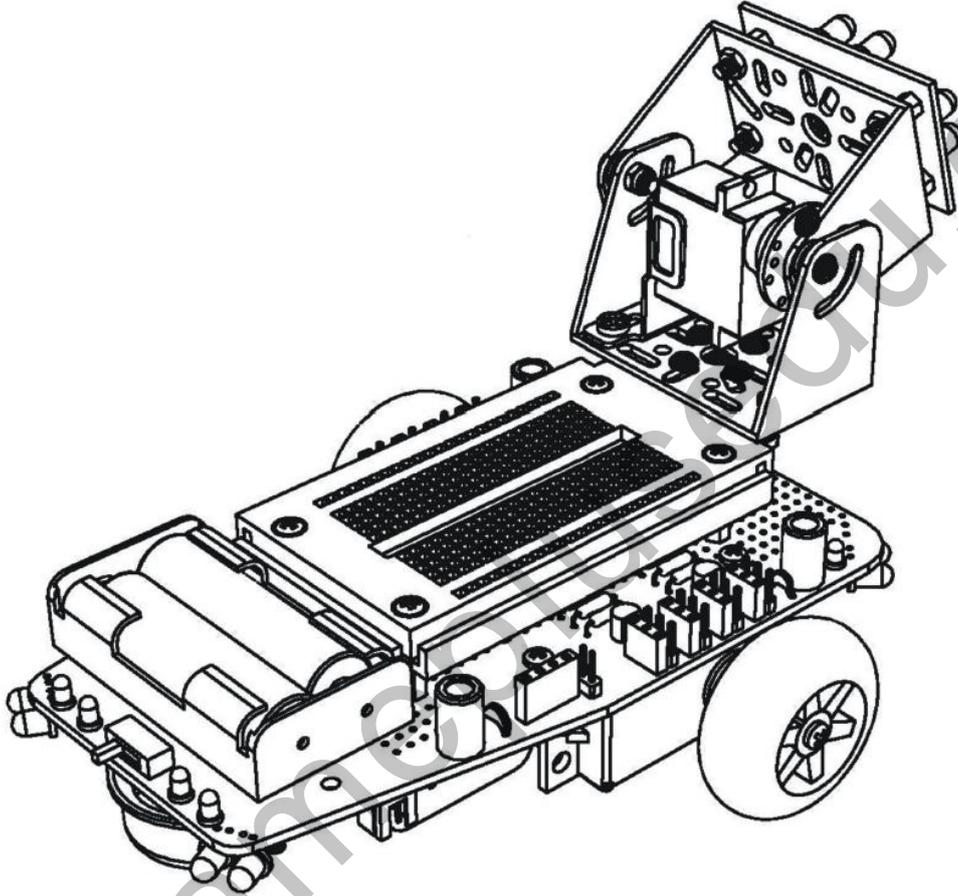


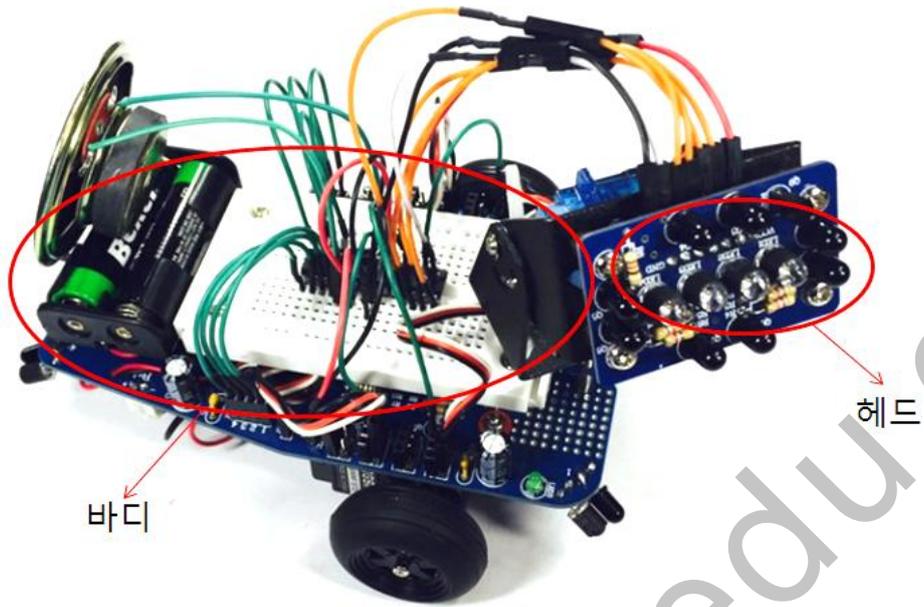
본제품의 한글본의 저작권은 게임플러스에듀에 있으며 무단도용을 금합니다

## MR. General programming

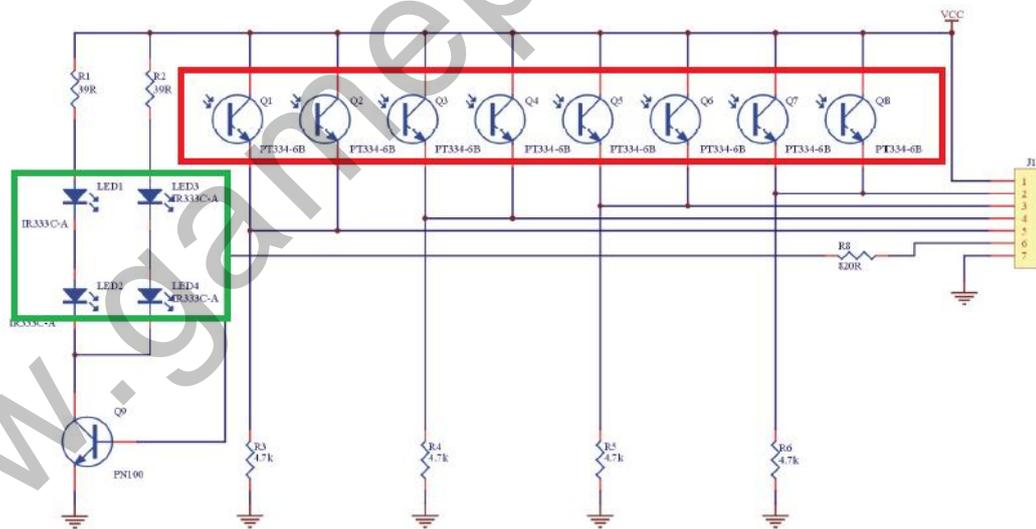


### 1. 소개

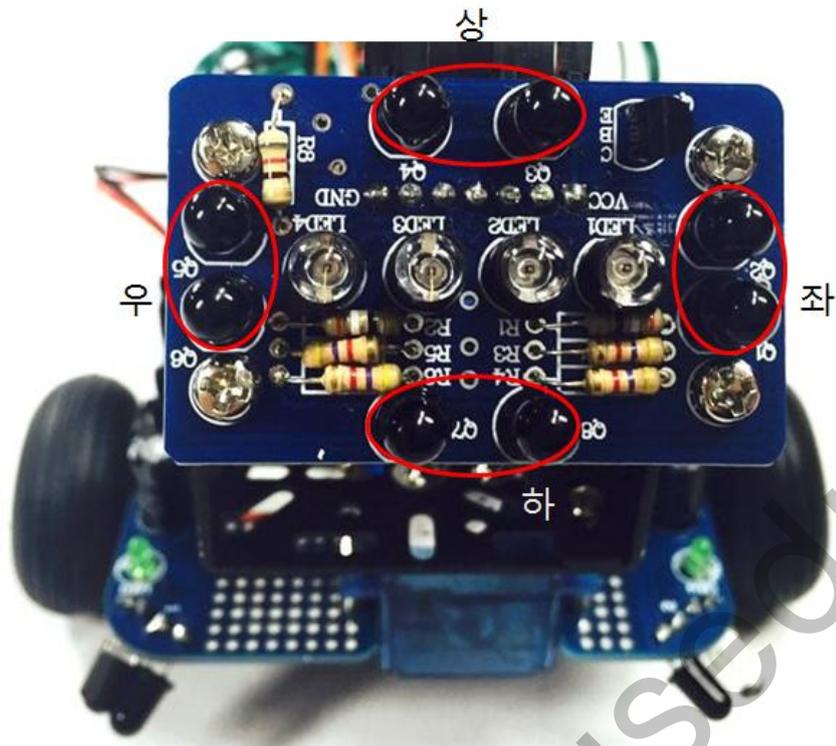
MR.General 키트는 IR센서를 이용해 동작을 제어하는 자동차로서 저항, 커패시터, 트랜지스터 등 기본 소자를 이용해 자동차를 조립하여 하드웨어에 대한 전반적인 이해를 할 수 있을 뿐만 아니라, 마이크로 컨트롤러를 연결할 수 있기 때문에 소프트웨어를 이용해 하드웨어를 제어하는 연습을 할 수 있는 키트입니다. MR.General 키트는 헤드와 바디 이렇게 두 부분으로 나누어져 있습니다.



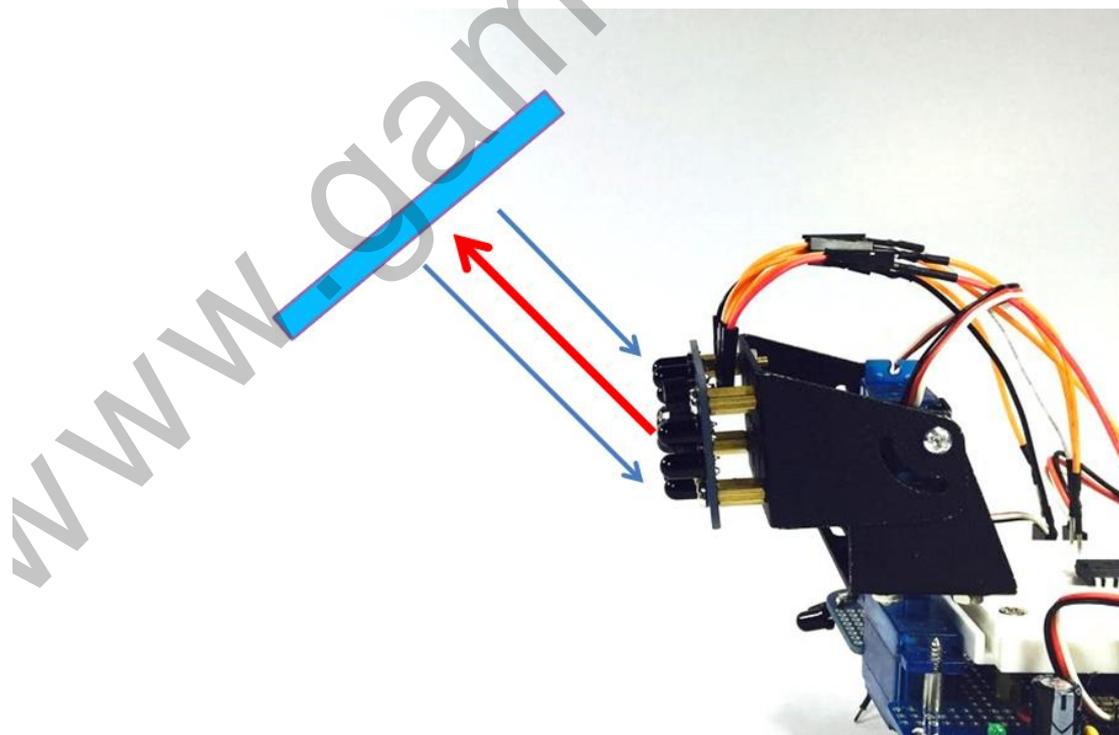
헤드 부분에는 IR 센서가 8개 달려 있으며, 센서를 이용해 물체의 위치를 파악할 수 있습니다. 헤드의 동작은 아래 compound eye PCB 회로도를 보며 설명하도록 하겠습니다.



초록색으로 표시된 IR LED를 통해 일정한 간격으로 적외선이 방출하게 됩니다. 방출된 적외선이 물체에 반사되면 다시 compound eye에 도달하게 되며, 빨간색으로 표시된 IR 센서를 통해 감지하게 됩니다.

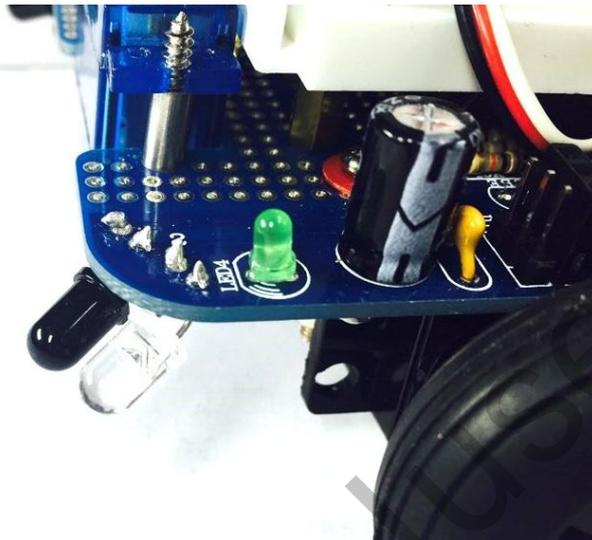


IR 센서는 위, 아래, 좌, 우 이렇게 4부분으로 나누어져 있는데 각 방향에 도달하는 적외선의 차이를 감지하여 물체의 위치를 판단하게 됩니다. 예를들어 물체가 센서의 위쪽에 위치하게 된다면 위쪽에 위치한 IR 센서가 감지하는 적외선의 수치가 아래쪽 IR 센서보다 더 크게 나오기 때문에 이 차이를 계산하여 물체의 상하 위치를 파악할 수 있습니다.



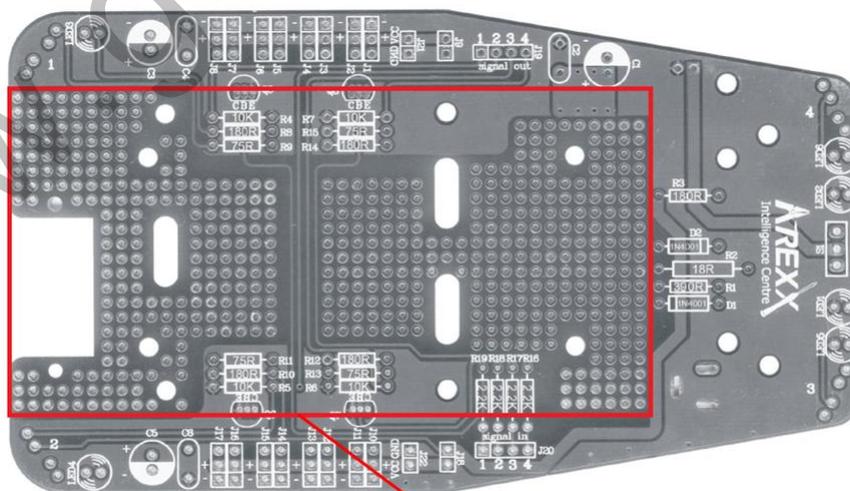
마찬가지로 좌우 IR센서를 이용해 물체의 좌우 위치를 파악할 수 있으며, 물체와 떨어진 거리는 4개의 센서의 평균치를 구해 떨어진 거리를 판단합니다.

바디에는 PCB 아랫쪽에 IR 센서와 IR LED가 4개씩 달려 있으며, 각각의 센서가 거리를 판단해 장애물을 감지 하거나, 테이블 가장자리에서 키트가 바닥으로 떨어지는 것을 막을 수 있게 합니다.



기본적으로 제공되는 소스코드를 통해 제어되는 MR.General의 동작은 헤드를 통해 감지된 물체와 일정한 거리를 유지하며, 장애물이 있을 때 부딪치는 것을 막아줍니다. 마지막 페이지에 링크된 동영상상을 통해 키트 동작을 확인할 수 있습니다. 아래 소개된 설명을 따라 MR.General을 프로그래밍 한다면, 동영상에서 시연된 동작을 실행 시킬 수 있습니다.

MR.General 키트 바디 PCB에는 여유 공간이 있습니다.



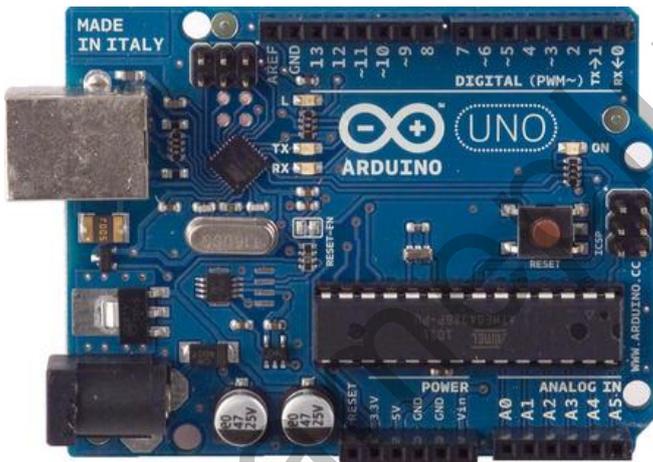
여유 공간

이 부분에 새로운 센서를 납땜하여 MR.General 동작에 새로운 기능을 추가할 수 있습니다. 지금부터 기본 제공되는 소스코드를 통해 MR.General의 동작을 제어해 보겠습니다. 아래의 과정이 모두 끝난다면, 지금까지 배운 내용을 통해 소스코드를 변경하거나 새로운 센서를 추가하여 하드웨어를 제어하는 연습을 해보시기 바랍니다!

## 2. 준비물

MR.General의 제작 매뉴얼을 통해 하드웨어 구성을 끝마쳤습니다. 다음은 MR.General을 프로그래밍 하여 작동시키도록 하겠습니다. 준비물은 아래와 같습니다.

### 1) Arduino UNO



MR.General을 동작 시키는 마이크로 컨트롤러로 사용됩니다.

2) Cable Pack



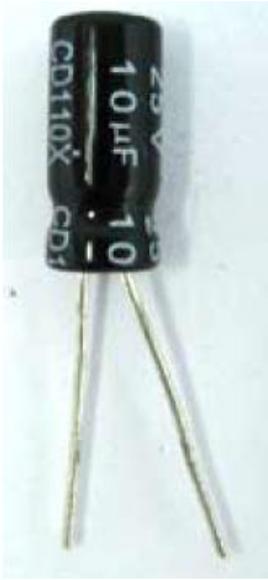
브레드보드와 MR.General, 마이크로 컨트롤러를 연결하는데 사용됩니다.

3) Speaker



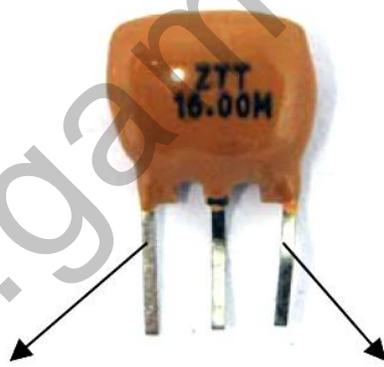
MR.General의 사운드를 출력합니다.

4) Electrolytic capacity



스피커의 회로 연결을 하는데 사용됩니다.

5) Crystal oscillator



right pin

left pin

마이크로 컨트롤러에 주파수를 할당하기 위해 사용됩니다. 16MHz oscillator 가 사용되며, right pin은 입력, left pin은 출력, 가운데 핀은 ground를 나타냅니다. 위의 사진은 정면에서 바라봤을 때 방향이므로 방향 설정에 주의하시기 바랍니다.

### 3. 서보 모터 센서 값 조절

MR.General을 프로그래밍 하도록 하겠습니다. 그 전에 서보 모터의 센서 값을 초기화 하는 작업을 진행 하도록 하겠습니다. 센서 값을 초기화 하는 이유는 서보 모터에 따라 정지할 때 값이 다르기 때문에 제공되는 소스코드에 설정된 센서 값과 실제로 사용되는 서보 모터의 센서 값이 다르기 때문입니다. 실제 센서 값을 측정하여 소스코드의 센서 값을 변경하도록 하겠습니다.

#### 1) 서보 모터 종류

MR.General에 사용되는 서보 모터의 종류는 바퀴 서보 모터, 센서 서보 모터 2가지 입니다. 2가지 서보 모터의 차이점은 360도 회전이 가능하냐 입니다. 바퀴 서보 모터는 바퀴를 동작 시키기 때문에 360도 연속 회전을 해야 합니다. 반면에 센서 서보 모터는 헤드의 움직임을 동작 시키기 때문에 일정한 각도 내에서 움직이게 됩니다. 이러한 차이 때문에 2개의 서보 모터는 입력 받은 센서 값에 따라 동작도 달라지게 됩니다.

##### ① 바퀴 서보 모터

바퀴 서보 모터의 다른 이름은 continuous rotation servo motor 입니다. 이 서보 모터는 값을 입력 받으면, 그 값에 해당하는 속도로 움직이게 됩니다. 예를 들어 1500을 입력 받을 때 모터가 정지하게 된다면 1600일 때 오른쪽으로 일정한 속도로 회전하게 되며, 숫자를 증가 시킬수록 속도는 더 빨라지게 됩니다. 1400을 입력 받으면 모터는 왼쪽으로 회전하게 되며, 숫자를 감소 시킬수록 속도는 더 빨라지게 됩니다. 증가할 수 있는 속도는 제한되어 있으며, 그 값은 서보 모터에 따라 다르게 정의되어 있습니다.



## ② 센서 서보 모터

센서 서보 모터의 다른 이름은 standard servo motor 입니다. 이 서보 모터는 값을 입력 받으면, 그 값에 해당하는 각도로 회전하게 됩니다. 예를 들어 1500일 때 서보 모터가 중앙에 위치하게 된다면 1600일 때 오른쪽으로 일정한 각도로 움직인 뒤 정지하며, 숫자를 증가 시킬수록 이동하는 각도는 더 커지게 됩니다. 1400을 입력 받으면 모터는 왼쪽으로 이동하며, 숫자를 감소 시킬수록 왼쪽으로 이동하는 각도는 더 커지게 됩니다. 일반적으로 standard servo motor는 모터의 종류에 따라 180~270도까지 이동할 수 있는 각도가 제한 되어 있으며, 360도 회전은 불가능 합니다.



일반적으로 서보 모터의 초기값은 1500으로 설정되어 있지만, 미세하게 그 값이 달라질 수 있습니다. 특히 바퀴 서보 모터를 동작 시킬 때 정지하는 초기값이 달라지게 되면, 처음 키트를 켜올 때 움직이게 되므로, 키트의 올바른 작동에 방해가 됩니다. 그렇기 때문에 정확한 초기값을 측정해야 합니다.

제공된 MR\_General\_ATmega328 소스코드에는 서보 모터에 따라 초기값이 설정되어 있습니다. 아래 사진은 MR\_General\_ATmega328 소스코드의 상수가 선언된 코드의 일부분을 나타냅니다.

```
Mr_General_ATmega328 | 아두이노 1.6.5
파일 편집 스케치 도구 도움말

Mr_General_ATmega328 IO_pins.h pitches.h

#include <Servo.h>
#include "pitches.h"
#include "IO_pins.h"

// define constants
byte LRscalefactor=6;
byte UDscalefactor=6;
int distancemax=200;
int bestdistance=550;
int neckLRcenter=1490;
int neckUDcenter=1300;
int leftmotorstop=1460;
int rightmotorstop=1480;
int LRmax=neckLRcenter+700;
int LRmin=neckLRcenter-700;
int UDmax=neckUDcenter+700;
int UDmin=neckUDcenter-200;

// define global variables
long time;
byte boredom;
int pan=neckLRcenter;
int tilt=neckUDcenter;
```

```
int neckLRcenter = 1490;
```

좌우 센서 서보 모터의 중간 값으로 키트가 처음 시작할 때 헤드의 좌우위치를 설정합니다.

```
int neckUDcenter= 1300;
```

위아래 센서 서보 모터의 중간 값으로 키트가 처음 시작할 때 헤드의 위아래위치를 설정합니다.

```
int leftmotorstop = 1460;
```

왼쪽 바퀴 서보 모터가 정지할 때 값입니다.

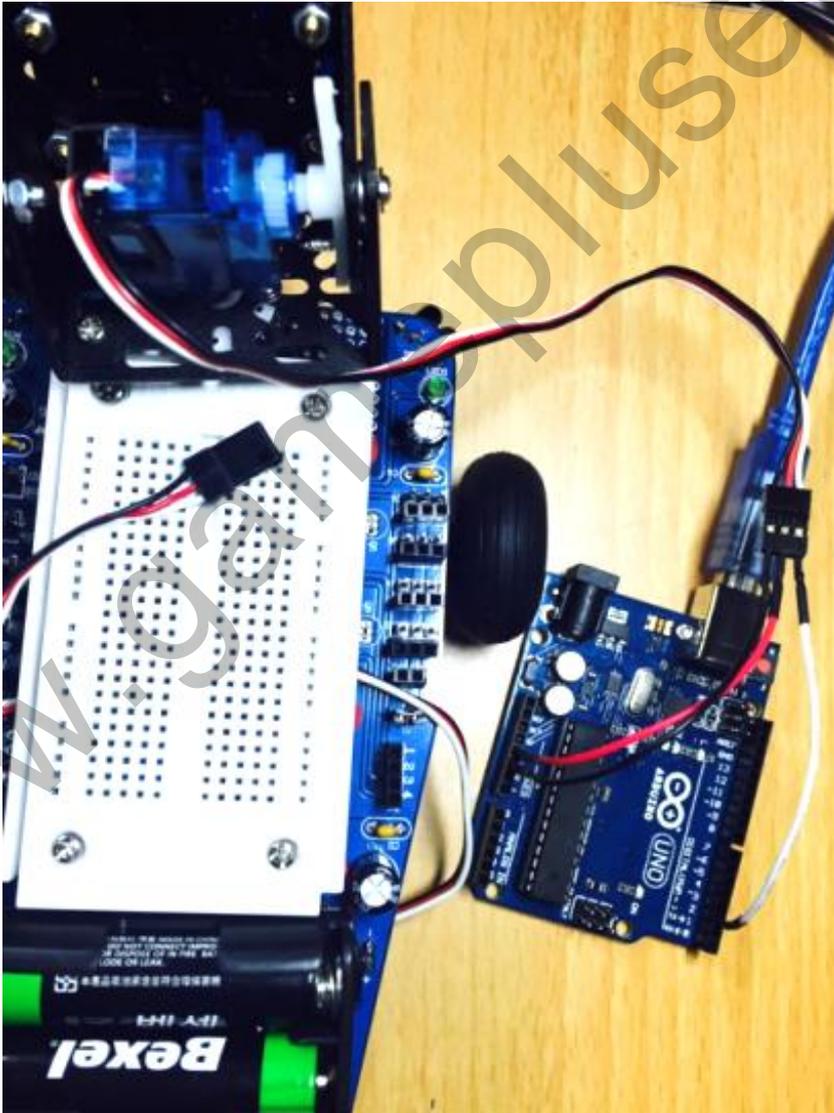
```
int rightmotorstop = 1480;
```

오른쪽 바퀴 서보 모터가 정지할 때 값을 의미합니다.

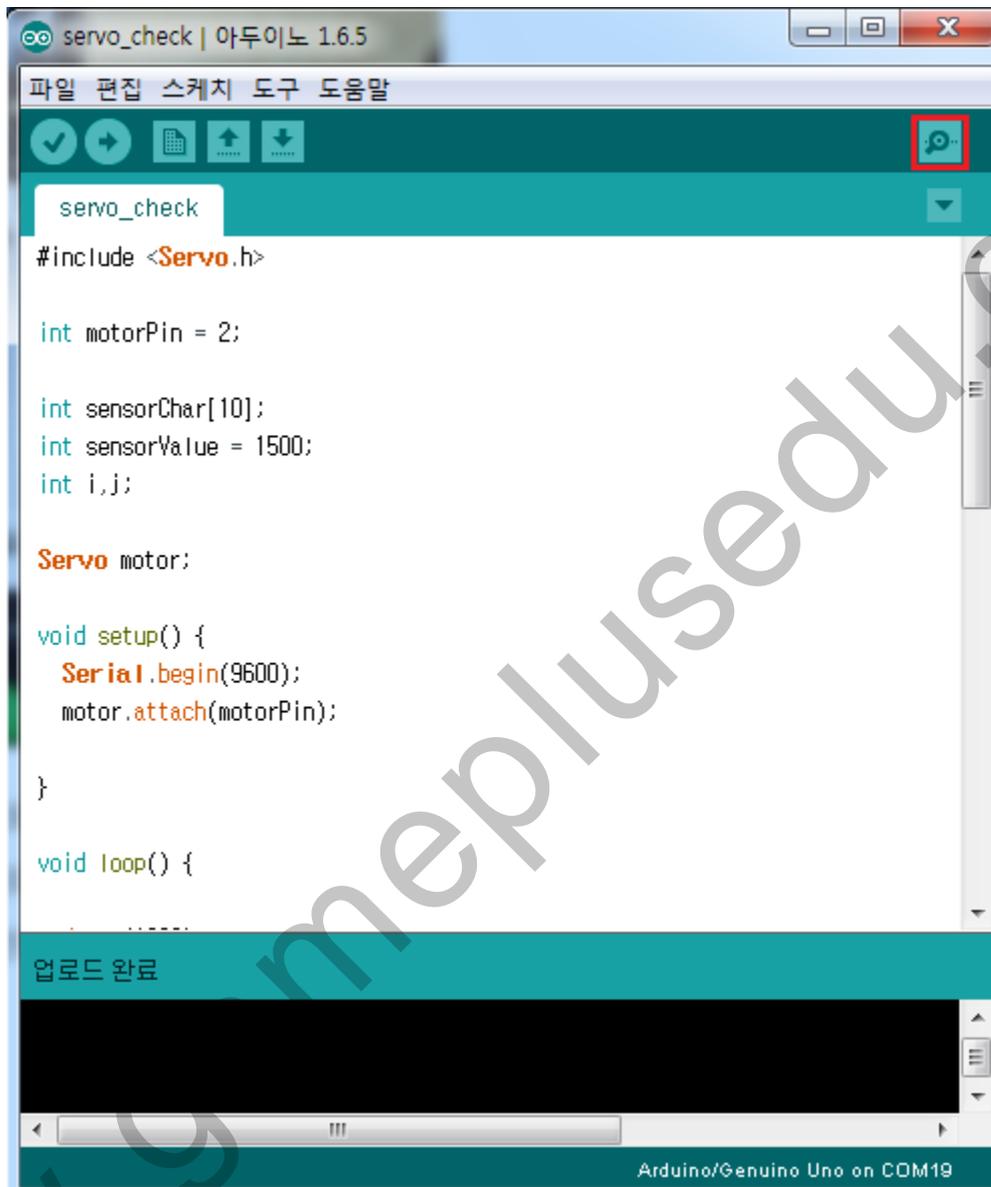
이 값들은 서보 모터에 따라 다르므로 값을 측정하여 변경해 주어야 합니다. 먼저 neckUDcenter, neckLRcenter 값을 측정해 보도록 하겠습니다.

## 2) neckUDcenter, neckLRcenter 값 측정

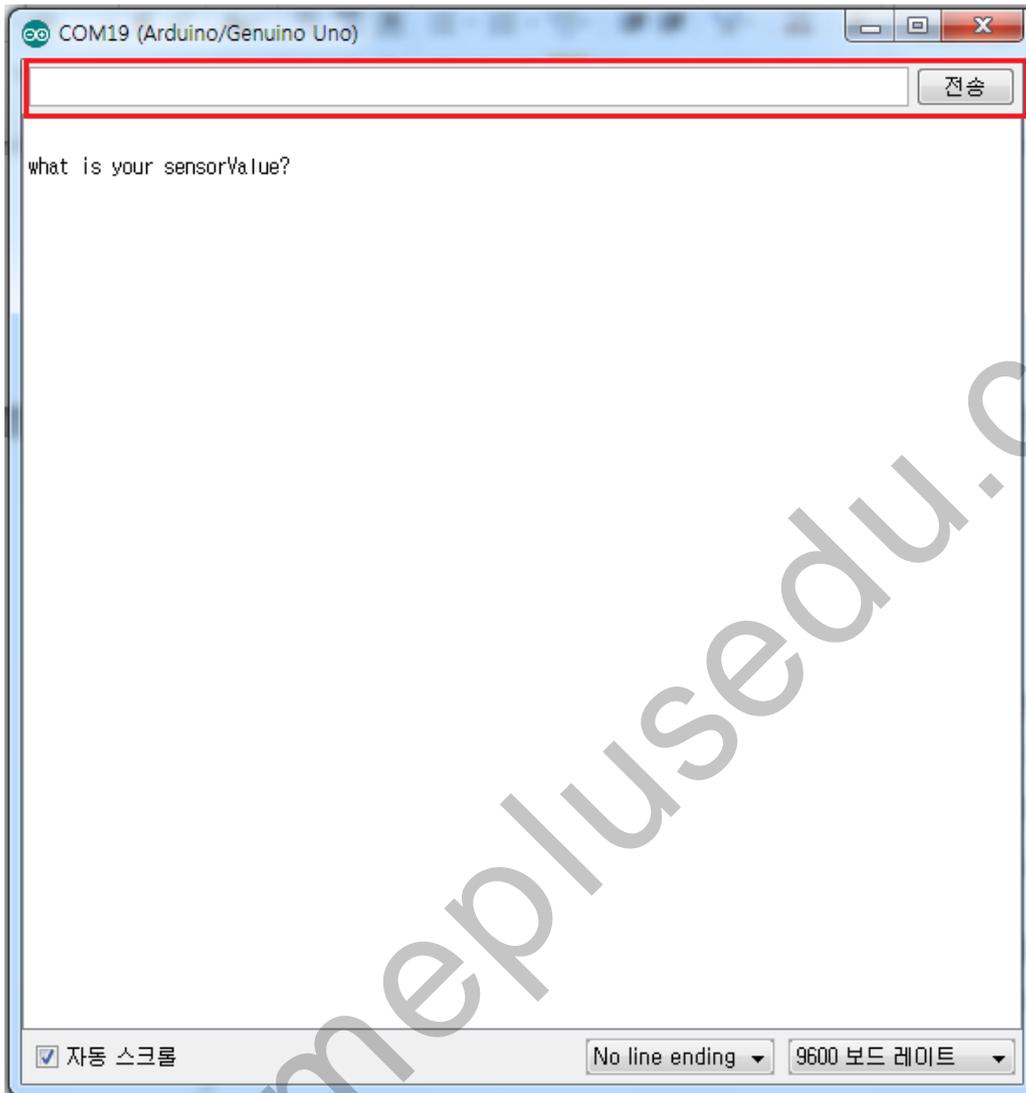
neckUDcenter 값을 먼저 측정해 보도록 하겠습니다. 아래 사진과 같이 센서 서보모터와 Arduino UNO를 서로 연결해 줍니다. 빨간선과 검정선을 각각 UNO에서 5V와 GND에 연결해 주며, 흰색 선을 DIGITAL 2번 핀과 연결 합니다.



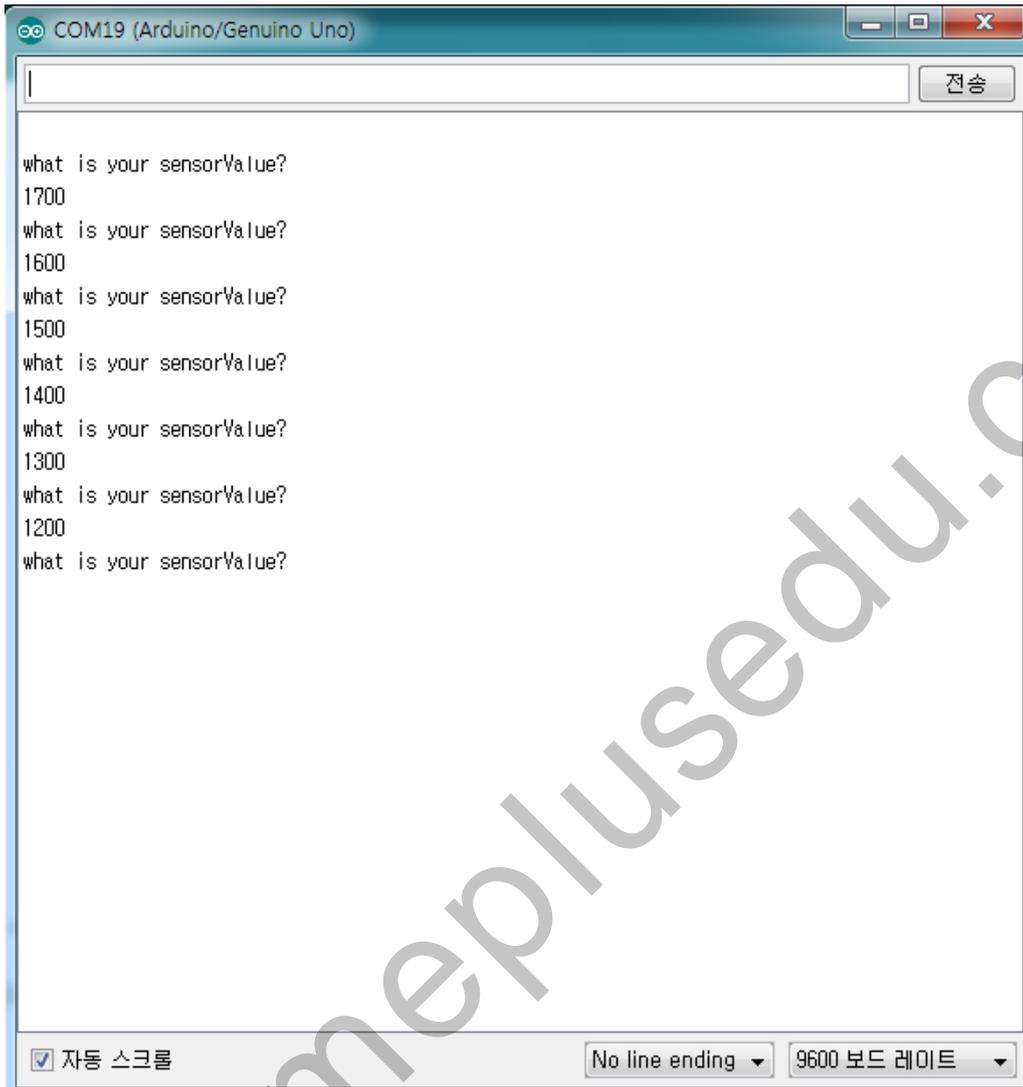
Arduino UNO를 컴퓨터와 연결하여 제공된 소스 코드인 servo\_check를 이용해 정확한 값을 측정합니다. 컴퓨터와 Arduino UNO를 연결하여 servo\_check를 업로드 해주세요.



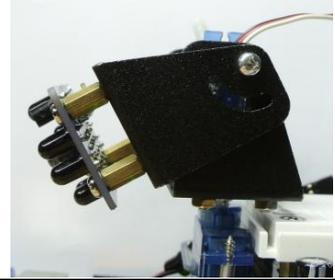
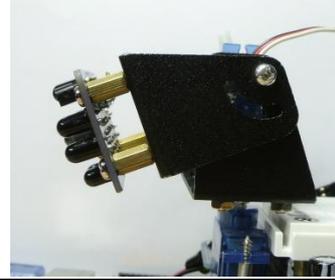
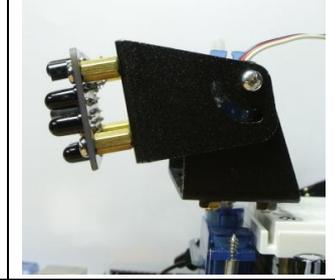
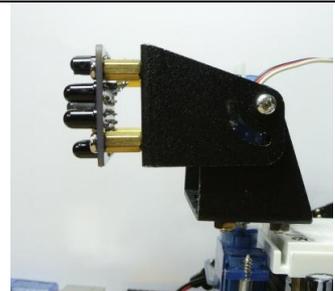
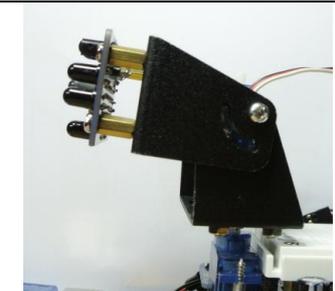
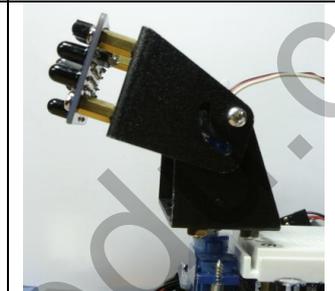
업로드가 성공하면 위의 사진에서 빨간 네모로 표시된 부분을 클릭해 시리얼 모니터를 열어줍니다.



시리얼 모니터의 화면은 위의 그림과 같습니다. 빨간색 부분에 숫자를 입력하여 sensorValue값을 변화 시킬 수 있습니다. sensorValue는 neckUDcenter, neckLRcenter에 입력되는 값을 의미하며, 초기값은 1500으로 설정되어 있습니다. 100단위로 값을 변화시켜 보겠습니다.

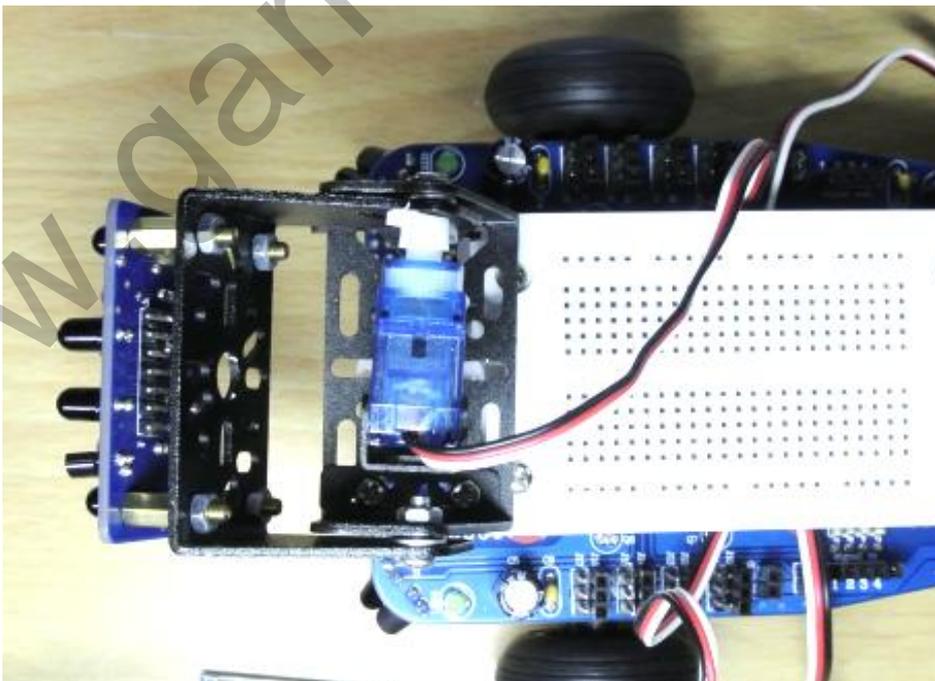


입력되는 값에 따라 서보 모터의 움직임은 아래 표와 같습니다.

		
sensorValue = 1700	sensorValue = 1600	sensorValue = 1500
		
sensorValue = 1400	sensorValue = 1300	sensorValue = 1200

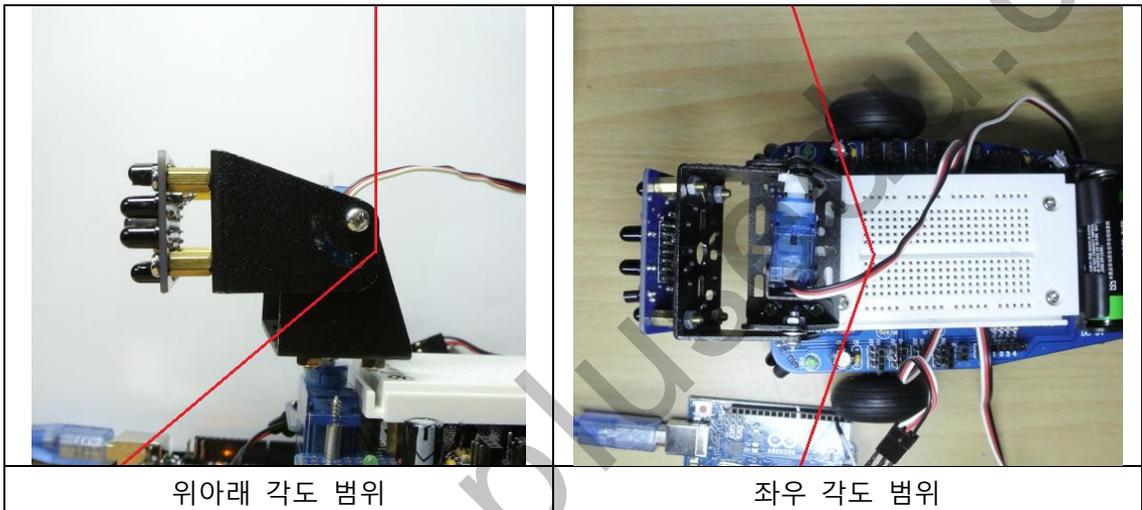
서보모터의 위치가 적당한 위치가 될 때 값을 선택하여 그 때의 sensorValue를 neckUDcenter 으로 선택합니다. 저는 sensorValue = 1400일 때 위치가 가장 적합하므로 neckUDcenter = 1400으로 설정하도록 하겠습니다. 이 값은 나중에 MR.General을 작동시킬 때 필요하므로 반드시 기록해 두세요.

같은 방법으로 neckLRcenter 값을 측정해 주세요. 적당한 위치는 아래 그림과 같습니다.



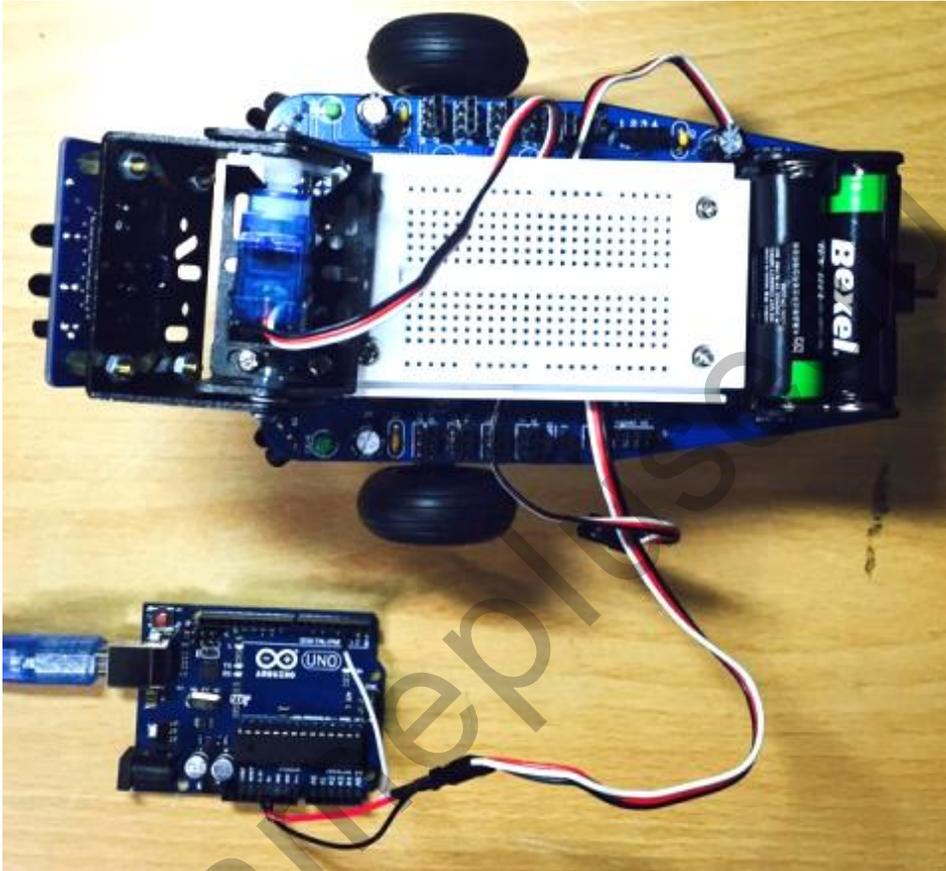
헤드가 정면을 바라보도록 neckLRcenter 값을 설정해 주세요. 이 값 역시 반드시 기록해 두시기 바랍니다.

서보모터는 움직임이 제한 돼 있습니다. sensorValue 값을 변화시켜 모터가 좌우, 위아래로 아래 그림에 나와있는 각도까지 움직일 수 있는지 확인해 주세요. 만약 sensorValue 값을 변화시켜도 서보모터가 작동하지 않는다면 모터의 제한 각도를 벗어난 것을 의미합니다. 헤드를 다시 조립해, 모터가 작동 범위 내의 각도로 들어올 수 있도록 조정해 주시기 바랍니다.

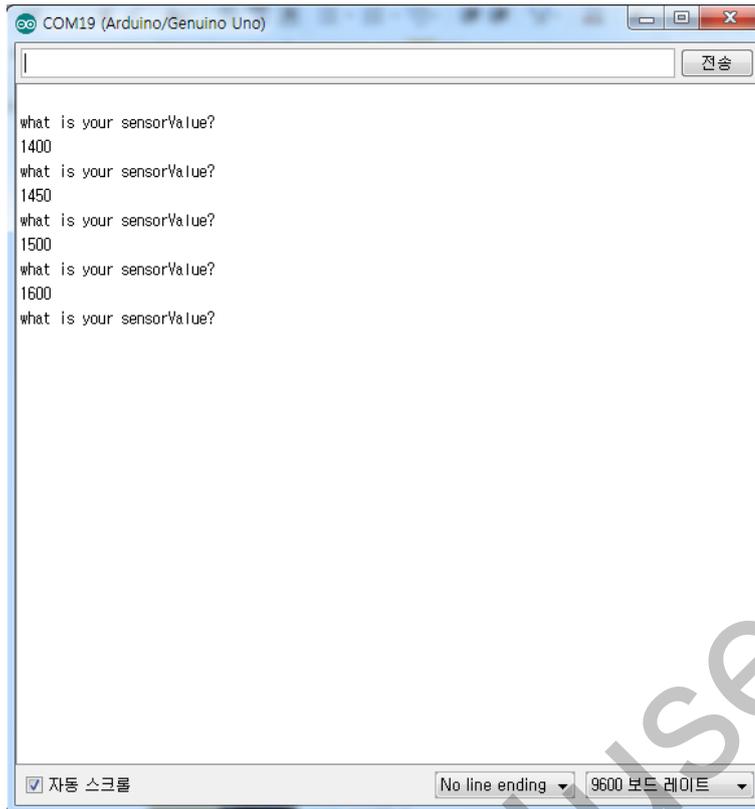


### 3) leftmotorstop, rightmotorstop 값 측정

leftmotorstop 값을 측정해 보도록 하겠습니다. 아래 사진과 같이 바퀴 서보모터와 Arduino UNO를 서로 연결해 줍니다. 빨간선과 검정선을 각각 UNO에서 5V와 GND에 연결해 주며, 흰색 선을 DIGITAL 2번 핀과 연결 합니다.



1)의 과정과 동일하게 servo\_check를 업로드 하여 시리얼 모니터를 열어줍니다.



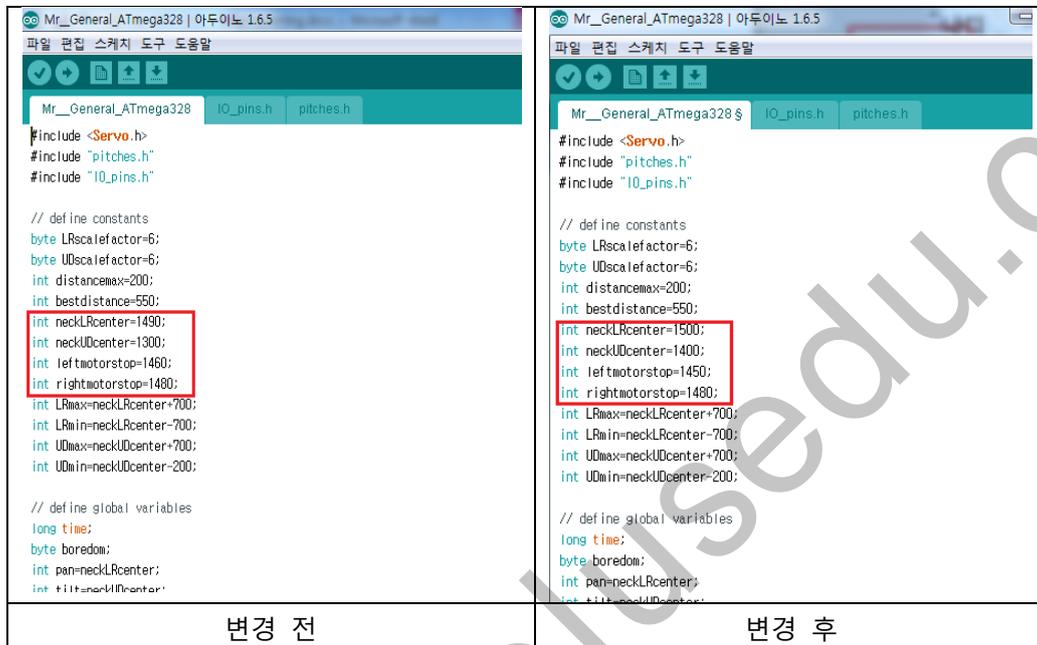
시리얼 모니터를 이용해 여러 개의 값을 입력해 줍니다. 저는 1400, 1450, 1500, 1600 의 값을 차례대로 입력해 주었습니다. 바퀴 서보모터의 동작을 살펴보면 1450에서 모터가 정지하며, 이 값을 기준으로 1400일때와 1500일 때 방향이 바뀌는 것을 확인할 수 있습니다. sensorValue = 1450일 때 바퀴가 정지 하므로 이 값이 바퀴 서보모터의 정지 값이라는 것을 알 수 있습니다. leftmotorstop = 1450으로 설정하도록 하겠습니다. rightmotorstop 역시 같은 방법으로 측정해 주시기 바랍니다. 두개의 값은 MR.General의 동작을 위해 반드시 기록 해두시기 바랍니다.

#### 4. 마이크로 컨트롤러 연결

MR.General과 마이크로 컨트롤러를 연결해보도록 하겠습니다. 마이크로 컨트롤러는 MR.General의 동작을 제어하는 두뇌와 같으므로 마이크로 컨트롤러를 잘못 연결한다면 MR.General이 작동하지 않을 수 있습니다. 이 점에 유의하여 마이크로 컨트롤러를 연결해 주시기 바랍니다.

1) Arduino UNO를 이용한 소스코드 업로드

제공된 MR\_General\_ATmega328를 업로드 하기 전, 측정한 서보모터 값을 변경하도록 하겠습니다. 아래 그림과 같이 초기값을 변경해 주세요.



변경이 완료되었다면, 컴파일 후 Arduino UNO에 업로드 해주세요.

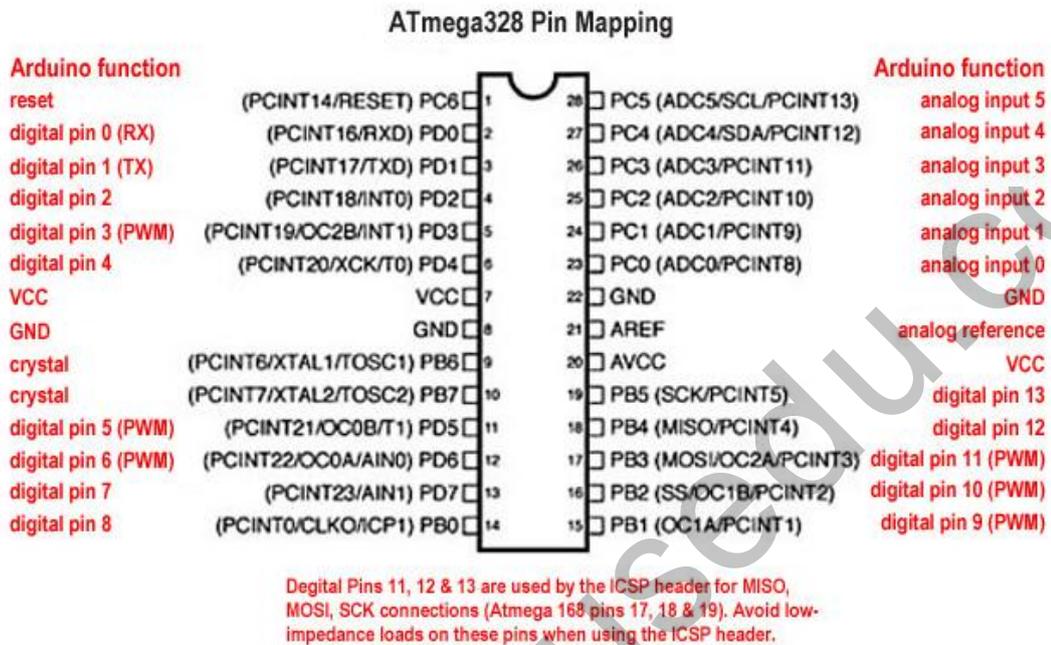
## 2) ATmega328 분리

제공된 코드가 업로드된 Arduino Uno 보드에서 ATmega328을 분리합니다. ATmega328을 분리하는 이유는 상대적으로 UNO보드 보다 ATmega328의 크기가 작기 때문에 MR.General의 이동성을 높이기 위해서 입니다.

ATmega328은 아래 그림에서 동그라미 친 부분입니다. 핀셋을 이용하면 쉽게 분리해 낼 수 있습니다. 분리 시 ATmega328의 핀이 부러지지 않도록 주의해서 분리해 주세요.



Atmega328의 pin mapping은 아래 그림과 같습니다. 브레드 보드 연결 시 참고해 주세요.

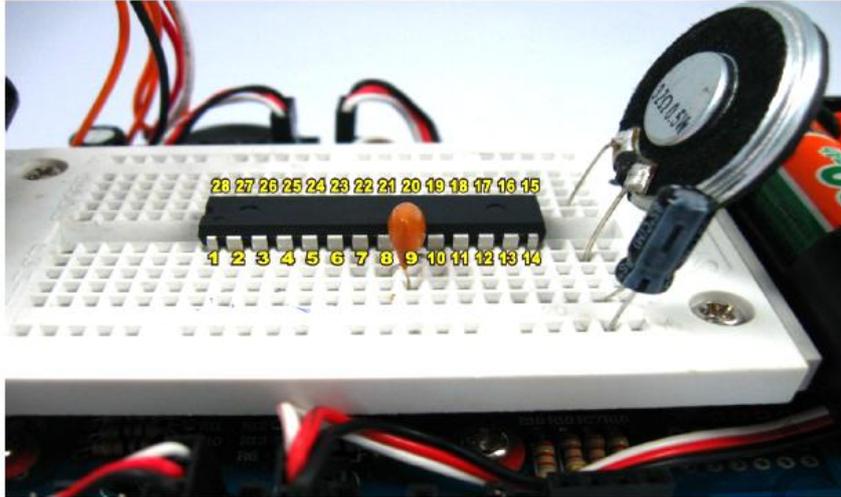


ATmega328 Pin Map

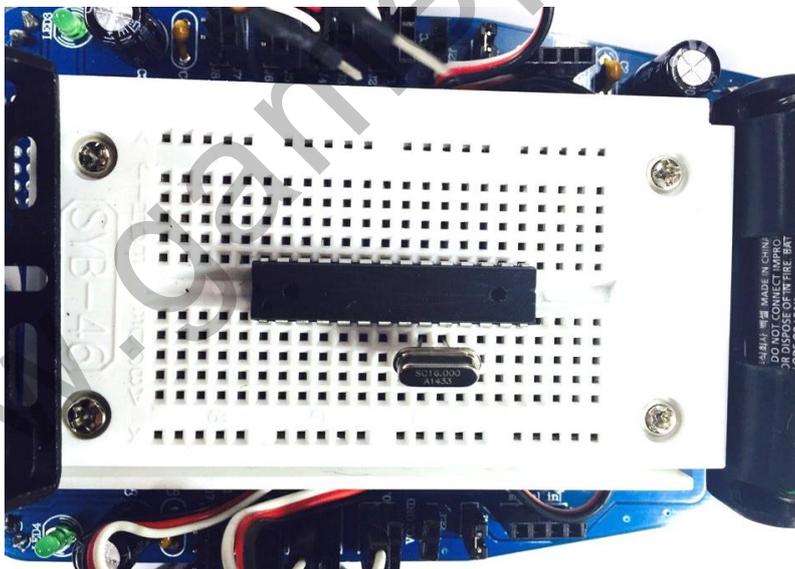
DEFINE	I O PINS	ANNOTATION
IRleft	analog input A3	Compound Eye Left
IRright	analog input A2	Compound Eye Right
IRup	analog input A5	Compound Eye Up
IRdown	analog input A4	Compound Eye Down
IRleds	digital output D0	Compound Eye LEDs
neckleftrighpin	digital output D1	PAN Servo
neckupdownpin	digital output D2	TILT Servo
leftmotorpin	digital output D3	Left Motor Servo
rightmotorpin	digital output D4	Right Motor Servo
rightfrontsenpin	digital input D12	Front Right Sensor
leftfrontsenpin	digital input D11	Front Left Sensor
leftrearsenpin	digital input D10	Rear Left Sensor
rightrearsenpin	digital input D9	Rear Right Sensor
rightfrontLED	digital output D5	Front Right LEDs
leftfrontLED	digital output D6	Front Left LEDs
leftrearLED	digital output D7	Rear Left LEDs
rightrearLED	digital output D8	Rear Right LEDs
Speaker	digital output D13	Speaker

Form 1

### 3) ATmega328 연결

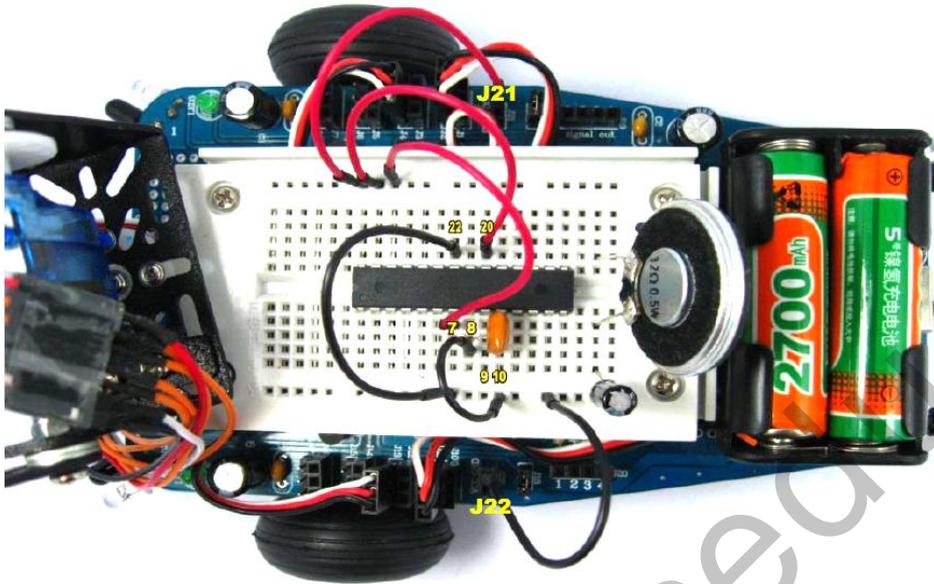


브레드 보드 위에 ATmega328을 연결해 주세요. 위의 그림과 같이 칩의 반원 부분이 MR.general의 헤드방향으로 오도록 합니다. 다음으로 crystal oscillator을 브레드 보드에 연결해 주세요. Left pin을 pin10, right pin을 pin9에 가운데 핀을 pin8에 연결합니다. crystal oscillator 를 연결 할 때 16MHz를 사용하여야 하며 2핀 crystal oscillator를 사용하면 pin8의 연결은 생략하셔도 좋습니다. 2핀일 때 연결은 아래 사진과 같습니다.



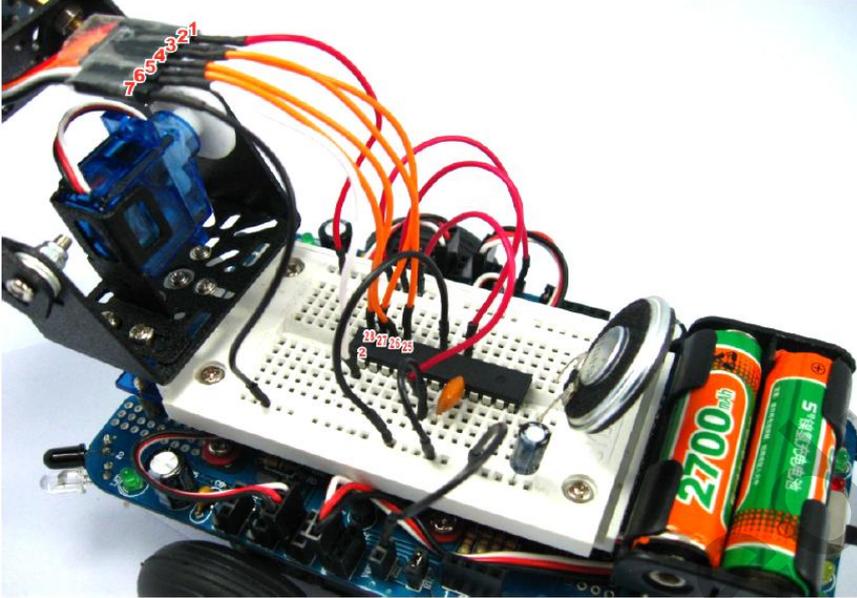
스피커와 Electrolytic capacity를 연결해 주세요. Electrolytic capacity는 극성을 가지며, -극은 ground에 연결해야 하므로 브레드 보드의 가장 바깥쪽 핀과 연결해 주세요.

#### 4) VCC, GND 연결

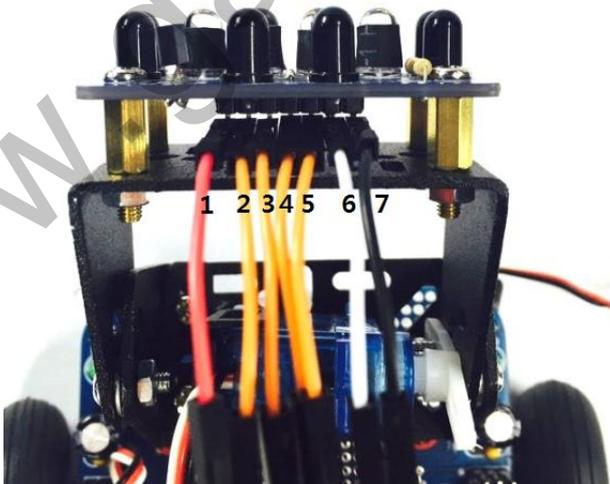


Atmega328의 pin mapping에 의하면 pin7과 pin20은 VCC, pin8과 pin22는 GND를 나타냅니다. ATmega328의 작동을 위해 VCC(+5V) 와 GND를 MR.General의 J21 또는 J22와 연결해 줍니다. J21와 J22 은 회로 상 서로 연결되어 있으므로 핀과 가까운 쪽으로 연결하면 됩니다. 점프선을 연결 할 때, VCC는 빨간색, GND는 검은색 선으로 연결하여 선의 구분이 쉽도록 합니다. 마지막으로 GND를 커패시터의 -극과 연결하여 스피커가 작동할 수 있도록 합니다.

## 5) Compound eye PCB 연결

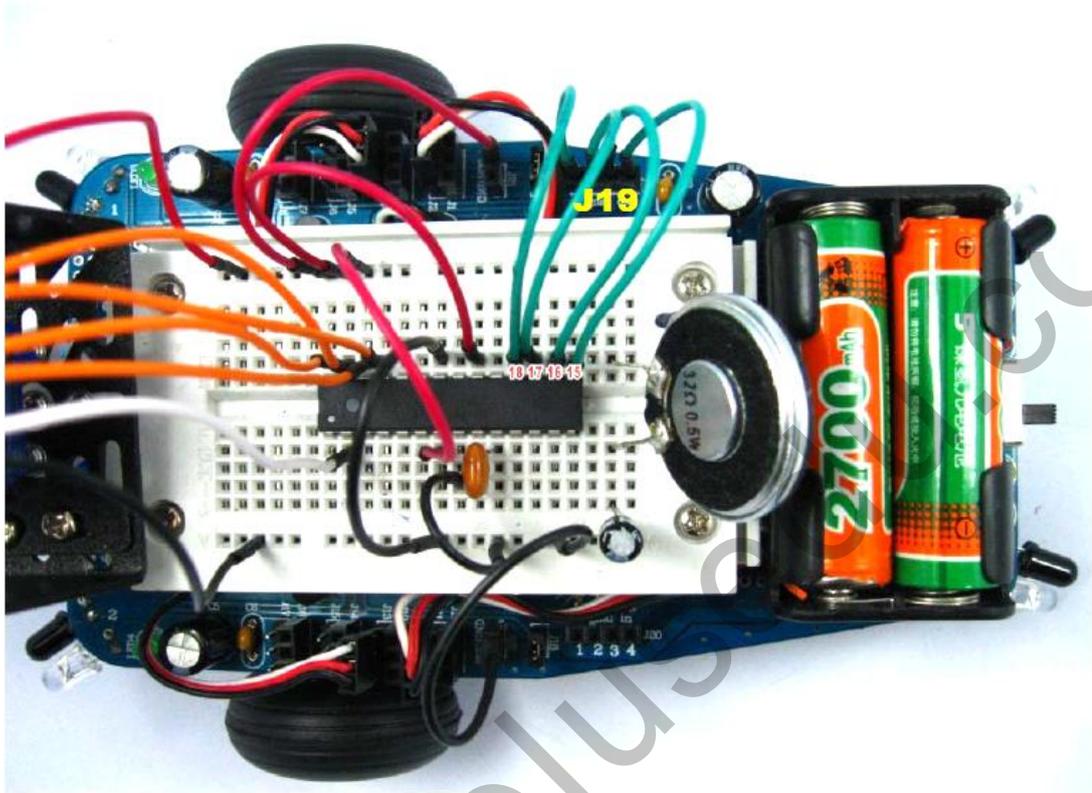


Compound eye PCB의 뒷면을 확인하면 7개의 핀은 VCC부터 GND까지 1~7번의 번호가 붙여져 있습니다. 1번과 7번을 각각 브레드보드에서 VCC와 GND에 연결해 줍니다. (1, pin7) (7, pin8)를 서로 연결합니다. 다음으로 2~5번은 순서대로 아래쪽, 오른쪽, 위쪽, 왼쪽 방향의 센서를 읽어냅니다. 이 핀을 순서대로 ATmega328의 pin28, pin26, pin27, pin25와 연결해 주세요. (2, pin28) (3, pin26) (4, pin27) (5, pin25) 를 서로 연결합니다. 연결 선은 Compound eye PCB의 연결 방향과 소스코드에 따라 연결단자가 바뀔 수 있습니다. 제공된 소스코드를 사용한다면 반드시 설명된 단자와 연결해 주세요. 6번은 센서의 on/off 단자로 pin2와 연결합니다. (6, pin2)



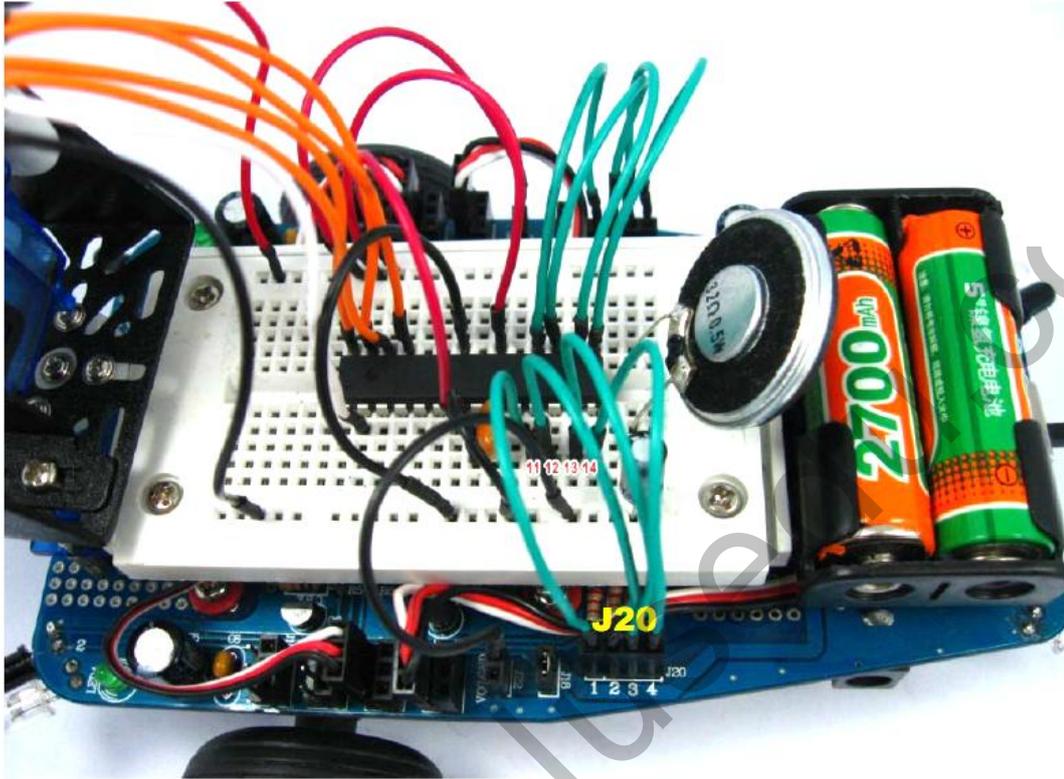
위의 사진과 같이 7pin header의 방향에 따라 실제 연결단자가 다를 수 있으니 주의하시기 바랍니다. VCC와 GND를 반드시 확인해 주세요.

6) Main PCB의 IR 센서 연결



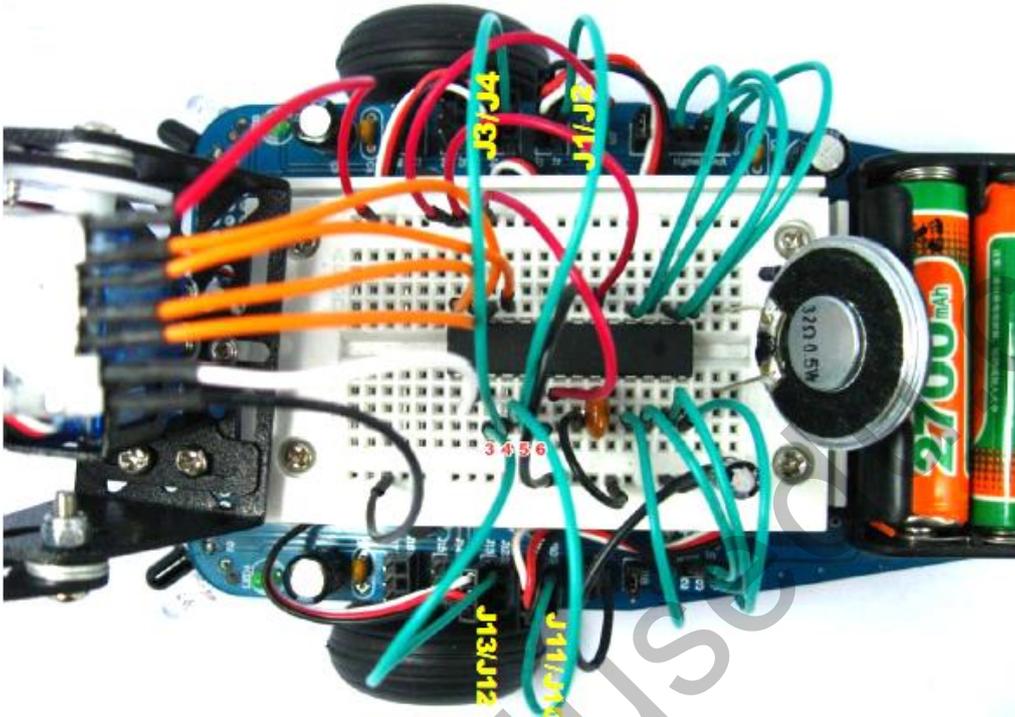
J19와 ATmega328을 연결해 주세요. J19는 Main PCB의 IR센서와 연결되어 있으며, J19를 통해 값을 읽어냅니다. J19의 1~4번핀은 Compound eye PCB가 앞쪽이라고 가정했을 때, 각각 오른쪽 앞, 왼쪽 앞, 왼쪽 뒤, 오른쪽 뒤 의 방향의 센서입니다. 이 핀을 순서대로 pin18, pin17, pin16, pin15와 연결해 주세요. (1, pin18) (2, pin17) (3, pin16) (4, pin15) 를 서로 연결합니다.

## 7) Main PCB의 LED 연결



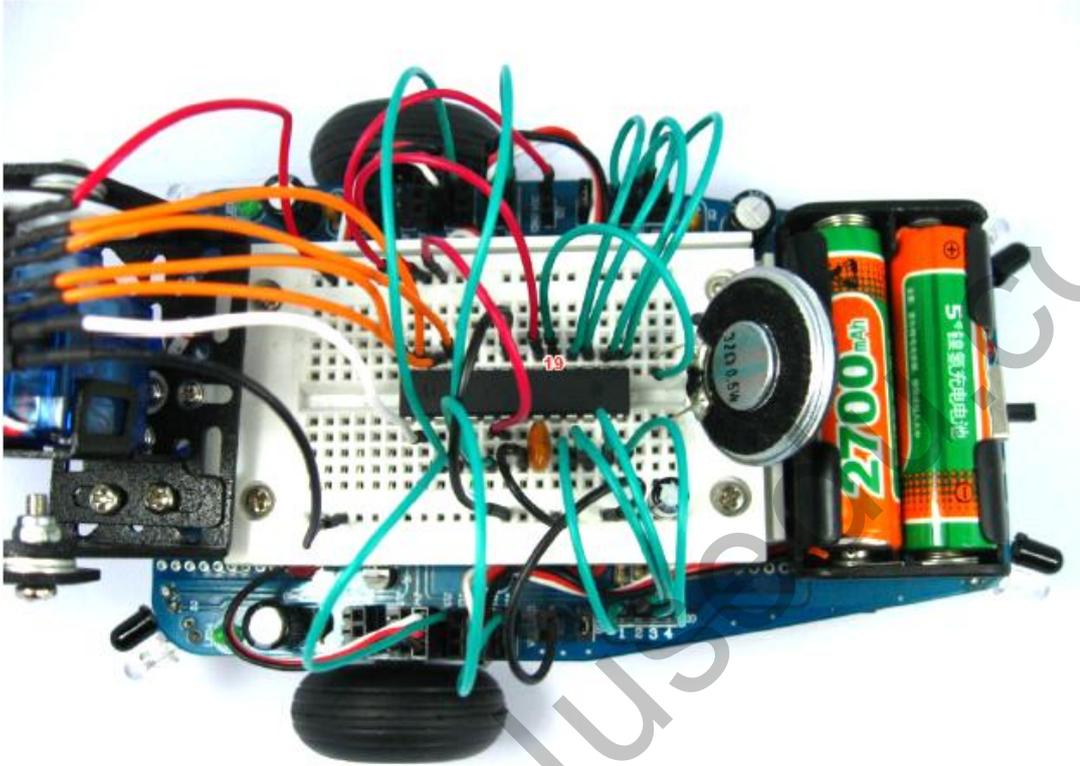
J20과 ATmega328을 연결해 주세요. J20은 Main PCB의 LED와 연결되어 있으며, J20을 통해 LED를 제어합니다. J20의 1~4번핀은 Compound eye PCB가 앞쪽이라고 가정했을 때, 각각 오른쪽 앞, 왼쪽 앞, 왼쪽 뒤, 오른쪽 뒤 의 방향의 LED입니다. 이 핀을 순서대로 pin11, pin12, pin13, pin14와 연결해 주세요. (1, pin11) (2, pin12) (3, pin13) (4, pin14) 를 서로 연결합니다.

## 8) 서보 모터 연결

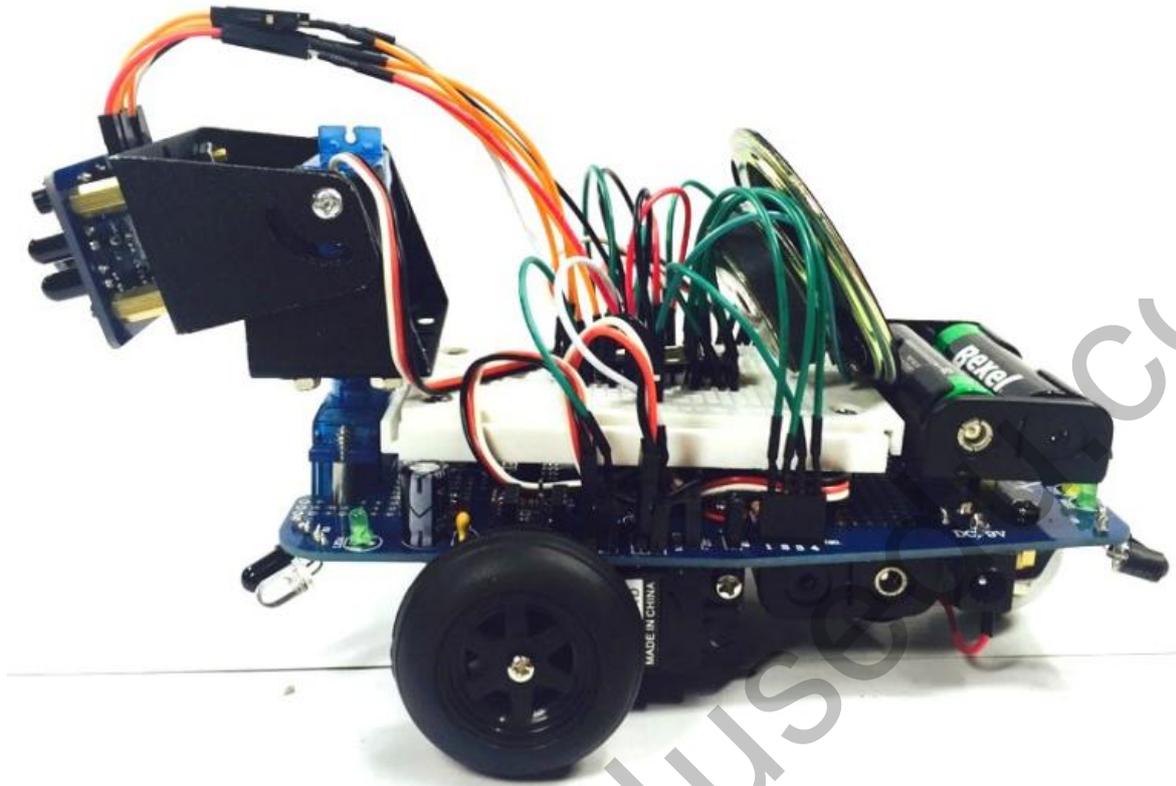


2개의 센서 서보 모터와 2개의 바퀴 서보 모터를 제어하기 위해 점프선을 연결합니다. 제공된 Main PCB 회로 도를 보면 J1~J8까지 서로 연결되어 있으며, J10~J17까지 서로 연결되어 있는 것을 확인할 수 있습니다. 이러한 특징을 이용해 서보 모터를 제어하는데 사용할 수 있습니다. 먼저 센서 서보 모터를 연결해 보도록 하겠습니다. 위 아래로 움직이는 서보 모터의 연결선을 J4와 연결합니다. 이때 흰색 선이 제어선 이므로 그림과 같이 제어 선이 브레드 보드를 향하도록 연결합니다. 다음으로 J3의 브레드 보드 쪽 단자와 pin4를 연결합니다. (J3, pin4) 다음으로 좌 우로 움직이는 서보 모터의 연결선을 J13과 연결하며 J12와 pin3을 연결합니다. (J12, pin3) 바퀴 서보 모터도 같은 방법으로 연결해 줍니다. 왼쪽 바퀴 모터를 J11과 연결하며, J10과 pin5를 연결합니다. (J10, pin5) 오른쪽 바퀴 모터를 J2와 연결하며, J2와 pin6을 연결합니다. (J2, pin6) 서보 모터와 ATmega328을 연결할 때 제어 선이 연결되어야 하므로 반드시 흰색선과 연결되어 있는지 확인하시기 바랍니다.

## 9) 스피커 연결



스피커의 커패시터와 연결된 반대쪽 단자와 pin19를 연결하여 MR.General이 동작 시 소리가 나올 수 있도록 합니다. 스피커의 출력에 따라 스피커의 크기가 달라질 수 있습니다.



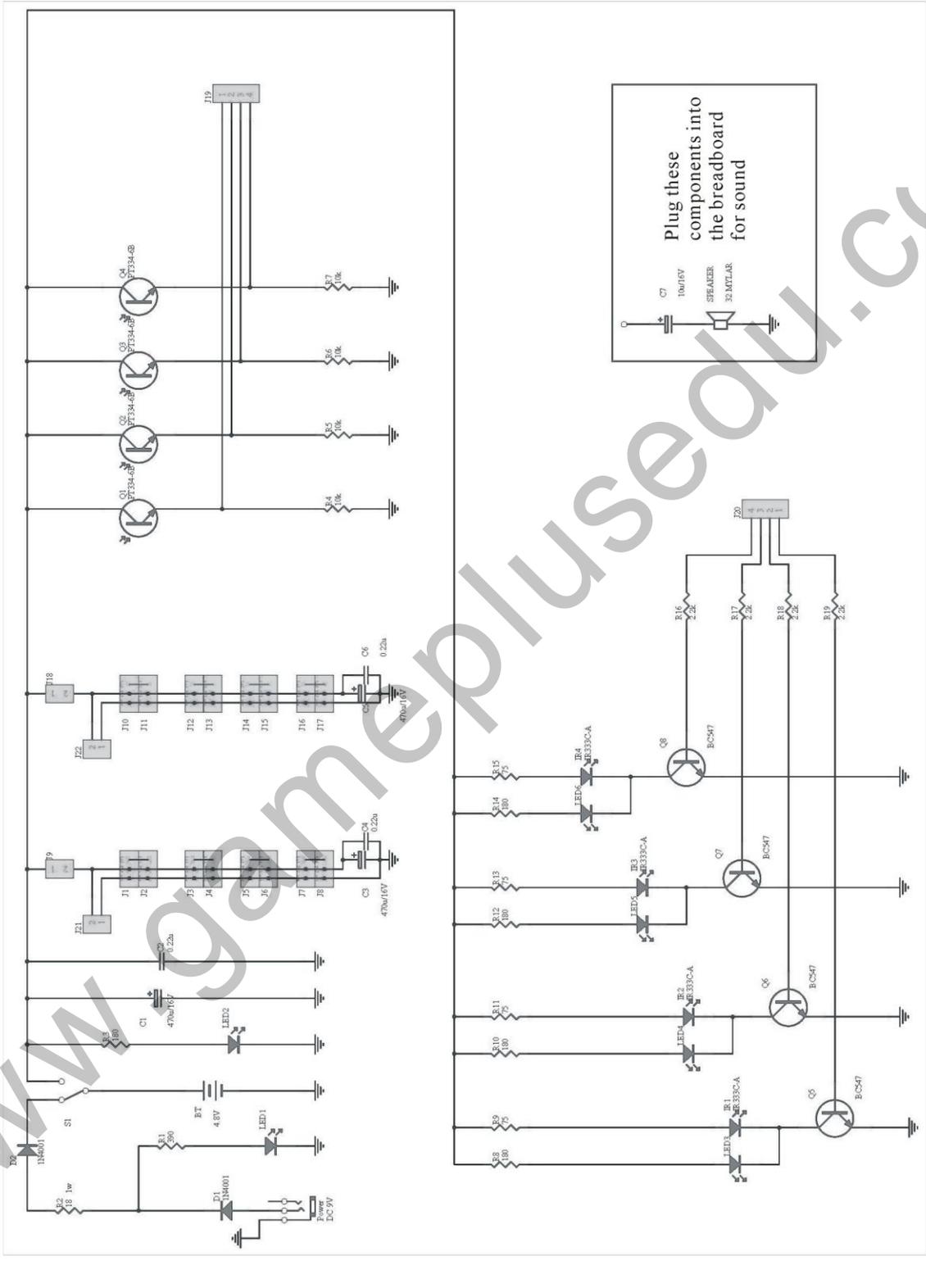
이제 MR.General의 작동을 위한 모든 작업이 끝났습니다. 뒤쪽에 위치한 on/off 스위치를 이용해 MR.General이 올바르게 동작하는 지 확인하시기 바랍니다.

시연 동영상은 아래 링크를 확인하시기 바랍니다.

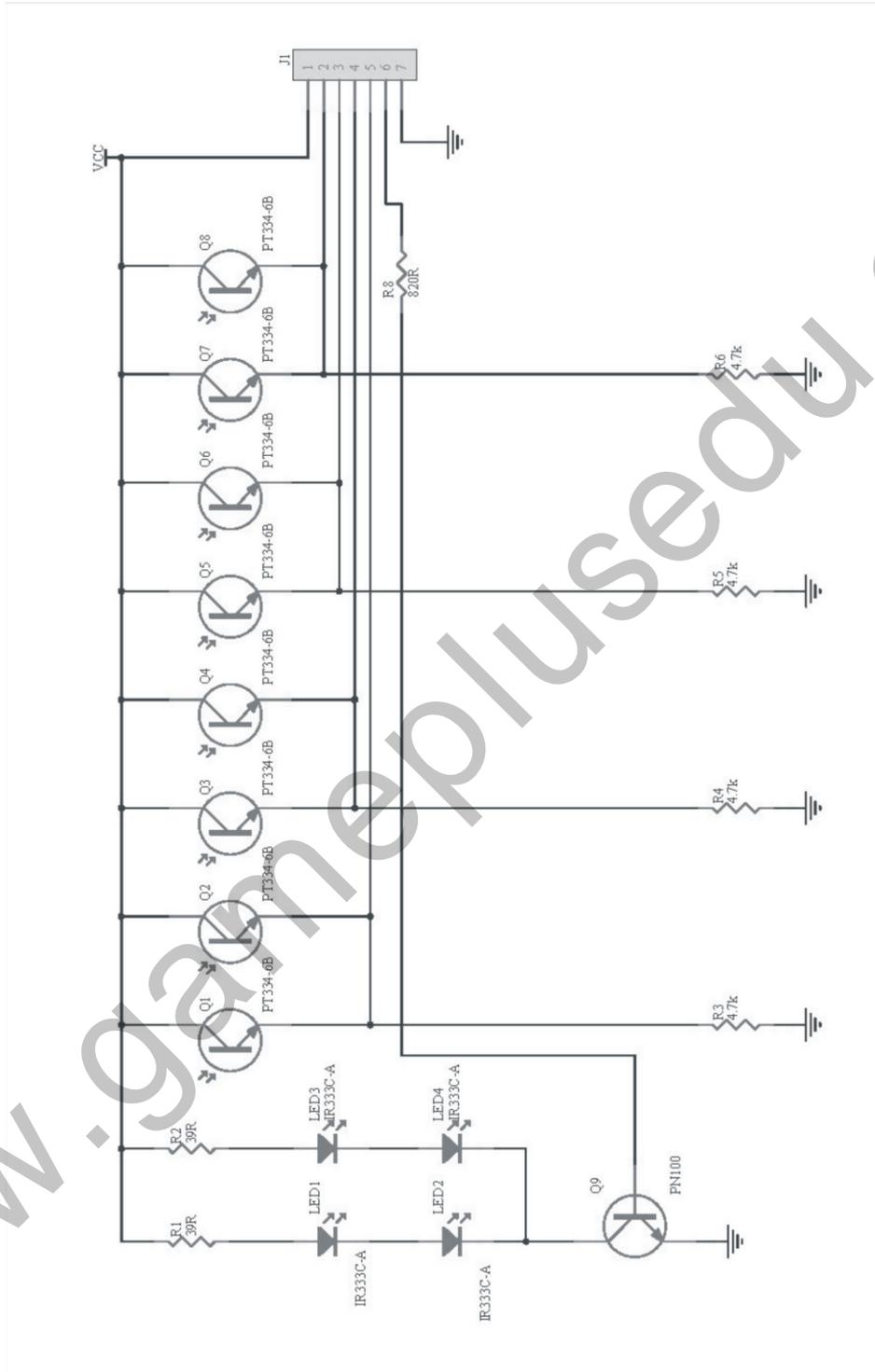
<https://youtu.be/fUMS2iGxeVU>

<https://youtu.be/fOtnMkpJMA>

# Main PCB



# Compound-eye PCB



www.gamereplusedu.cc