



**Ardublock**





# Module **1** : LEDs



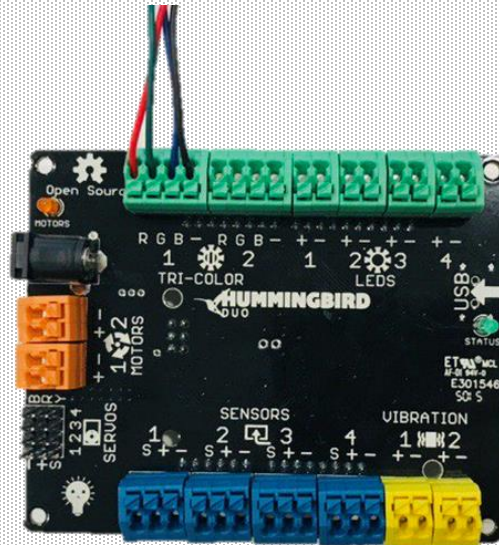
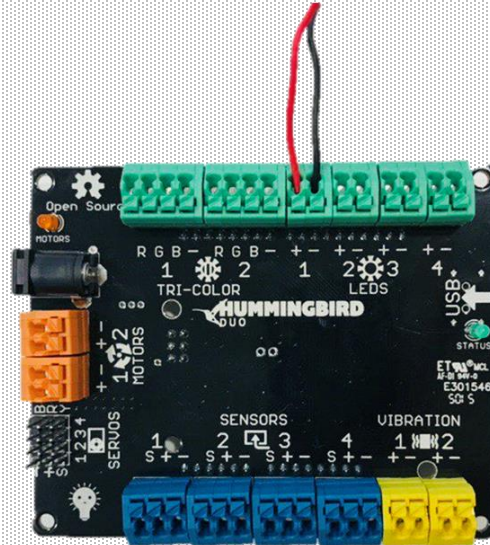
# HUMMINGBIRD 단색 LED , 삼색 LED



**단색 LED**

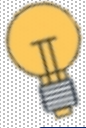


**삼색 LED**

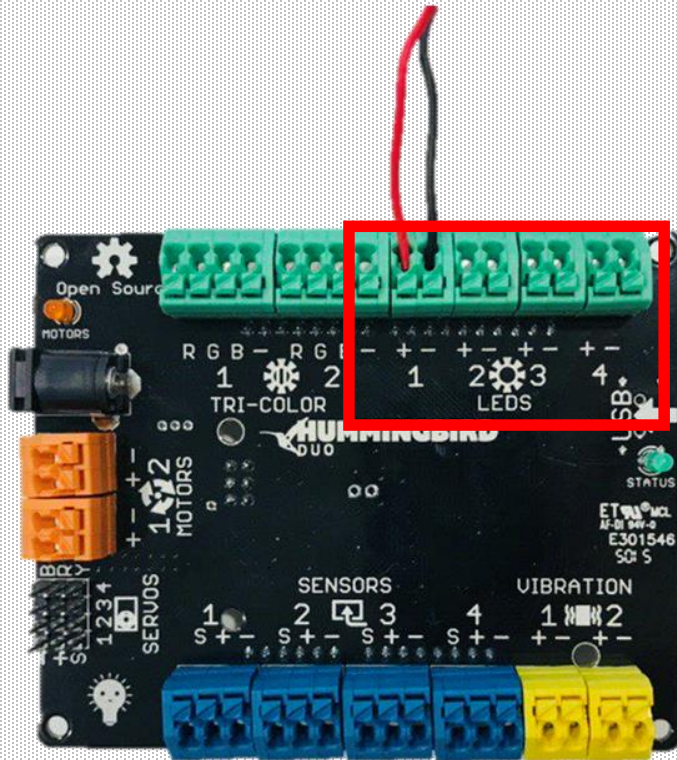


허밍버드 키트에는 두 종류의 LED(Light Emittted Diode)가 있습니다.  
두 개의 선으로 이루어진 **단색LED**와  
네 개의 선으로 이루어진 **삼색LED**

# 단색 LED



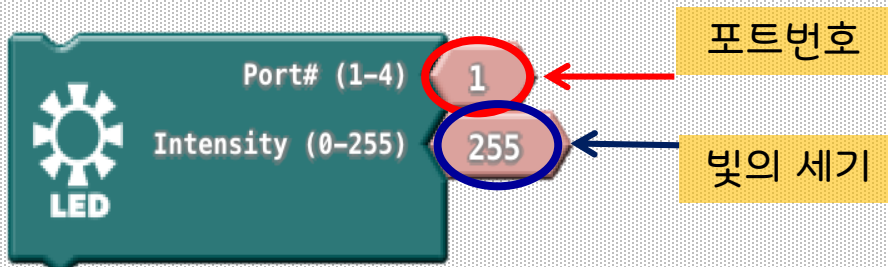
## 단색 LED



단색LED는 'LEDS'부분 1~4포트에 4개를 연결 할 수 있습니다.  
검은색 선은 '-' 터미널에, 색이 있는 선은 '+' 터미널에 연결합니다.  
선의 색은 LED 빛의 색과 동일합니다.

현재 단색LED는 포트1번에 연결되어있습니다.

# Ardublock 블록 설명

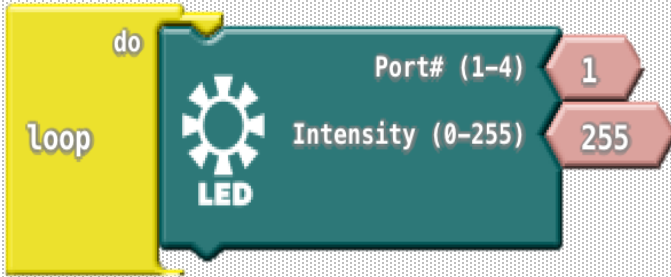


Hummingbird Duo 메뉴의 첫 번째 블록은 LED 블록입니다. LED 블록은 단색LED를 제어하는데 사용합니다.

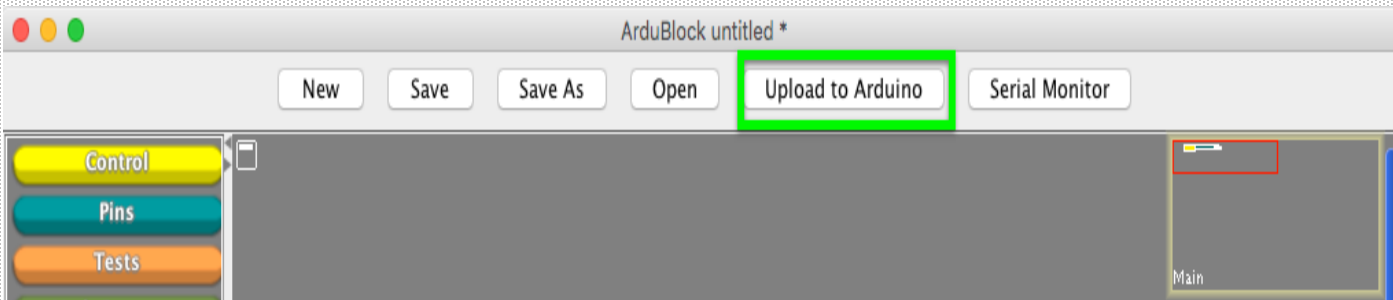
이 블록에는 두 가지를 입력해야 하는데, 첫 번째 동그라미에는 허밍버드 컨트롤러 'LEDS' 부분에 적혀있는 포트 번호(1~4)를 입력합니다. 다음에는 LED 밝기의 세기(0~255)를 입력합니다. 0은 LED 꺼짐을 의미하고, 255은 LED 최대 밝음을 의미합니다. 단색 LED를 LEDS 포트1에 연결합니다. LED블록을 스크립트 영역으로 드래그해 가져옵니다.



# Ardublock x 허밍버드 키트



Loop블록은 Setup 함수가 main() 함수입니다. LED블록을 Loop블록 안에 넣어주세요.



다음으로, 아두이노에 업로드 버튼을 클릭해주세요!

```
LEDtest | A
#include <Hummingbird.h>

Hummingbird bird;

void setup()
{
  bird.init();
}

void loop()
{
  bird.setLED(1,255);
}
```

그럼, 아두이노 창의 텍스트가 생기는 것을 확인할 수 있습니다. **아두블록 프로그램이 아두이노 명령어로 변환되었습니다!** 예를 들어 LED블록 'setLED()' 라는 함수로 변환 되었습니다. 아두블록 프로그램이 변환되면 새로운 아두이노 프로그램을 저장하라는 메시지가 표시됩니다. **잊어버리지 말고 저장하세요!**



# Ardublock x 허밍버드 키트



```
LEDtest
#include <Hummingbird.h>

Hummingbird bird;

void setup()
{
  bird.init();
}

void loop()
{
  bird.setLED(1,255);
}

Done uploading.
SKETCH USES 5102 BYTES (17%)
```

아두이노 창에서 **업로드 버튼을 클릭**하세요. 아두이노 창의 하단에서 업로드 상태를 확인해보세요. 프로그램 업로드가 끝나면 **“업로드완료”** 라고 표시되어야 합니다.

포트 1번의 단색 LED가 켜지는 것을 확인해보세요.

## Exercise 1.1

블록 속의 숫자를 255에서 0으로 바꾼 뒤 블록을 눌러 실행시켜봅시다. LED 빛이 꺼지는 것을 볼 수 있습니다. 블록 속의 숫자 0을 255으로 바꾼 뒤 블록을 눌러 실행시켜봅시다. LED 빛이 가장 밝게 켜지는 것을 볼 수 있습니다. 블록 속에 0 ~ 255 사이의 숫자를 자유롭게 넣어봅시다. 다양한 밝기의 빛을 볼 수 있습니다.

# Ardublock x 허밍버드 키트



아두블럭 에서 LED를 깜박이려면 컨트롤 블록에 LED 블록과 기다리기 블록이 있어야 합니다. 두 기다리기 블록은 일정 시간 동안 프로그램을 일시 중지합니다. MILLIS 기다리기 블록을 사용하여 프로그램을 수 밀리 초 동안 일시 중지해 봅시다.



LED 블록과 MILLIS 기다리기 블록을 추가하여 LED를 0.5 초 동안 반복해서 켜 다음 0.5 초 동안 LED를 끕니다. 위와 같은 코드를 프로그램을 업로드 하여 작동시켜 봅시다.



# Ardublock x 허밍버드 키트



프로그램을 업로드 한 후에는 USB 케이블을 뽑은 경우에도 LED가 계속 깜박이는 것을 확인 할 수 있습니다. **USB 케이블을 뽑아도 프로그램이 실행되는 이유는 허밍버드 컨트롤러의 아두이노 모드에서 실행되기 때문입니다!** USB 케이블을 뽑을 경우에 허밍버드 컨트롤러에 추가 전원 공급이 필요합니다. 추가 보조배터리를 사용해도 좋습니다. 컨트롤러에 전원이 공급되는 한 새 프로그램을 업로드 할 때까지 프로그램이 계속 실행됩니다.

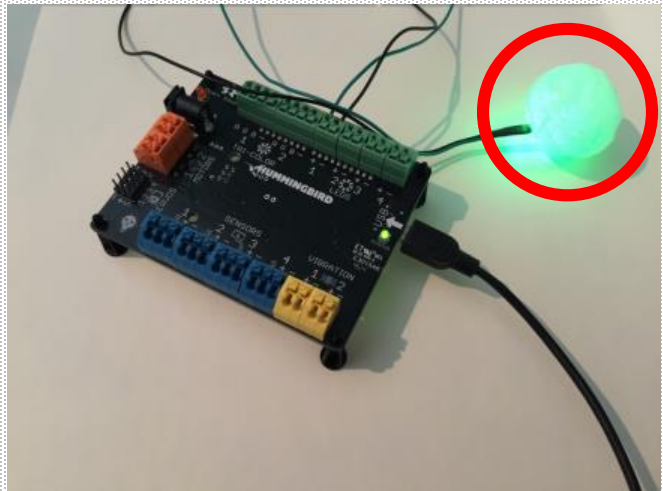
## Exercise 1.2

두 개의 LED를 허밍버드 컨트롤러에 연결해 주세요, 첫 번째 LED가 꺼져있을 때 두 번째 LED 켜지도록 프로그램을 작성해 보세요. 새 프로그램을 업로드 하려면 USB 케이블을 다시 허밍버드 컨트롤러와 컴퓨터에 연결해야 합니다.



### 메이킹 TIP

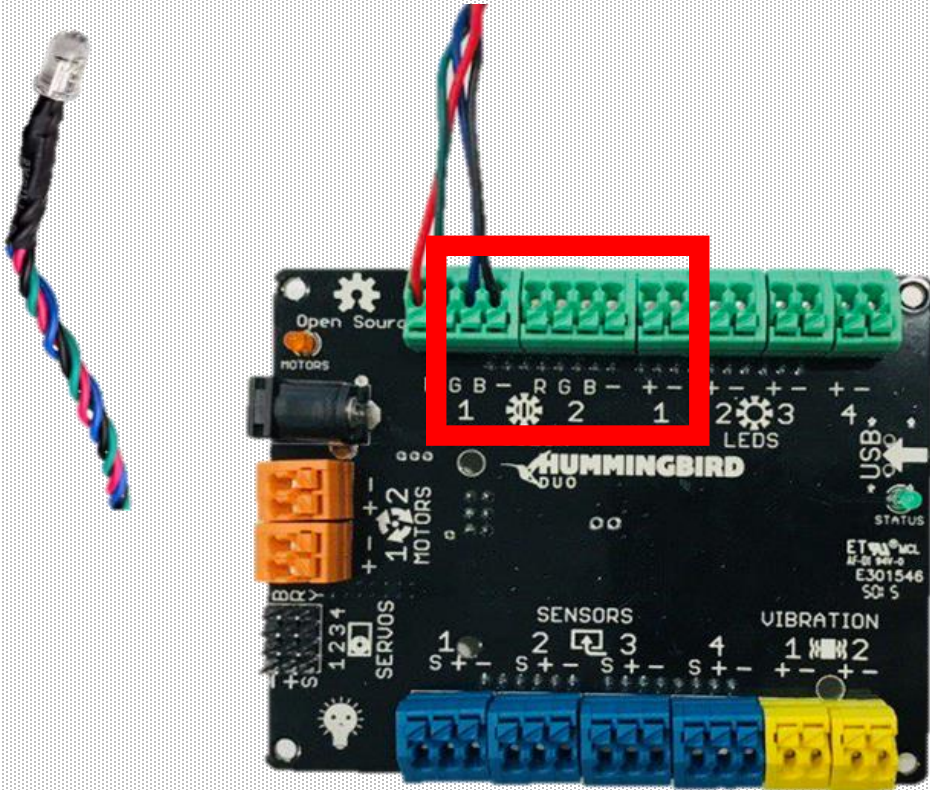
LED 빛은 작게 나타나지만, 스티로폼 공을 이용하면 빛이 퍼지는 것을 볼 수 있습니다.



# 삼색 LED



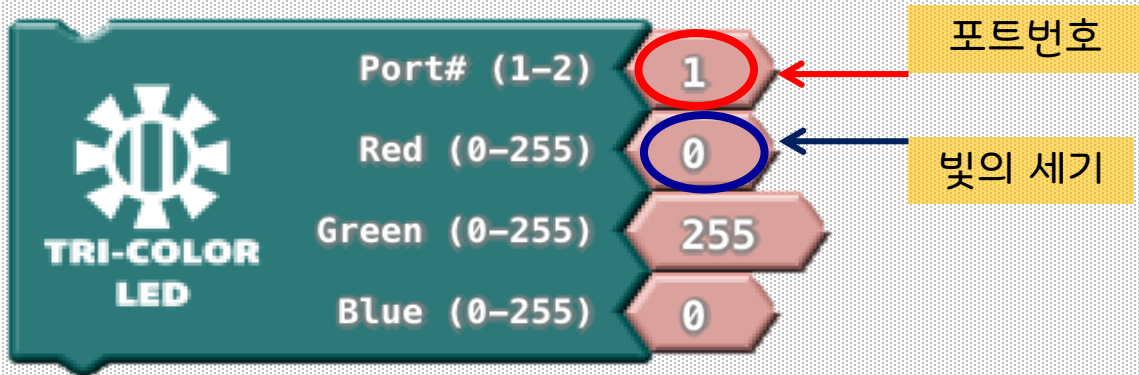
## 삼색 LED



삼색LED는 'TRI-COLOR'부분 1~2포트에 2개를 연결 할 수 있습니다.  
검은색 선은 '-'터미널에, 빨간색 선은 'R'터미널, 초록색 선은 'G'터미널,  
파란색 선은 'B'터미널에 연결합니다.

현재 삼색LED는 포트1번에 연결되어 있습니다.

# Ardublock x 허밍버드 키트



삼색LED 안에는 빨간색, 초록색, 파란색의 세 가지 작은 빛이 들어있습니다. 삼색LED를 제어할 때

**Hummingbird TRI-LED** 블록이 사용됩니다.

이 블록에는 4 가지를 입력해야 합니다.

가장 왼쪽 동그라미에는 TRI-COLOR가 연결되어있는 쪽의 허밍버드 컨트롤러에 적혀있는 포트 번호(1,2)를 입력합니다.

다음 동그라미에는 차례대로 R,G,B라고 적혀있습니다.

R은 빨간색 **빛의 양(0(없음)~255(최대 밝음))**을 조절합니다.  
G, B는 각각 초록색과 파란색 빛의 양(0~255)을 조절합니다.

현재 삼색LED는 TRI-COLOR 포트1에 연결되며 초록색 빛을 켜고 있습니다.

# Ardublock x 허밍버드 키트

## Exercise 1.3

아래 스크립트가 어떤 동작을 하나요? 결과를 상상해보고 따라 해보세요! 삼색 LED를 어떻게 끄면 될까요? 아두이노 코드와 아두블럭 코드는 어떤 차이가 있는지 눈으로 확인해 봅시다.



## Exercise 1.4

삼색 LED 가 최소 10번 이상 5초 동안 깜박이도록 프로그램을 만들어 봅시다. 삼색 LED는 동일한 시간 동안 켜지고 꺼져야 합니다. 동일한 시간 동안 꺼지는 구문은 아두이노 코드로 직접 변경해 봅시다.