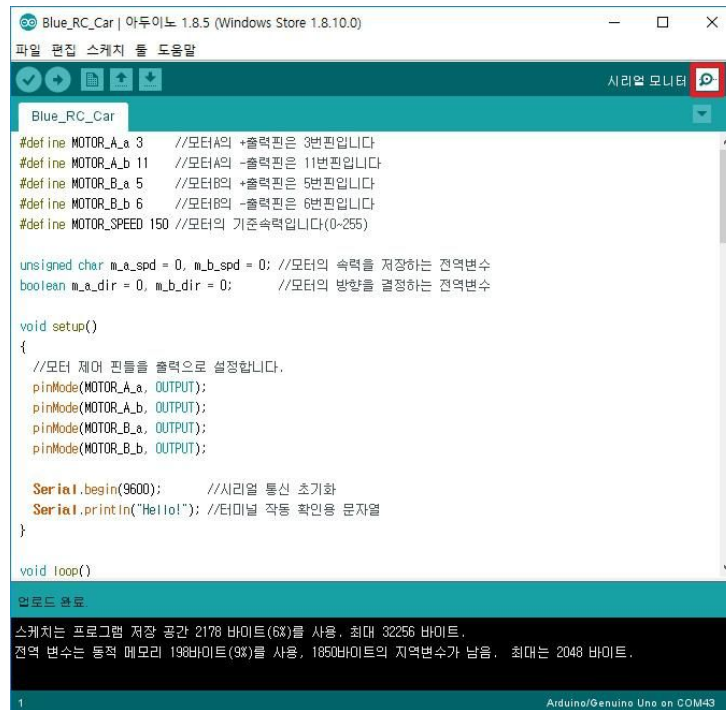


USB 케이블을 이용해 PC와 아두이노를 연결한 뒤 아두이노 IDE에서  
[툴] => [포트]를 선택하셔서 아두이노가 연결된 COM포트를 선택합니다.  
[툴] => [보드] => [Arduino/Genuino Uno]로 보드도 정상적으로 선택되어 있는지 체크합니다.



왼쪽 상단의 업로드 버튼을 클릭하여 업로드합니다.  
블루투스를 연결하셨다면 업로드전에 블루투스모듈을 제거하고 업로드합니다.

## 9. 코드 업로드



```
Blue_RC_Car | 아두이노 1.8.5 (Windows Store 1.8.10.0)
파일 편집 스케치 툴 도움말
시리얼 모니터

Blue_RC_Car

#define MOTOR_A_a 3 //모터A의 +출력핀은 3번핀입니다
#define MOTOR_A_b 11 //모터A의 -출력핀은 11번핀입니다
#define MOTOR_B_a 5 //모터B의 +출력핀은 5번핀입니다
#define MOTOR_B_b 6 //모터B의 -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);

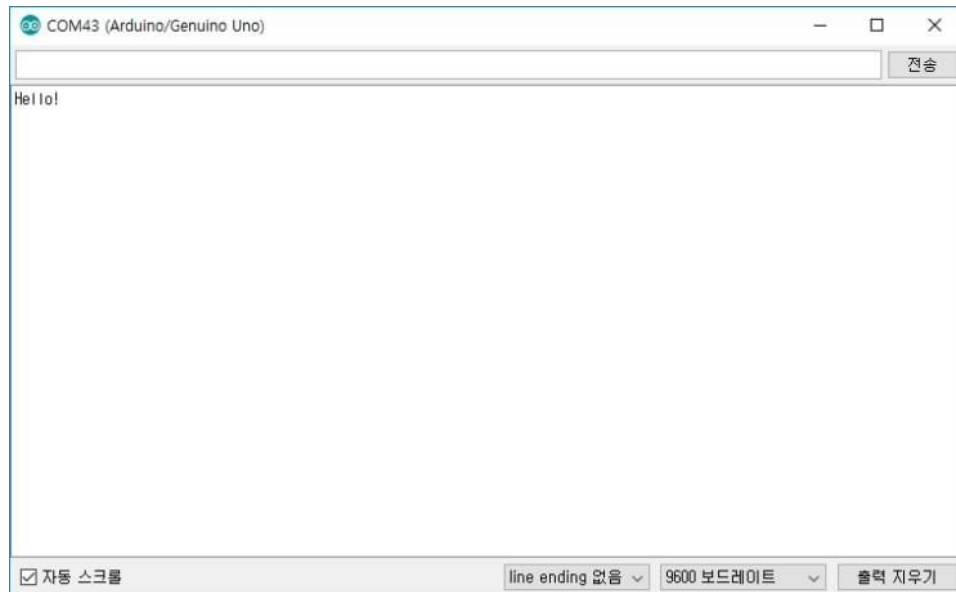
  Serial.begin(9600); //시리얼 통신 초기화
  Serial.println("Hello!"); //터미널 작동 확인용 문자열
}

void loop()

업로드 완료

스케치는 프로그램 저장 공간 2178 바이트(6%)를 사용, 최대 32256 바이트.
전역 변수는 동적 메모리 198바이트(9%)를 사용, 1850바이트의 지역변수가 남음. 최대는 2048 바이트.
1 Arduino/Genuino Uno on COM43
```

업로드가 완료되었다면 아두이노가 연결된 상태 그대로 시리얼모니터를 실행합니다.  
모터에 전원이 공급되도록 블루RC카의 전원스위치를 켭니다.



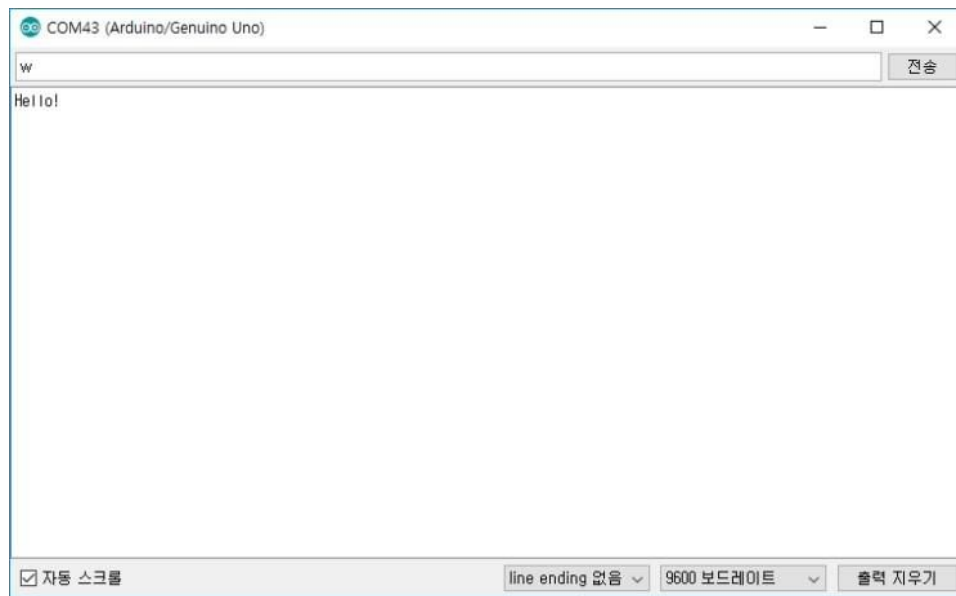
```
COM43 (Arduino/Genuino Uno)
전송

Hello!

[x] 자동 스크롤 line ending 없음 9600 보드레이트 출력 지우기
```

정상적으로 업로드되었다면 시리얼모니터 실행시 'Hello!'가 표시됩니다.

## 9. 코드 업로드



소문자 'w'를 쓰고 엔터키를 눌러 명령을 입력합니다.

블루RC카가 전진한다면 정상적인 동작입니다.

소문자 'x'를 쓰고 엔터키를 누르면 정지합니다.

명령용 문자는 이하와 같습니다.

w - 전진

a - 좌회전

s - 후진

d - 우회전

x - 정지

여기까지 테스트 하셨다면 블루투스나 다른 센서들을 연결할 준비가 된 상태입니다.

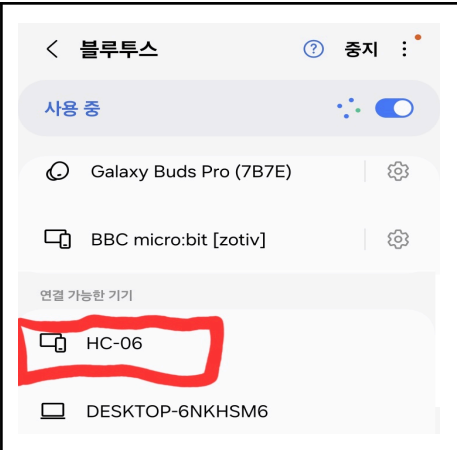
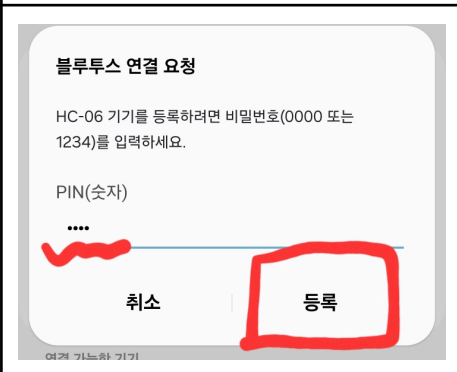
## 10. 블루투스 어플 설치와 설정













[Arduino Bluetooth Controller - Google Play 앱](#)

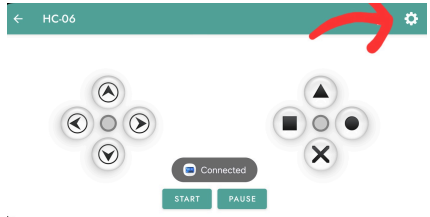
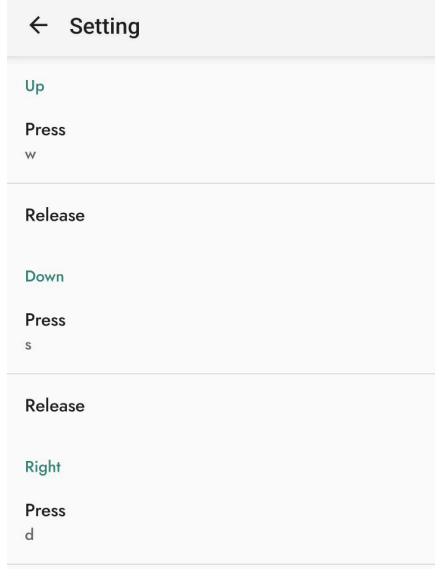
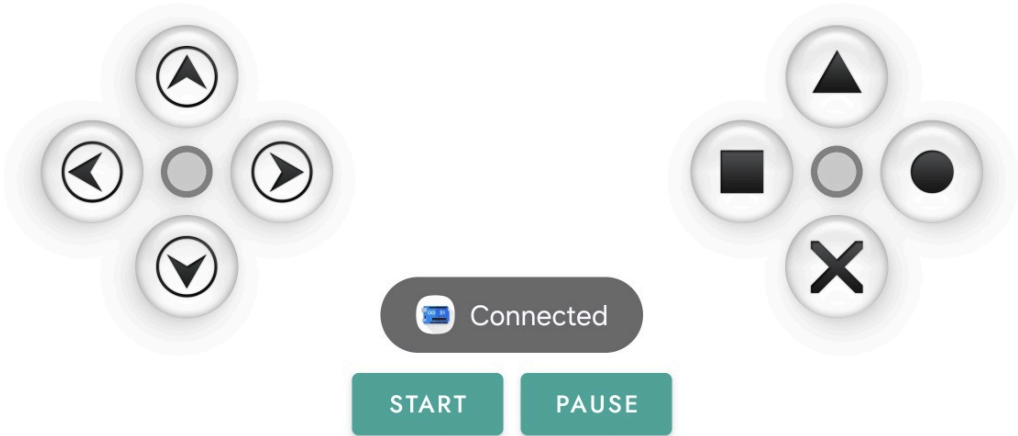


위의 URL 또는 QR코드를 통해 블루투스 어플리케이션을 다운로드합니다.

아두이노에 USB 케이블이 연결되어 있다면 제거해주시고 블루투스 모듈을 제 위치에 꽂고 블루RC카의 전원을 켭니다. 블루투스모듈과 테더링을 하기 위해 스마트폰을 켜 블루투스를 활성화 해주시고 블루투스 설정창으로 접속합니다

	<p>연결 가능한 디바이스에서 'HC-06'을 찾습니다. 'HC-06'으로 표시되지 않고 'XX:XX:XX:XX'의 형식으로 표시될 수도 있습니다. 이런경우 한번 연결해주시면 정상적으로 디바이스명이 표시됩니다.</p> <p>HC-06을 찾아 등록합니다.</p>
	<p>비밀번호 또는 PIN을 입력하라고 표시되는데 초기 암호는 [0000] 또는 [1234]입니다.</p>

<p>있습니다.</p> <p>등록된 기기</p> <p> HC-06</p> <p> Galaxy Buds (D473)</p>	<p>올바른 비밀번호를 입력하면 그림과 같이 등록된 디바이스에 표시됩니다.</p>
<p>검색</p> <p> Arduino Bluetooth...</p>	<p>설치했던 어플을 실행합니다.</p>
<p>Arduino Bluetooth Control</p> <p>PAIRED DEVICES</p> <p> HC-06 98:D3:C1:FD:C0:50</p> <p> Galaxy Buds (D473) C0:DC:DA:FD:D4:73</p> <p> BBC micro:bit</p>	<p>HC-06 연결된 블루투스를 선택합니다.</p>
<p>Connect to HC-06</p> <p> <b>LED/Lamp Control</b> Control LED or lamp with a bu</p> <p> <b>RGB LED Control</b> Control RGB LED or LED strip</p> <p> <b>Gamepad</b> Used as a gamepad or car con</p> <p> <b>Switches</b> Control lights fans etc. remote</p> <p> <b>Terminal</b> Used to send serial data</p> <p> <b>Voice Control</b></p>	<p>'Gamepad'로 접속합니다.</p>

	<p>우측 상단 설정에 접속합니다.</p>
	<p>각각 다음과 같이 설정해줍니다.</p> <p>Up(▶) - w  Down(▼) - s  Right(▶) - d  Left(◀) - a  Square(■) - x</p> <p>설정 후 뒤로가기 버튼을 통해 조이스틱으로 이동합니다.</p>
	
<p>조이스틱 버튼을 통해 RC카를 동작해보세요.</p>	