

# 라두이노 미니 드론 교육 자료

준비물

- 파트 I. 드론의 이해 및 조정 실습
  - 드론의 이해
  - 드론의 종류
  - 드론의 구성요소
  - 드론 조정 실습(자이로짐)
- 파트 II. 드론 설계 및 드론 조립
  - 드론 3D 프레임 설계
  - 드론 조립
  - 드론 조정기 조립
- 파트 III. 드론에 코딩 하기
  - 개발환경 구축
  - 사운드 드론 만들기 (부저, RGB LED)
  - 무지개 드론 만들기 (Neopixel LED)
  - MIT 인벤터를 이용한 드론 조정 앱 만들기(BLE 모듈)
  - 고도 유지 드론 만들기 (적외선 센서)

# **PART 1. 드론의 이해 및 조정 실습**

# 드론의 이해

인스파이어

[https://www.youtube.com/watch?v=XQH\\_dVn9kCU](https://www.youtube.com/watch?v=XQH_dVn9kCU)

맵핑 및 슬램

<https://www.youtube.com/watch?v=IMSozUpFFkU>

비콘을 이용한 드론

<https://www.youtube.com/watch?v=MvRTALJp8DM>

마빅 관련 자료

[https://www.youtube.com/watch?v=p1d\\_ptE6yrc](https://www.youtube.com/watch?v=p1d_ptE6yrc)

사람이 타는 드론

<https://www.youtube.com/watch?v=hrO8ERbzof4>

수정

물속 드론

<https://www.youtube.com/watch?v=XGW5OOND9lc>

가변 피치 드론

<https://www.youtube.com/watch?v=TnGhEInTXYc&list=PLAE3vYvoMAETdEM1SBgmRCTc0cPIHEJTs>

내가 생각하는

드론이란?

멀티콥터란 ?

UAV(Unmanned aerial Vehicle)란?

# 드론이란?

- 무인 비행체
- 사전적의미로 벌 등이 웅웅 거리는 소리 or 낮게 웅웅거리는 소리
- 기체에 사람이 타지 않고 지상에서 원격 조종한다는 점에서 무인항공기(UAV)라는 표현으로도 쓰임
- 헬리콥터형과 기체형으로 나누어짐
- 손바닥만한 크기의 나노급 드론에서부터 카메라 장착된 촬영용 드론, 15m 에 달하는 군사 목적용 기체형 드론
- 3D 프린터로 제작된 드론

드론은 어디에 무슨 용도로 사용 될까?

# 드론 어디에 사용 되는가?



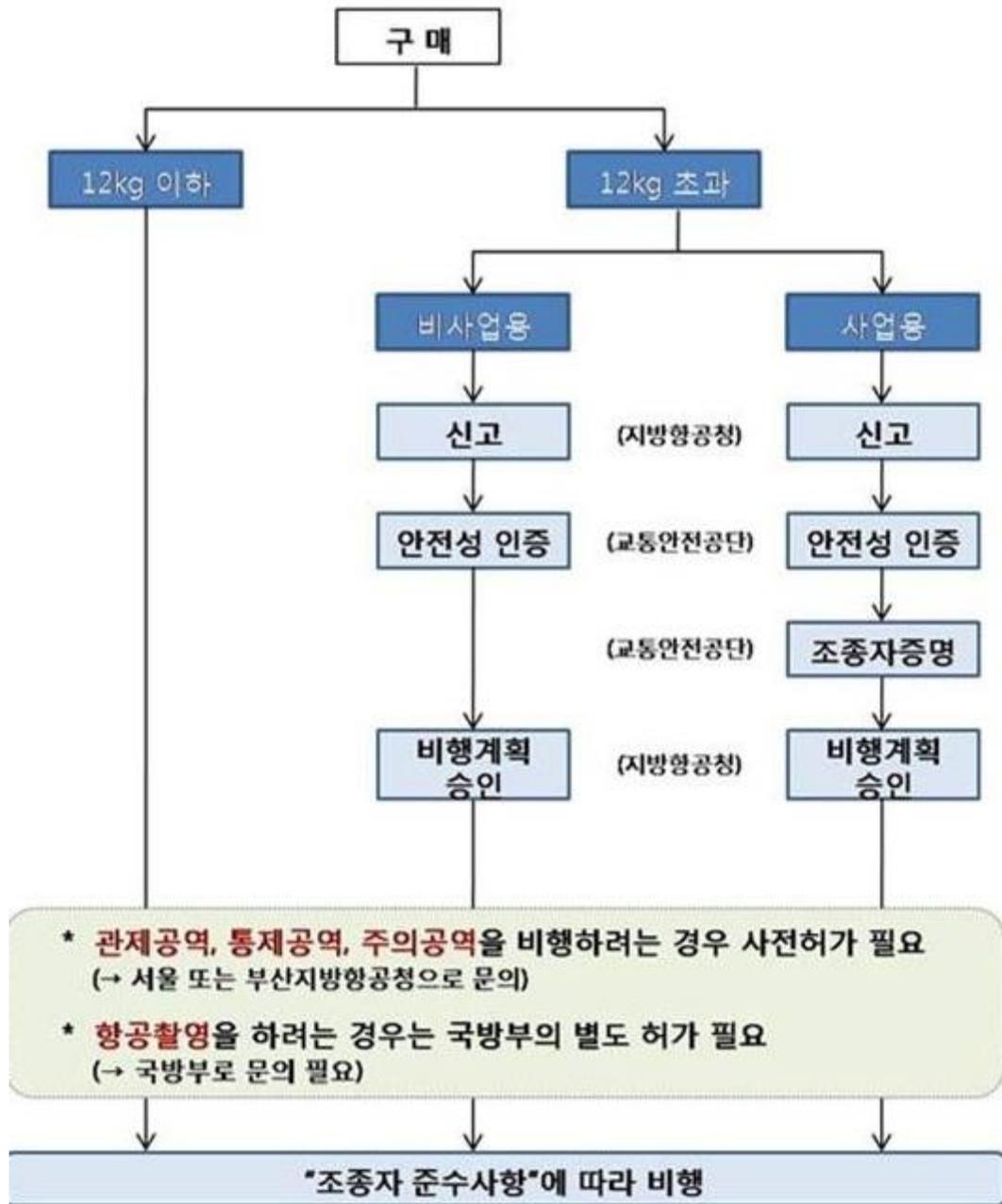
좌)나노급 드론 / 중)물류, 촬영용 드론 / 우)군사용 드론

드론의 비행에 따로 필요한 것은?

규정, 규칙, 등? (생각나는대로)

# 규정

수정



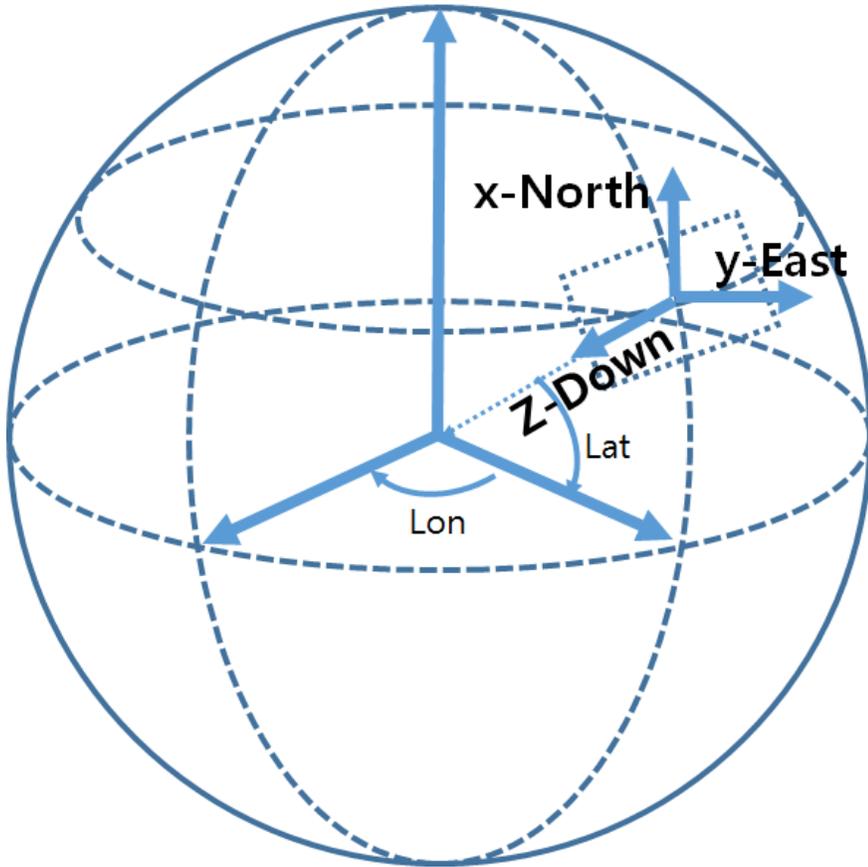
일반 목적의 드론은 12KG미만

# 규정 수정

주요 법규 위반 행위		처벌 처분기준
조종자 준수 사항 위반	비행금지구역을 허가 없이 침범	200만원 이하 과태료
	야간비행	
	사람이 많이 모인 곳 상공에서 비행	
사업등록을 하지 않고 무인비행장치를 영리 목적으로 사용한 경우		1년 이하 징역 또는 3천만원 이하 벌금

자료: 국토교통부, 무인비행장치(드론, 이것만 지키면 모두가 안전해요. 2015.5.27

# 드론의 종류

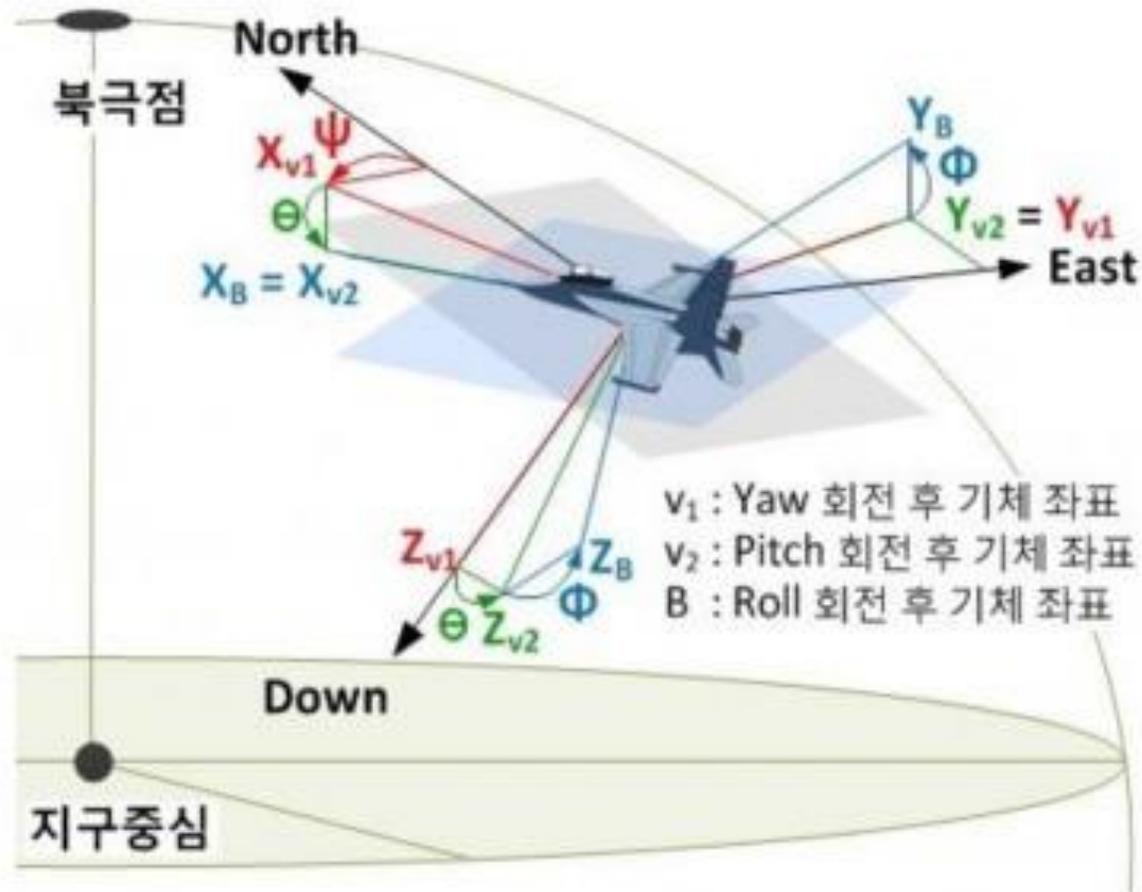


## NED (North-East-Down)

드론의 조정은

**롤(Roll) 피치(Pitch) 요(Yaw) 추력(Thrust)**

의 네 가지 파라미터를 이용한다



# 멀티콥터 이해하기

## 멀티콥터의 종류 구별하기



고정익



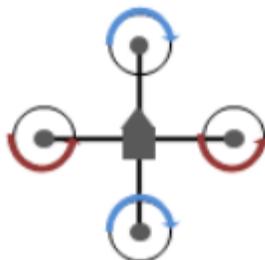
회전익

# 멀티콥터의 종류

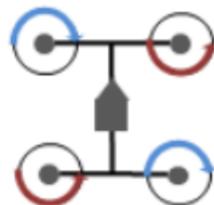
X4 or X  
Quadcopter



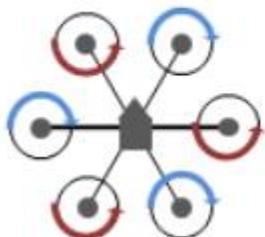
I4 or +  
Quadcopter



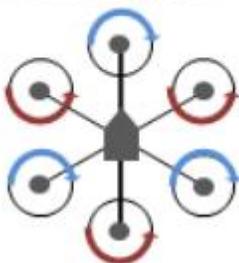
H4 or H  
Quadcopter



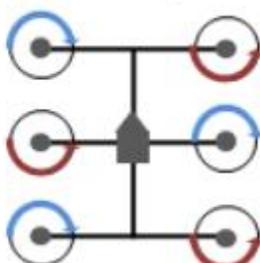
X6 or X  
Hexacopter



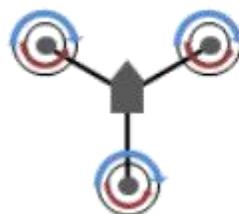
I6 or I  
Hexacopter



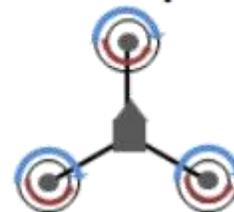
H6 or H  
Hexacopter



Y6 or Y  
Hexacopter



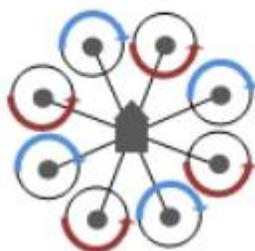
IY6 or IY  
Hexacopter



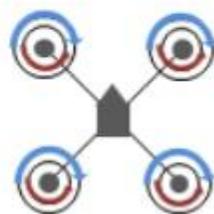
I8 or +  
Octocopter



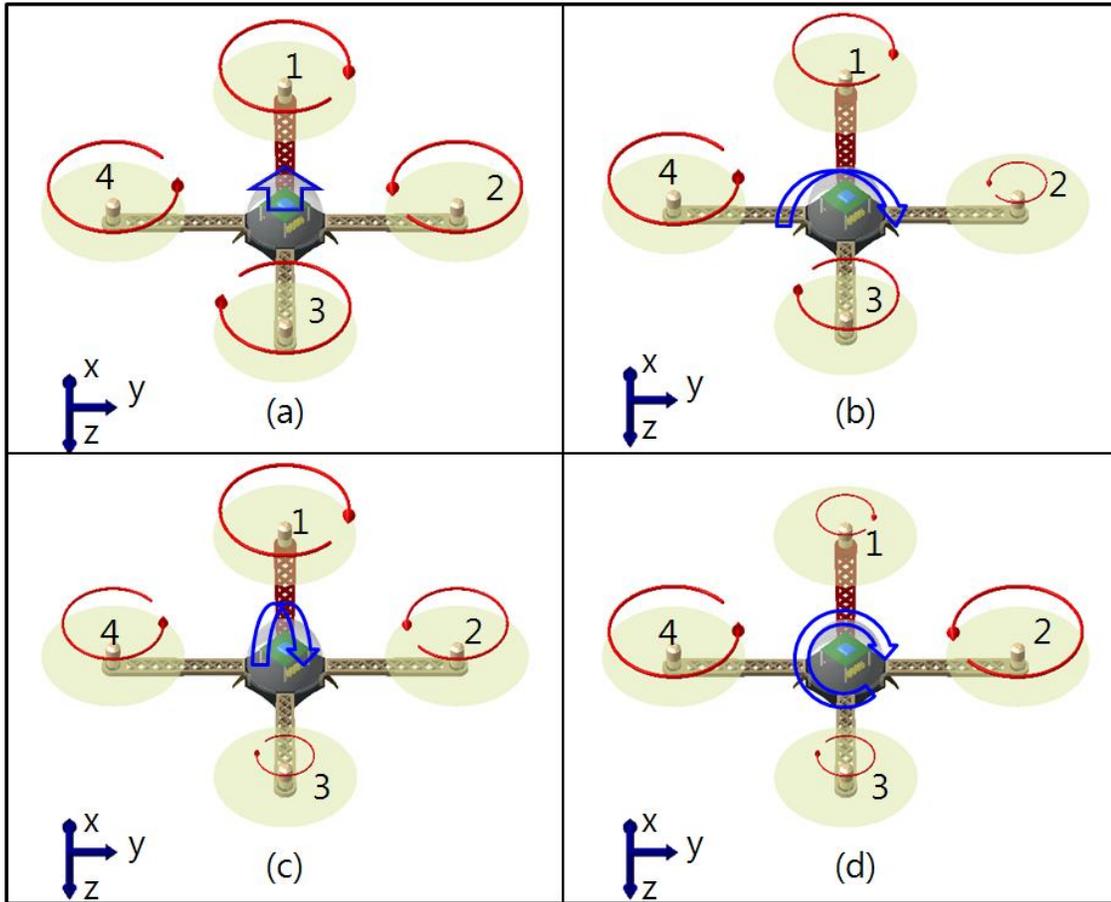
V8 or V  
Octocopter



X8 or X  
Octocopter



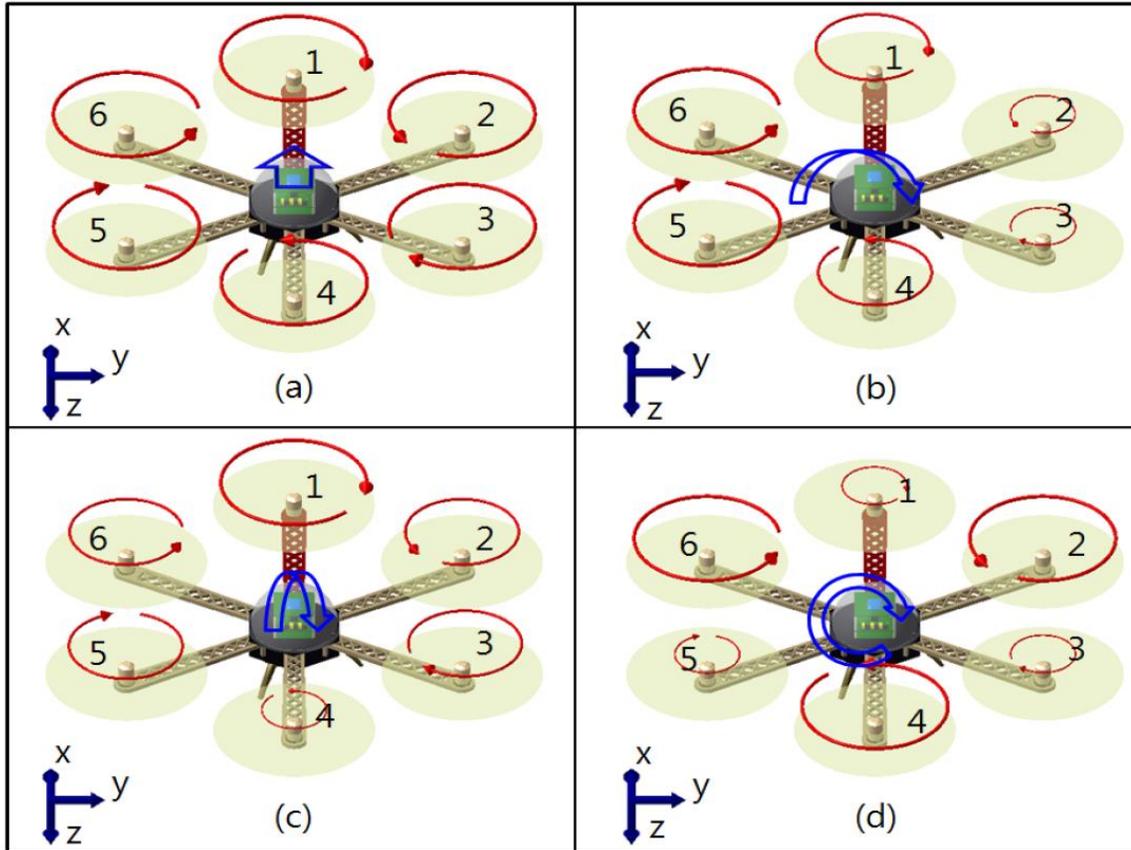
# 멀티콥터의 구동방식



운동	A	B	C	D
Hovering				
Rotor 1	H	M	H	L
Rotor 2	H	L	M	H
Rotor 3	H	M	L	L
Rotor 4	H	H	M	H

쿼드콥터 (크로스 [+])

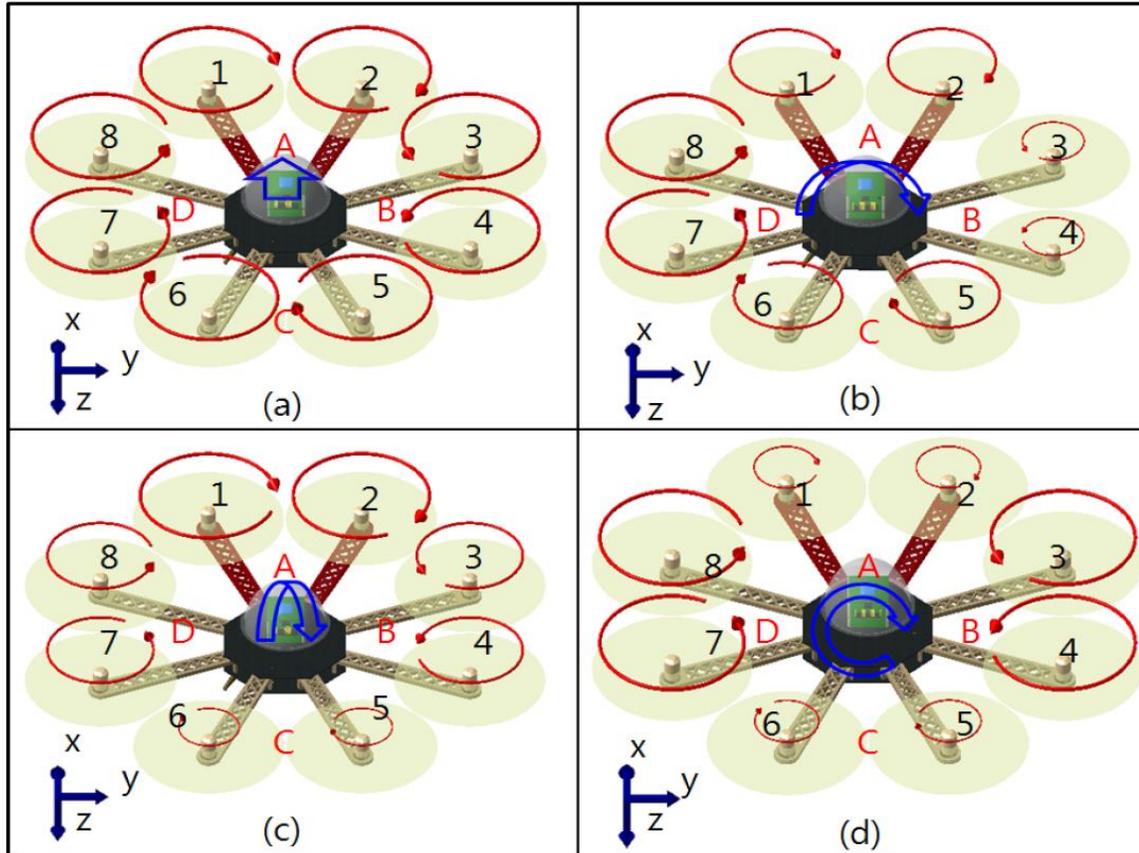
# 멀티콥터의 구동방식



운동	A	B	C	D
	Hovering	Roll	Pitch	Yaw
Rotor 1	H	M	H	L
Rotor 2	H	L	M	H
Rotor 3	H	L	M	L
Rotor 4	H	M	L	H
Rotor 5	H	H	M	L
Rotor 6	H	H	M	H

헥사콥터 (I타입)

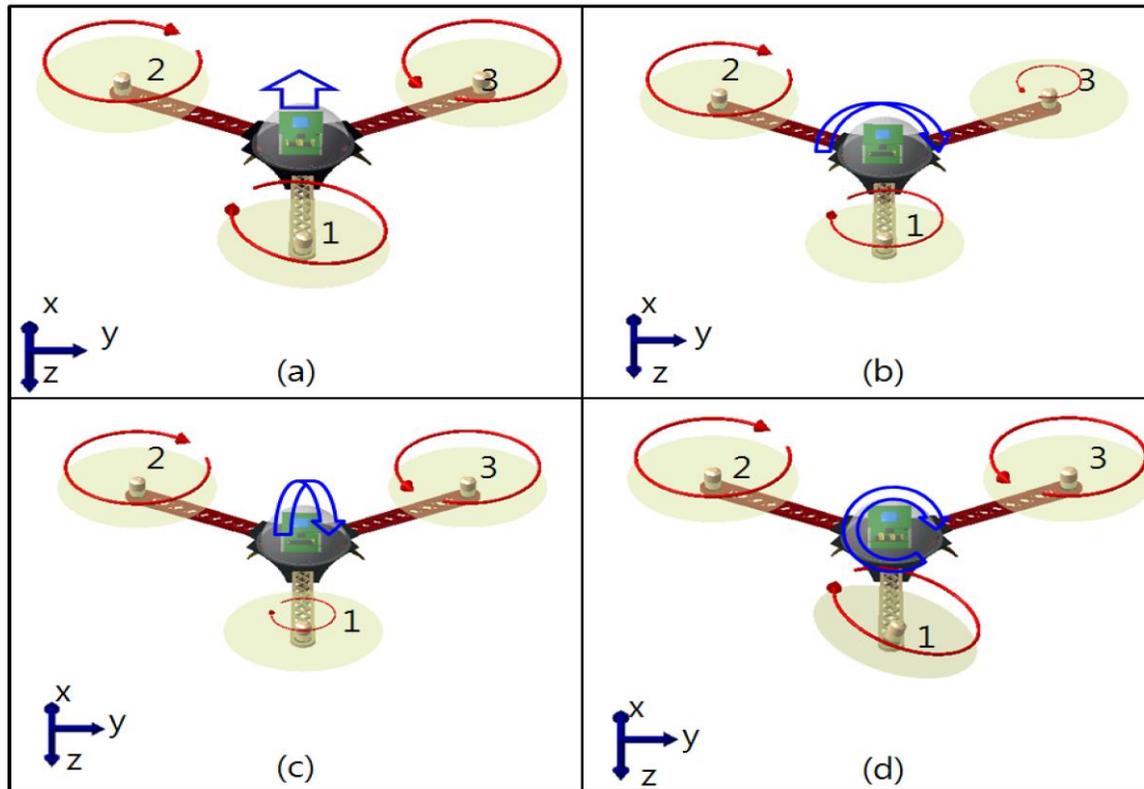
# 멀티콥터의 구동방식



운동	Hovering	Roll	Pitch	Yaw
Rotor1	H	M	H	L
Rotor2	H	M	H	L
Rotor3	H	L	M	H
Rotor4	H	L	M	H
Rotor5	H	M	L	L
Rotor6	H	M	L	L
Rotor7	H	H	M	H
Rotor8	H	H	M	H

옥토크터 (V타입)

# 멀티콥터의 구동방식

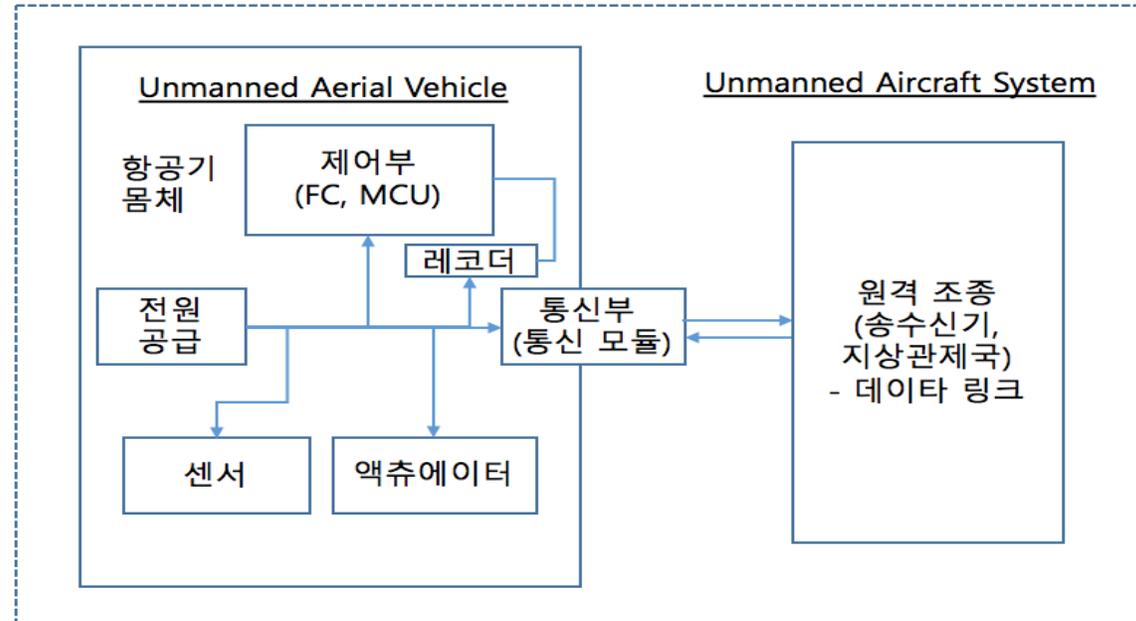


운동	Hovering	Roll	Pitch	Yaw
Rotor 1	H	M	L	L
Rotor 2	H	H	H	L
Rotor 3	H	L	H	H

트라이콥터 (Y타입)

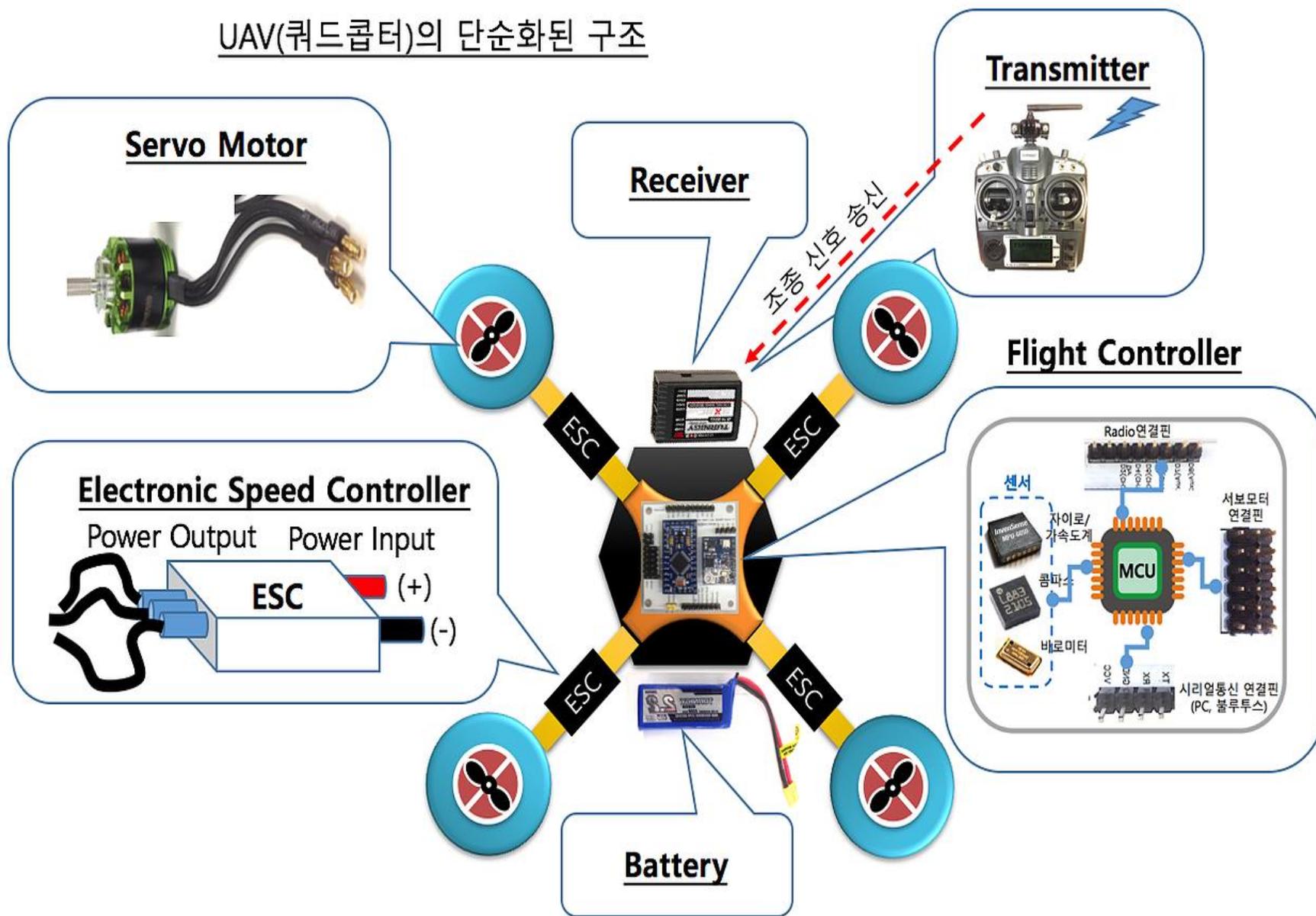
# 드론의 구성요소

# 드론의 구성요소



- 제어부
- 센서부
- 액츄에이터
- 통신부
- 전원부

# UAV(쿼드콥터)의 단순화된 구조



# 비행 기체, 프레임(Frame)



멀티콥터의 형태를 결정하는 중요한 요소이다.



# 변속기(ESC) 와 모터



ESC(전자 변속기)는 드론에 사용되는 모터의 회전 속도 조절하기 위한 부품  
ESC(전자 변속기)와 모터는 드론의 크기와 용도에 맞추어 신중히 선택

# 프로펠러



[그림] 슬로우 타입 프로펠러

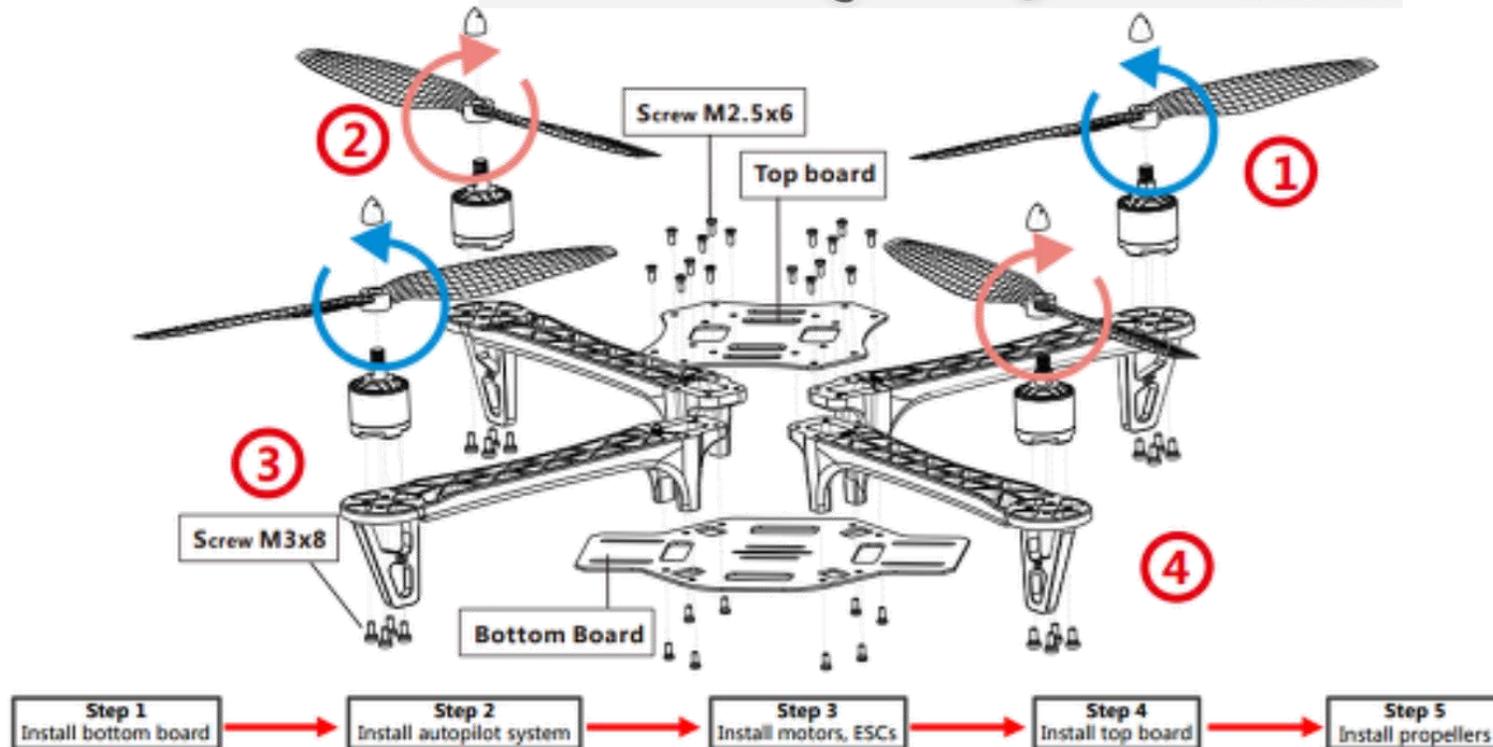
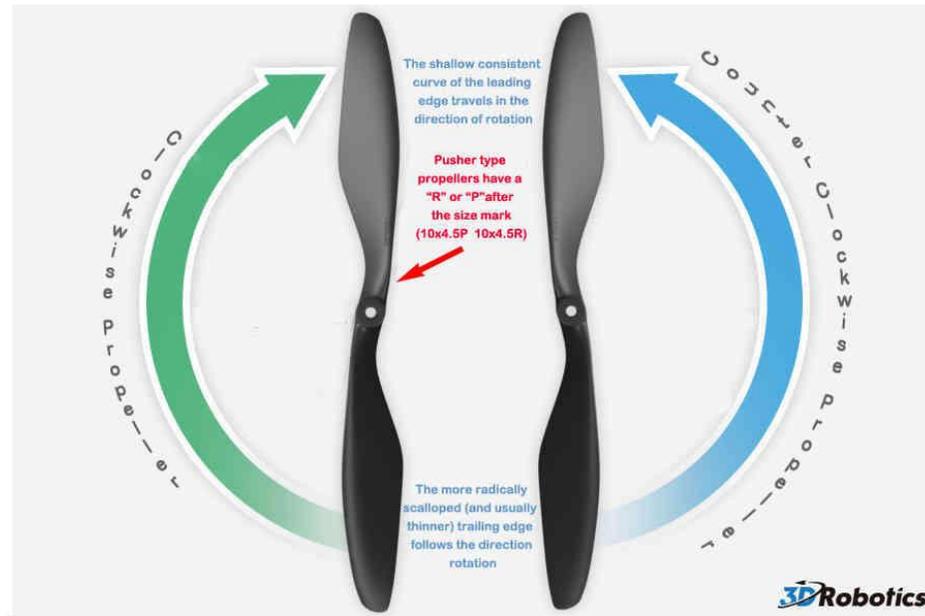


[그림] 나무재질



[그림] APC 제품

# 프로펠러의 회전 방향



# 배터리

[그림] 3셀 11.1V 1300mAh 20C 리포 배터리



3S :  $3.7V \times 3 = 11.1V$

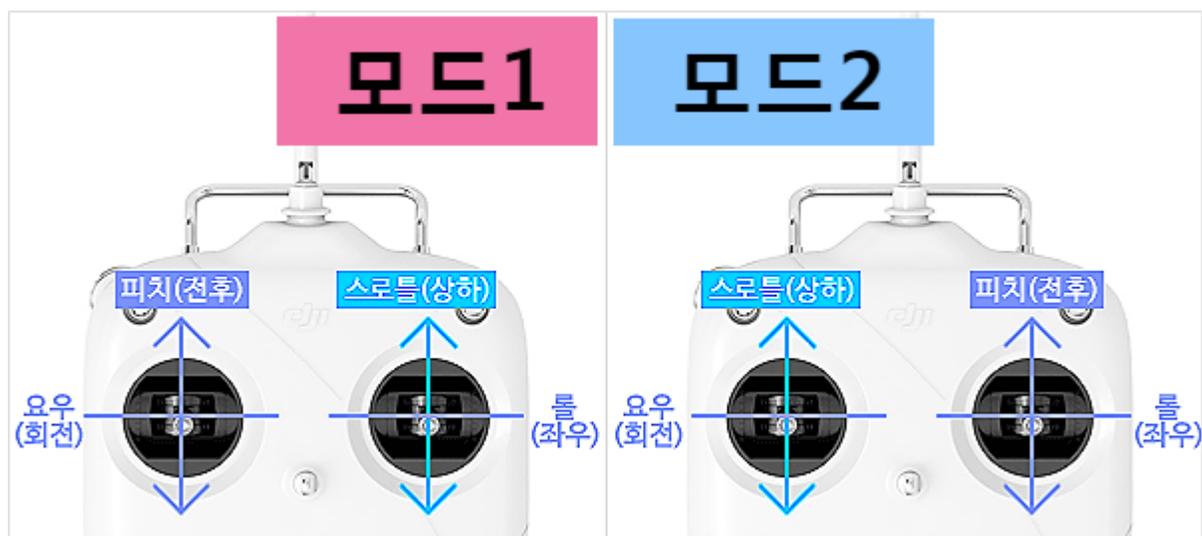
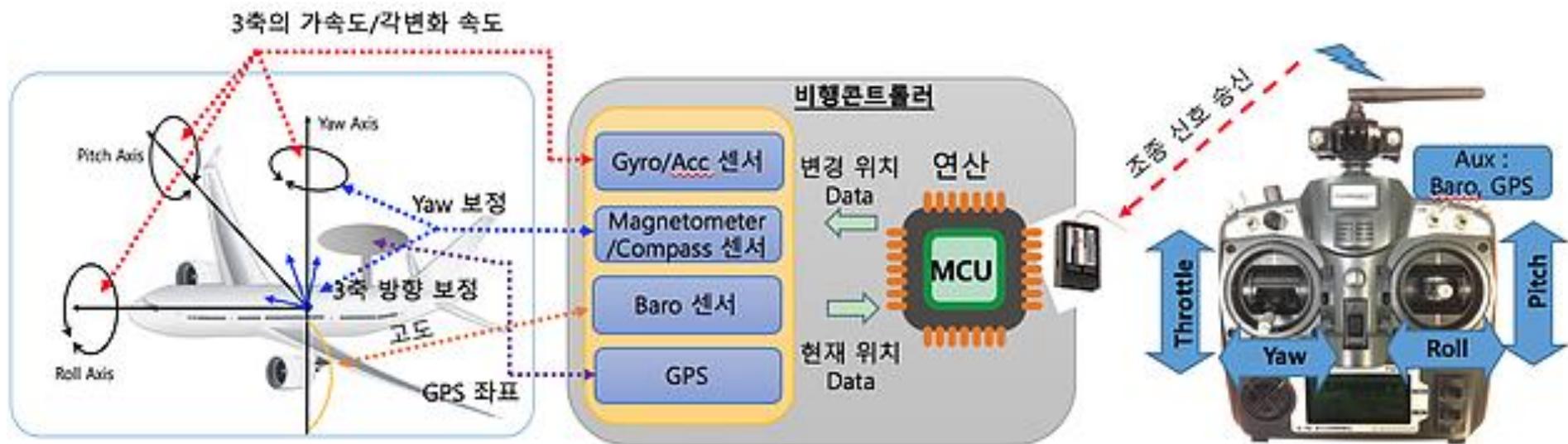
1300 mAh: 1시간에 1.3A 전류를 지속적으로 방출할 수 있는 용량

20C: 방전율  $20C \times 1.3A = 26A$

# 조종기



## 조종기 실습(자이로짐)



드론 교육  
드론을 실내외 어디서나  
실험 및 훈련이 가능합니다.

비행시간 ∞  
장시간 비행할 가정용 전기를  
공급받아 수 있습니다.

확장성  
다양한 타입의 드론을  
탑재하고 비행해보세요

데이터 모니터링  
아두이노, 프로세싱 환경에서  
자신의 비행 데이터를 시각적 확인하세요.

웹사이트  
아두이노, 프로세싱 환경에서  
자신의 비행 데이터를 시각적 확인하세요.

전원공급  
최대 1kW까지 안정적인  
전원공급과 선 고정방지



# 자이로짐

<https://youtu.be/8ZINAvnlv-g>

## D. GyroGym

드론 비행실험 및 훈련장치

- 드론의 역학 이해와 실습을 위한 장비
- 드론의 비행을 준비하고 실험 및 실습을 위한 장비
- 220V AC 전원을 활용한 지속적인 실내외 비행실습
- 드론의 자세값을 블루투스 통신으로 받아 모니터링 가능

## **PART II. 드론 설계 및 조립**

## 드론 3D 프레임 설계

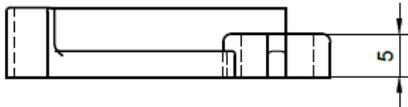
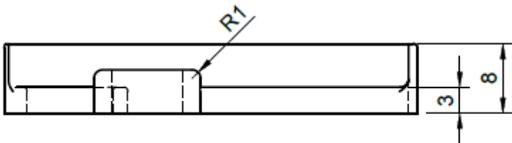
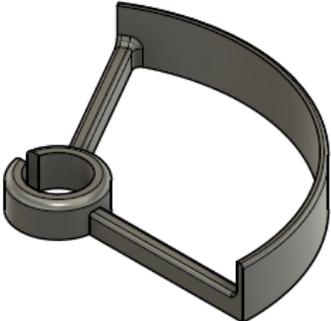
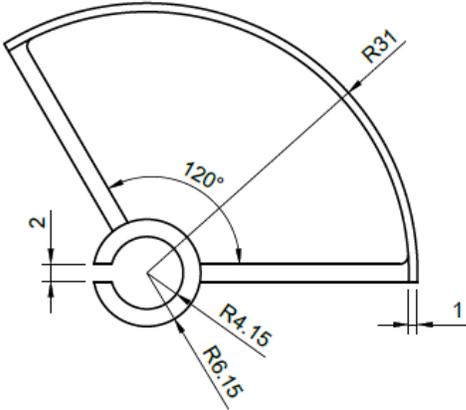
- 준비물 : Window 환경의 PC
- 프로그램 : Fusion 360

# 3D 프레임 설계

<https://youtu.be/5IY5TJnriDk>

# 연습문제

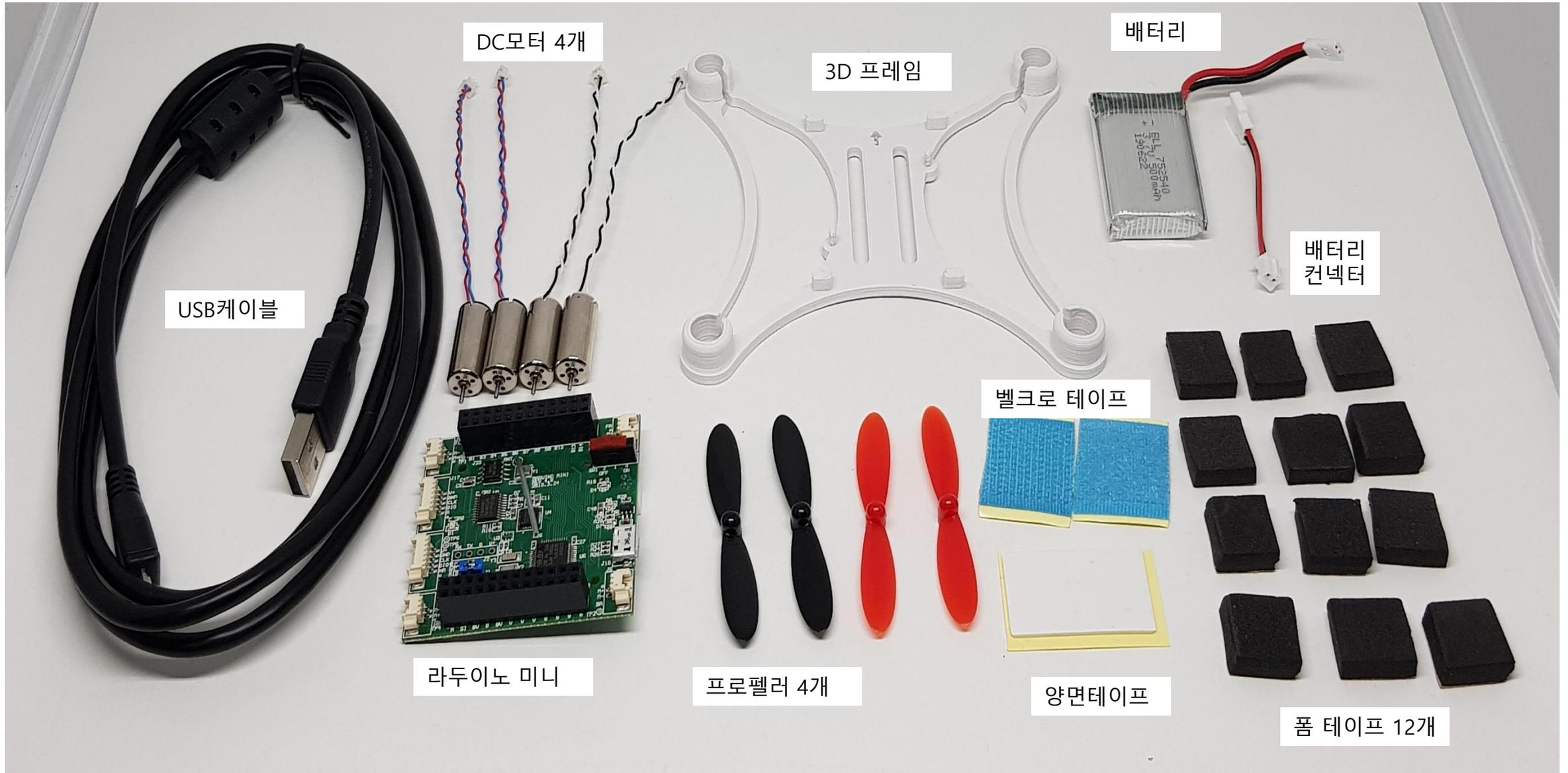
-프로펠러 가드 만들기



## 드론 조립

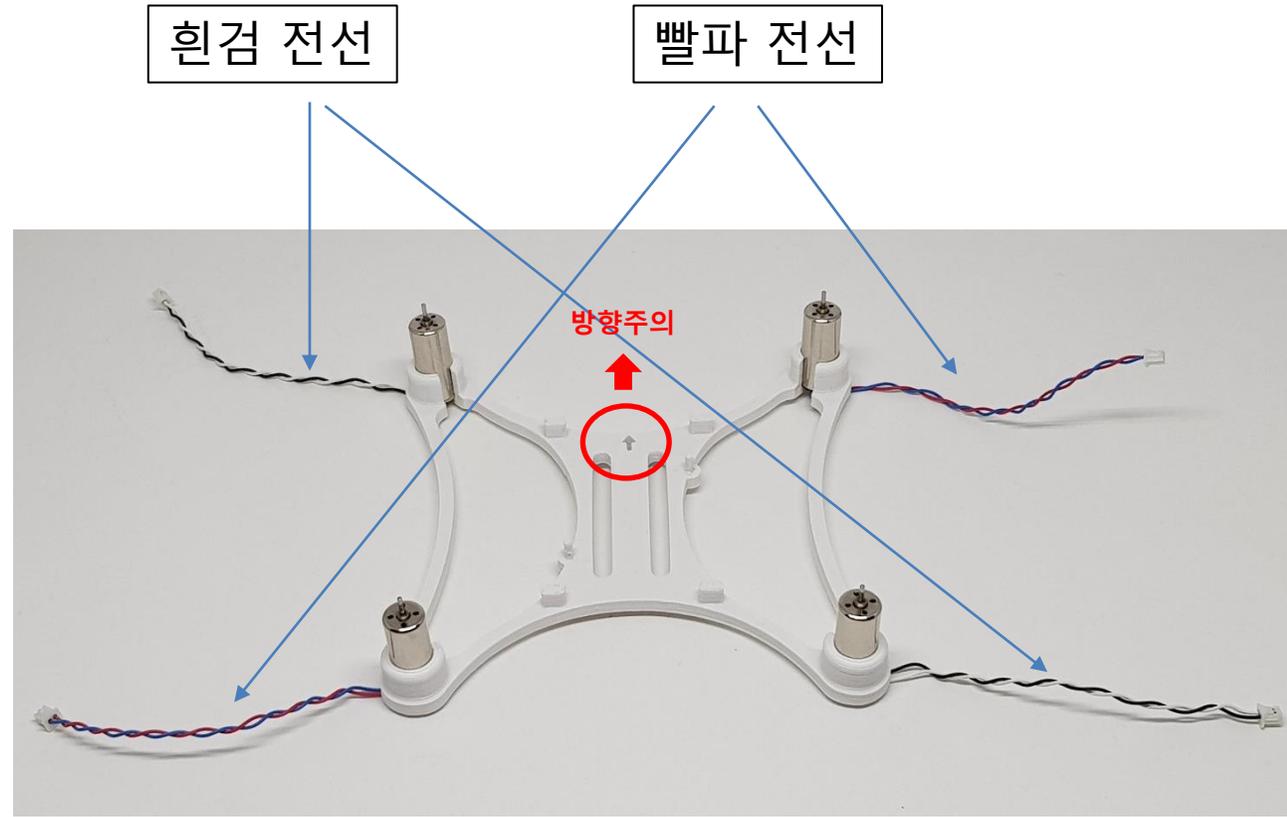
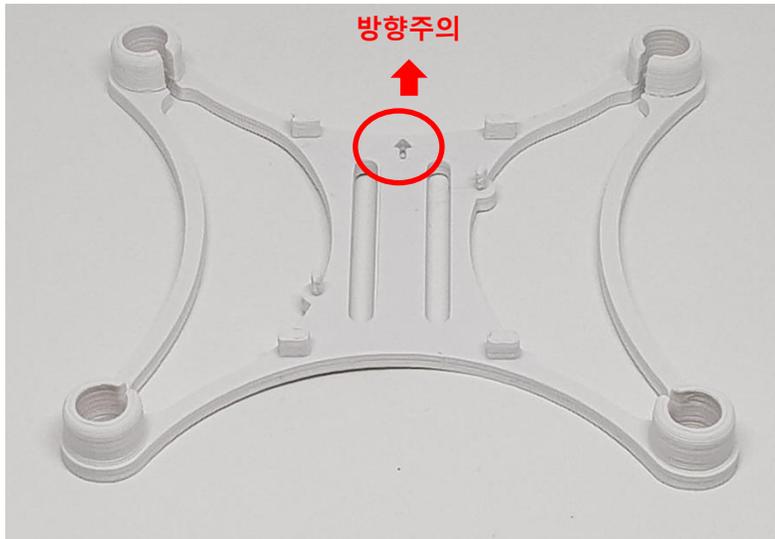
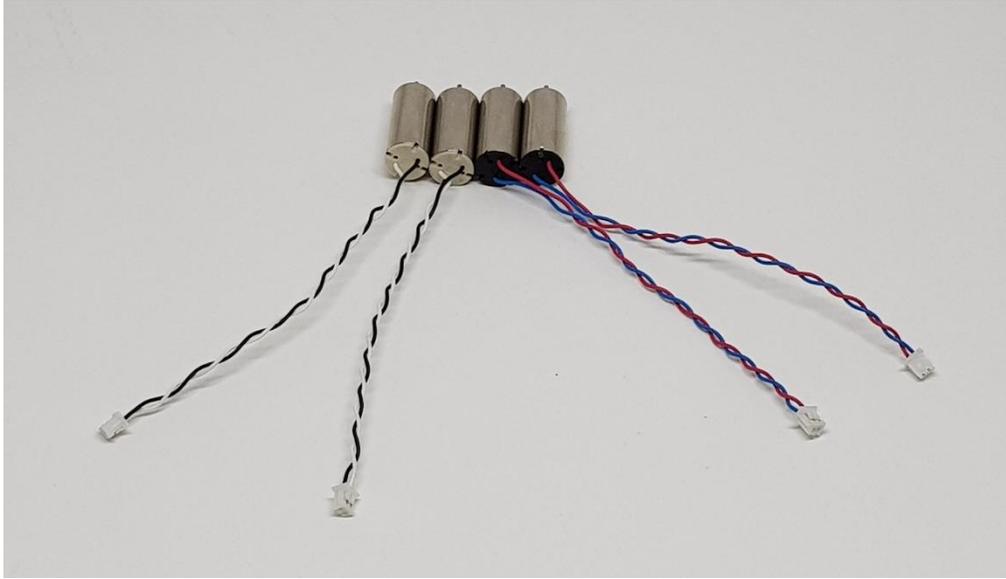
- 준비물 : 라두이노 미니 드론

# 라두이노 미니 드론 구성품



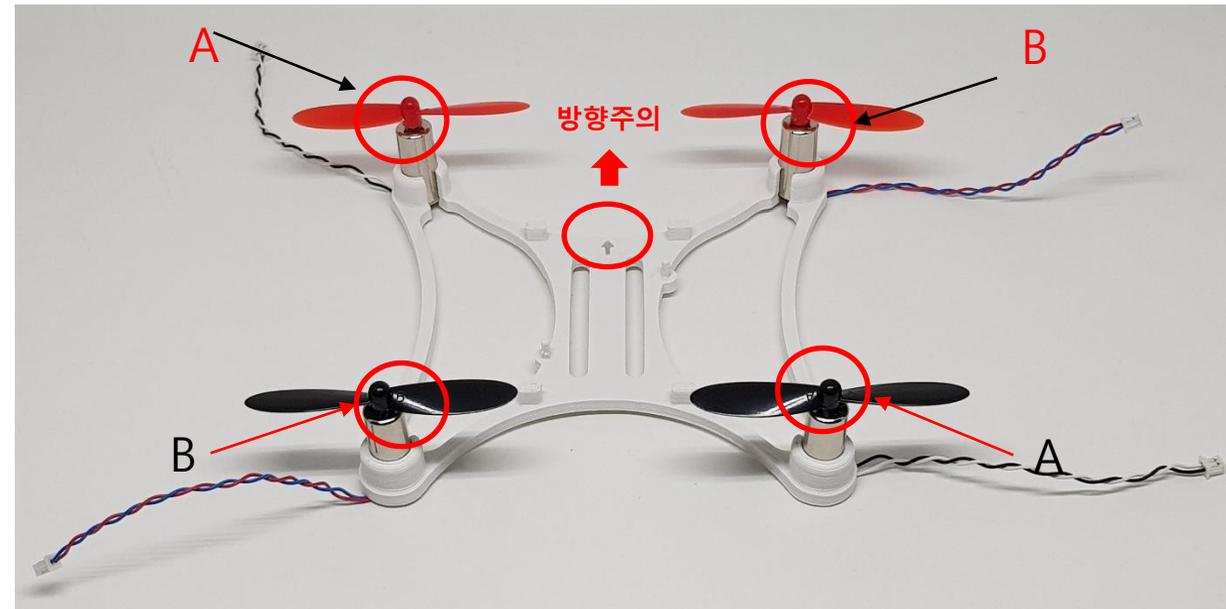
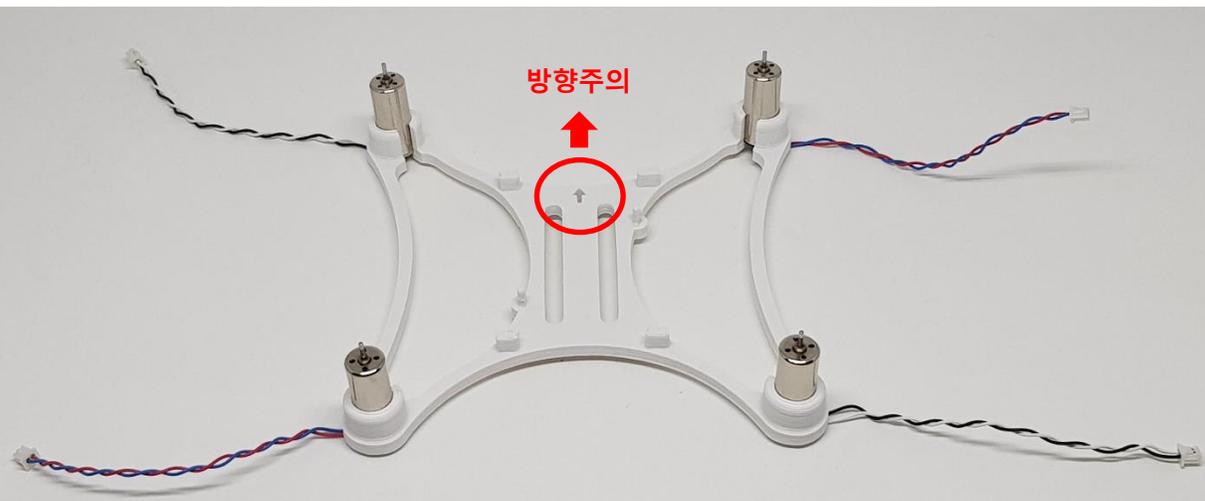
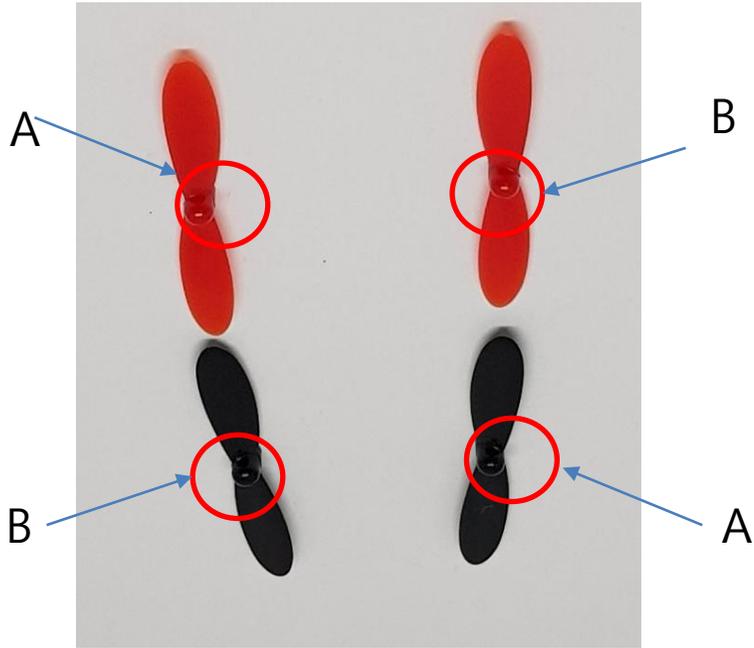
# DC모터 조립

DC모터에 달린 전선 색깔에 맞춰서 조립



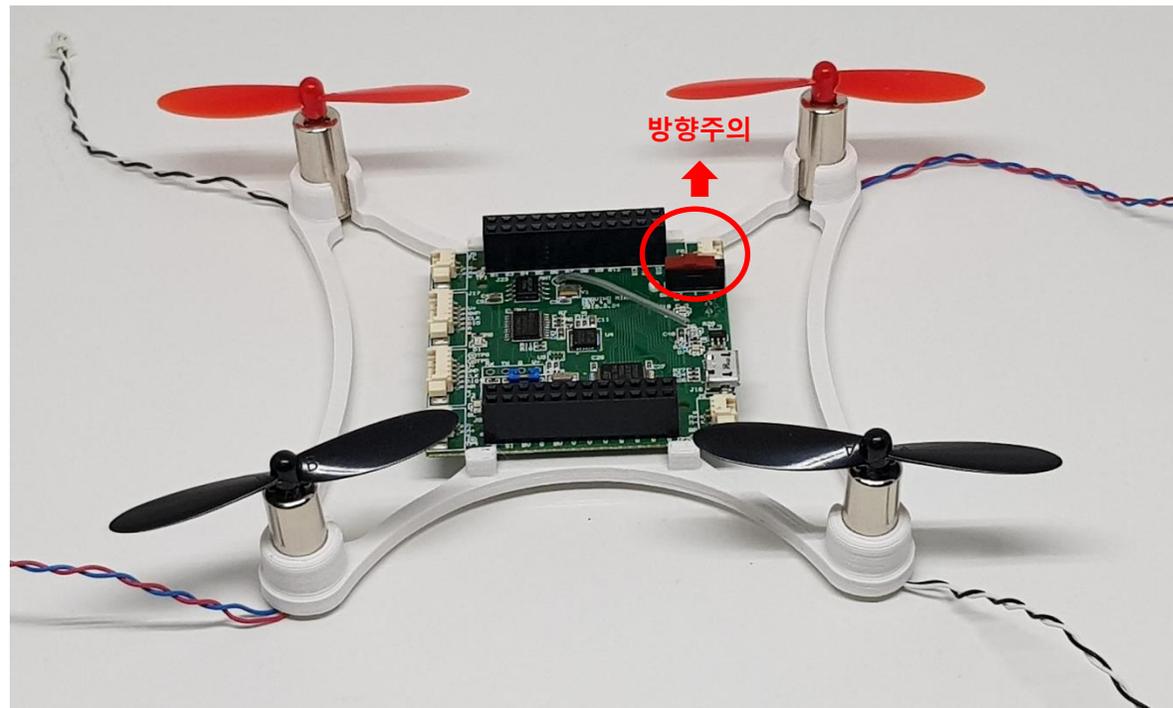
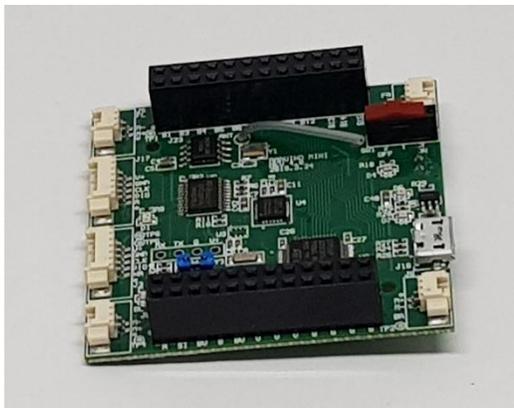
# 프로펠러 조립

프로펠러에 색깔과 기호에 맞춰 그림과 같이 조립



# 라두이노 미니 조립

양면테이프 두개를 라두이노 미니 뒷편에 접착후 방향에 맞춰 조립

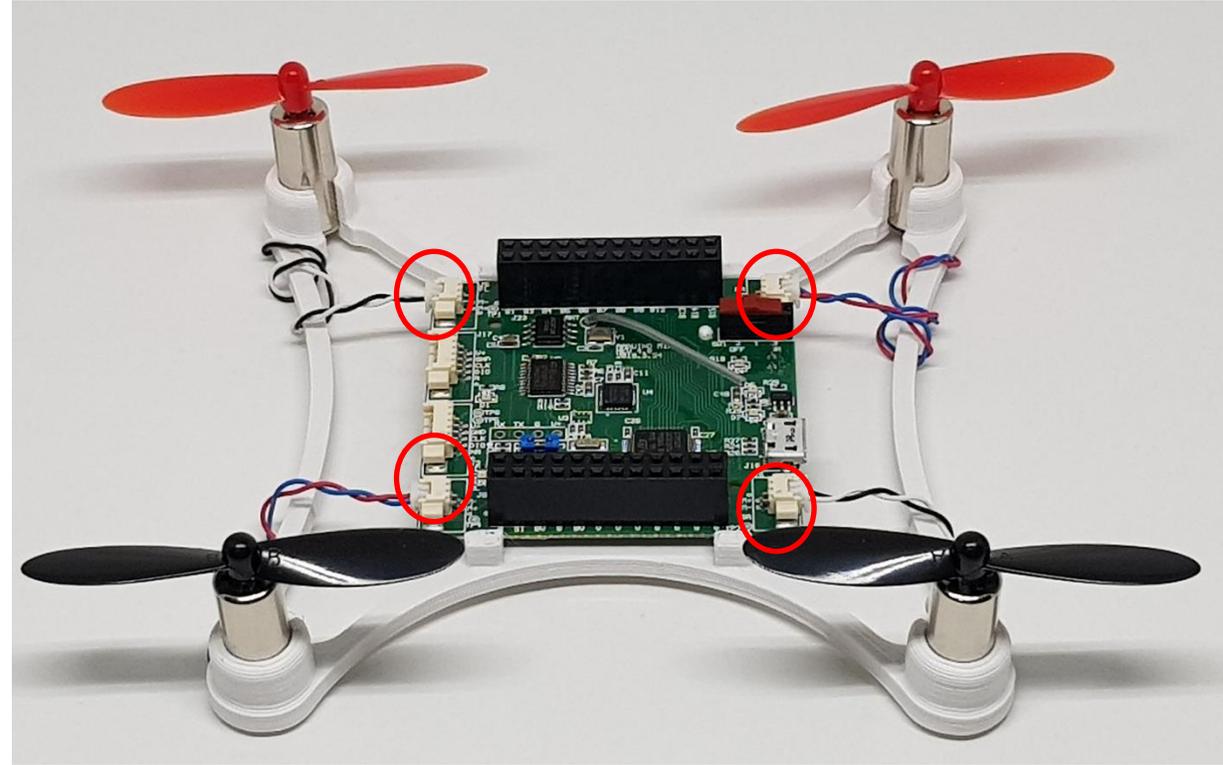
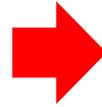
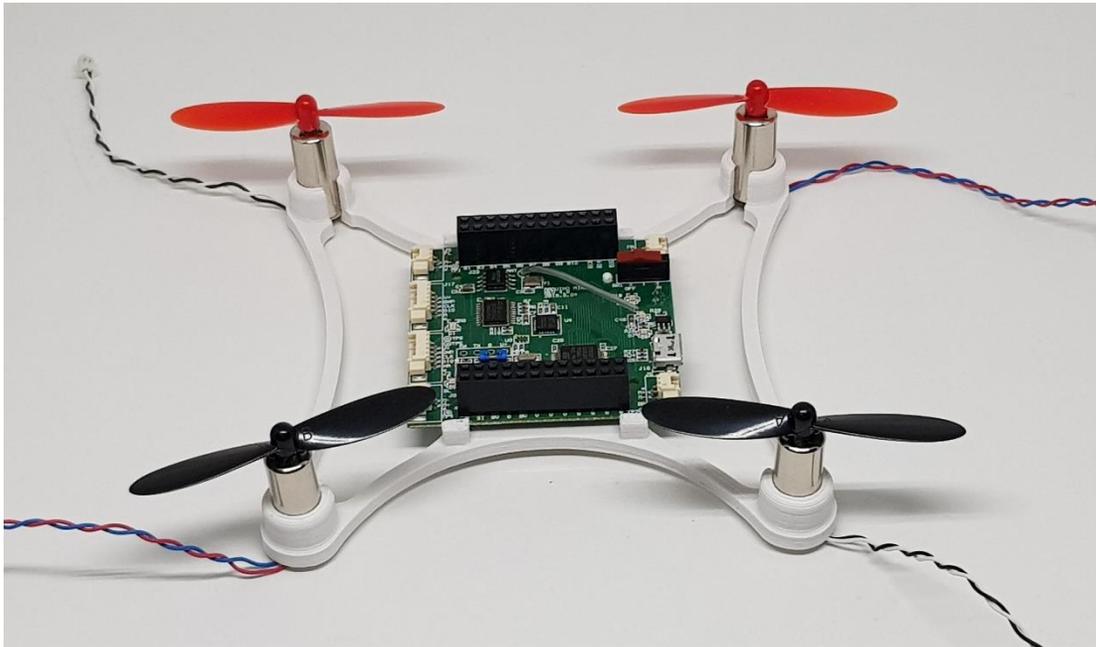


보드가 프레임에 가운데 위치하도록 함.



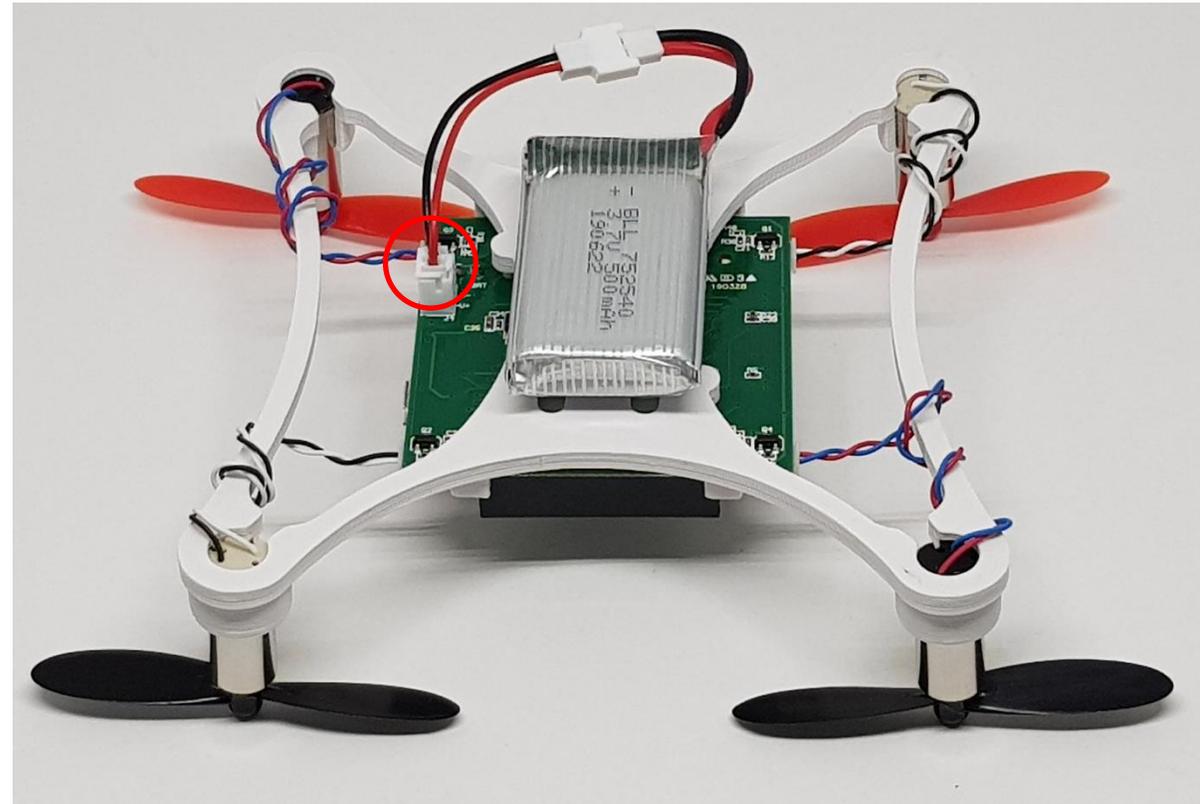
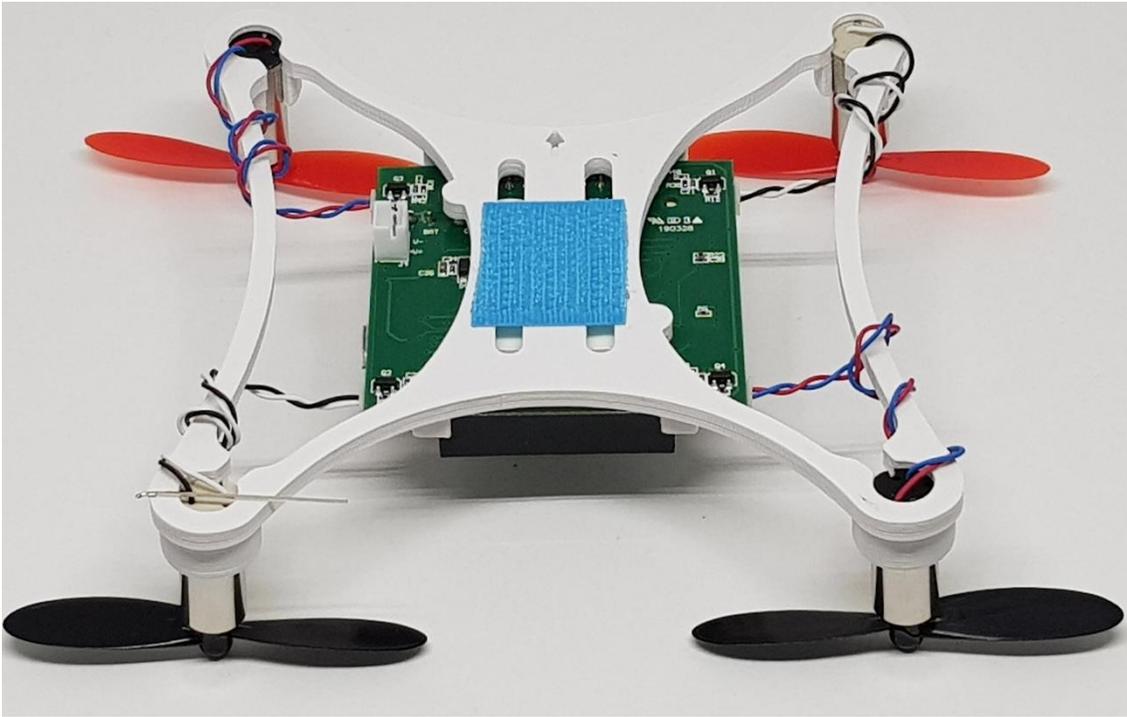
## DC모터 전선 정리 후 커넥터에 조립

DC모터 전선을 프레임에 감아서 정리 후 보드의 커넥터에 조립



# 배터리 조립

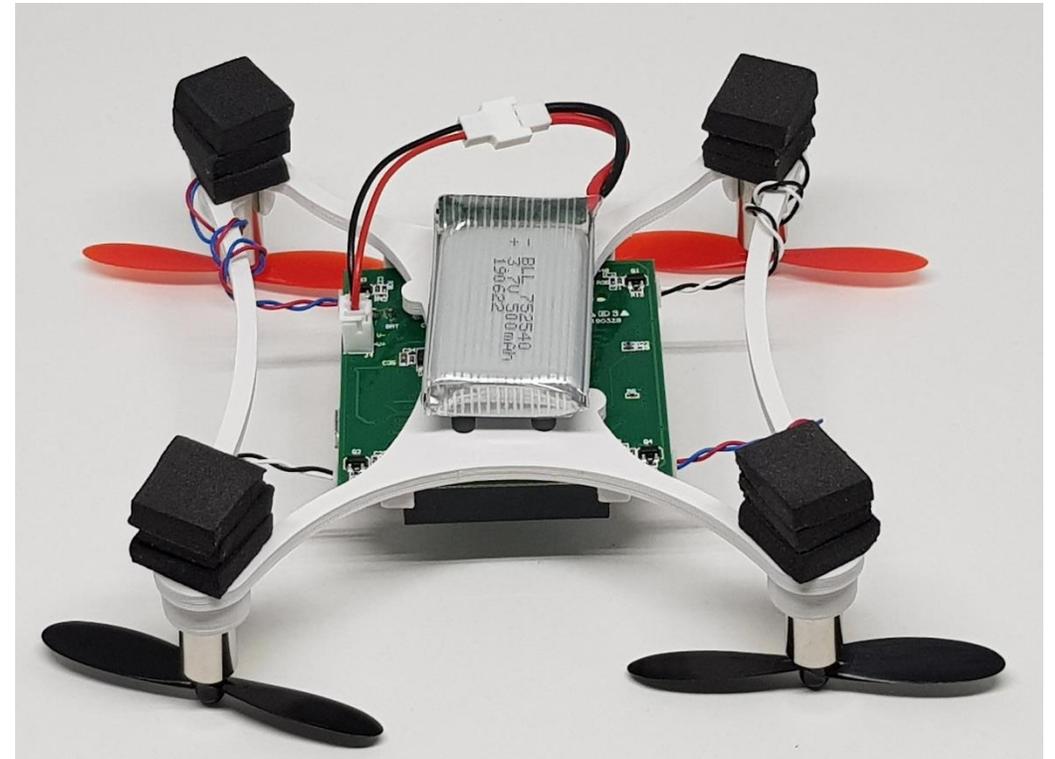
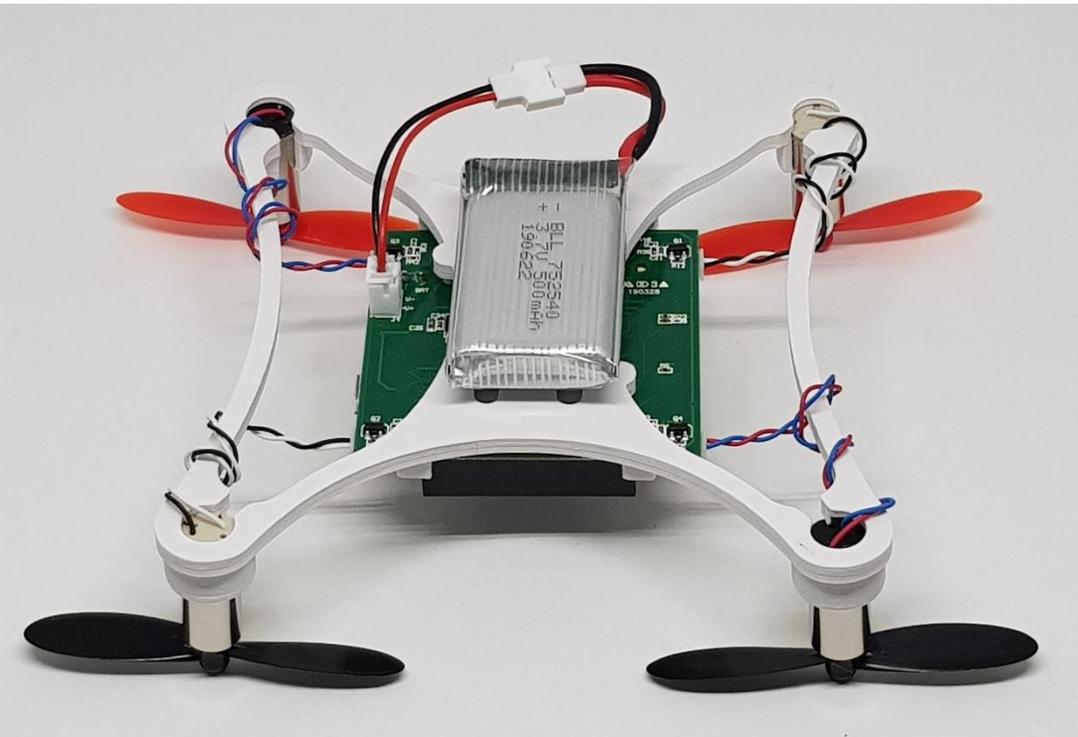
배터리와 프레임 뒷면에 한 쌍의 벨크로 테이프를 각각 접착 후 조립



## 다리 만들기



폼테이프를 이용해서 다리 만들기, 각각 네개의 다리를 3개의 폼 테이프를 적층하여 만듦.



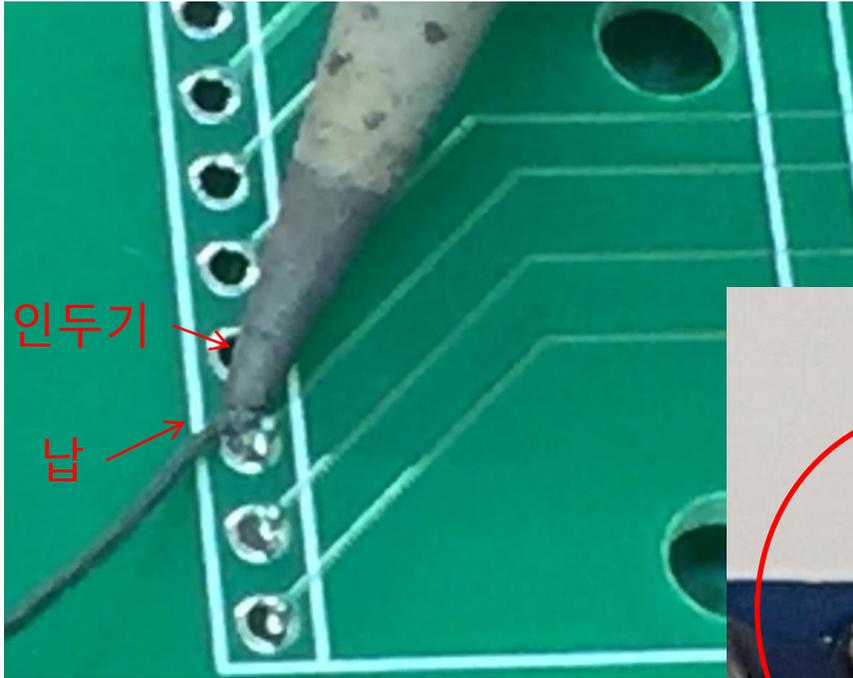
# 라두이노 미니 드론 완성



## 드론 조정기 조립

- 준비물 : 라두이노 미니 조정기

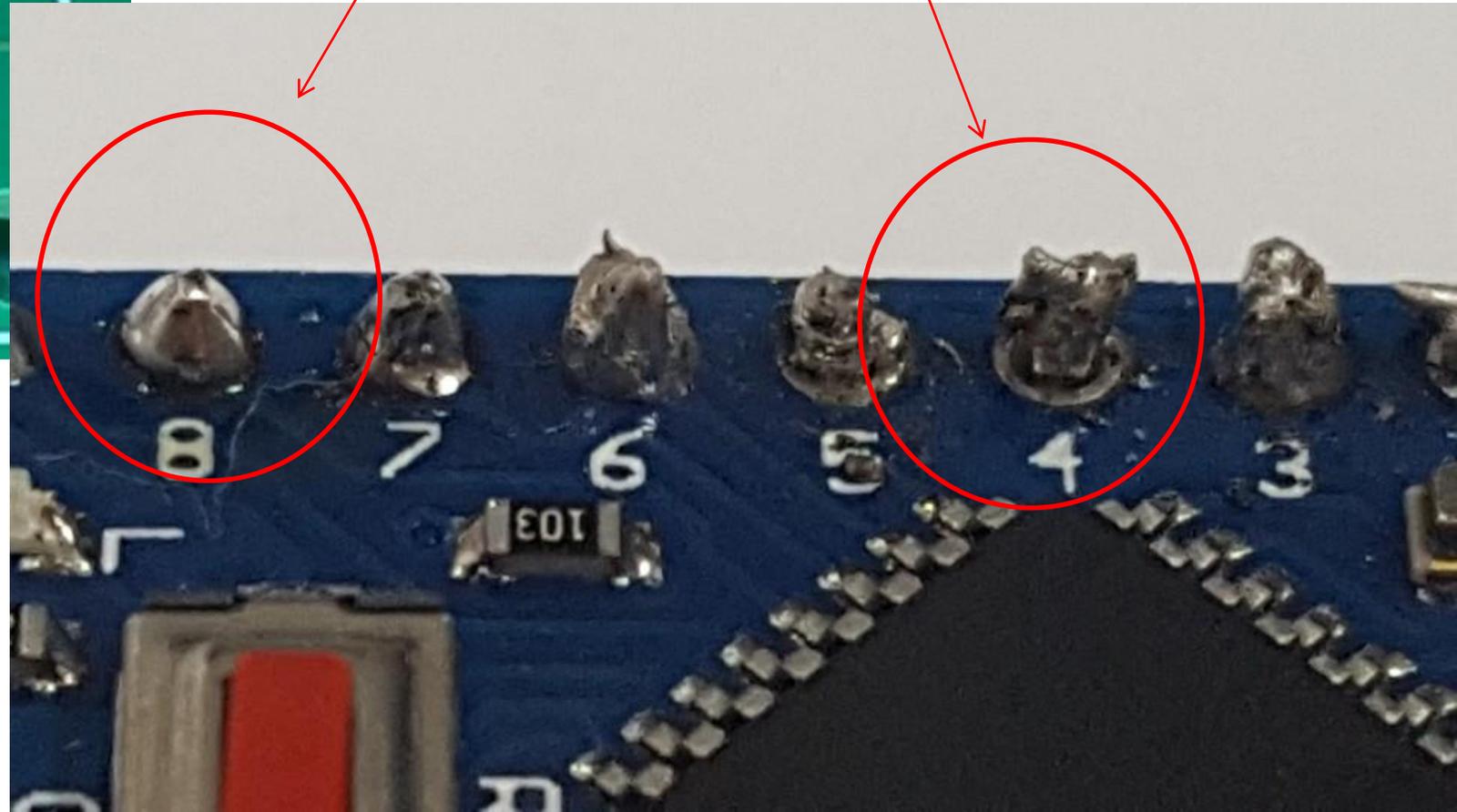
# 납땜 주의사항



인두기의 열이 충분히 오른 후 그림과 같이 납땜할 위치에 인두기를 대고 납을 녹여 납땜 부위에 납이 스며들도록 한다.

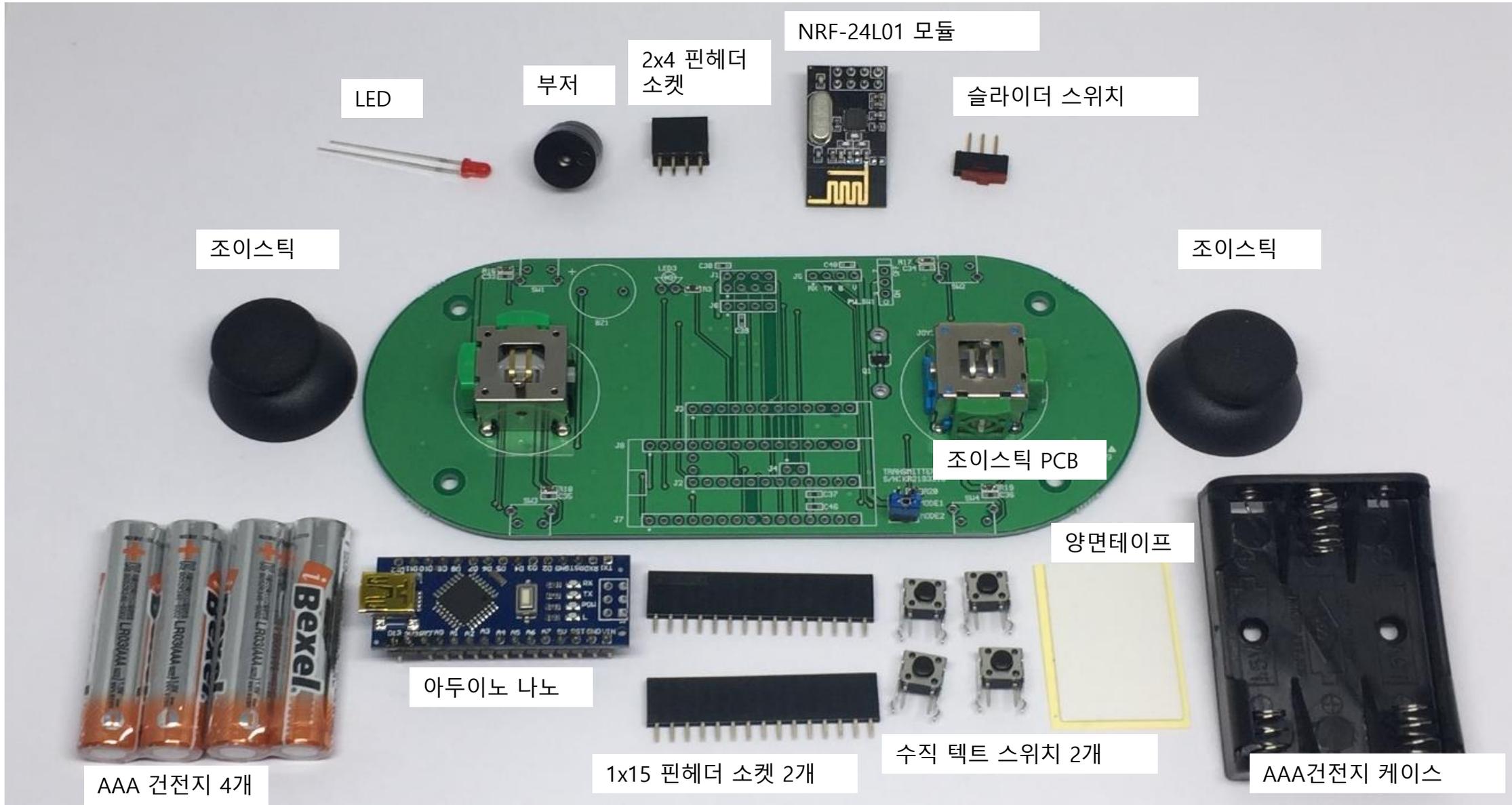
잘된 납땜

잘못된 납땜



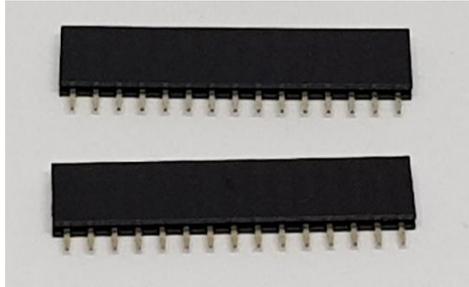
\*화상 주의, 환기 필요\*

# 라두이노 미니 조종기 구성품

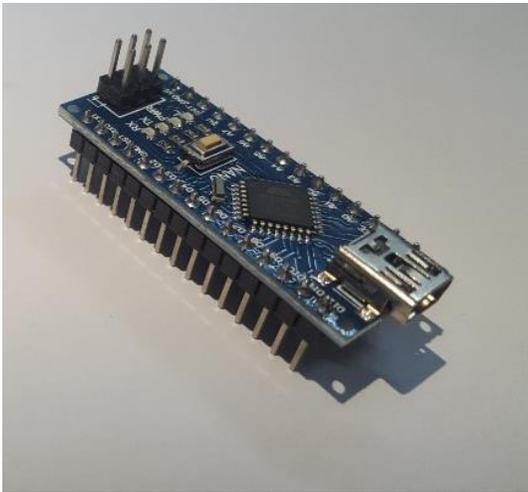
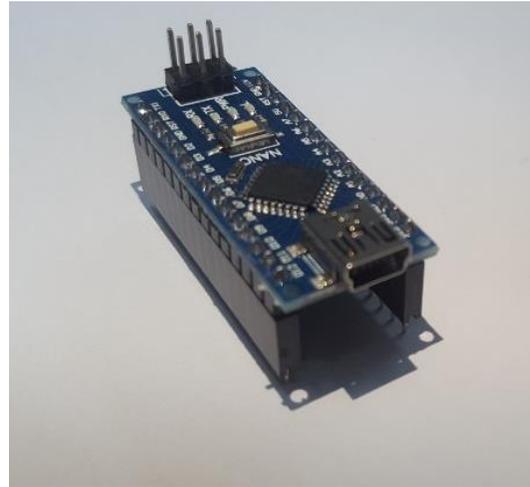


# 핀헤더 조립 팁

아두이노 나노에 핀헤더 소켓을 끼운 후 납땜을 하면 편리하다.

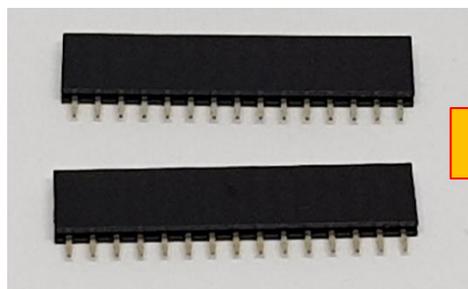


+

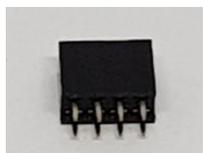


# 핀헤더 조립

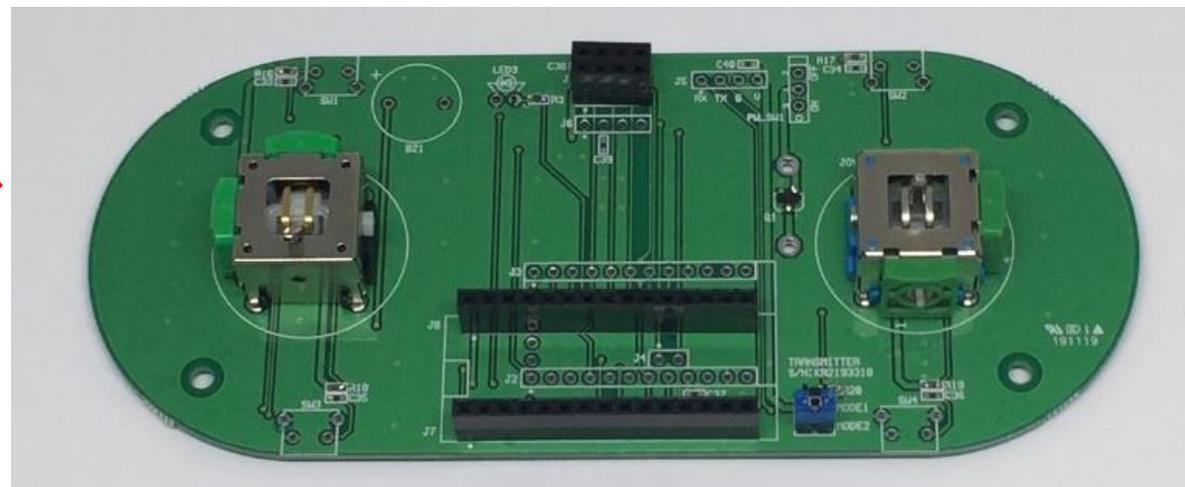
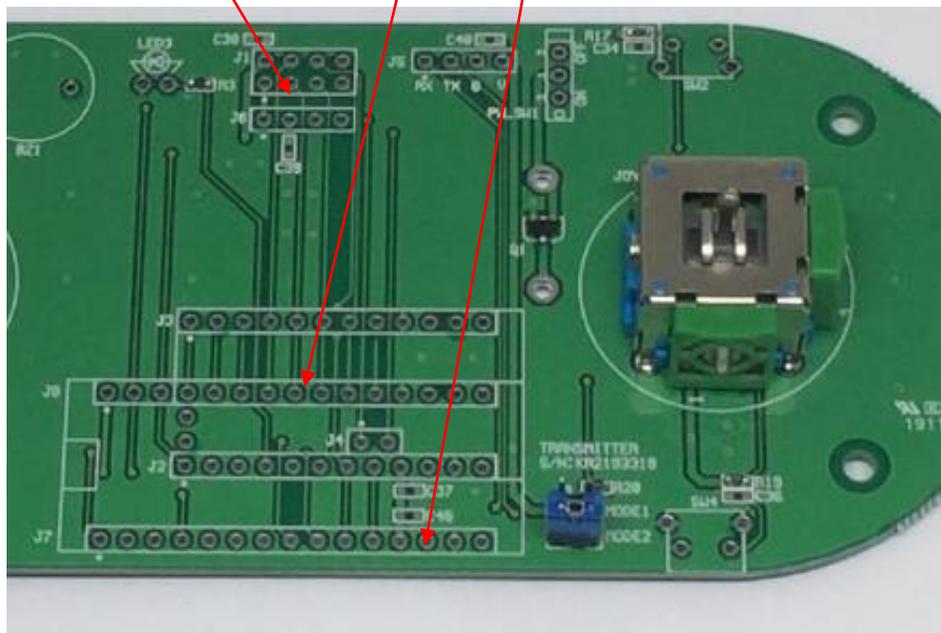
2x4 핀 헤더 소켓 1개, 1x15 핀 헤더 소켓 2개 삽입 후 납땜



J7, J8에 삽입



J1에 삽입



\*각각의 핀 헤더 소켓이 PCB에 직각이 되도록 납땜.  
그러지 않을 경우 부품이 조립되지 않을 수 있음.  
다음 장에 "핀헤더 조립 팁"을 참고하자

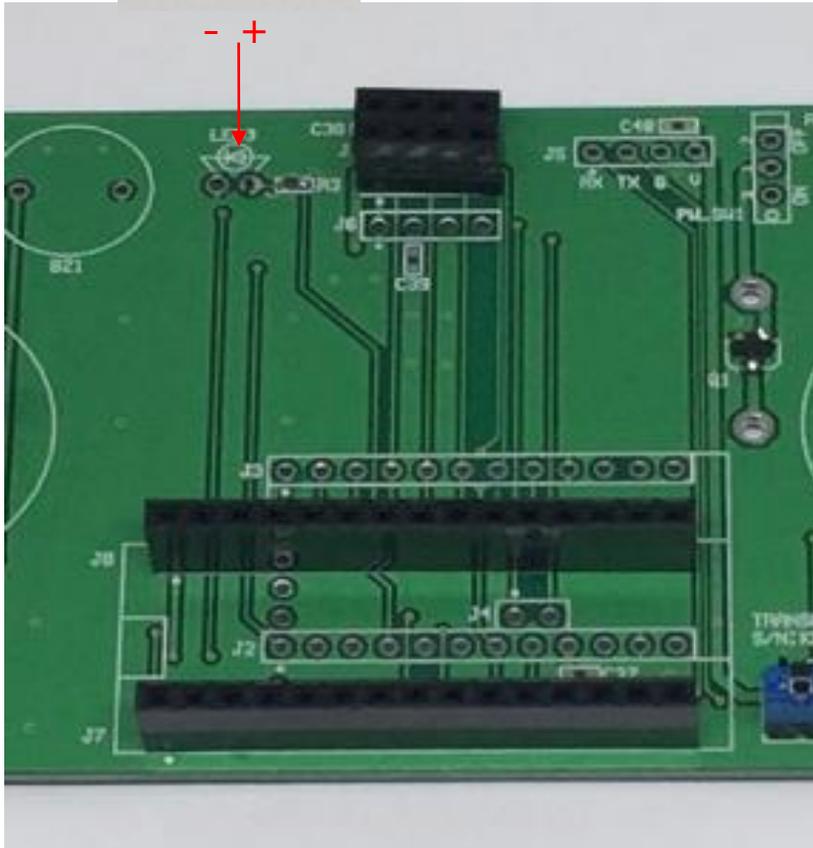
PCB을 뒷면으로 돌려 납땜 한다.

# LED 조립

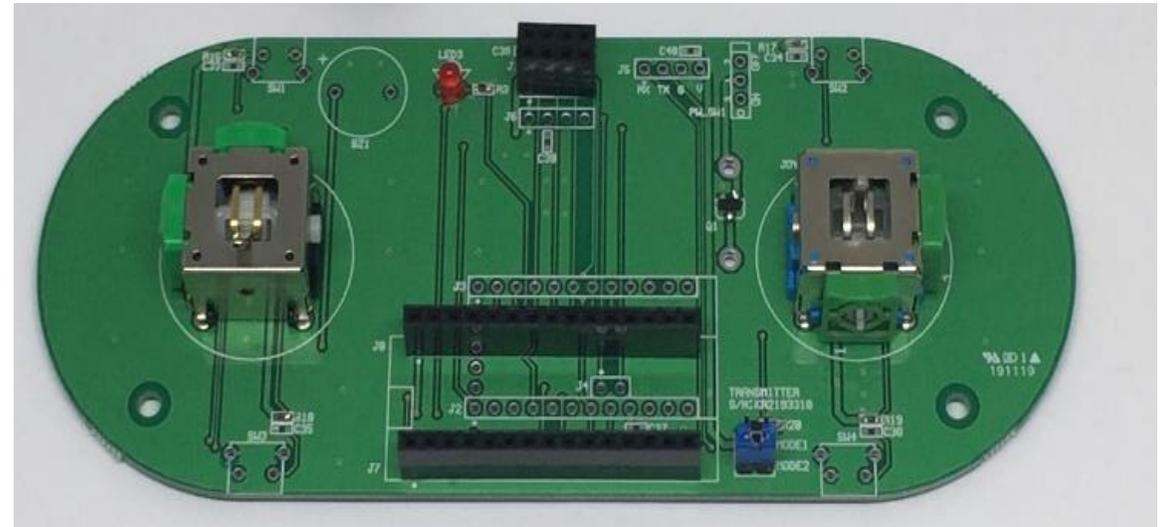
LED 극성(+, -)을 확인하고 삽입 후 납땜



LED3에 삽입



PCB을 뒷면으로 돌려 납땜 한다.

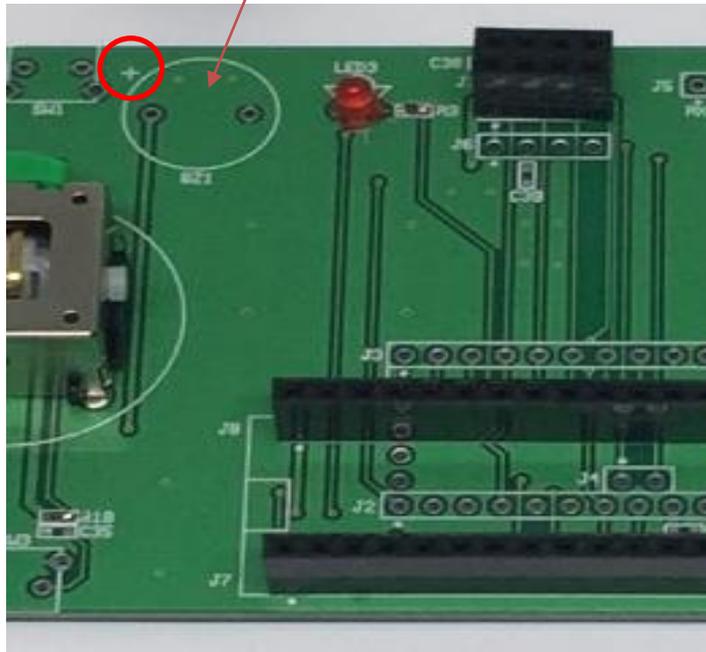


# 부저 조립

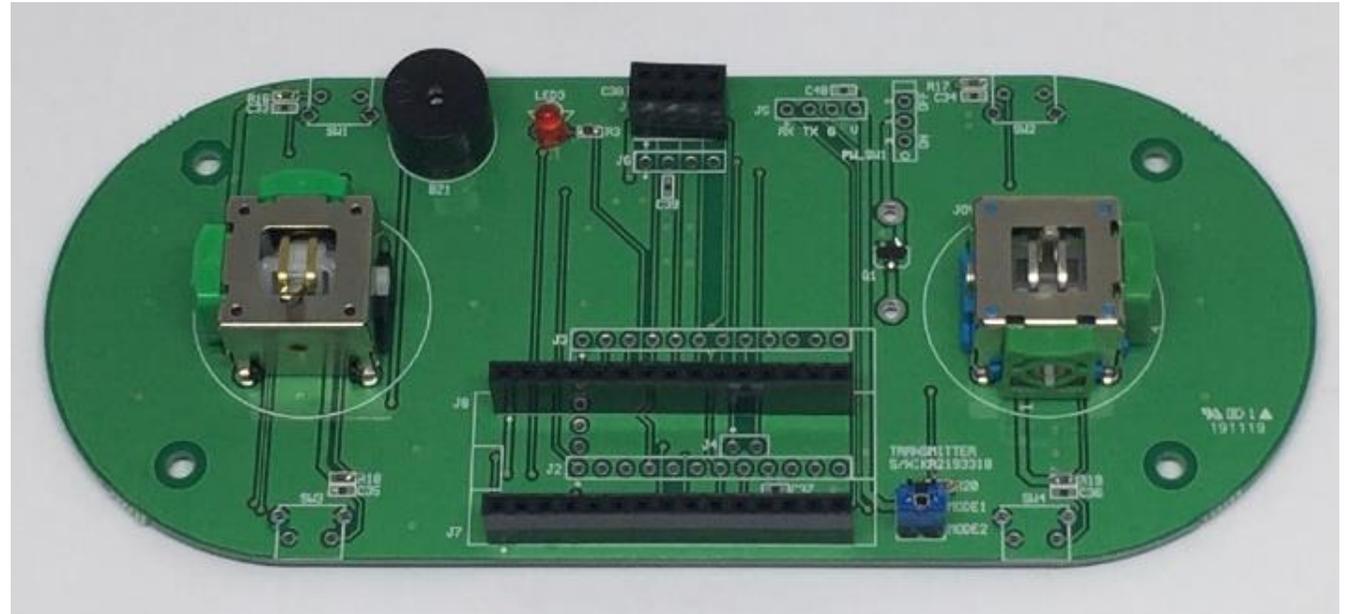
부저의 +가 PCB의 +와 일치 하도록 삽입 후 납땜



BZ1에 삽입

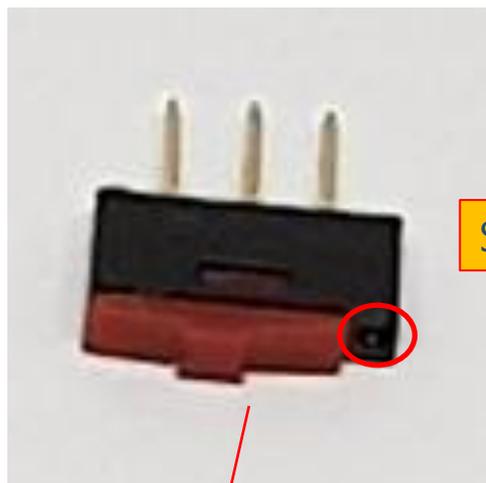


PCB을 뒷면으로 돌려 납땜 한다.



# 슬라이드 스위치 조립

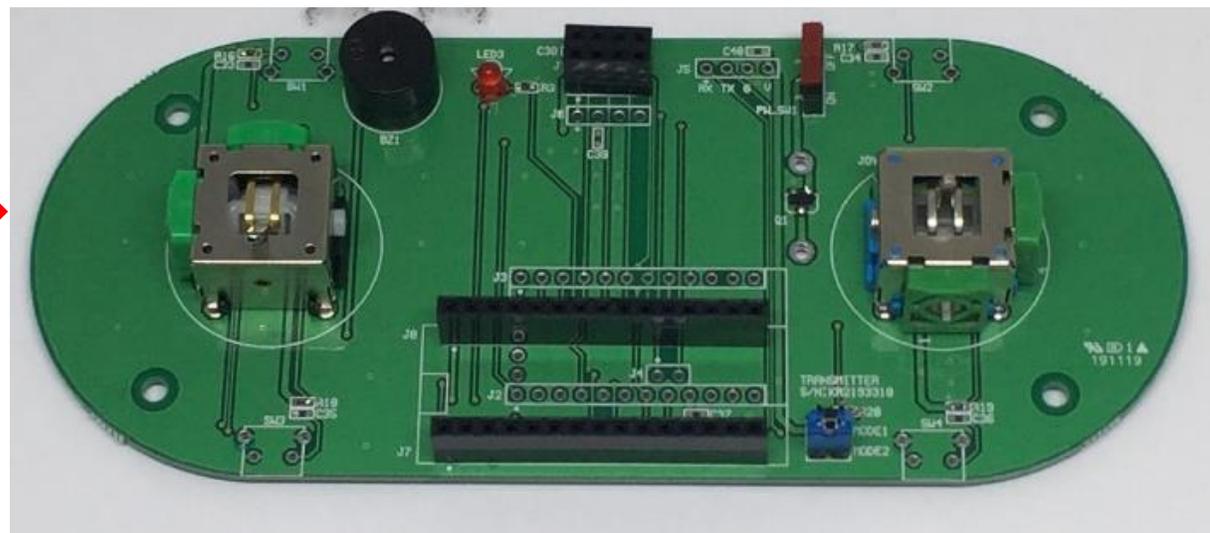
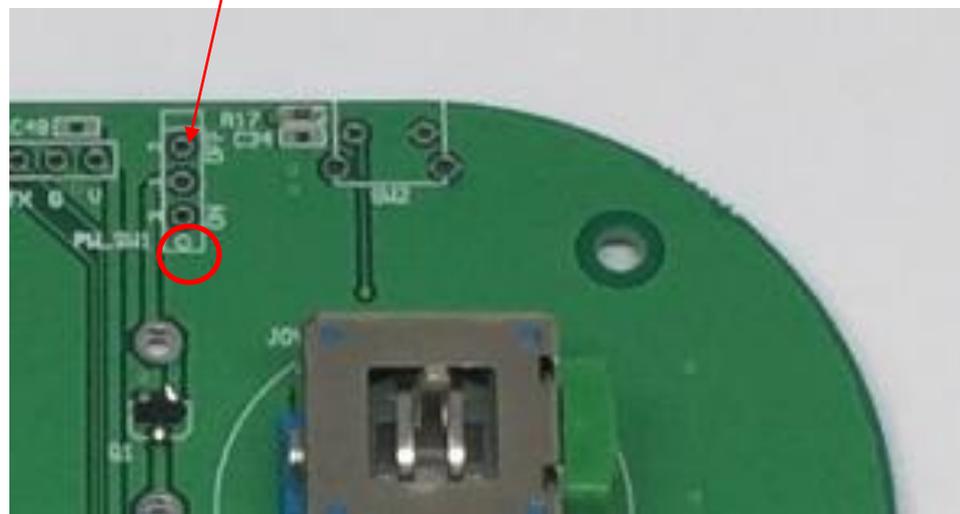
슬라이드 스위치 on off 방향에 유의해서 삽입 후 납땜



SW1에 삽입

빨간색 원 표시된 쪽으로 방향을 일치

PCB을 뒷면으로 돌려 납땜 한다.

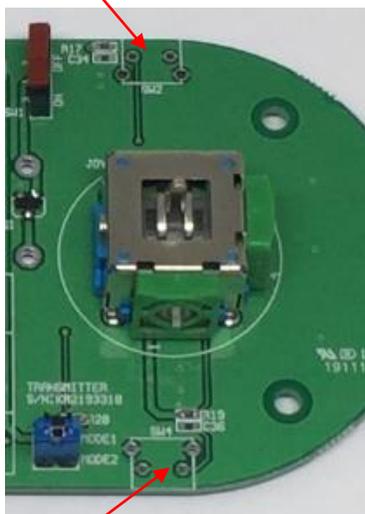


# 텍트 스위치 조립

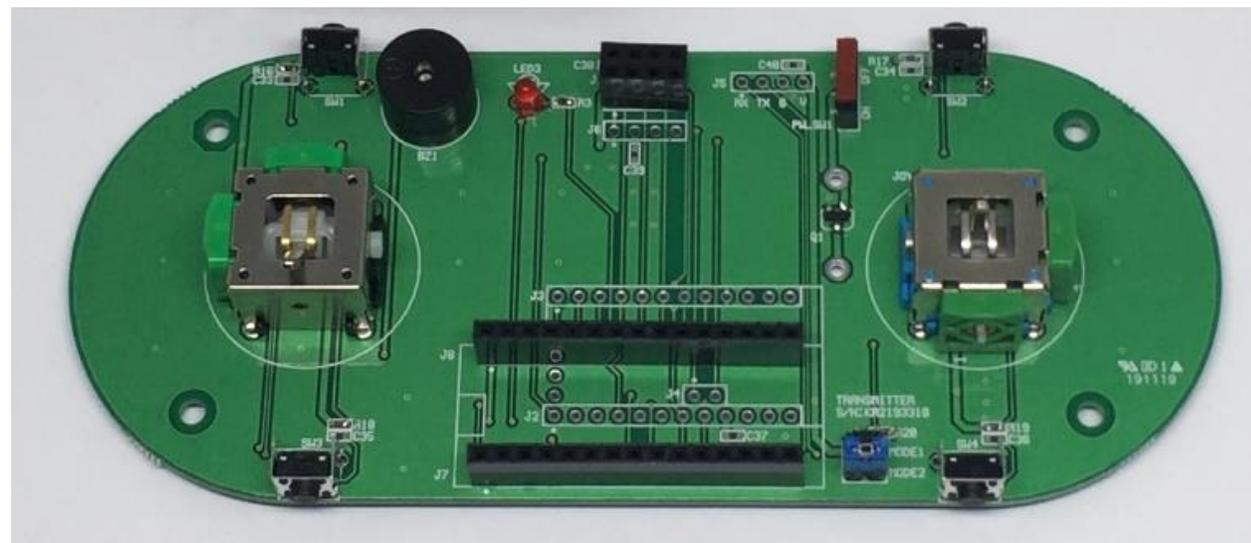
텍트 스위치 A타입, B타입 일치 하도록 삽입 후 납땜



HEADLESS, FLIP에 삽입



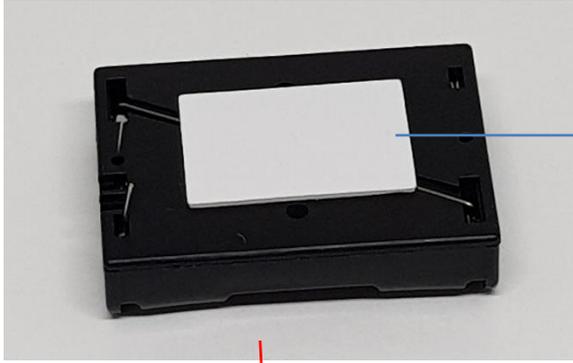
HEADLESS, FLIP에 삽입



PCB을 뒷면으로 돌려 납땜 한다.

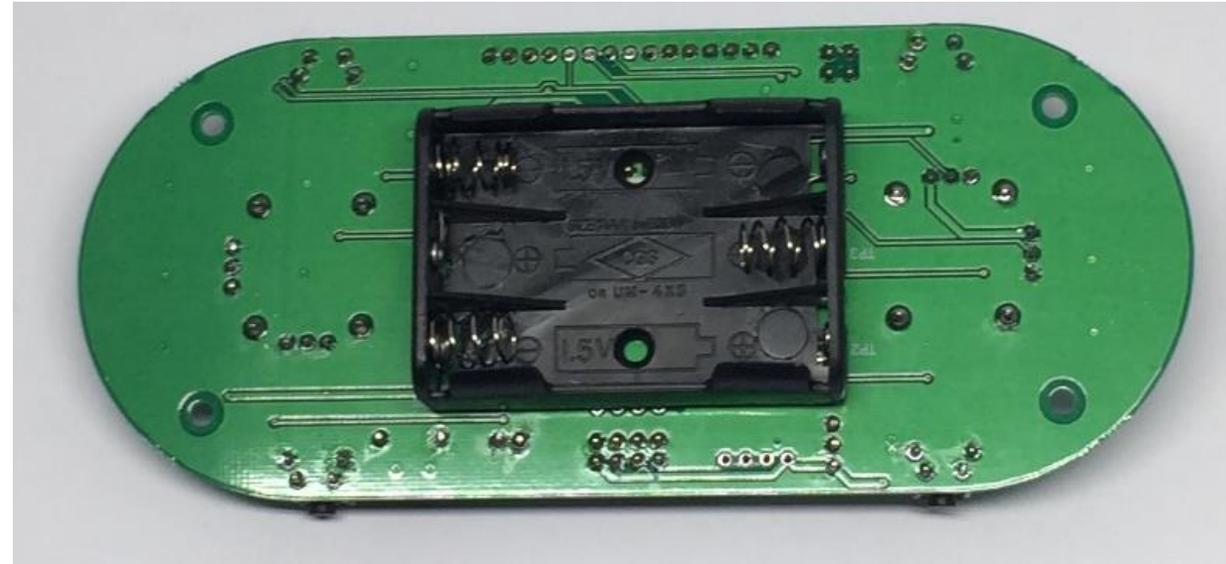
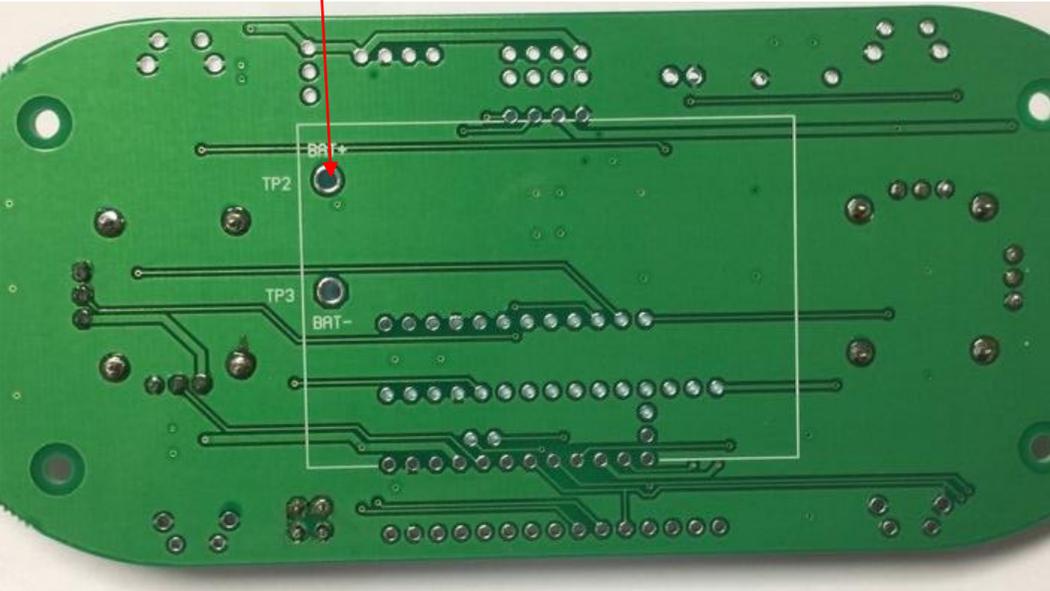
# AAA건전지 케이스 조립

양면 테이프 접착, + - 일치 삽입 후 납땜



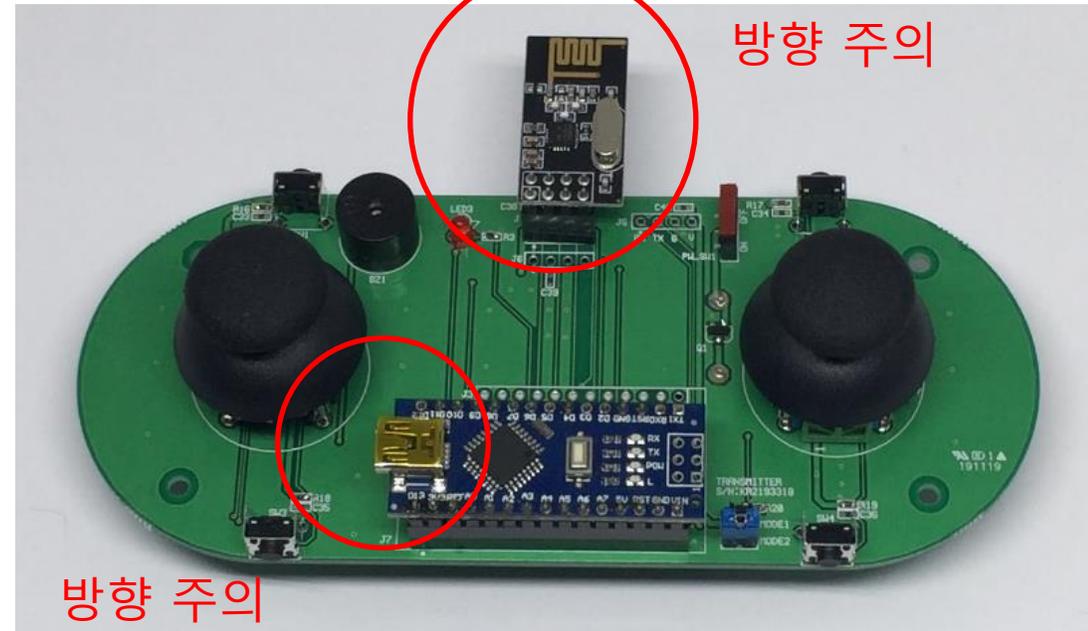
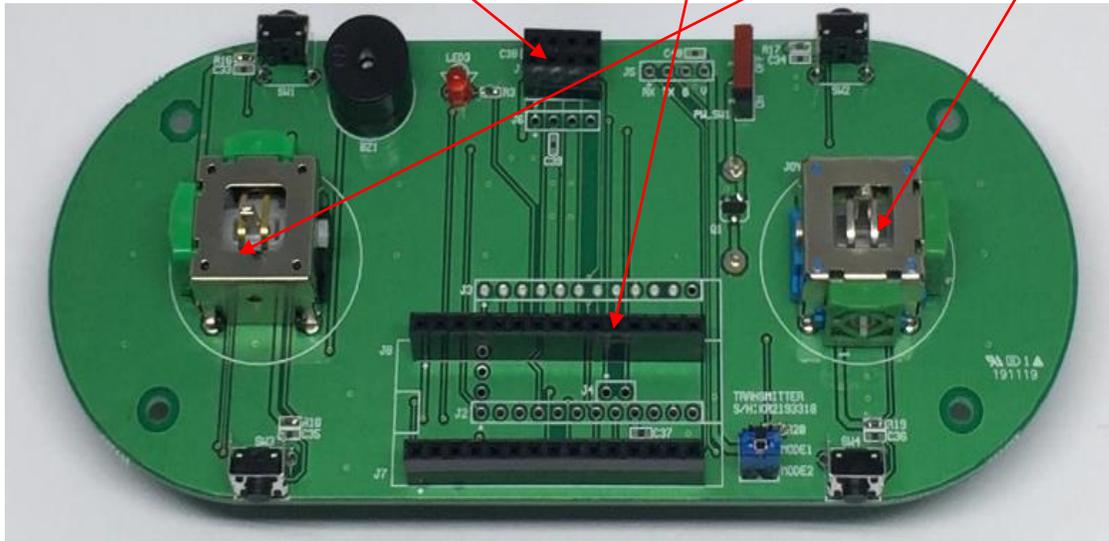
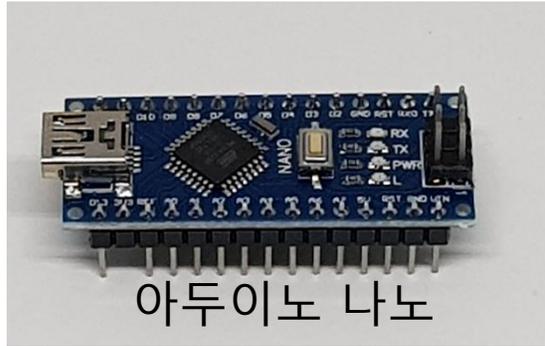
양면테이프

건전지 케이스를 삽입 후 PCB을 앞면으로 돌려 납땜 한다.



# NRF통신, 아두이노 나노, 조이스틱 커버 조립

NRF-24L01 모듈, 아두이노 나노 및 조이스틱 커버를 방향에 맞춰서 조립한다.

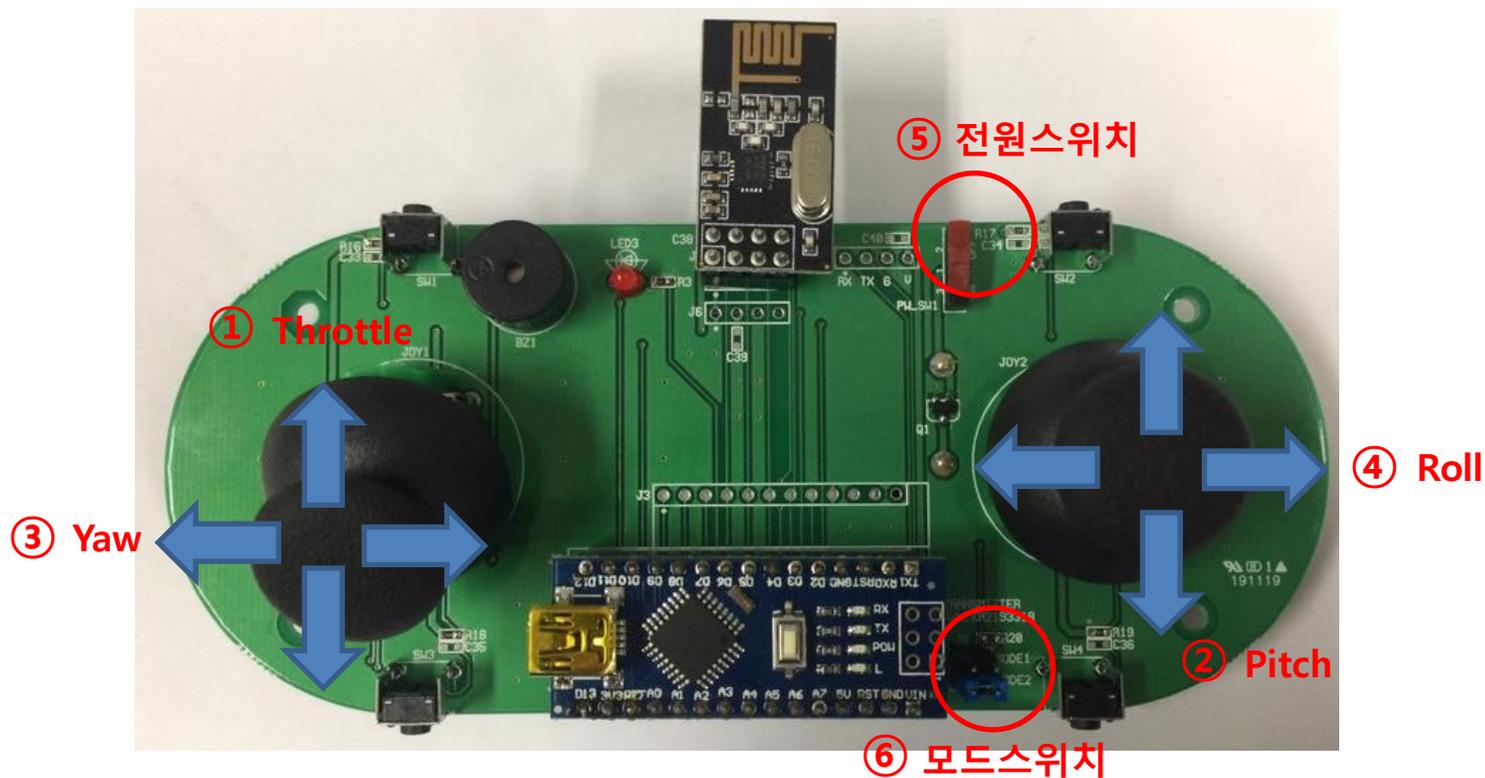


## 비행조작하기

- 준비물 : 라두이노 미니 드론  
라두이노 미니 조종기

## 조종기 용어 설명

- ① 상하타(Throttle, 스로틀): 스틱을 앞뒤로 밀어 드론을 승하강 시킨다.
- ② 전후타(Pitch, 피치): 스틱을 앞으로 밀면 전진하고, 뒤로 당기면 뒤로 후진합니다.
- ③ 회전타(Yaw, 요): 스틱을 좌로 움직이면 제자리에서 반 시계방향으로 회전하고, 우로 움직이면 제자리에서 시계방향으로 회전합니다.(수평 회전 정도)
- ④ 좌우타 ( Roll, 롤 ) : 스틱을 좌로 움직이면 좌측으로, 우로 움직이면 우측방향으로 기울면서 그 방향으로 나아갑니다. (좌우 기울어짐의 정도)
- ⑤ 전원 스위치 : 조종기의 스로틀을 제일 밑으로 내린 후 전원 스위치를 켜다.
- ⑥ 모드 스위치 : 조종기의 조종 모드를 1과 2로 선택한다. 모드 1로 선택 시 조종기를 180도 돌려서 잡는다.



## 드론 상태 설명

### 1. 수신대기 상태

배터리를 연결하고 드론의 전원 스위치를 켜준 뒤 상태 LED가 2초뒤 매우 빠르게 점멸합니다. 조종기의 RF(radio frequency) 신호를 받을 준비가 된 수신 대기 상태입니다

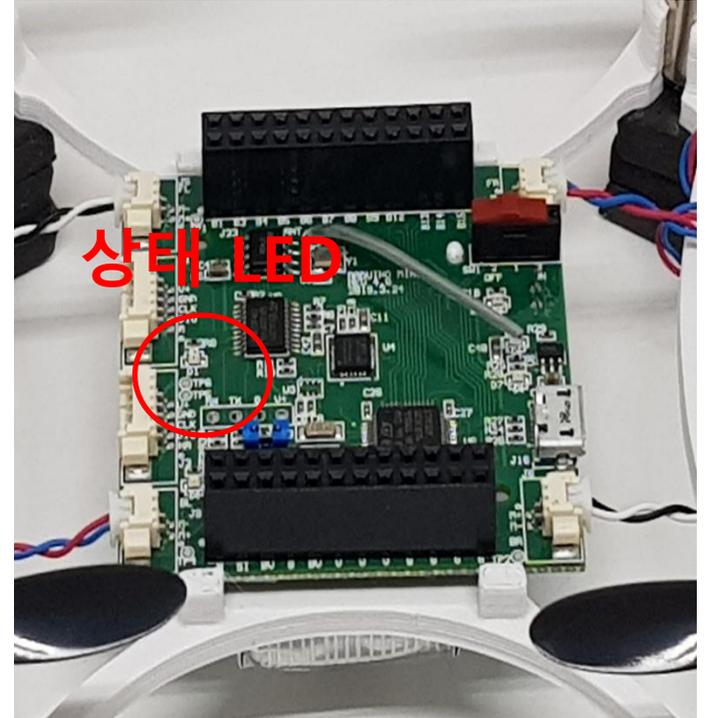
### 2. 페어링(pairing)

페어링은 드론과 조종기가 무선통신을 위해 필요한 작업입니다. 최초 전원을 켜면, 빨간 원 안의 LED가 매우 빠르게 깜박거립니다. 이때, 조종기의 전원을 켜서 페어링이 완료가 된 후에는 상태 LED의 빨간 불이 지속적으로 켜 집니다.

### 3. 센서보정(Calibration)

드론 내 관성센서의 (X, Y, Z)축을 재조정하는 작업입니다, 만약 센서보정을 하지 않으면 드론이 정상적으로 작동 하지 않을 수 있습니다. 그렇기 때문에, 처음 사용 전 반드시 센서보정 작업을 하는 것을 필요합니다.

센서보정 방법은 평평한 바닥면에 드론을 놓고 조종기와 드론이 페어링 된 상태에서 조종기의 Pitch를 아래 (↓↓↓)로 빠르게 세번 내리면, 빨간 원 안의 LED가 5번 점멸한후 켜지면 캘리브레이션이 완료 됩니다.



### ※ LOW Battery 가 되었을 경우

▶ 비행 중 빨간 원 안의 LED가 1초 간격으로 점멸 후, 자동으로 착륙합니다.

# 드론 비행하기



## 1. 드론 전원 켜기

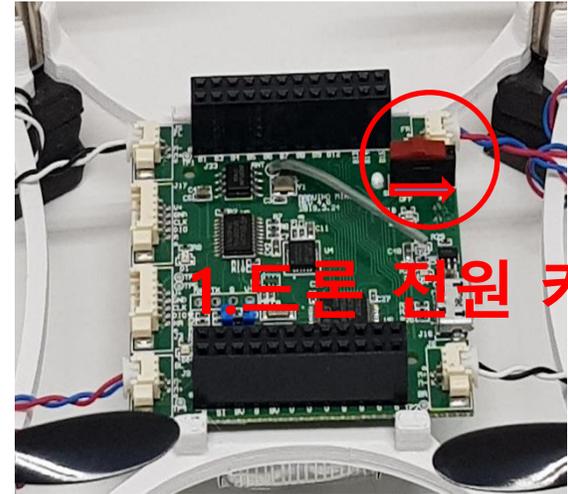
드론의 제어보드에 3.7V 배터리를 연결하고 빨간색 전원 스위치(바깥쪽 방향)를 켭니다.

## 2. 조종기 전원 켜기

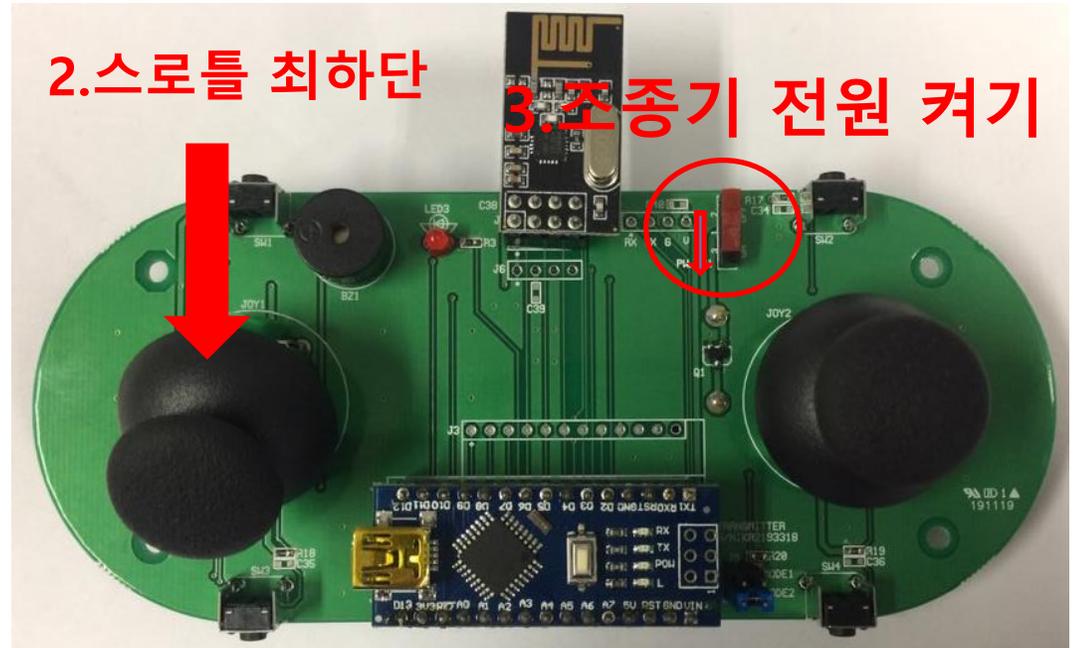
페어링 모드로 진입한 드론을 확인한 뒤 조종기의 스로틀(throttle)을 내려주고 조종기의 전원을 켭니다. 만약 스로틀이 최하단 상태가 아니면 시동이 걸리지 않습니다. 조종기를 켜면 한번의 삐- 소리와 함께 드론과 조종기가 페어링 되면서 자동으로 시동이 걸립니다.

## 3. 비행하기

배터리 소진상태로 인한 자동 착륙까지 페어링 된 조종기를 이용하여 드론을 비행합니다.



1. 드론 전원 켜기



2. 스로틀 최하단

3. 조종기 전원 켜기

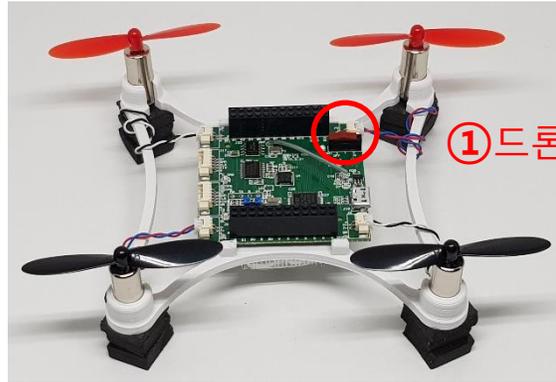
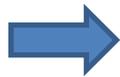
※ 비행 조정이 익숙하기 전까지는 낮은 고도에서 날리자.

## 다수의 드론 비행하기

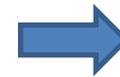
다수의 드론을 같은 공간에서 비행하기 위해 순차적으로 드론과 조종기를 페어링 해야 한다.  
그림과 같이 하나의 드론 전원을 먼저 켜고 그 다음 해당 조종기 전원을 켜다.  
하나의 페어링이 완료된 후 다음 드론의 전원을 켜고 해당 조종기 전원을 켜서 페어링 한다.  
이와 같이 여러 개의 드론을 각각의 조종기에 페어링할 수 있다.



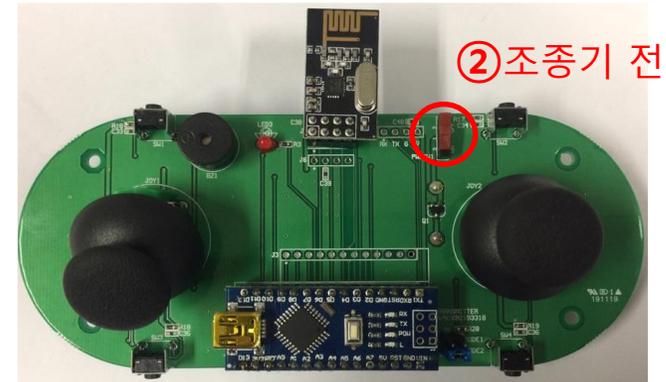
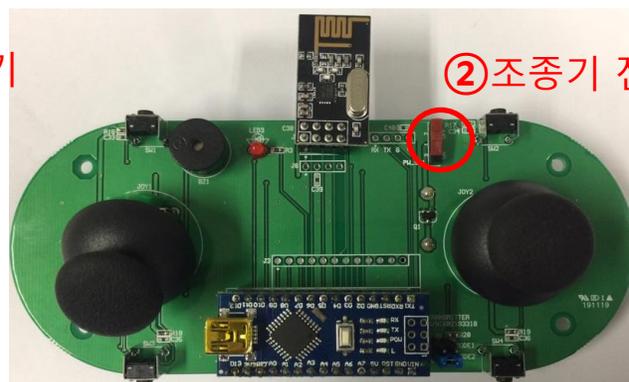
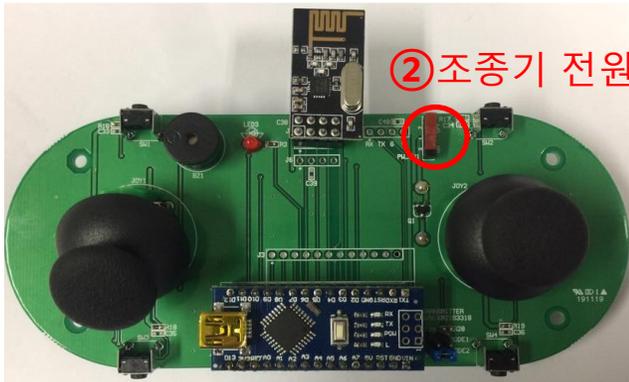
↕ 페어링



↕ 페어링



↕ 페어링



## 주의사항 및 참고사항

# 라두이노 미니 드론 이용시 주의사항

## 제품사용 주의사항

1. 반드시 허가와 안전이 확보된 장소에서만 비행하세요.
2. 직사광선 및 습기가 없는 곳에 보관하세요.
3. 상승레버(throttle)를 천천히 작동하여 급속도로 상승하지 않도록 유의하세요.
4. 상승, 비행, 착륙 시 반드시 주변 3미터 이내에 사람 또는 동물, 물건 등이 없는지 확인하세요.
5. 어린이들이 작동할 시 반드시 보호자의 지도하에 사용하세요.
6. 반드시 가시거리 내에서만 작동하세요.
7. 키트에 포함된 조종기를 이용할 경우, 반경 30m 이내 공간에서 2대 이상 드론의 조종을 피하세요. 피아구분이 되지 않아 일 대 다수의 드론이 페어링 되어 비행될 수 있습니다.

## 배터리 취급 주의사항

1. 리튬 폴리머(Li-Po) 배터리 특성 상 취급 주의사항을 반드시 숙지한 후 이용하세요.
2. 배터리를 임의로 분해하거나 개조하지 마세요.
3. 배터리 과충전 시 과부하로 인해 배터리가 부풀리거나 화재가 발생할 수 있습니다.
4. 드론의 제어보드와 같이 과충전 보호회로가 없는 충전기를 충전 시 과부하로 인해 배터리가 부풀리거나 화재가 발생할 수 있습니다.
5. 리튬 폴리머 배터리 특성 상 완전 방전된 배터리는 저전압으로 인해 배터리 수명이 현저히 떨어져 재사용이 불가능해질 수 있으므로 완전 방전 시 반드시 재충전 하세요.
6. 키트에 포함된 조종기용 배터리를 혼용해서 사용하지 마세요. 사용한 배터리와 새배터리의 혼용 또는 서로 다른 종류의 배터리 혼용 등은 오작동의 원인이 될 수 있습니다.
7. 배터리는 소모품이므로 충/방전이 거듭될수록 수명이 줄어들며 과방전 및 과충전은 배터리의 수명을 저하시키는 주요인입니다. 여분 배터리를 구비하여 과사용 방지를 권장합니다.
8. 장기간 배터리를 사용하지 않을 시 배터리를 60% 정도 방전시켜 보관하세요.
9. 리튬 폴리머 배터리를 충전 중에는 외출하지 마세요.
10. 리튬 폴리머 배터리는 열과 수분에 취약하므로 고온 및 습한 곳에 보관하지 마세요.
11. 휴대폰 전용 고속 충전 어댑터를 사용하지 마십시오. 고장의 원인이 됩니다.

## 정비 및 취급 방법

1. 반드시 전용 부품을 사용하세요.
2. 제어보드는 물과 접촉 시 작동하지 마시고 전원을 끈 뒤 배터리를 분리한 상태로 완전 건조시키세요.
3. 이물질 및 장애물 등으로 인해 프로펠러에 간섭이 발생 할 시 모터 고장의 원인이 됩니다. 드론의 작동을 즉시 멈추고 이물질 및 장애물 등을 완전히 제거한 후에 사용하세요.
4. 비행 전 항상 부속품의 상태를 확인하고 손상된 부품 등은 새것으로 교체하세요.
5. 손상된 부품을 재사용할 시 오작동 및 고장의 원인이 됩니다. 여러분의 부품을 구비하여 드론의 상태가 항상 최상의 컨디션을 유지하는 것을 권장합니다.
6. 반드시 가시거리 내에서만 작동하세요.
7. 키트에 포함된 조종기를 이용할 경우, 반경 30m 이내 공간에서 2대 이상 드론의 조종을 피하세요. 피아구분이 되지 않아 일 대 다수의 드론이 페어링 되어 비행될 수 있습니다.

**\* 반드시 취급상 주의사항을 숙지한 후 사용하세요**

주의사항 미숙지로 인해 발생한 문제 등은 당사에서 책임지지 않습니다.

사용자의 관리 소홀 및 부주의로 인한 제품 손상은 불량으로 인정되지 않으며, 유상수리 대상입니다. 모든 부품은 소모품입니다.

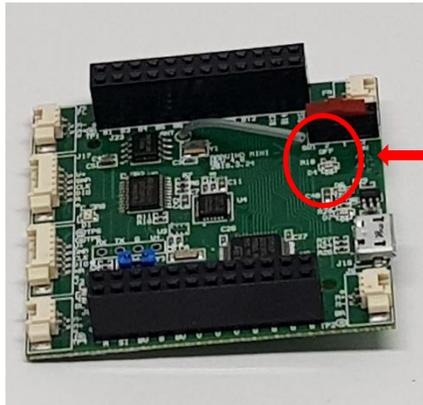
## 참고사항

### 드론 운용 안내

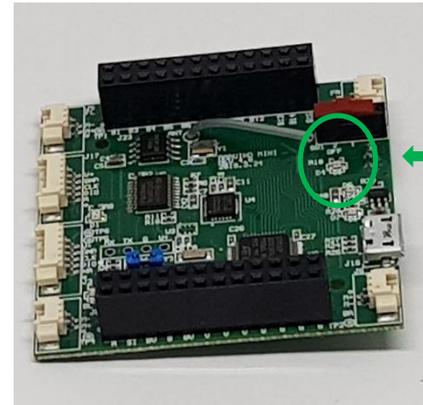
1. 라두이노 미니 드론 제품은 오토 호버링 기능이 없으므로 왼쪽 레버가 중간에 고정되지 않습니다. 따라서 사용자가 직접 드론의 호버링을 유지해야 합니다.
2. 비행 중 드론의 LED가 점멸한다면 배터리가 부족상태이므로 LED점멸 인지 시 즉시 가까운 곳으로 드론을 착륙시키세요. 드론의 배터리 부족 신호를 인지 하였음에도 계속해서 비행할 경우 드론의 분실 및 파손 등을 야기합니다.
3. 프로펠러 교체시 타입 확인을 정확히 하세요. 잘못된 타입을 조립하였을 경우 드론의 정상비행이 불가능합니다.
4. 관련법령에 의거하여 야간비행, 관제권 내에서의 비행, 150m 고도 이상으로 비행, 음주비행, 시가지비행, 가시거리 범위 외에서의 비행등은 금지되어 있습니다.
5. 드론은 실내비행 및 가급적 바람이 불지 않는 날씨의 실외 비행을 권장합니다.
6. 드론 등의 무선설비는 전파 혼선의 가능성이 있으므로 인명 안전과 관련된 사고 책임은 사용자에게 있습니다.
7. 드론의 모든 부품은 소모품입니다.

### 리튬 폴리머 배터리 충전상태

1. 리튬 폴리머(Li-Po) 배터리가 방전 시 제어보드에 Micro USB를 이용하여 PC 또는 충전 어댑터에 연결하여 충전합니다.
2. 그림과 같이 LED가 초록색으로 변하면 충전이 완료된 상태입니다.
3. 배터리 표준 충전 시간은 60분 이내입니다. 반드시 충전시간을 지켜주세요. 과충전 시 과부하로 인해 배터리가 부풀리거나 화재가 발생 할 수 있습니다



방전 시  
빨간색 점멸

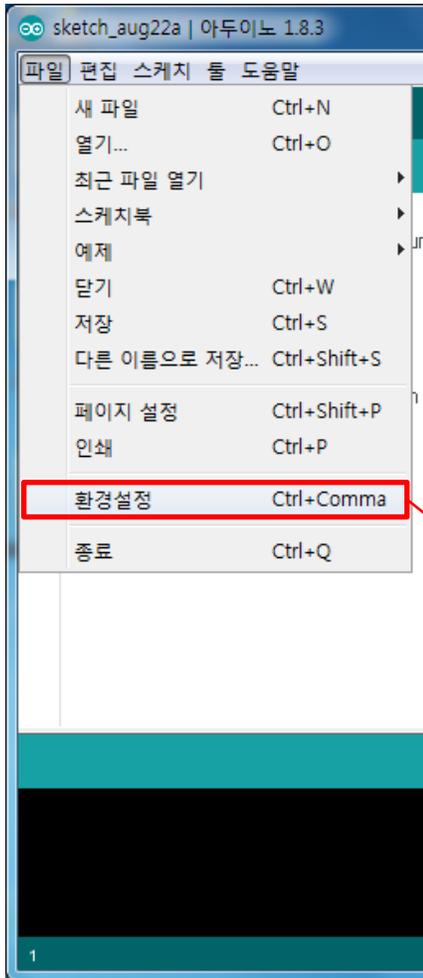


완충 시  
초록색

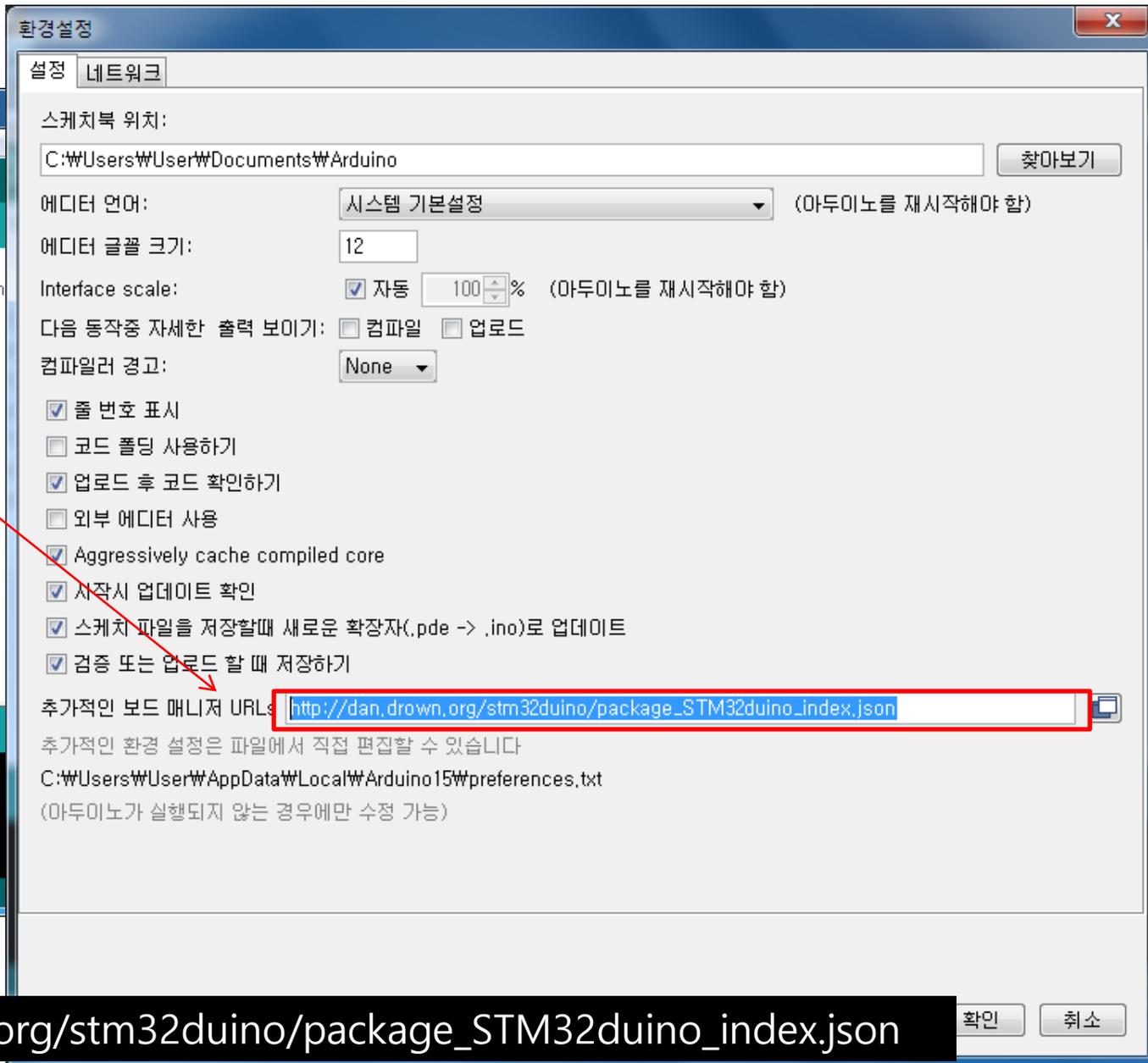
# **PART III. 드론에 코딩하기**

# 개발환경 구축하기

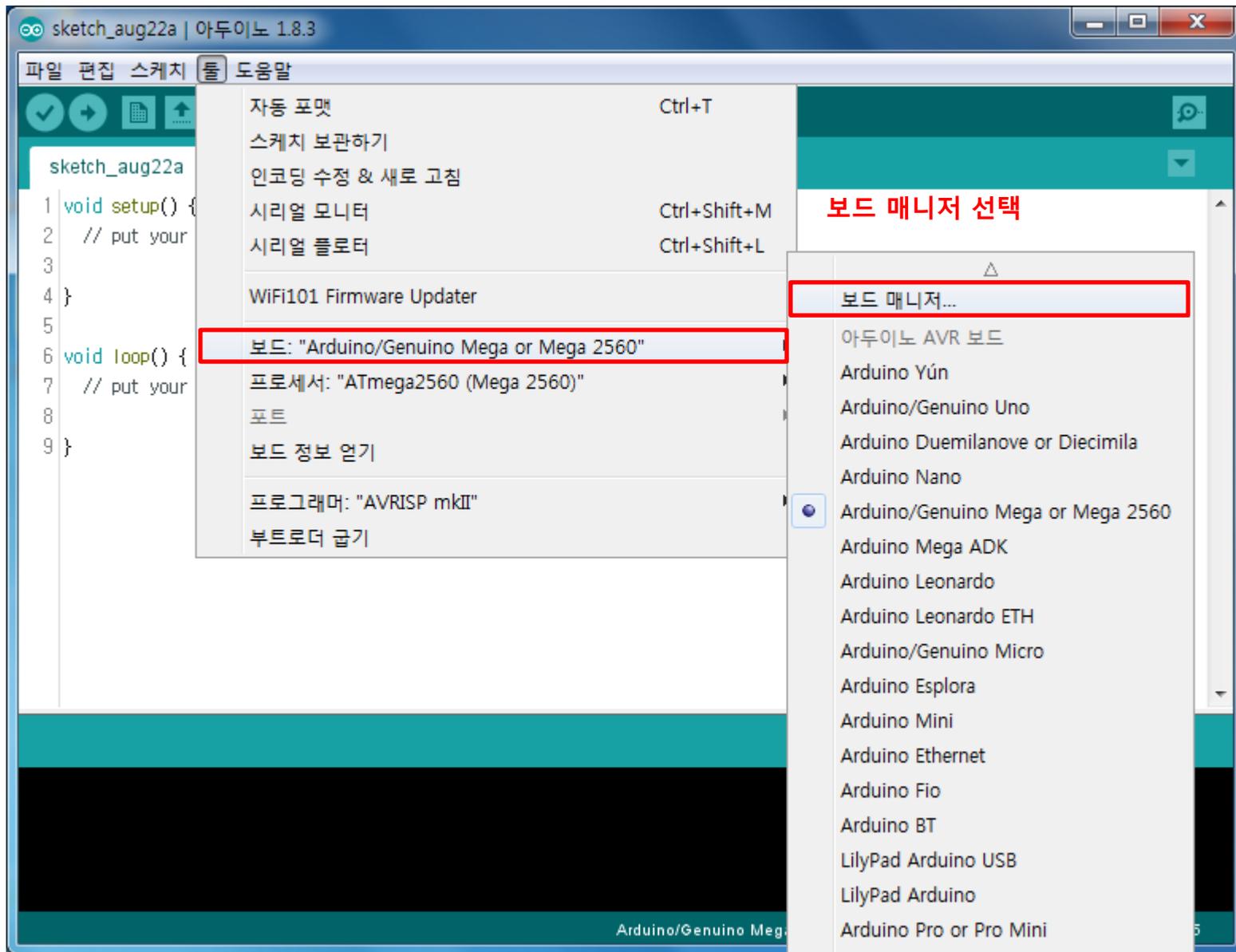
- 준비물 : Window 환경의 PC
- 프로그램 : 아두이노 IDE(스케치)

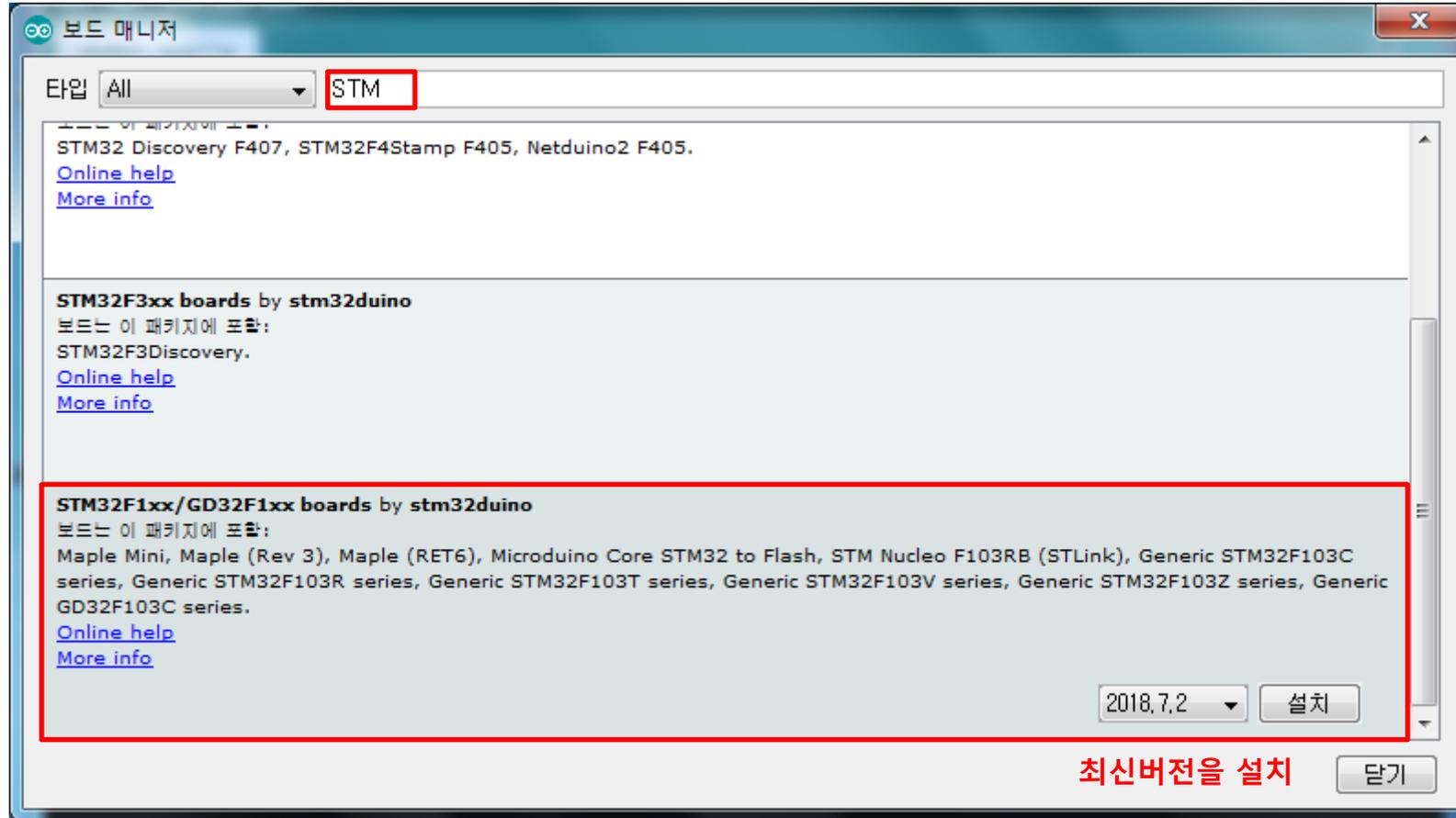


보드 매니저 URL을 추가



[http://dan.drown.org/stm32duino/package\\_STM32duino\\_index.json](http://dan.drown.org/stm32duino/package_STM32duino_index.json)





Browser address bar: http://www.raduino.co.kr/sub.php?menu=4

Navigation menu: [제품소개](#) [교육](#) [블로그](#) [자료실](#) [라두위키](#) [외주제작](#)

Search bar:

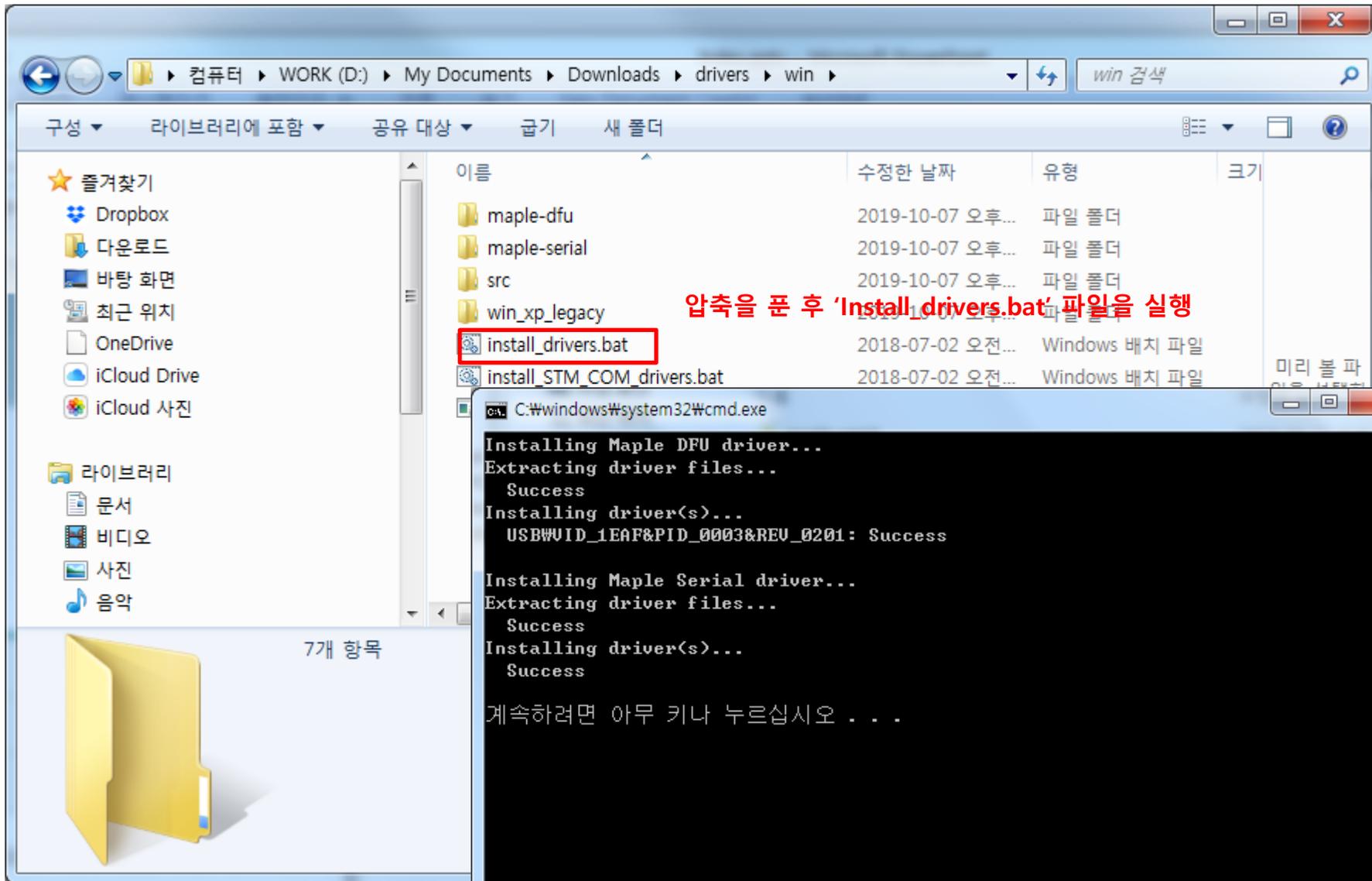
URL: <http://www.raduino.co.kr>

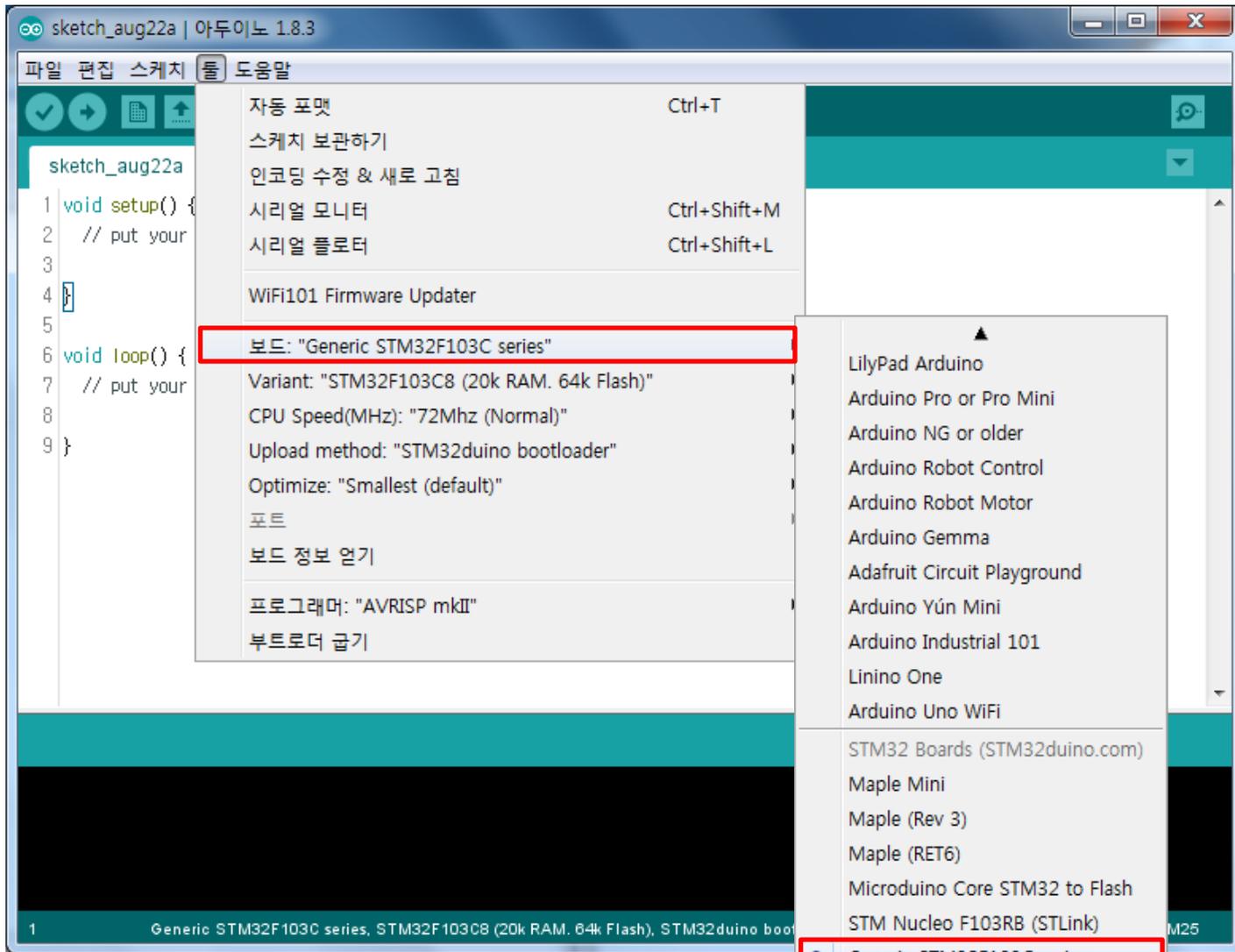
Section: **자료실에서 STM32 USB 드라이버 다운로드**

## 자료실

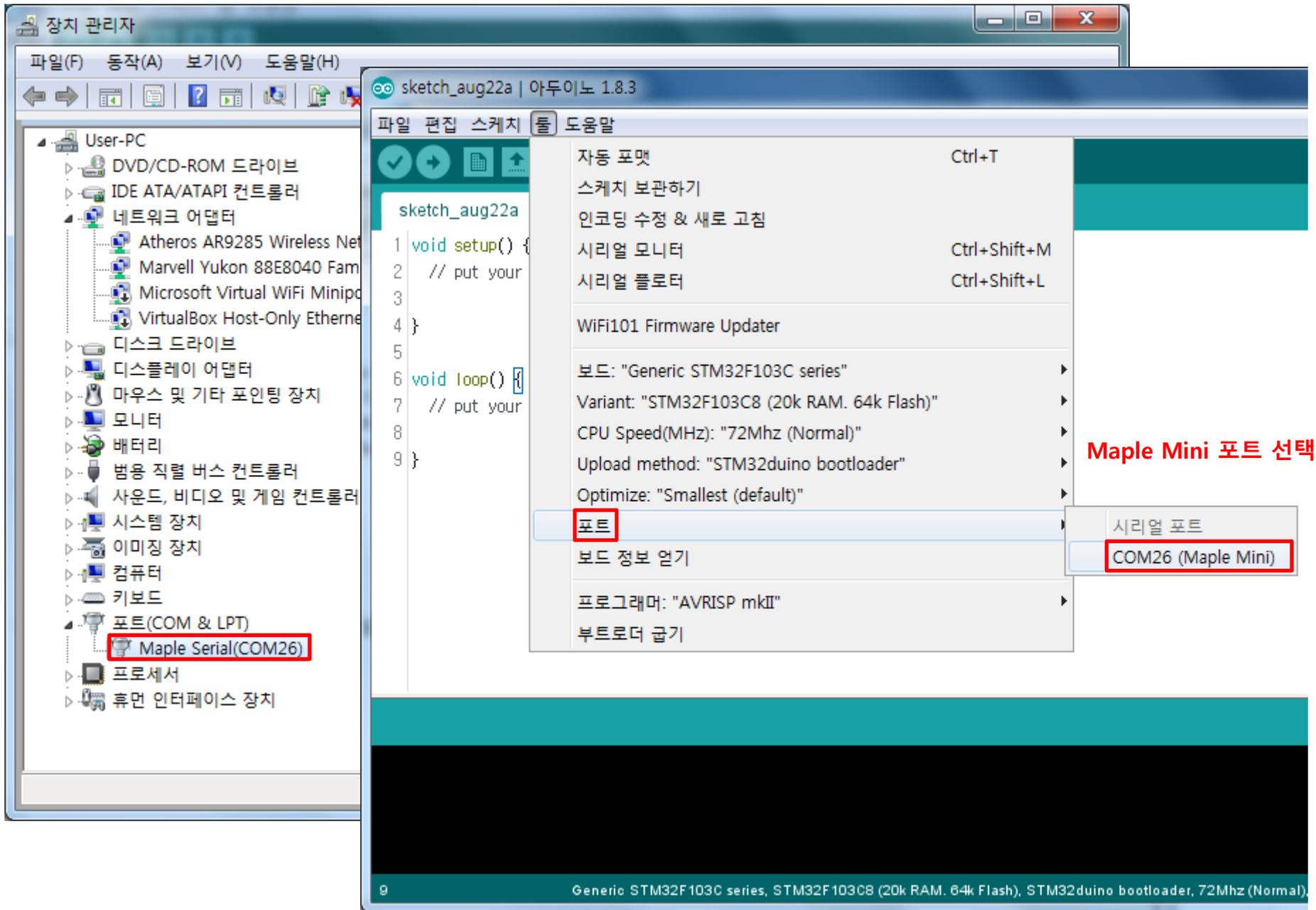
제목

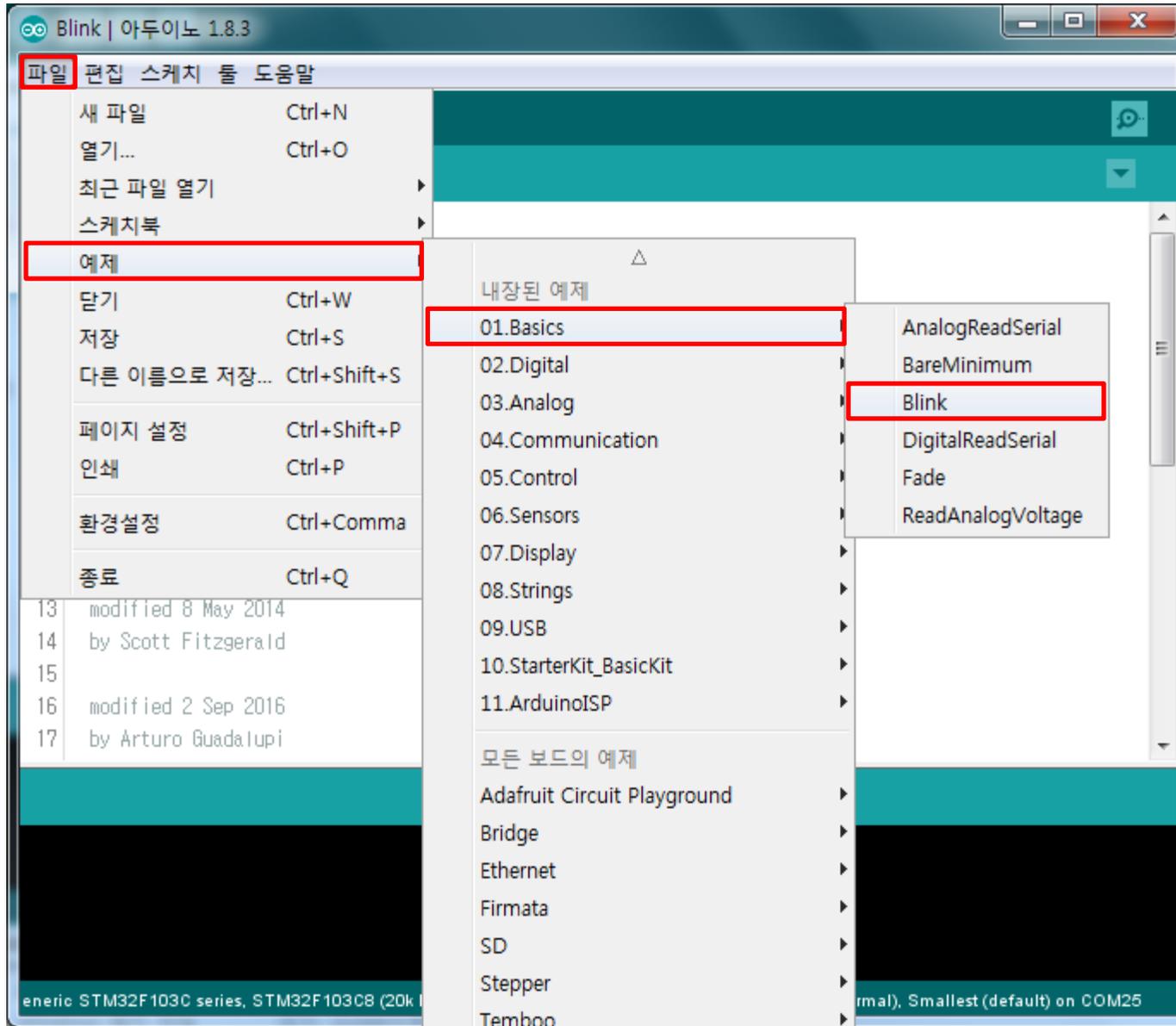
번호	제목	작성자	작성일
13	[라두이노 미니 자료] STM32 USB드라이버 <a href="#">📄</a>	디*유	2019-09-17
12	[라두이노 미니 자료] 라두이노 미니 플로우센서와라이다센서를 이용한 고도유지 및 호버링 <a href="#">📄</a>	디*유	2019-08-22
11	[라두이노 미니 자료] 라두이노 미니 BLE5.0 코드 및 apk <a href="#">📄</a>	디*유	2019-04-30
10	[라두이노 미니 강의자료] 라두이노 미니 BLE조종 <a href="#">📄</a>	디*유	2018-11-09





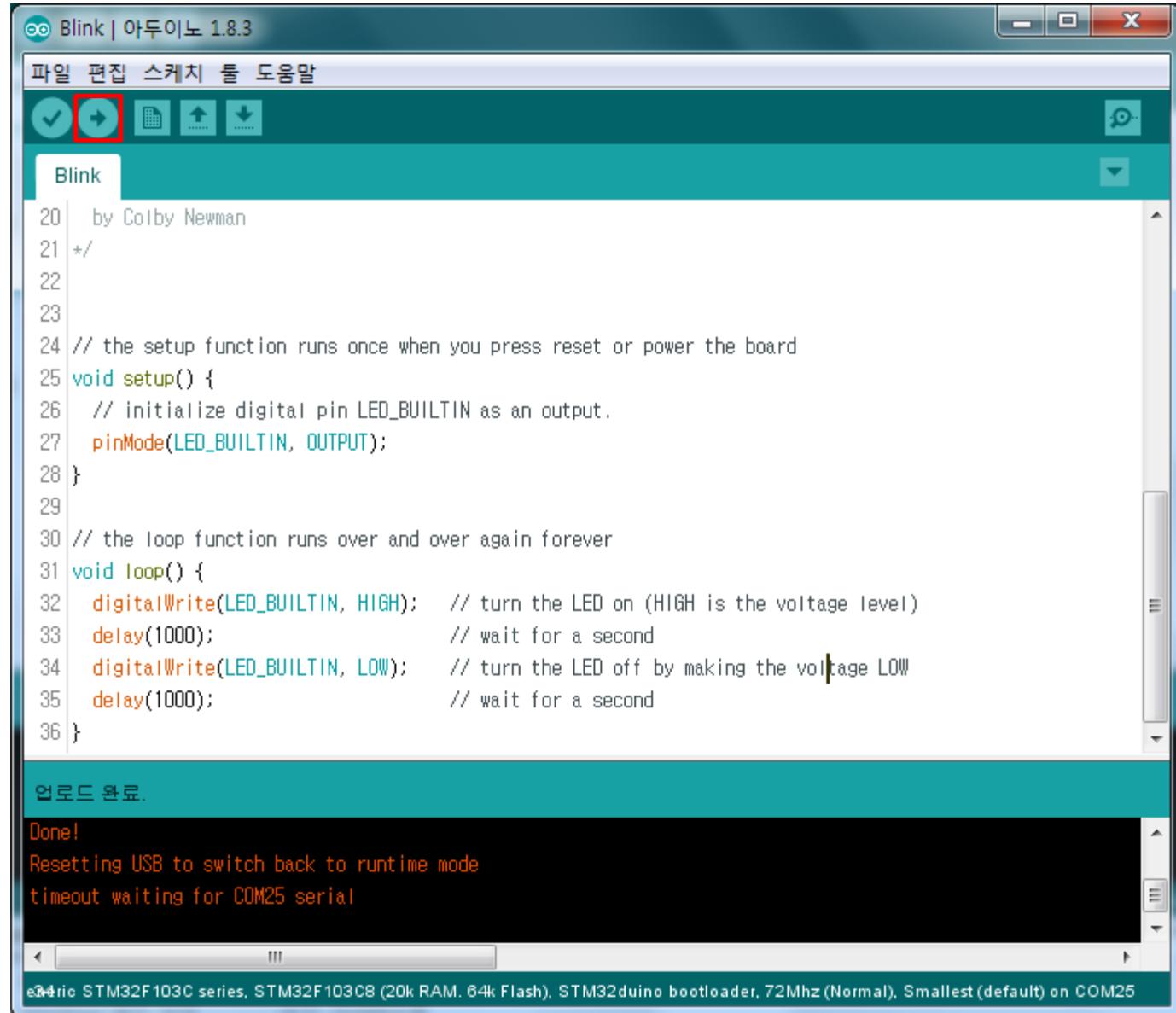
라두이노 미니의 보드 설정  
(Generic STM32F103C series)





아두이노 기본 예제 'Blink' 열기

## 아두이노 기본 예제 'Blink' 업로드



The screenshot shows the Arduino IDE window titled "Blink | 아두이노 1.8.3". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar contains icons for "확인" (check), "업로드" (upload, highlighted with a red box), "새 스케치" (new sketch), "열기" (open), and "저장" (save). The main editor displays the following code:

```
20 by Colby Newman
21 */
22
23
24 // the setup function runs once when you press reset or power the board
25 void setup() {
26   // initialize digital pin LED_BUILTIN as an output.
27   pinMode(LED_BUILTIN, OUTPUT);
28 }
29
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
33   delay(1000); // wait for a second
34   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
35   delay(1000); // wait for a second
36 }
```

Below the code editor, the status bar shows "업로드 완료." (Upload complete.). The serial monitor displays the following output:

```
Done!
Resetting USB to switch back to runtime mode
timeout waiting for COM25 serial
```

The status bar at the bottom indicates the board and port: "eM4ric STM32F103C series, STM32F103C8 (20k RAM, 64k Flash), STM32duino bootloader, 72Mhz (Normal), Smallest (default) on COM25".

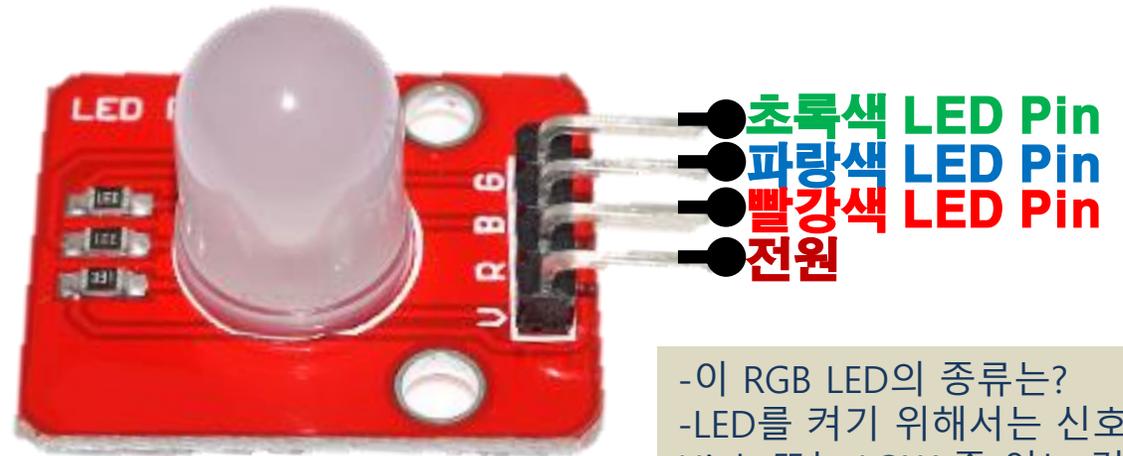
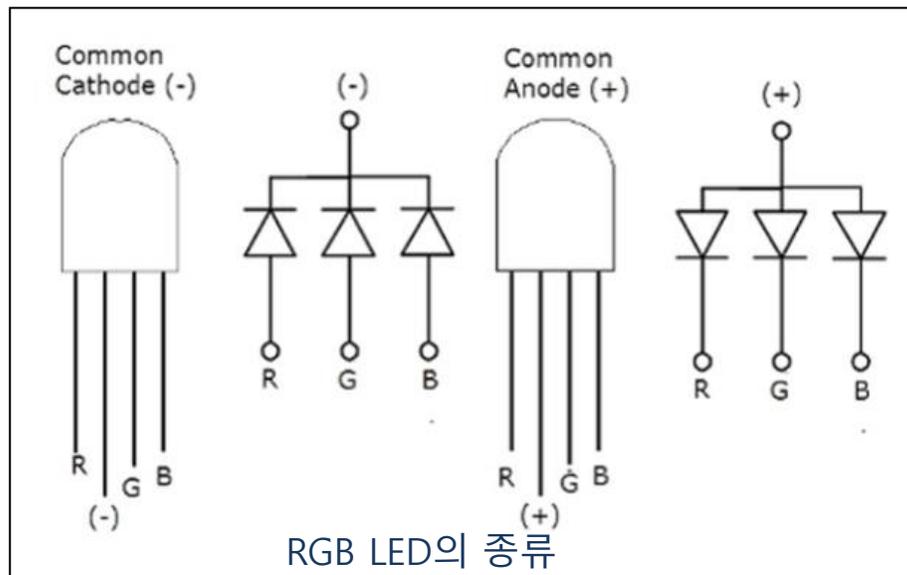
## 사운드 드론 만들기

- 준비물 : RGB LED, BUZZER
- 조정기 : RC 조정기

# RGB LED

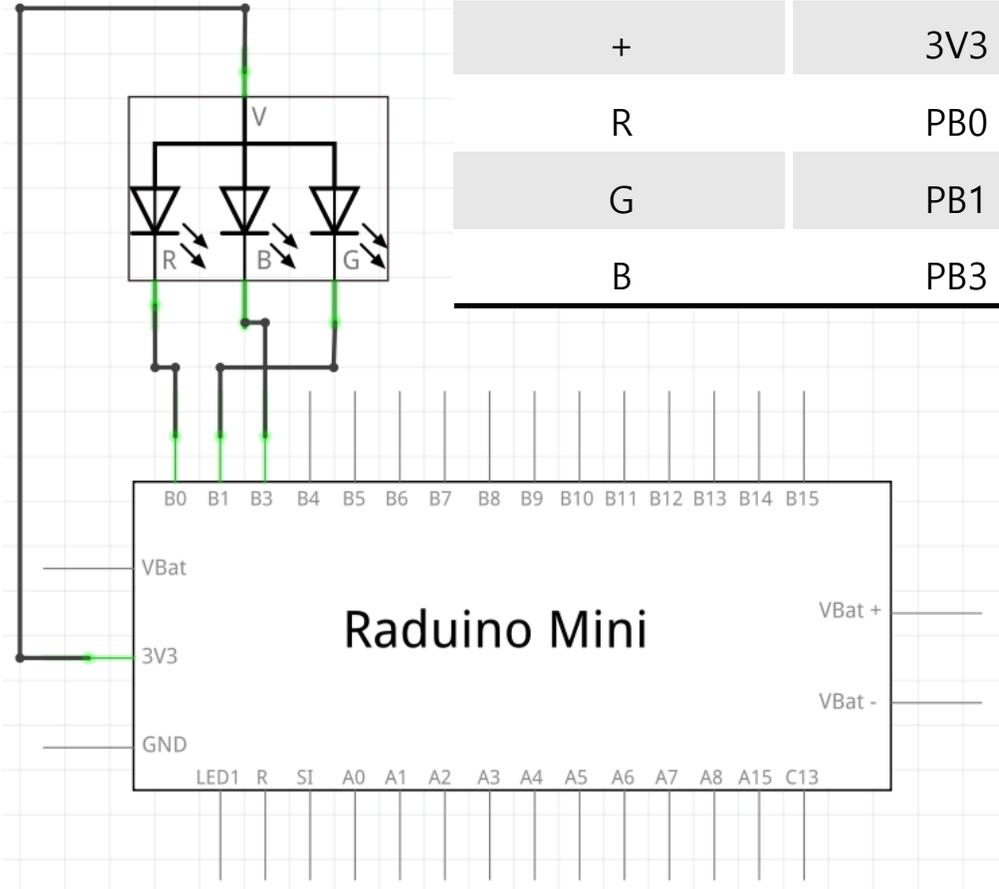
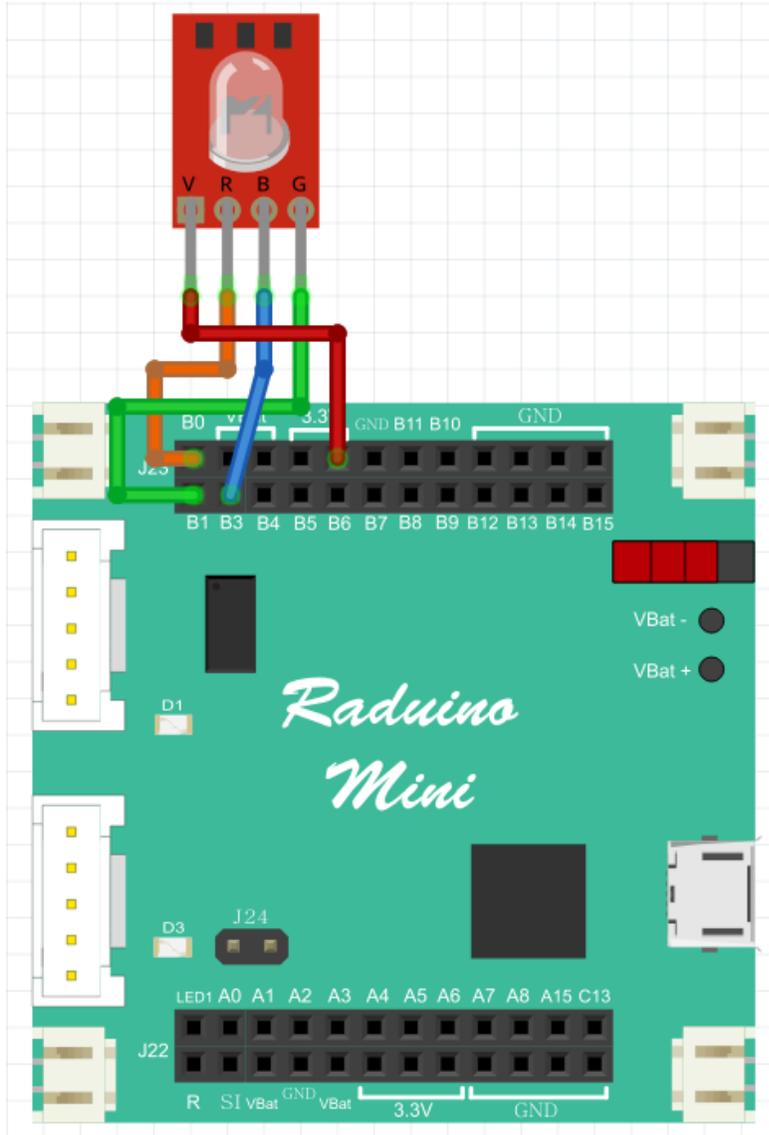
RGB LED 모듈은 특정한 한 가지 색을 내는 단색 LED 모듈과 다르게 일반 LED 3개를 한로 합친 LED가 들어있습니다. LED는 빨강색, 초록색, 파란색이 있으며 세가지 색을 조합하여 총 265가지의 색을 표현할 수 있다.

R(RED), G(Green), B(Blue), 1개의 공통 애노드(Common Anode) 또는 공통 캐소드(Common Cathode)의 4개의 리드선이 있다. 공통 애노드는 전원(VCC)부분이 단일 회선으로 연결하고, 공통 캐소드의 경우 그라운드(GND)가 연결된다.



-이 RGB LED의 종류는?  
-LED를 켜기 위해서는 신호단을 High 또는 LOW 중 어느 것?

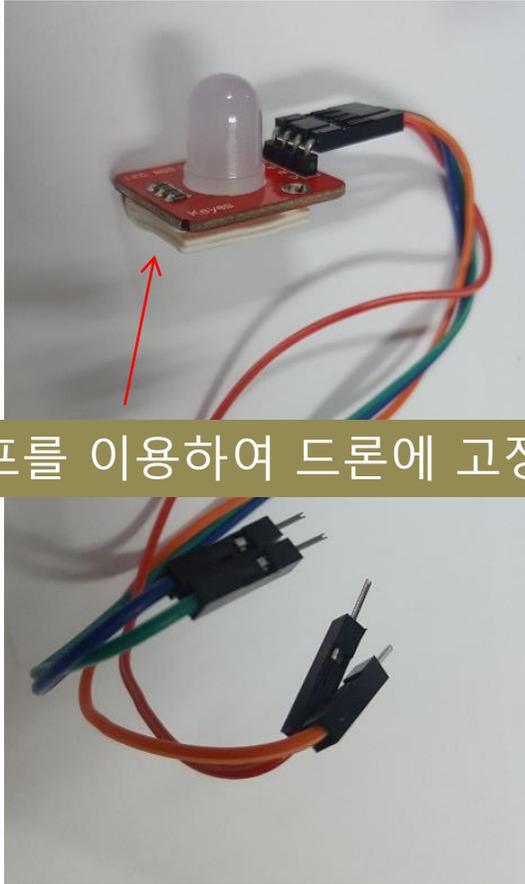
# 연결 회로도



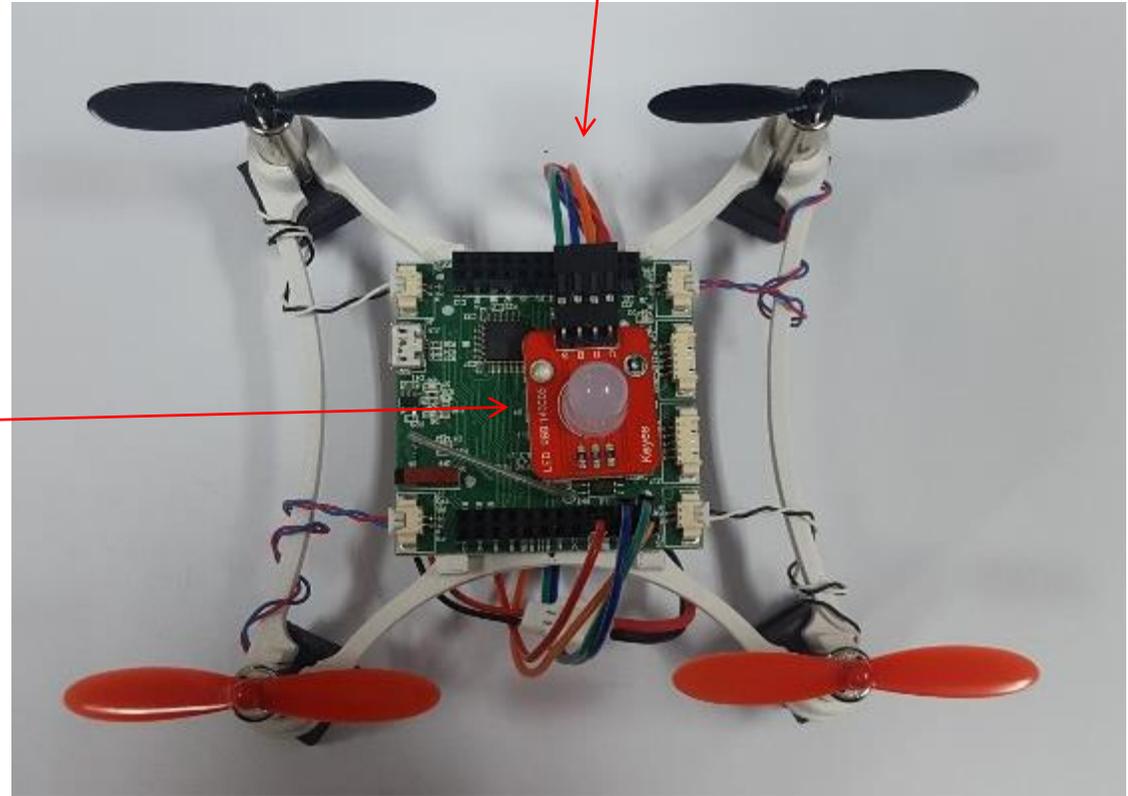
RGB LED Module	라두이노 미니
+	3V3
R	PB0
G	PB1
B	PB3

# 연결 하기

회로에 맞게 RGB LED 연결



양면테이프를 이용하여 드론에 고정하자



선이 프로펠러에게 걸리지 않도록 주의하자

# 스케치코드 EX1

```
#define RED PB0
#define GREEN PB1
#define BLUE PB3

void setup() // 실행 시 한번 실행
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
}
```

RGB LED Module	라두이노 미니
R	PB0
G	PB1
B	PB3

```
//RED(PB0)을 Digital Output 핀으로 설정
//GREEN(PB1)을 Digital Output 핀으로 설정
//BLUE(PB2)을 Digital Output 핀으로 설정
```

## pinMode() 함수

### 설명

특정 핀을 입력 또는 출력으로 동작하도록 설정합니다.

### 문법

```
pinMode(pin, mode)
```

### 매개변수

pin: 모드를 설정하려는 핀 번호

mode: INPUT, OUTPUT, 또는 INPUT\_PULLUP

```

void loop() // 무한 반복
{
  digitalWrite(RED, LOW); digitalWrite(GREEN, HIGH); digitalWrite(BLUE,HIGH); // RED
  delay(1000); //1000msec, 1초 기다리기
  digitalWrite(RED, HIGH); digitalWrite(GREEN, LOW); digitalWrite(BLUE,HIGH); // GREEN
  delay(1000);
  digitalWrite(RED, HIGH); digitalWrite(GREEN, HIGH); digitalWrite(BLUE,LOW); // BLUE
  delay(1000);
}

```

Loop() 함수에 다음 색을 추가 해보자

Color	RED	GREEN	BLUE
YELLOW	ON	ON	OFF
SKY BLUE	OFF	ON	ON
PINK	ON	OFF	ON
WHITE	ON	ON	ON

### digitalWrite() 함수

#### 설명

HIGH 또는 LOW 값을 디지털 핀에 출력합니다. pinMode() 를 OUTPUT 으로 설정하지 않고 LED를 핀에 연결하면, digitalWrite(HIGH) 가 불릴 때 LED 불빛이 흐리게 나타날 수 있습니다.

#### 문법

digitalWrite(pin, value)

#### 매개변수

pin: 핀 번호

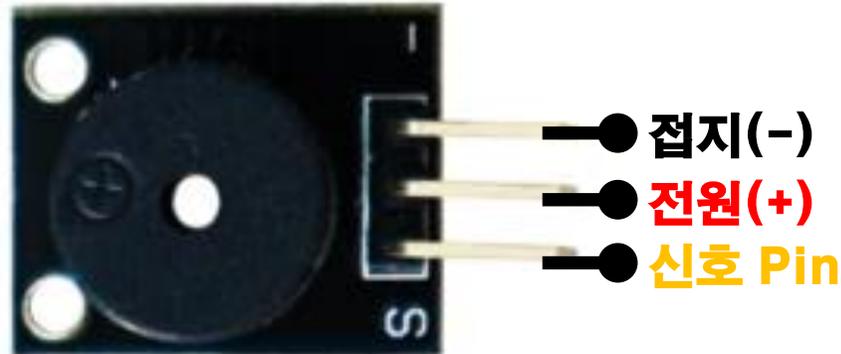
value: HIGH 또는 LOW

# BUZZER

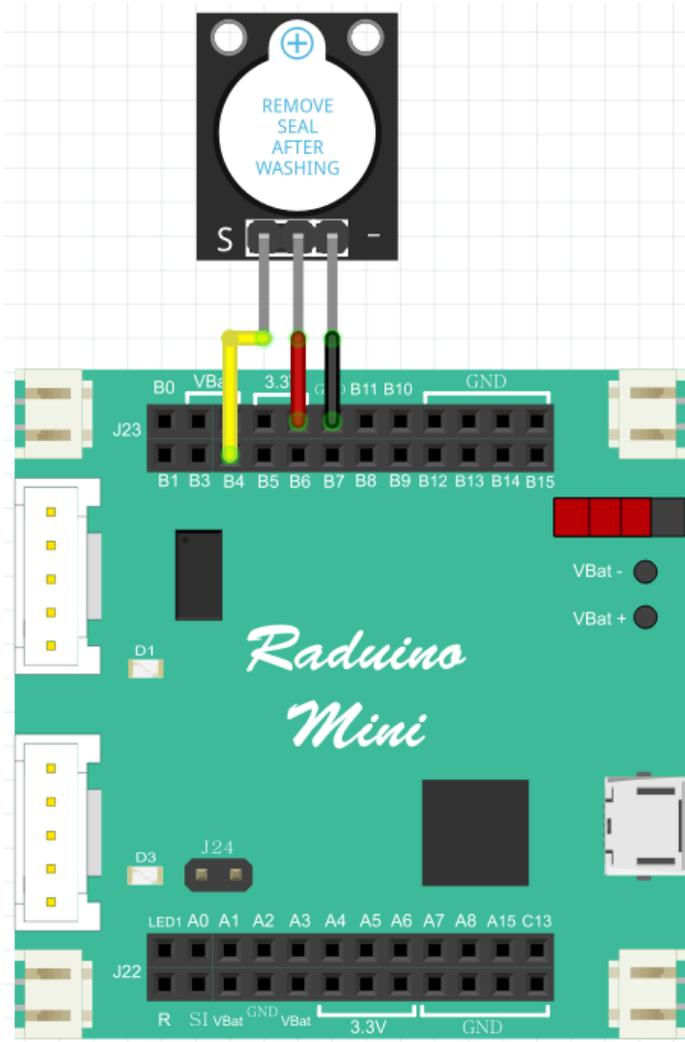
- 전원 입력만으로 소리가 나는 능동형 부저(Active buzzer)와 전원과 신호를 입력할 경우 소리가 나는 수동형 부저(Passive buzzer)가 있습니다.
- 수동형 부저는 PWM 신호를 이용해 주파수에 맞는 신호를 입력할 경우 간단한 배경음을 재생할 수도 있습니다.

## ❖ 사양

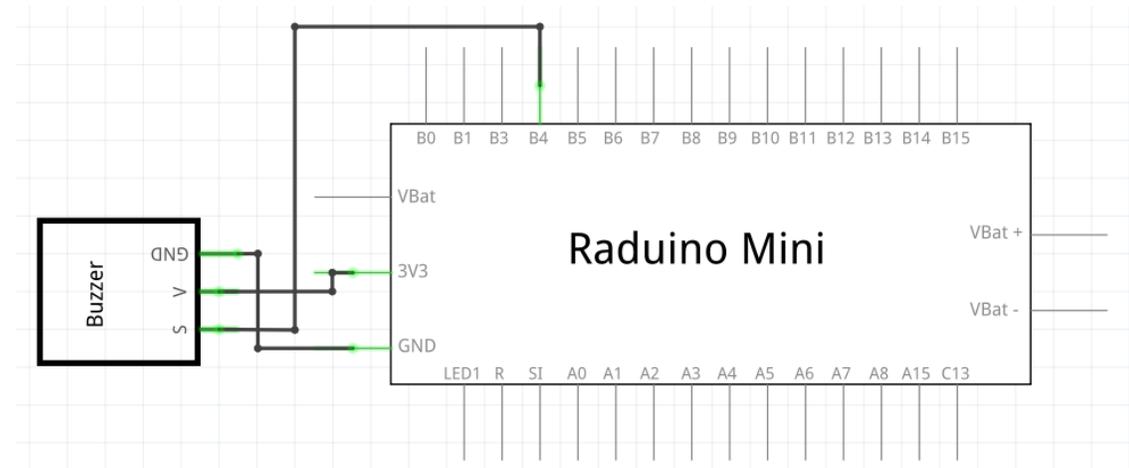
- 작동전압: 3.3 ~ 5V
- 작동전류: 25mA
- 용도: 호신용 경고음 장치, DIY 생일 카드 등
- 핀 구성: 3핀
- 크기: 15(L)x25(W)x10(H) mm
- 무게: 3g



# 연결 회로도



Passive Buzzer	라두이노 미니
+	3V3
-	GND
S	PB4

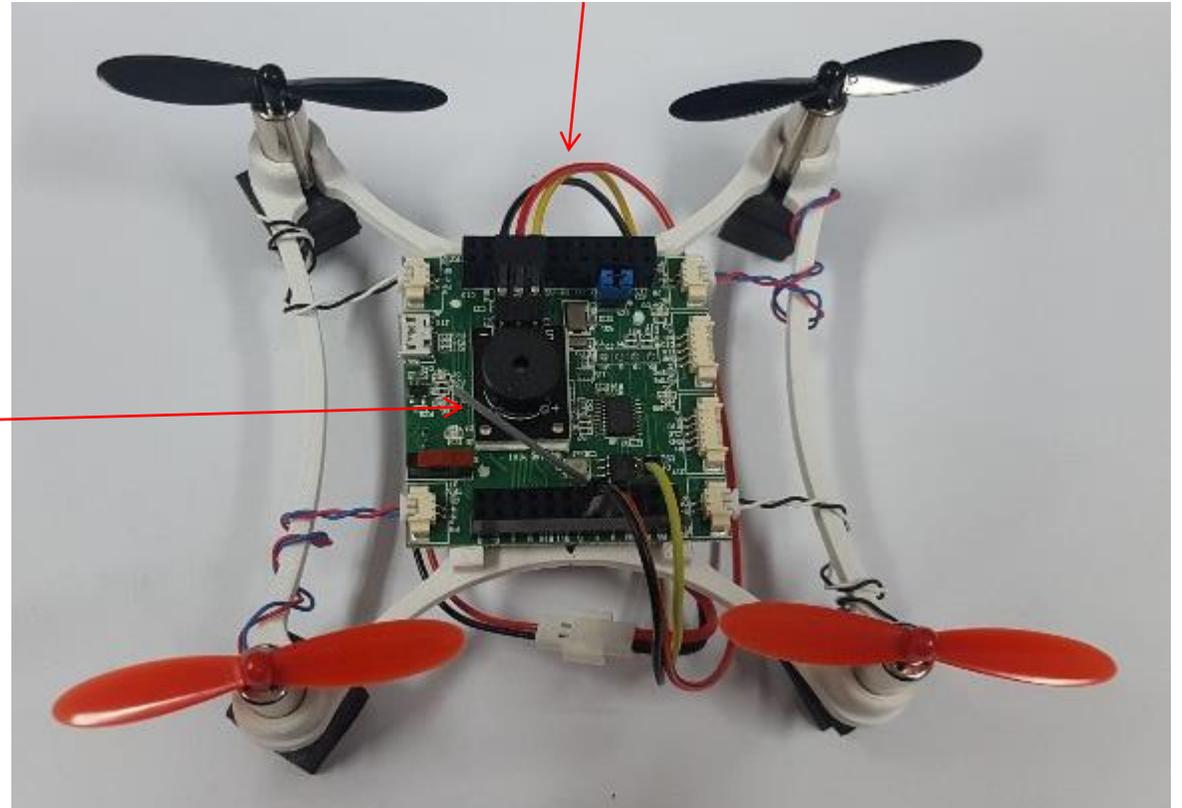


# 연결 하기

회로에 맞게 부저 연결



양면테이프를 이용하여 드론에 고정하자



선이 프로펠러에게 걸리지 않도록 주의하자

# 스케치코드 EX2

```
#define C4 262 //도
#define D4 294 //레
#define E4 330 //미
#define F4 349 //파
#define G4 392 //솔
#define A4 440 //라
#define B4 494 //시
```

이탈리아	영문
도	C
레	D
미	E
파	F
솔	G
라	A
시	B

```
#define SOUND PB4
```

Passive Buzzer	라두이노 미니
SOUND	PB4

```
int melody[] = {G4, E4, E4, F4, D4, D4, C4, D4, E4, F4, G4, G4, G4};
// 멜로디를 배열로 저장함
```

```
int noteDurations[] = {4, 4, 2, 4, 4, 2, 4, 4, 4, 4, 4, 4, 2}; //해당 음의 박자 수
```

**나비야**

♩ = 100

1                      2                      3                      4  
 나   비   야            나   비   야            이   리   날   아            오   너   라  
 솔   미   미            파   레   레            도   레   미   파            솔   솔   솔

문음표	
2분음표	
점2분음표	
4분음표	
점4분음표	
8분음표	
점8분음표	
16분음표	

```

void setup() {
  pinMode(SOUND, OUTPUT); //SOUND(PB4)을 Digital Output 핀으로 설정
}
void loop() {
  for (int thisNote = 0; thisNote < 13; thisNote++) {
    int noteDuration = 1800/noteDurations[thisNote]; // 박자 시간 변환
    tone(SOUND, melody[thisNote],noteDuration);
    int pauseBetweenNotes = noteDuration * 1.10 ; // 음 길이를 계산
    delay(pauseBetweenNotes); // 음 출력 중지
    noTone(SOUND);
  }
}

```

한 박자의 크기를 결정 한다.(1800msec)  
크기가 클 수록 한 박자의 크기가 크다

노래의 총 음표의 수 : 13개

음표와 음표 사이의 구분을 위해 1보다  
조금 크게 한다.

## tone() 함수

### 설명

핀에 특정 주파수(50% 듀티 사이클)의 구형파를 발생시킵니다. 지속 시간을 정할 수 있으며, 따로 정하지 않는다면 [noTone\(\)](#)을 부를 때까지 구형파가 지속됩니다.

### 문법

tone(pin, frequency)

tone(pin, frequency, duration)

### 매개변수

pin: tone을 발생시킬 핀

frequency: tone의 주파수 (Hz 단위) - unsigned int

duration (옵션) : tone의 지속 시간 (밀리초 단위) - unsigned long

# 연습문제

- 악보를 보고 노래를 완성하자
- 완성된 부저 예제와 RGB LED 예제를 합쳐보자

이탈리아	영문
도	C
레	D
미	E
파	F
솔	G
라	A
시	B

**나비야**

♩ = 100

1 C 2 G 3 C 4

나 비 야 나 비 야 이 리 날 아 오 너 라

5 C 6 G 7 C 8

노 랑 나 비 흰 나 비 춤 을 추 며 오 너 라

9 G 10 11 C 12

봄 바 람 에 꽃 잎 도 방긋 방긋 웃 으 며

13 C 14 G 15 C 16 G C

참 새 도 짹 짹 짹 노래 하 며 춤 춘 다

온음표	
2분음표	
점2분음표	
4분음표	
점4분음표	
8분음표	
점8분음표	
16분음표	

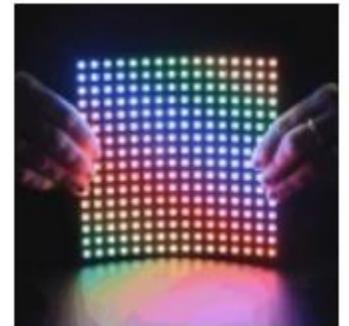
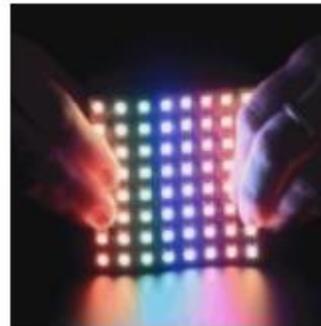
실행예제 유튜브 <https://youtu.be/UM6Q49FYFI0>

## 무지개 드론 만들기

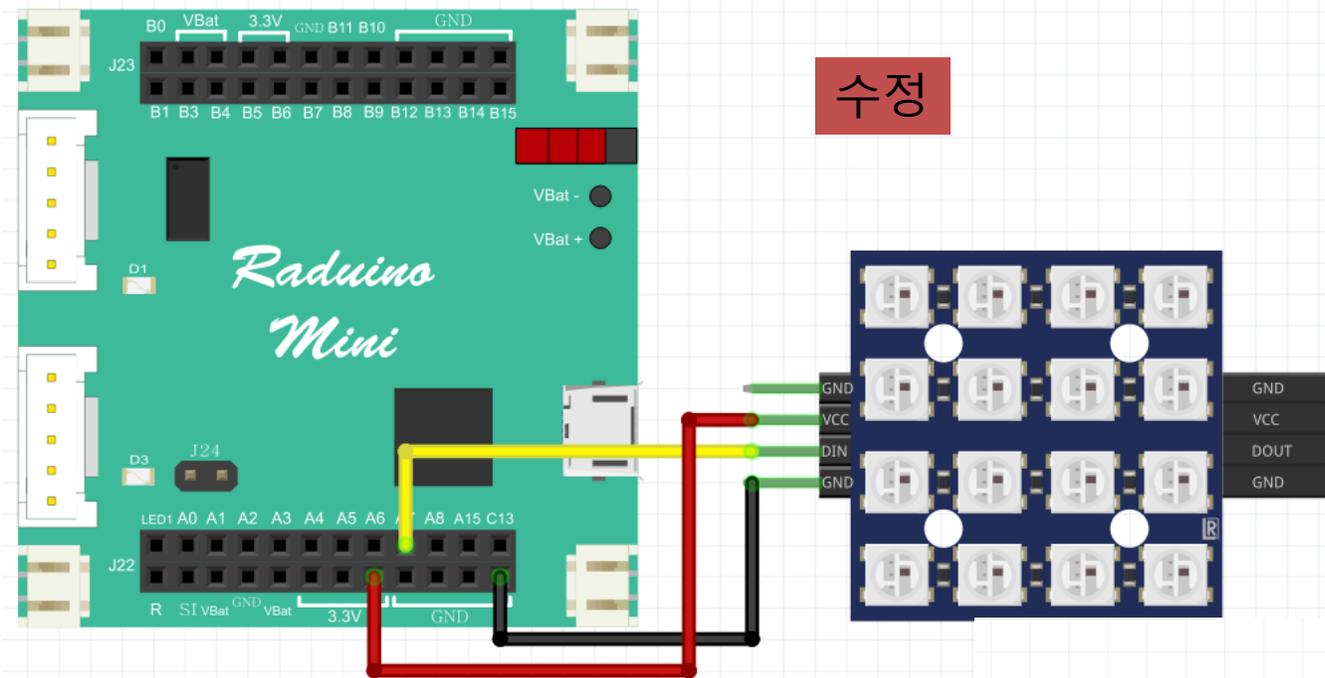
- 준비물 : 4X4 NEOPIXEL
- 조정기 : RC 조정기

# NeoPixel

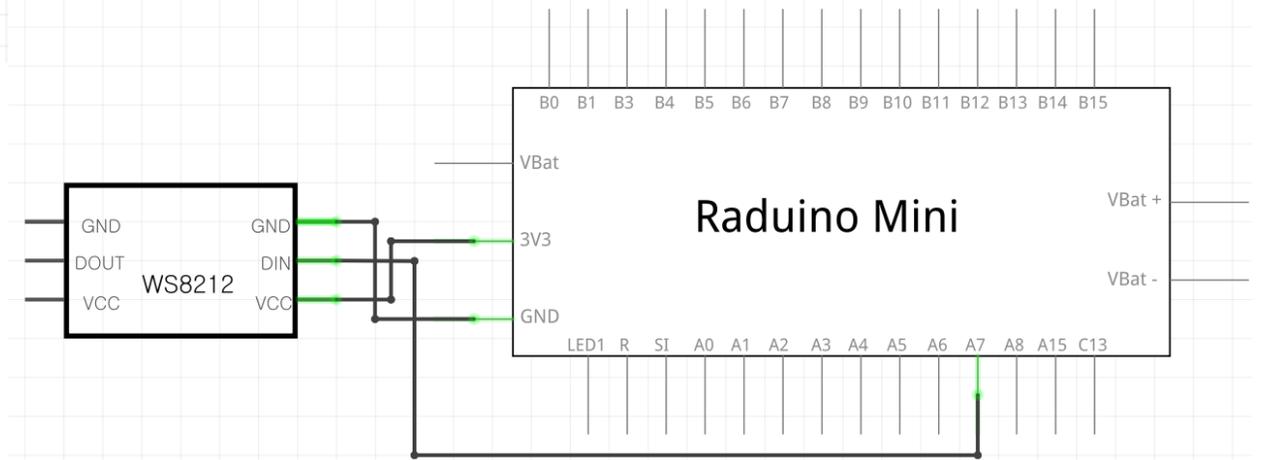
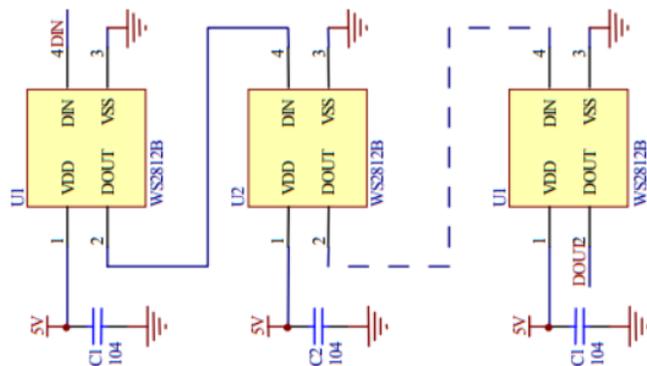
- Adafruit 회사에 만든 LED, 빛의 3원색인 R,G,B 색상을 단색별 또는 혼합하여 표현 수 있음.
- 장점: 회로 구성이 간단하며 네오피셀은 선 3개(VCC,DI,GND)만으로 수십개의 LED를 개별 제어 할 수 있는 강력한 기능 제공, 다양한 모양의 가공(Strip, Bar, Dot)
- 단점: 기본 RGB LED보다 가격이 비싸고, 제어가 필요



# 연결 회로도



네오 픽셀	라두이노 미니
VCC	3V3
DIN	PA7
GND	GND

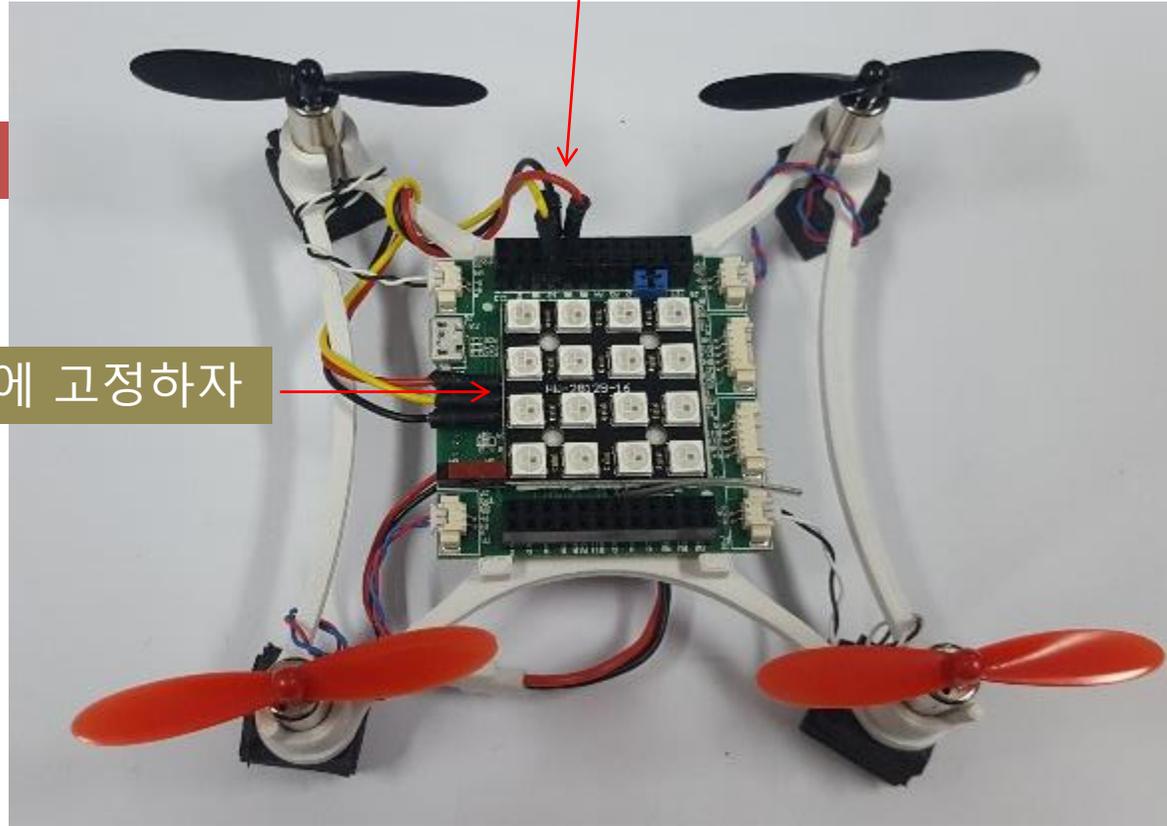


# 연결 하기

회로에 맞게 네오픽셀 연결

수정

양면테이프를 이용하여 드론에 고정하자



선이 프로펠러에게 걸리지 않도록 주의하자

# 스케치코드 EX3

```
#include "WS2812B.h"
```

```
#define NUM_LEDS 16
```

```
/*
```

```
* Note. Library uses SPI1
```

```
* Connect the WS2812B data input to PA7 on your board.
```

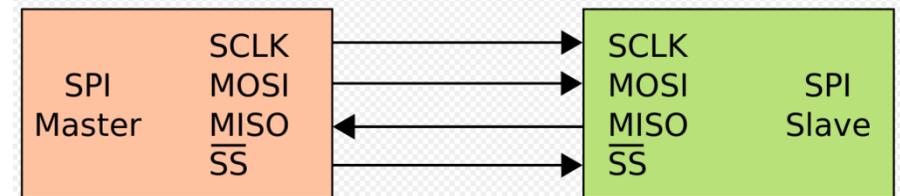
```
*
```

```
*/
```

```
WS2812B strip = WS2812B(NUM_LEDS);
```

레인보우 삭제 후 예제소스 업로드

**SPI**(Serial Peripheral Interface Bus) 는 장치간 동기화 직렬 데이터 연결 표준이다. 장치들은 마스터, 슬레이브 모드로 통신하며 여기서 마스터 장치는 데이터 프레임을 초기화한다. 여러 슬레이브 장치들은 개별 슬레이브 선택 (칩 선택) 라인과 함께 동작할 수 있다.



```
void setup()
```

```
{
```

```
  strip.begin();// Sets up the SPI
```

```
  strip.show();// Clears the strip, as by default the strip data is set to all LED's off.
```

```
}
```

```
void loop()
{
  colorWipe(strip.Color(0, 255, 0), 1); // Green
  delay(500);
  colorWipe(strip.Color(255, 0, 0), 1); // Red
  delay(500);
  colorWipe(strip.Color(0, 0, 255), 1); // Blue
  delay(500);
}
```

// Fill the dots one after the other with a color

```
void colorWipe(uint32_t c, uint8_t wait)
{
  for(uint16_t i=0; i<strip.numPixels(); i++)
  {
    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}
```

**colorWipe()**는 표현할 색, 네오픽셀 간 딜레이를 변수로 받는다  
·  
**colorWipe(strip.Color(255,0,0), 1)**이라하면 빨간색을 각 LED가 칠해지는 간격을 1ms로 하여 표현한다는 의미이다.  
**strip.Color(R,G,B)**에서는 인자값으로 0부터 255사이의 숫자를 3개 받는데 각각 R,G,B값을 나타낸다. 그 색을 표현하는 값이 커질수록 그 색의 밝기는 점점 밝아진다.

# 연습문제

-loop() 함수 안에 다음 코드를 추가해보고 실행 해 보자

```
//RAINBOW-----  
colorWipe(strip.Color(0, 255, 0), 20); // Green  
colorWipe(strip.Color(255, 0, 0), 20); // Red  
colorWipe(strip.Color(0, 0, 255), 20); // Blue  
rainbow(10);  
rainbowCycle(10);  
whiteOverRainbow(20,75,5);  
//-----
```

실행예제 유튜브

<https://youtu.be/2pJ2-YEC28U>

## 드론 조정 앱(APP) 만들기

- 준비물 : BLE 모듈  
스마트폰(안드로이드)
- 프로그램 : MIT 앱 인벤터

# MIT 앱 인벤터

Designer 파트 : <https://youtu.be/lt-fVRZYA0>

Blocks 파트 : <https://youtu.be/vp9eMIITDWc>

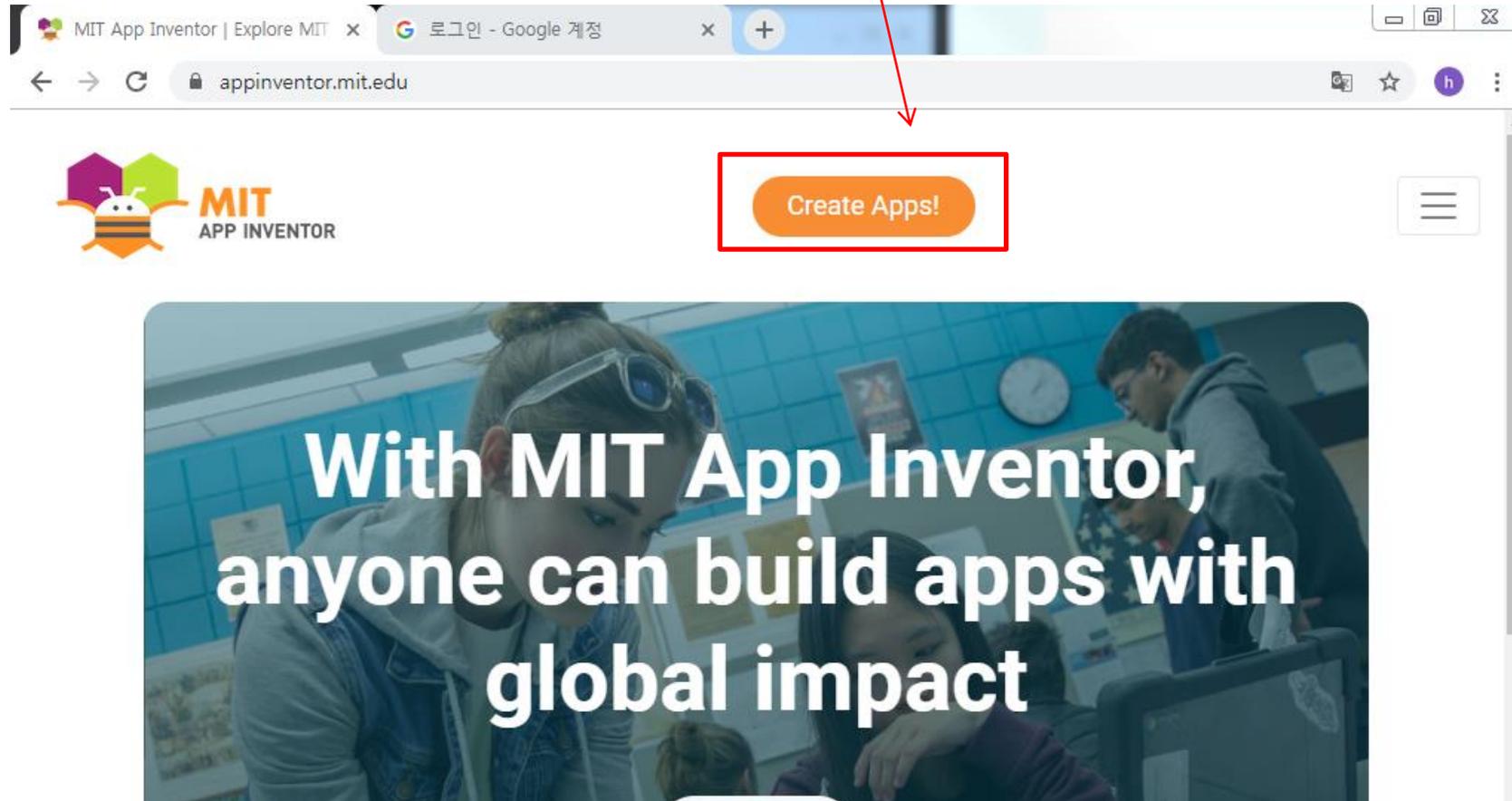
TaifunTools 다운로드: <http://puravidaapps.com/tools.php>

BLE 다운로드 : <https://mit-cml.github.io/extensions/>

# MIT 앱 인벤터

구글의 크롬(Chrome)을 사용. 구글 계정으로 로그인 후

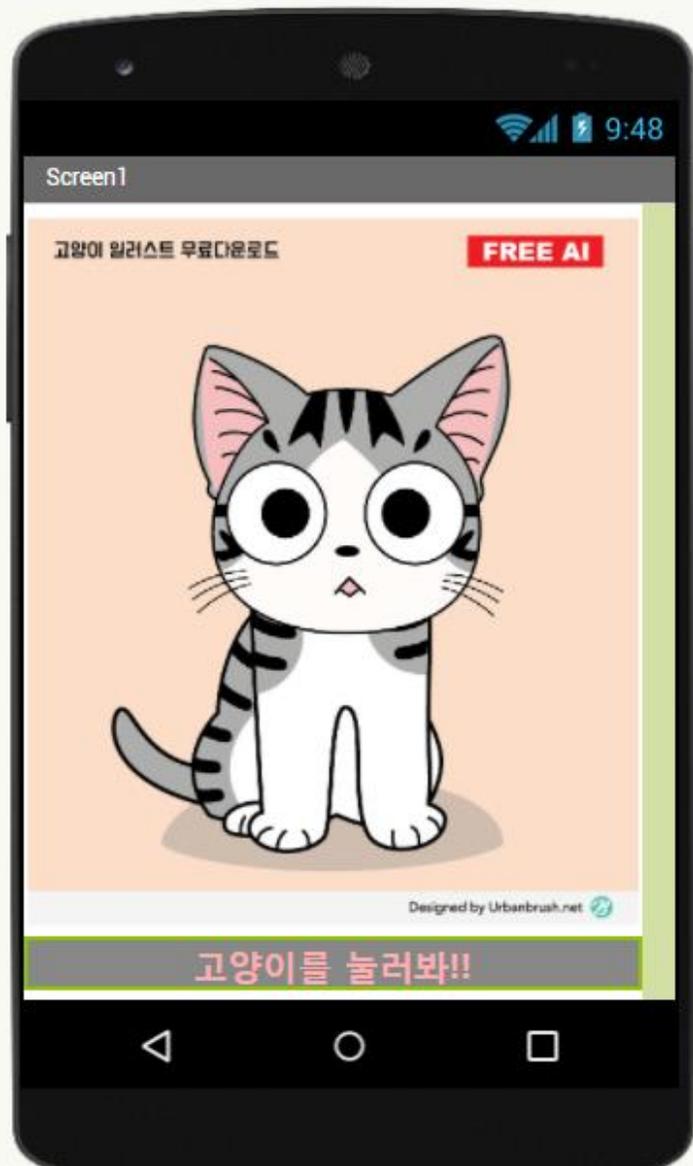
<http://appinventor.mit.edu> 으로 접속한 후 'Create apps!' 버튼 선택



**고양이 앱**

뷰어에 숨겨진 컴포넌트 나타내기

전화 크기 (505,320) ▾



보이지 않는 컴포넌트



- Screen1
  - 고양이버튼
  - 메시지레이블
  - 고양이소리

이름 바꾸기

삭제

Cat.mp3

Cat.png

파일 올리기 ...

버튼	고양이버튼	높이	부모 요소에 맞추기
		너비	부모 요소에 맞추기
		이미지	자동
		텍스트 정렬	가운데 : 1
메시지레이블		글꼴크기	20
		높이	자동
		너비	부모 요소에 맞추기
		텍스트	고양이를 눌러봐!!
소리	고양이소리	텍스트 정렬	가운데 : 1
		최소간격	500
		소스	Cat.mp3

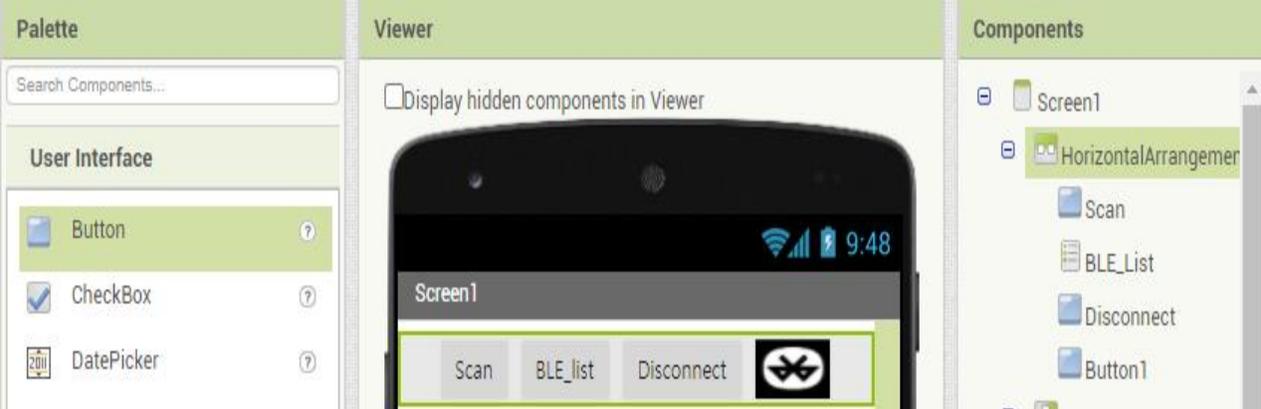
언제 고양이버튼 ▾ .클릭했을때

실행 호출 고양이소리 ▾ .재생하기

호출 고양이소리 ▾ .진동하기

밀리초 300

**라두이노 조정기 앱**



이름	이름수정	속성	속성 수정
수평배치		수평 정렬	가운데 : 3
		수직 정렬	가운데 : 2
		높이	자동
		너비	부모 요소에 맞추기
버튼	Scan	글꼴크기	14
		높이	자동
		너비	자동
		텍스트	Scan
		텍스트 정렬	가운데 : 1
목록선택버튼	BLE_List	글꼴크기	14
		높이	자동
		너비	자동
		텍스트	BLE_list
		텍스트 정렬	가운데 : 1
버튼	Disconnect	글꼴크기	14
		높이	자동
		너비	자동
		텍스트	Disconnect
		텍스트 정렬	가운데 : 1
버튼	Button1	글꼴크기	14
		높이	30 픽셀
		너비	50 픽셀
		이미지	BT_Black
		텍스트	
		텍스트 정렬	가운데 : 1



이름	이름수정	속성	속성 수정
수직배치		수평 정렬	가운데 : 3
		수직 정렬	가운데 : 2
		높이	자동
		너비	부모 요소에 맞추기
슬라이더	Yaw_슬라이더	왼쪽색상	기본값
		오른쪽색상	기본값
		너비	80 퍼센트
		최댓값	1024
		최솟값	0
		섬네일 위치	512



이름	이름수정	속성	속성 수정
수직배치		수평 정렬	가운데 : 3
		수직 정렬	가운데 : 2
		높이	부모 요소에 맞추기
		너비	부모 요소에 맞추기
슬라이더	Roll_슬라이더	왼쪽색상	파랑
		오른쪽색상	파랑
		너비	부모 요소에 맞추기
		최댓값	1024
		최솟값	0
		섬네일 위치	512
		슬라이더	Pitch_슬라이더
		오른쪽색상	파랑
		너비	부모 요소에 맞추기
		최댓값	1024
		최솟값	0
		섬네일 위치	512

**팔레트**

Search Components...

**사용자 인터페이스**

- 버튼
- 체크박스
- 날짜선택버튼
- 이미지
- 레이블
- 목록선택버튼
- 목록뷰
- 알림
- 암호텍스트박스
- 슬라이더
- 스피너
- 스위치
- 텍스트박스

**뷰어**

뷰어에 숨겨진 컴포넌트 나타내기

**컴포넌트**

- Screen1
  - HorizontalArrangemer
    - Scan
    - BLE\_List
    - Disconnect
    - Button1
  - VerticalArrangement1
    - Yaw\_슬라이드
  - VerticalArrangement3
    - Roll\_슬라이더
    - Pitch\_슬라이더
  - VerticalArrangement5
    - Thro\_슬라이더
  - HorizontalArrangemer
    - Button2
    - HorizontalArrangen
    - Button3

이름	이름수정	속성	속성 수정
수직배치		수평 정렬	가운데 : 3
		수직 정렬	가운데 : 2
		높이	자동
		너비	부모 요소에 맞추기
슬라이더	Thro_슬라이더	왼쪽색상	기본값
		오른쪽색상	기본값
		너비	80 퍼센트
		최댓값	1024
		최솟값	0
		섬네일 위치	0

팔레트

Search Components...

사용자 인터페이스

- 버튼
- 체크박스
- 날짜선택버튼
- 이미지
- 레이블
- 목록선택버튼
- 목록뷰
- 알림
- 암호텍스트박스
- 슬라이더
- 스피너
- 스위치
- 텍스트박스
- 시간선택버튼

뷰어

뷰어에 숨겨진 컴포넌트 나타내기

Screen1

Scan BLE\_list Disconnect

C B A

컴포넌트

- Scan
- BLE\_List
- Disconnect
- Button1
- VerticalArrangement1
- Yaw\_슬라이더
- VerticalArrangement3
- Roll\_슬라이더
- Pitch\_슬라이더
- VerticalArrangement5
- Thro\_슬라이더
- HorizontalArrangemr
- Button2
- HorizontalArrangen
- Button3
- HorizontalArrangen
- Button4

이름 바꾸기 삭제

이름	이름수정	속성	속성 수정
수평배치		수평 정렬	가운데 : 3
		수직 정렬	가운데 : 2
		높이	자동
		너비	부모 요소에 맞추기
버튼	버튼C	글꼴크기	14
		높이	40 픽셀
		너비	40 픽셀
		텍스트 정렬	가운데 : 1
수평배치		수평 정렬	왼쪽 : 1
		수직 정렬	위 : 1
		높이	40 픽셀
		너비	40 픽셀
버튼	버튼B	글꼴크기	14
		높이	40 픽셀
		너비	40 픽셀
		텍스트 정렬	가운데 : 1
수평배치		수평 정렬	왼쪽 : 1
		수직 정렬	위 : 1
		높이	40 픽셀
		너비	40 픽셀
버튼	버튼A	글꼴크기	14
		높이	40 픽셀
		너비	40 픽셀
		텍스트 정렬	가운데 : 1

팔레트

Search Components...

사용자 인터페이스

- 버튼
- 체크박스
- 날짜선택버튼
- 이미지
- 레이블
- 목록선택버튼
- 목록뷰
- 알림
- 암호텍스트박스
- 슬라이더
- 스피너
- 스위치
- 텍스트박스
- 시간선택버튼
- 웹뷰어

레이아웃

미디어

그리기 & 애니메이션

지도

뷰어

뷰어에 숨겨진 컴포넌트 나타내기

Screen1

Scan BLE\_list Disconnect

Yaw\_슬라이더

Roll\_슬라이더

Pitch\_슬라이더

C B A

보이지 않는 컴포넌트

TaifunTools1 BluetoothLE1 Notifier1 AccelerometerSensor1 Clock1

컴포넌트

- Yaw\_슬라이더
- VerticalArrangement3
- Roll\_슬라이더
- Pitch\_슬라이더
- VerticalArrangement5
- Thro\_슬라이더
- HorizontalArrangement
- 버튼C
- HorizontalArrangement
- 버튼B
- HorizontalArrangement
- 버튼A
- TaifunTools1
- BluetoothLE1
- Notifier1
- AccelerometerSensor1
- Clock1

이름 바꾸기 삭제

미디어

- BT\_Black.png
- BT\_Blue.png

파일 올리기 ...

이름	속성	속성 수정
TaifunTools1		
BluetoothLE1	ConnectionTimeout	10
알림		
가속도 센서	최소간격	100
시계	타이머간격	100

# BLE관련 블럭

전역변수 만들기 device 초기값

전역변수 만들기 SERVICE\_UUID 초기값 " 6E400001-B5A3-F393-E0A9-E50E24DCCA9E "

전역변수 만들기 CHARACTERISTIC\_UUID 초기값 " 6E400002-B5A3-F393-E0A9-E50E24DCCA9E "

언제 Scan 클릭했을때

실행 호출 BluetoothLE1 .StartScanning

언제 BluetoothLE1 .DeviceFound

실행 지정하기 BLE\_List 요소문자열 값 BluetoothLE1 DeviceList

언제 BLE\_List 선택후에

실행 호출 BluetoothLE1 .StopScanning

지정하기 전역변수 device 값 텍스트에서 문자열 추출하기 텍스트 BLE\_List 선택된항목  
시작위치 1  
문자열길이 17

호출 BluetoothLE1 .ConnectWithAddress  
주소 가져오기 전역변수 device

지정하기 Yaw\_슬라이드 섬네일위치 값 512

지정하기 Thro\_슬라이더 섬네일위치 값 0

언제 BluetoothLE1 .Connected

실행 호출 BluetoothLE1 .WriteStrings  
serviceUuid 가져오기 전역변수 SERVICE\_UUID  
characteristicUuid 가져오기 전역변수 CHARACTERISTIC\_UUID  
utf16 참

값 " BIND123456Wn "

지정하기 Button1 이미지 값 " BT\_Blue.png "

BLE연결을 위한 변수선언  
Service UUID, Characteristic UUID 설정

Scan 버튼을 클릭하면 주변 BLE장치 스캔

BLE\_List에 스캔 된 BLE장치 표시

BLE\_List에서 스캔 된 장치 중 한가지 클릭 시  
BLE장치 스캔 중지  
'device'변수에 선택된 BLE주소저장 및 연결

Yaw\_슬라이더, Thro\_슬라이더 지정된 값으로 초기화

BLE가 연결된 경우  
연결된 BLE에 "BIND123456Wn" 문자열 전송

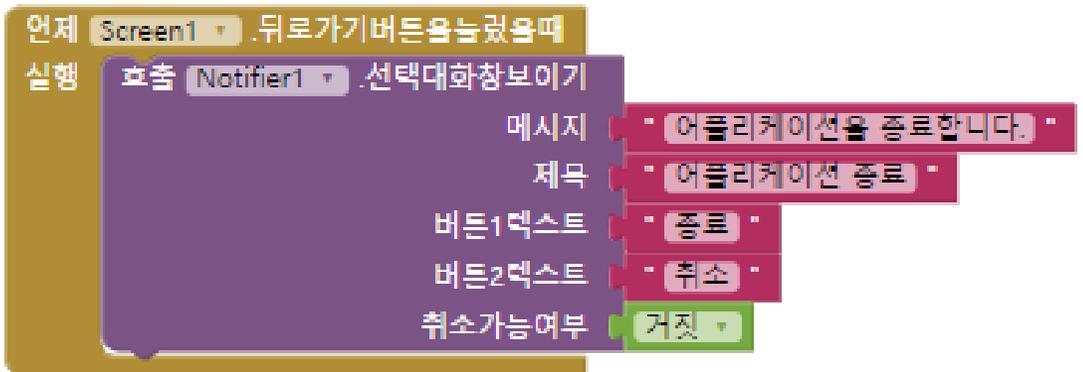
Button1에 들어간 이미지 변경

Disconnect버튼 클릭 시 BLE연결종료  
Button1에 들어간 이미지 변경



```
언제 Disconnect .클릭했을때
실행 호출 BluetoothLE1 .연결끊기
지정하기 Button1 . 이미지 값 " BT_Black.png "
```

어플리케이션 종료 관련 블럭



```
언제 Screen1 .뒤로가기버튼을눌렀을때
실행 호출 Notifier1 .선택대화창보이기
    메시지 " 어플리케이션을 종료합니다. "
    제목 " 어플리케이션 종료 "
    버튼1텍스트 " 종료 "
    버튼2텍스트 " 취소 "
    취소가능여부 거짓
```

뒤로 가기 버튼을 눌렀을 경우

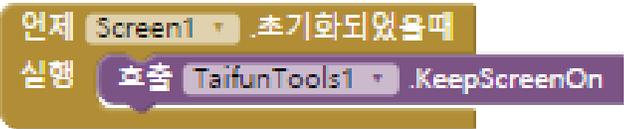
"어플리케이션을 종료합니다!" 라는 메시지와 함께  
'종료' , '취소' 버튼 표시



```
언제 Notifier1 .선택한후에
선택된항목
실행 만약
    텍스트 비교하기 가져오기 선택된항목 = " 종료 "
    이라면 실행 앱 종료
```

'종료' 버튼을 누르면 어플 종료

화면이 꺼지지않게 해주는 블럭



```
언제 Screen1 .초기화되었을때
실행 호출 TaifunTools1 .KeepScreenOn
```

전역변수 만들기 Roll 초기값 512

전역변수 만들기 Pitch 초기값 512

전역변수 만들기 Thro 초기값 0

전역변수 만들기 Yaw 초기값 512

전역변수 만들기 X가속도 초기값 0

전역변수 만들기 Y가속도 초기값 0

전역변수 만들기 데이터길이 초기값 0

전역변수 만들기 버튼 초기값 0

언제 Yaw\_슬라이더 . 위치가변경되었을때

섬네일위치

실행 지정하기 전역변수 Yaw 값 내림 가져오기 섬네일위치

언제 Thro\_슬라이더 . 위치가변경되었을때

섬네일위치

실행 지정하기 전역변수 Thro 값 내림 가져오기 섬네일위치

언제 버튼A . 클릭했을때

실행 지정하기 전역변수 버튼 값 1

언제 버튼B . 클릭했을때

실행 지정하기 전역변수 버튼 값 2

언제 버튼C . 클릭했을때

실행 지정하기 전역변수 버튼 값 3

FC에 전송될 각 변수의 초기값 설정

X축, Y축 각가속도의 초기값과 전송될 데이터의 길이,  
버튼의 상태 값 초기 설정

Yaw\_슬라이더, Thro\_슬라이더 변수에 각각의 Slider의  
움직임에 따른 값 저장

각 버튼이 클릭되었을 때의 값을 '버튼' 변수에 저장

```

언제 Clock1 .타이머가작동할때
실행
지정하기 전역변수 X가속도 값 AccelerometerSensor1 . X가속도
지정하기 전역변수 Y가속도 값 AccelerometerSensor1 . Y가속도
지정하기 전역변수 Roll 값 1024 - 내림 * 가져오기 전역변수 X가속도 * 50 + 500
지정하기 전역변수 Roll 값 내림 * 가져오기 전역변수 Y가속도 * 50 + 500

```

X가속도, Y가속도 변수에 센서 값 저장  
이 후 변수 값이 0~1024값을 갖도록 변환 후  
Roll, Pitch 변수에 저장

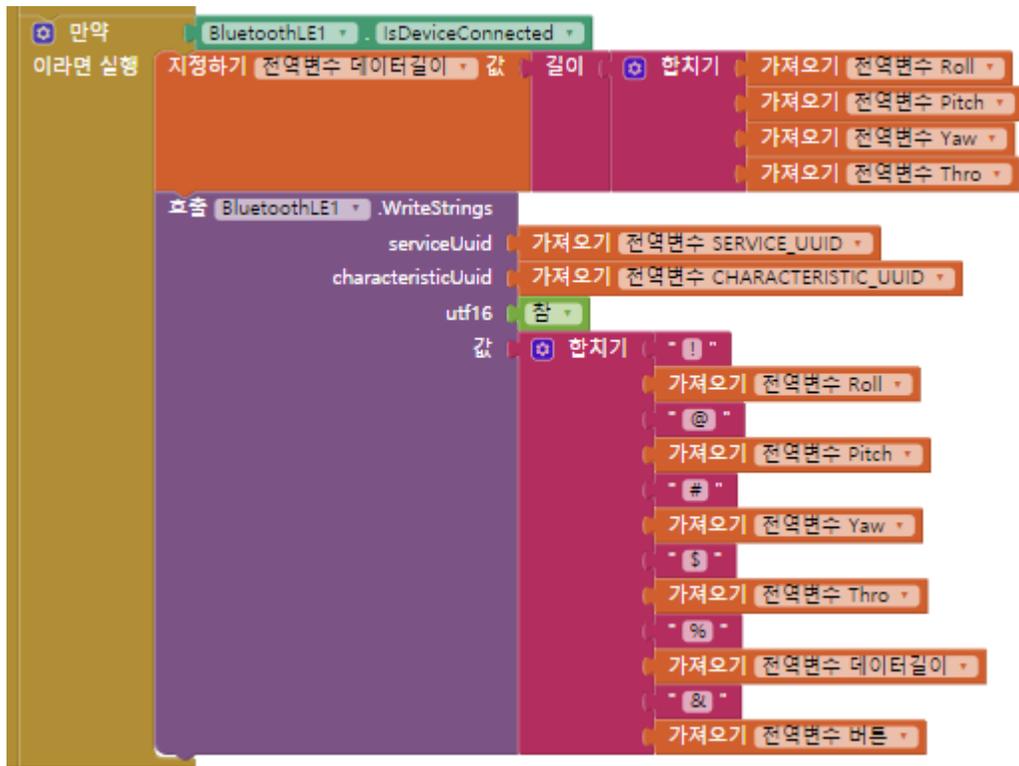
```

만약 가져오기 전역변수 Pitch < 0
이러면 실행 지정하기 전역변수 Pitch 값 0
만약 가져오기 전역변수 Roll < 0
이러면 실행 지정하기 전역변수 Roll 값 0
만약 가져오기 전역변수 Roll ≤ 562 그리고 가져오기 전역변수 Roll ≥ 462
이러면 실행 지정하기 전역변수 Roll 값 512
만약 가져오기 전역변수 Pitch ≤ 562 그리고 가져오기 전역변수 Pitch ≥ 462
이러면 실행 지정하기 전역변수 Pitch 값 512

```

저장된 변수가 -값이 나오지 않도록 범위를 지정해주고

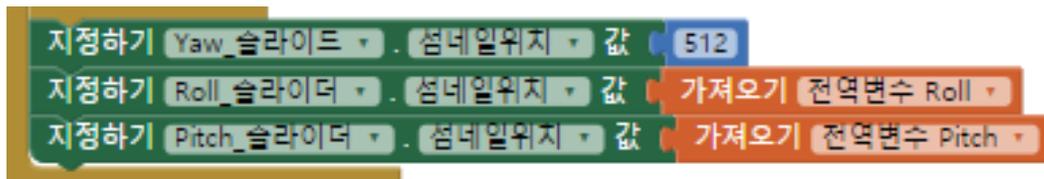
기준이 되는 수평상태의 값 512에서 앞,뒤,좌,우로 50씩  
변하는 값은 무시할 수 있도록 범위 설정



BLE가 연결된 경우 동작

Ch1~ch4 까지 4가지 변수의 길이 합을 lengh\_TX변수에 저장

이후 ch1~ch4까지의 값과 변수 길이 합을 나타내는 lengh\_TX, 버튼 상태값을 저장하는 변수인 Button을 하나의 문자열로 전송



Yaw\_슬라이더 위치는 항상 중앙에 위치할 수 있도록 값(512) 설정  
Roll\_슬라이더, Pitch\_슬라이더 의 위치 값을 각 변수에 저장

# BLE (Bluetooth Low Energy)

Bluetooth Low Energy 또는 줄여서 BLE은 블루투스 4.0(Bluetooth Smart) 스펙이 2010년 6월 30일에 채택된 이후로 배포되는 저전력 블루투스이다.

저전력 기능과 쉬운 상호연결성을 모두 가진 통신 기술로써 스마트 워치, 스마트 안경, 스마트 신발, 스마트 가방 등 여러 사물 인터넷 기기들에 적용

헬스케어 및 운동/건강 시장을 중심으로 BLE 시장이 성장해 왔고, 최근 2~3년 동안은 비콘(Beacon) 시장이 폭발적으로 성장하고 있음.

비콘, Beacon? BLE 기반으로 10미터 내외의 거리에 약 30바이트 이내의 데이터를 방송(Broadcasting) 방식으로 정보를 제공하는 서비스로 SK플래닛의 전자지갑 서비스인 '시럽'(Syrup)과 같이 스마트폰 앱과 연동하여 할인 정보 제공 등 여러 분야에서 활용

# 노르딕의 nRF52 시리즈

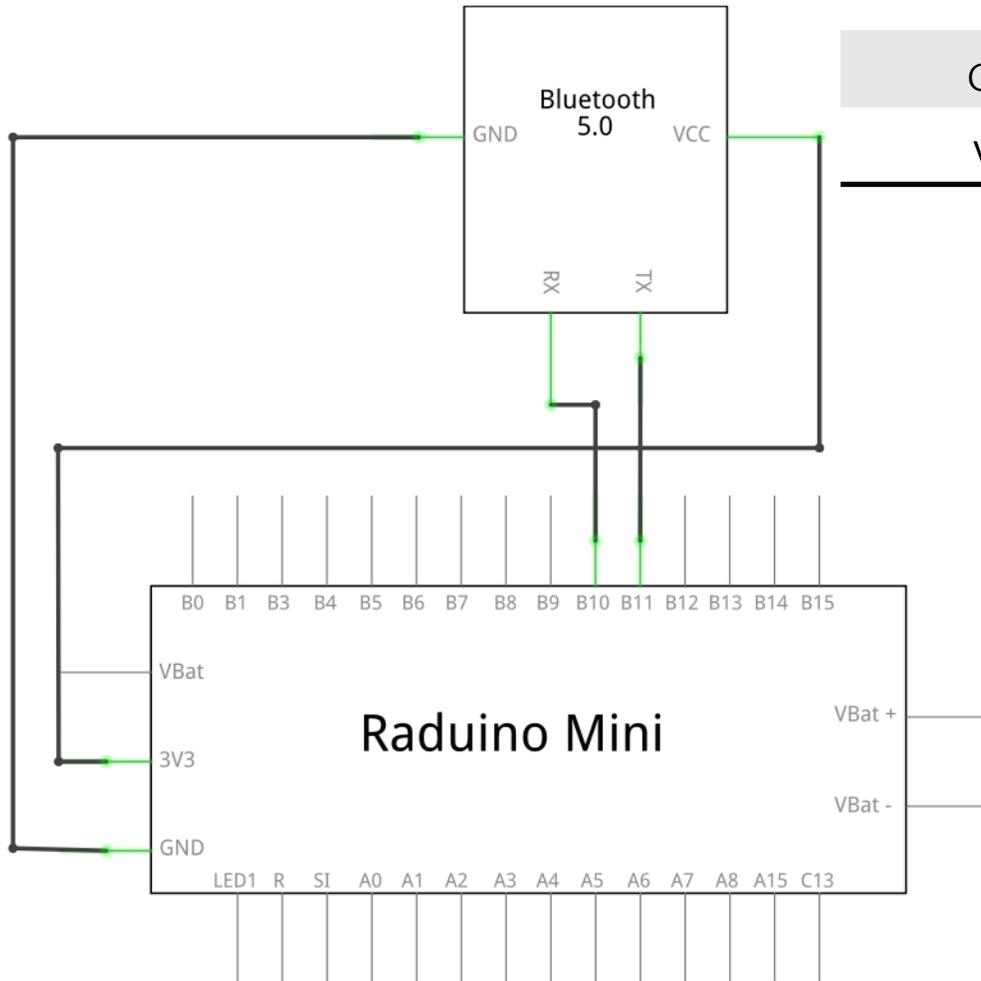
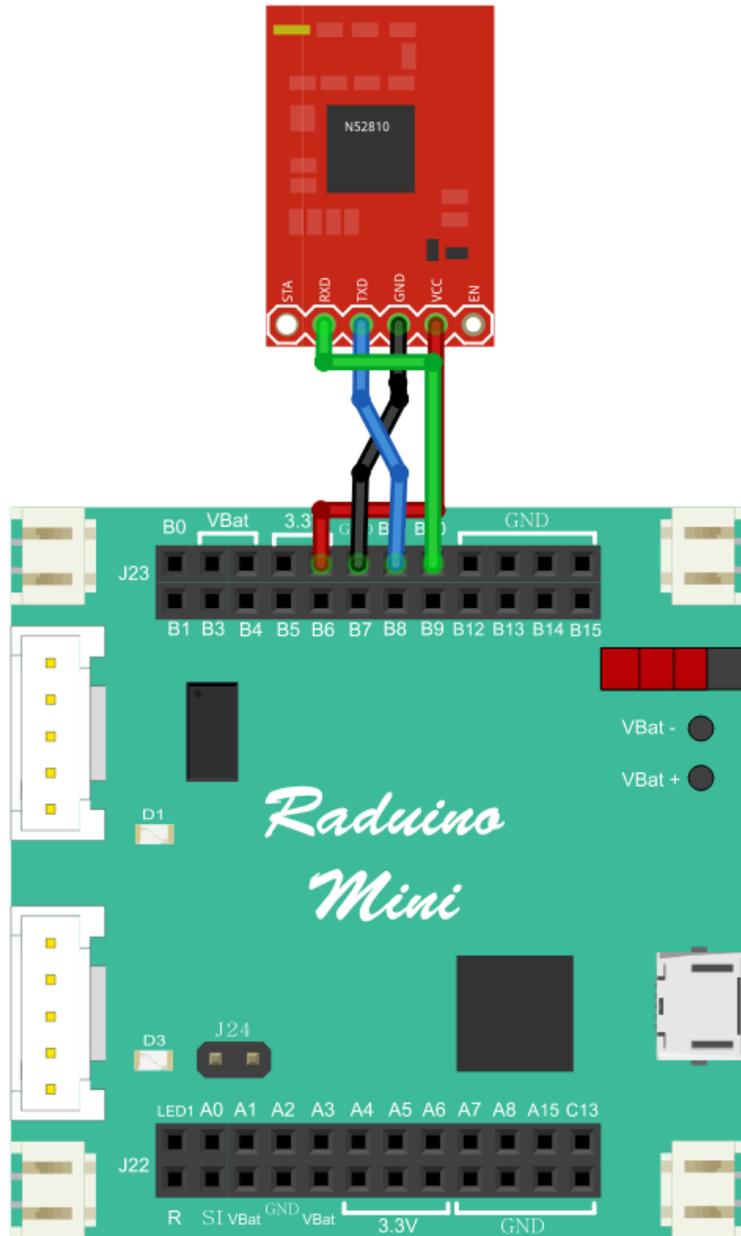
Bluetooth 5 에 최적화된 노르딕 nRF52시리즈

Bluetooth 5는 BLE(bluetooth low energy)에 새로운 기능을 추가합니다:

- 속도 두 배
- 범위 네 배
- 브로드캐스팅 용량 여덟 배

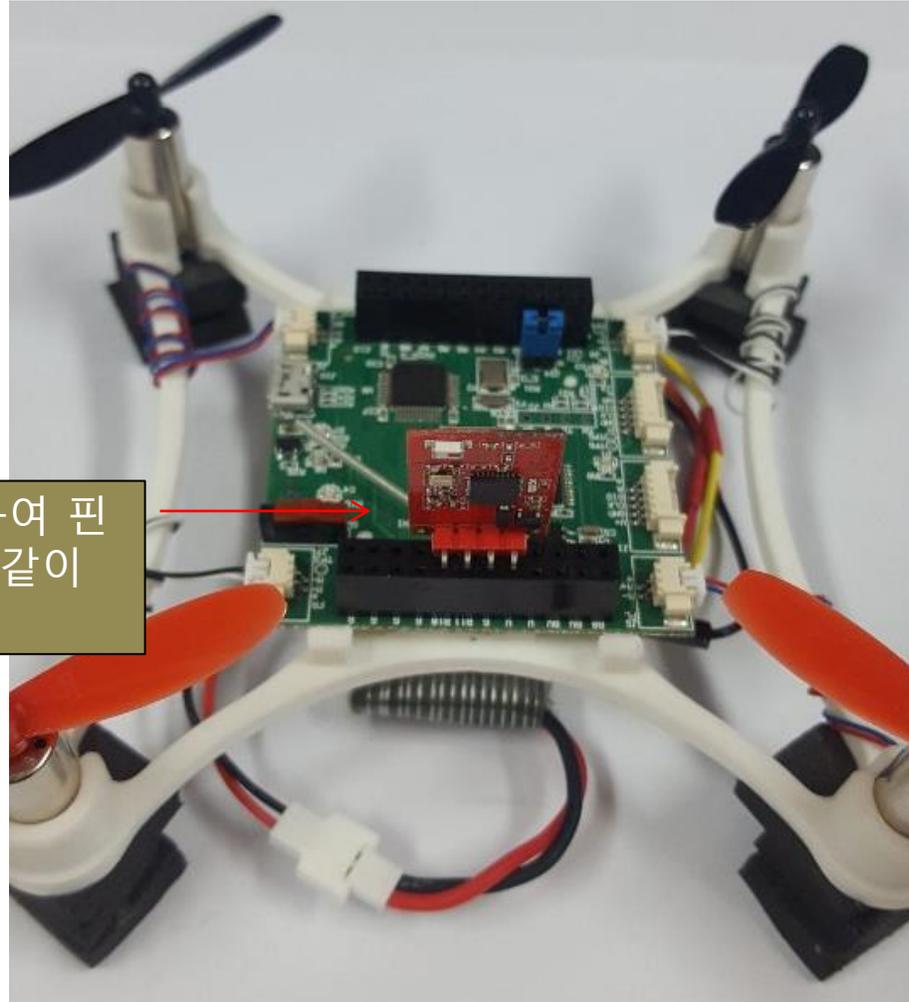


# 연결 회로도



BLE Module	라두이노 미니
RXD	PB10
TXD	PB11
GND	GND
VCC	3V3

# 연결 하기



BLE 모듈을 회로도를 참고하여 핀의 위치를 확인하고 그림과 같이 조립한다.

BLE Module	라두이노 미니
RXD	PB10
TXD	PB11
GND	GND
VCC	3V3

선이 프로펠러에게 걸리지 않도록 주의하자

# 스케치코드 EX4

```
void setup()
{
  raduino_ble_init();//라두이노미니 BLE 연결을 초기화 및 BLE 조정 프로세서 생성
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

# 연습문제

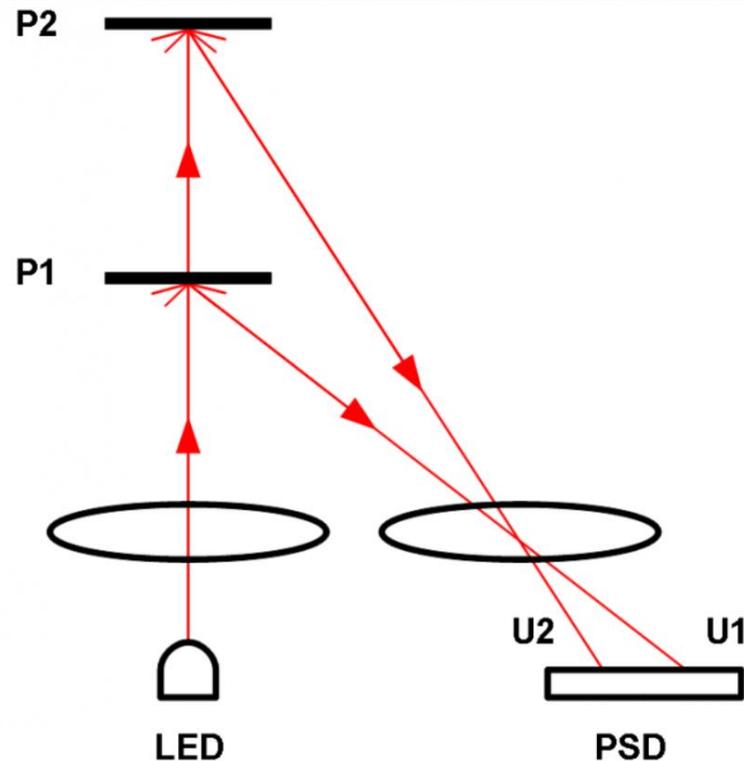
- RGB LED 예제를 적용해보자
- 사운드 예제를 적용해보자

# 고도유지 드론 만들기

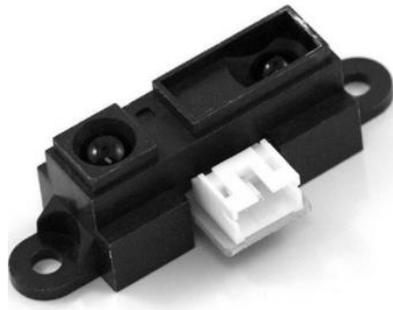
- 준비물 : BLE 모듈, 적외선 센서
- 조정기 : 스마트폰 조정 앱(APP)

# 적외선 센서란

적외선 센서는 초음파 센서에 비해 범위가 좁기는 하지만 정확도는 높다.  
음파 센서와 달리 광삼각법으로 거리를 측정한다.  
음파가 아닌 광선을 사용하므로 대상 물체에 초점이 만들어지고, 초점과의 각도를  
계산해 물체와의 거리를 환산한다.



# 적외선 센서 | GP2Y0A41SK0F



동작전압	4.5 ~ 5V
동작전류	40mA
동작 주파수	40HZ
발생 주파수	40kHz
측정 거리	4cm ~ 30cm
정밀도	0.1cm
크기	29.5x13.0x13.5mm

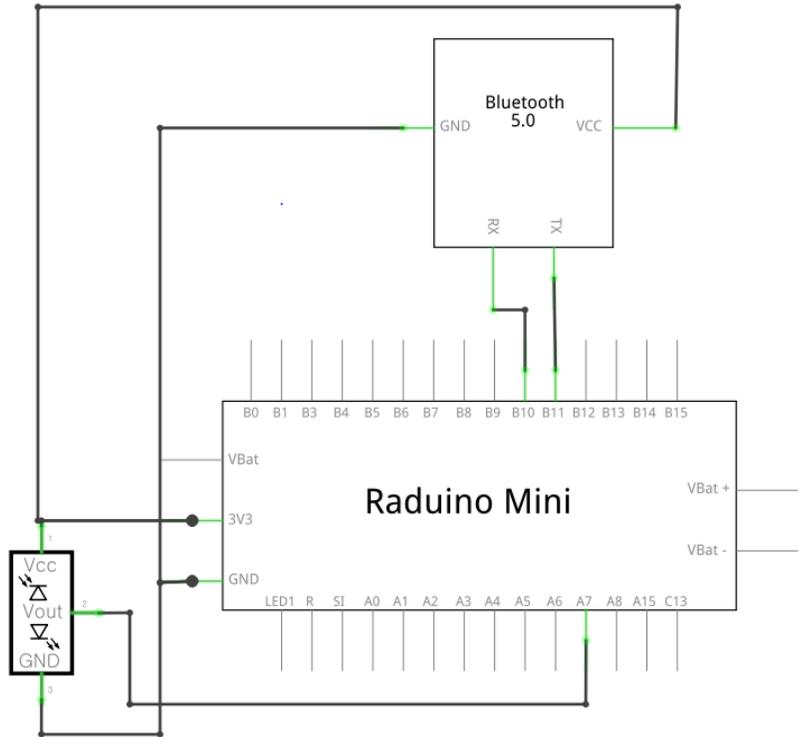
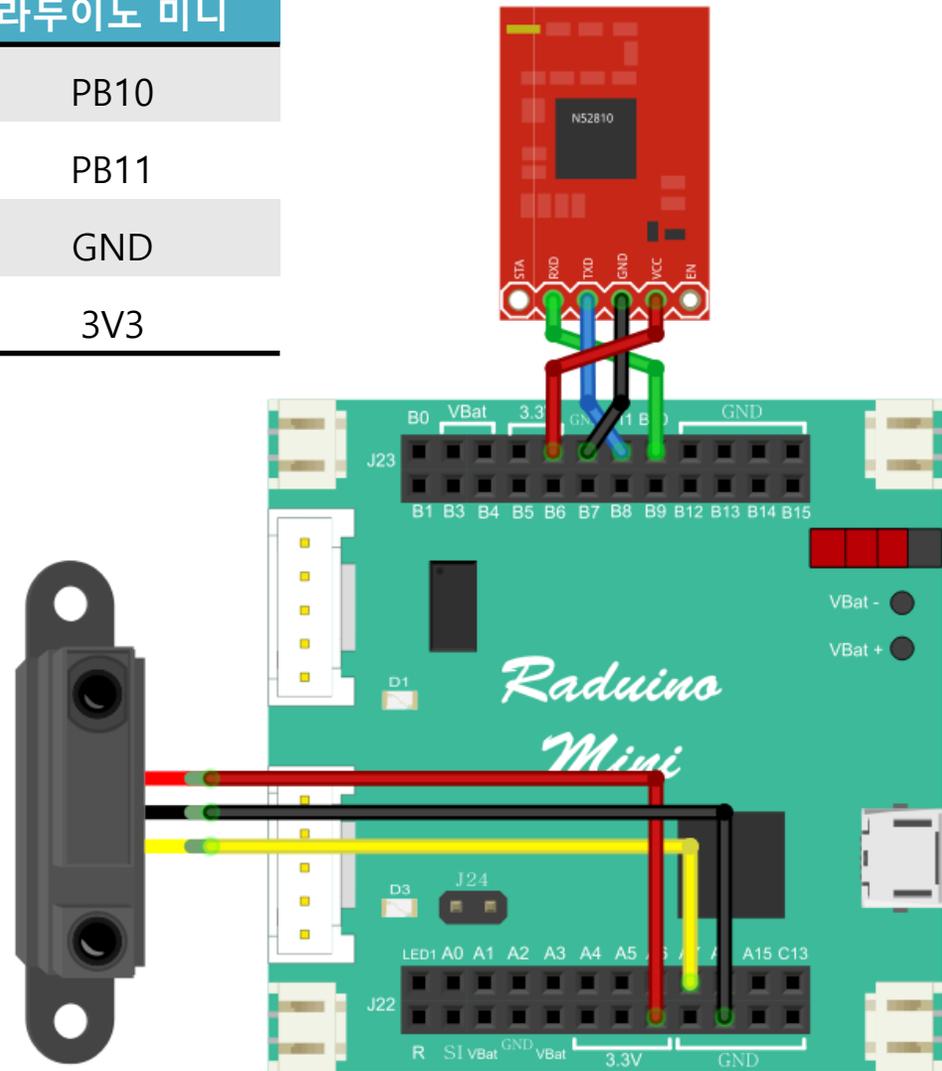
## GP2Y0A41SK0F 의 사양

### 핀맵 GP2Y0A41SK0F

GP2Y0A41SK0F	VCC	GND	OUT
라두이노 미니	VCC	GND	PA7

# 연결 회로도

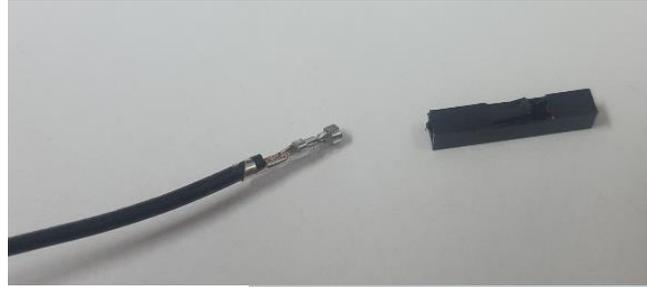
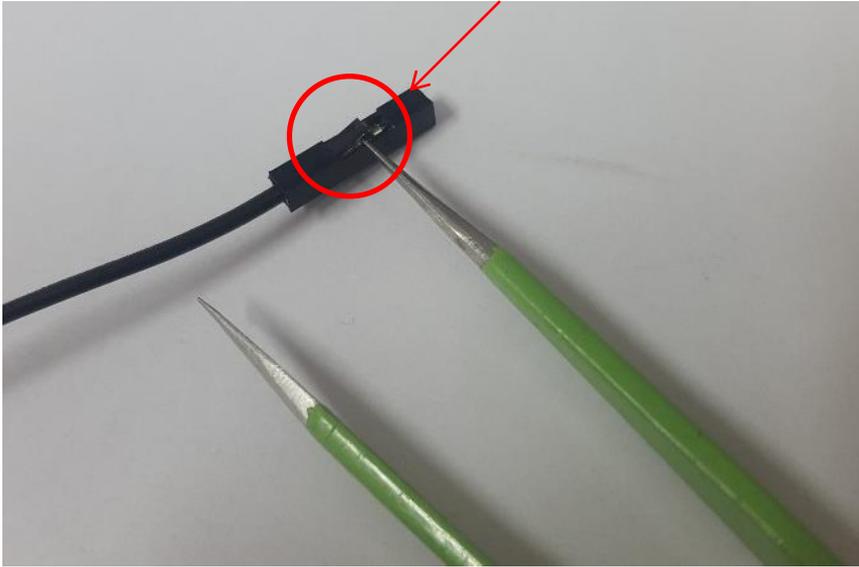
BLE Module	라두이노 미니
RXD	PB10
TXD	PB11
GND	GND
VCC	3V3



적외선	라두이노 미니
VCC	3V3
GND	GND
Vout	PA7

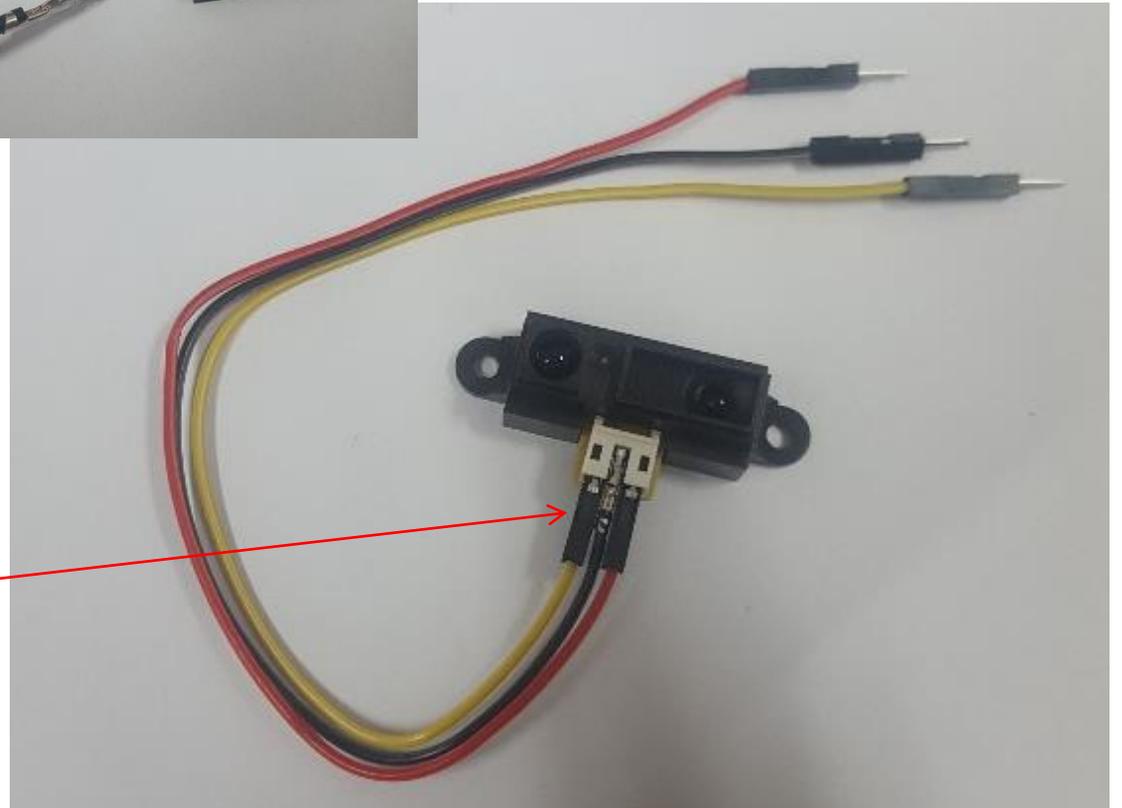
# 연결 하기

핀셋을 이용하여 케이블의 암 커넥터의 이음 부분을 벌려서 커넥터를 제거한다.



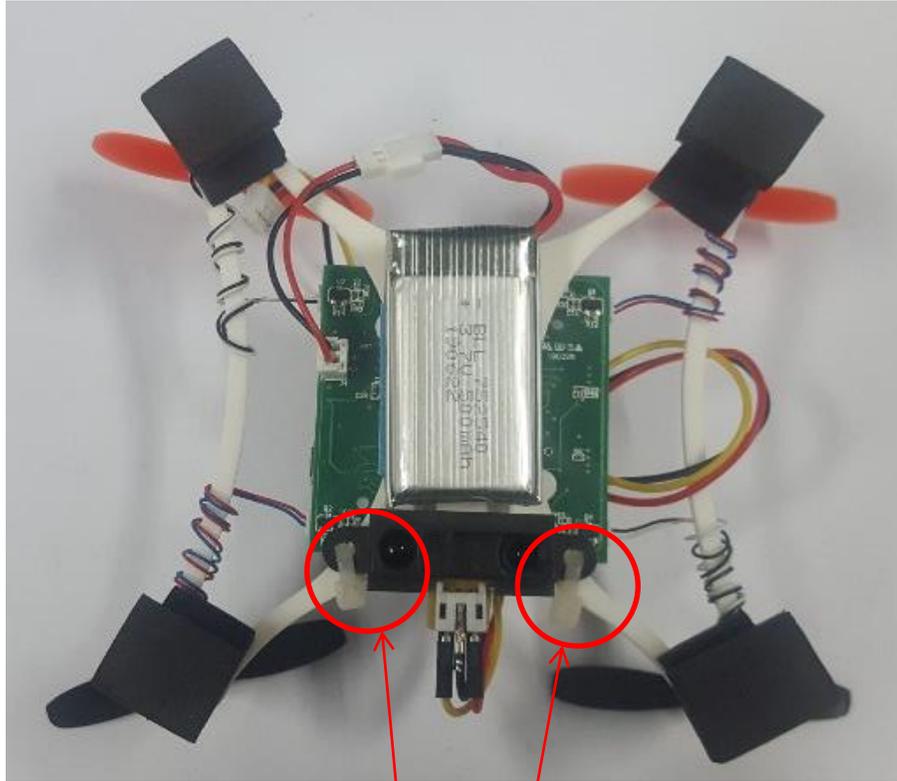
적외선 센서에 기본적으로 제공되는 케이블은 끝에 핀 작업이 되어 있지 않아서 키트에 제공되는 별도의 케이블을 이용하여 다음 작업을 통해 손 쉽게 회로에 연결할 수 있다.

선 색깔과 핀 위치를 그림과 같게 적외선 커넥터에 삽입한다.



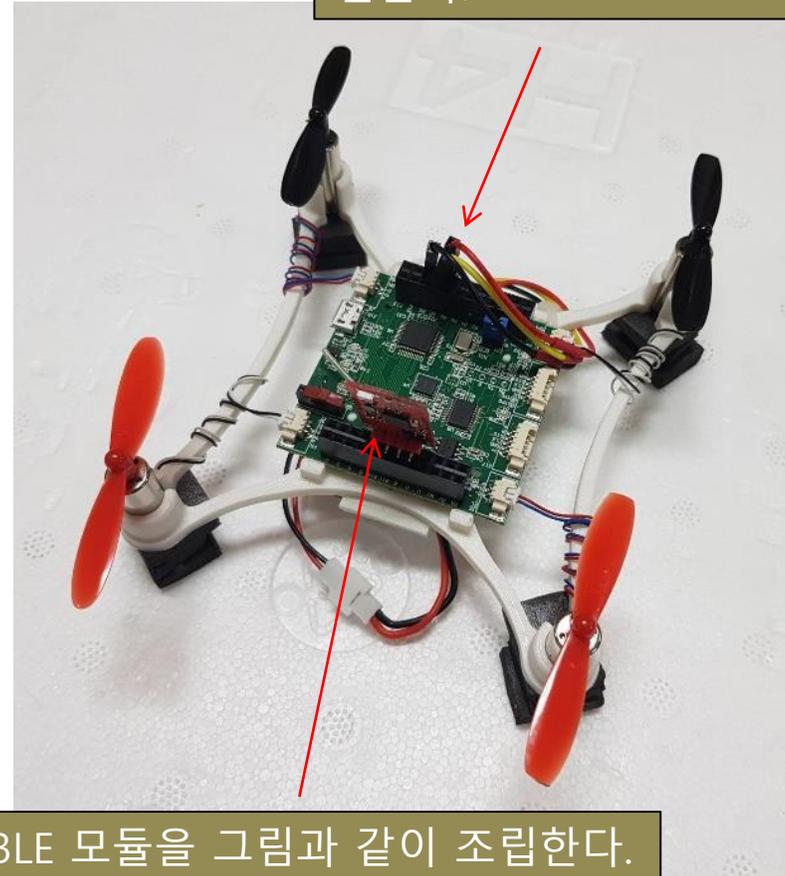
커넥터에서 선이 빠지지 않게 단단히 체결한다.

# 연결 하기



밴드타이를 이용하여 적외선 센서를 바닥을 보도록 그림과 같이 조립한다.

적외선 센서를 회로와 맞게 연결한다.



BLE 모듈을 그림과 같이 조립한다.

선이 프로펠러에게 걸리지 않도록 주의하자

# 스케치코드 EX5

```
#include "raduino_ble.h"  
#include "raduino_fc.h"  
#include "PID_v1.h"
```

```
BLE ble(Serial3);  
FC Flight(Serial1);  
boolean startFlag = false;
```

BLE 모듈이 라두이노 미니의 Serial3에 연결되어 있음  
라두이노 미니 FC와 STM32duino가 Serial1에 연결되어 있음

```
int ledPin = PC13;
```

← 보드에 내장 되어 있는 LED가 PC13에 연결되어 있음

```
double Setpoint, Input, Output;
```

```
double Kp=0.2, Ki=0.3, Kd=0.07;
```

PID의 제어기의 P, I, D의 계수를 정의한다.

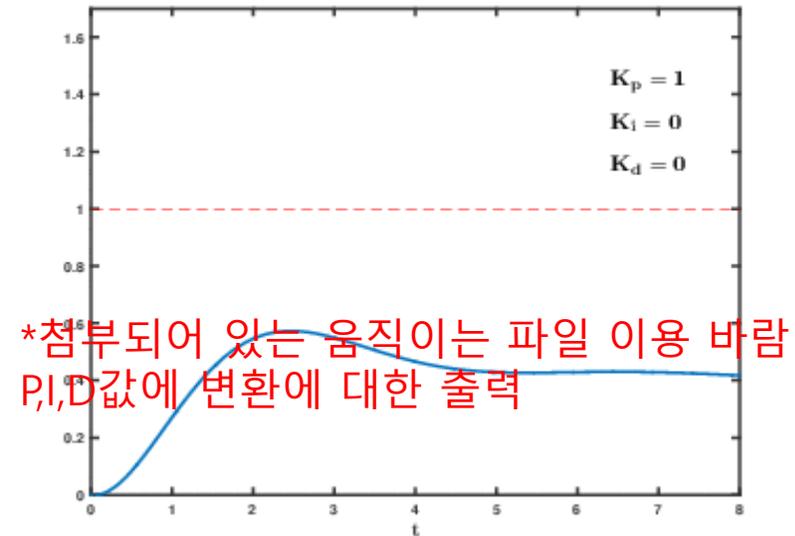
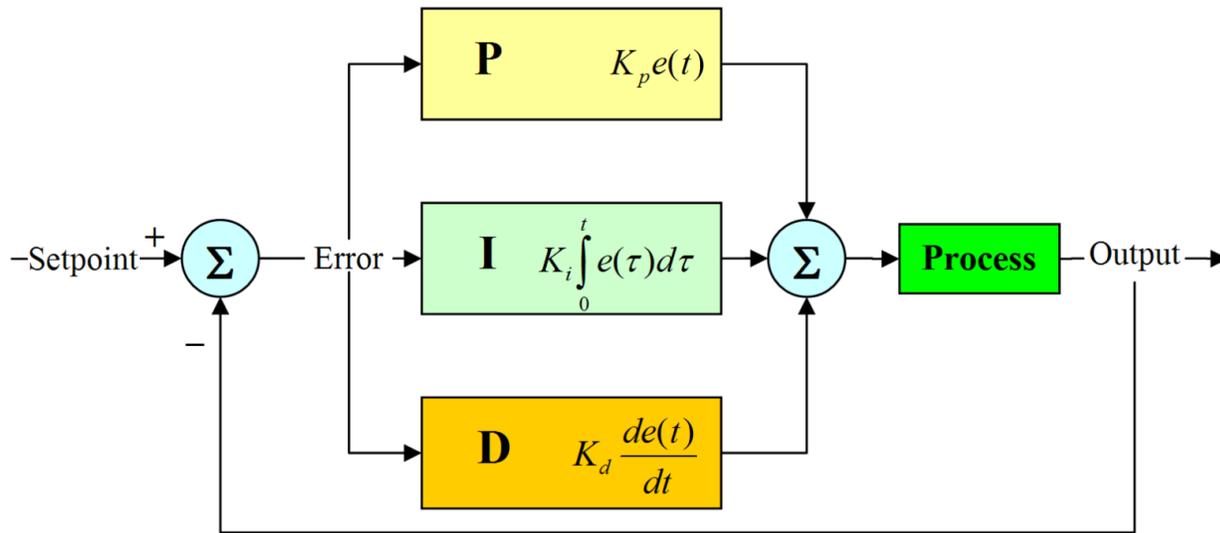
```
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
```

## PID 제어(PID control)

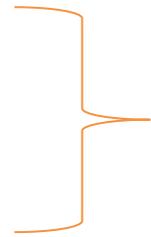
**비례-적분-미분 제어기(Proportional-Integral-Differential controller) 또는 PID 제어(PID control)**는 실제 응용분야에서 가장 많이 사용되는 대표적인 형태의 제어기법이다.

PID 제어기는 기본적으로 **피드백(feedback)** 제어기의 형태를 가지고 있으며, 제어하고자 하는 대상의 출력값(output)와 설정값(setpoint)과 비교하여 오차(error)를 계산하고, 이 오차값을 이용하여 제어에 필요한 제어값을 계산하는 구조로 되어 있습니다.

- 비례항(P): 현재 상태에서의 오차값의 크기에 비례한 제어작용을 한다.
- 적분항(I) : 정상상태(steady-state) 오차를 없애는 작용을 한다.
- 미분항(D) : 출력값의 급격한 변화에 제동을 걸어 **오버슈트(overshoot)**을 줄이고 안정성(stability)을 향상시킨다.



```
const int numReadings = 5;
int readings[numReadings];
int readIndex = 0;
int total = 0;
int average = 0;
```



적외선 센서 값을 위한 변수 선언

```
#define IR_Sensor PA7 // Sharp IR GP2Y0A41SK0F (4-30cm, analog)
```

적외선	라두이노 미니
Vout	PA7

```
void setup() {
  //-----PID Control-----//
  //Setpoint 설정
  Setpoint = 200; // 20cm 고도 유지

  //PID OUTPUT 범위 세팅
  myPID.SetOutputLimits(300, 500); //300~500 사이로 범위 제한

  //PID SampleTime 설정 : 50 msec
  myPID.SetSampleTime(50);

  //PID 시작
  myPID.SetMode(AUTOMATIC);
  //-----PID Control-----//
}
```

```
ble.begin();
```

BLE모듈과 통신 시작

```
// Pause the timer while we're configuring it  
Timer1.pause(); // FC control
```

```
// Set up period  
Timer1.setPeriod(10000); // in 10 microseconds
```

```
// Set up an interrupt on channel 1  
Timer1.setChannel1Mode(TIMER_OUTPUT_COMPARE);  
Timer1.attachCompare1Interrupt(handler_10ms);
```

```
// Refresh the timer's count, prescale, and overflow  
Timer1.refresh();
```

```
// Start the timer counting  
Timer1.resume();
```

Timer Interrupt  
타이머 : Timer1  
주기 : 10 msec  
인터럽트 서비스 함수: handler\_10ms()

\*타이머 인터럽트에 대한 설명은 다음 페이지를 참고

```
Flight.begin();
```

FC와 통신 시작

```
pinMode(LED_BUILTIN, OUTPUT);  
digitalWrite(LED_BUILTIN, HIGH);
```

```
delay(100);
```

```
// roll positive right  
// pitch positive forward  
//Flight.Trim(20,10); //roll: 0.0 , pith: -5.0  
Serial.println("Triming...");
```

```
delay(5000);
```

```
// binding.....  
Flight.Bind();  
Serial.println("binding...");
```

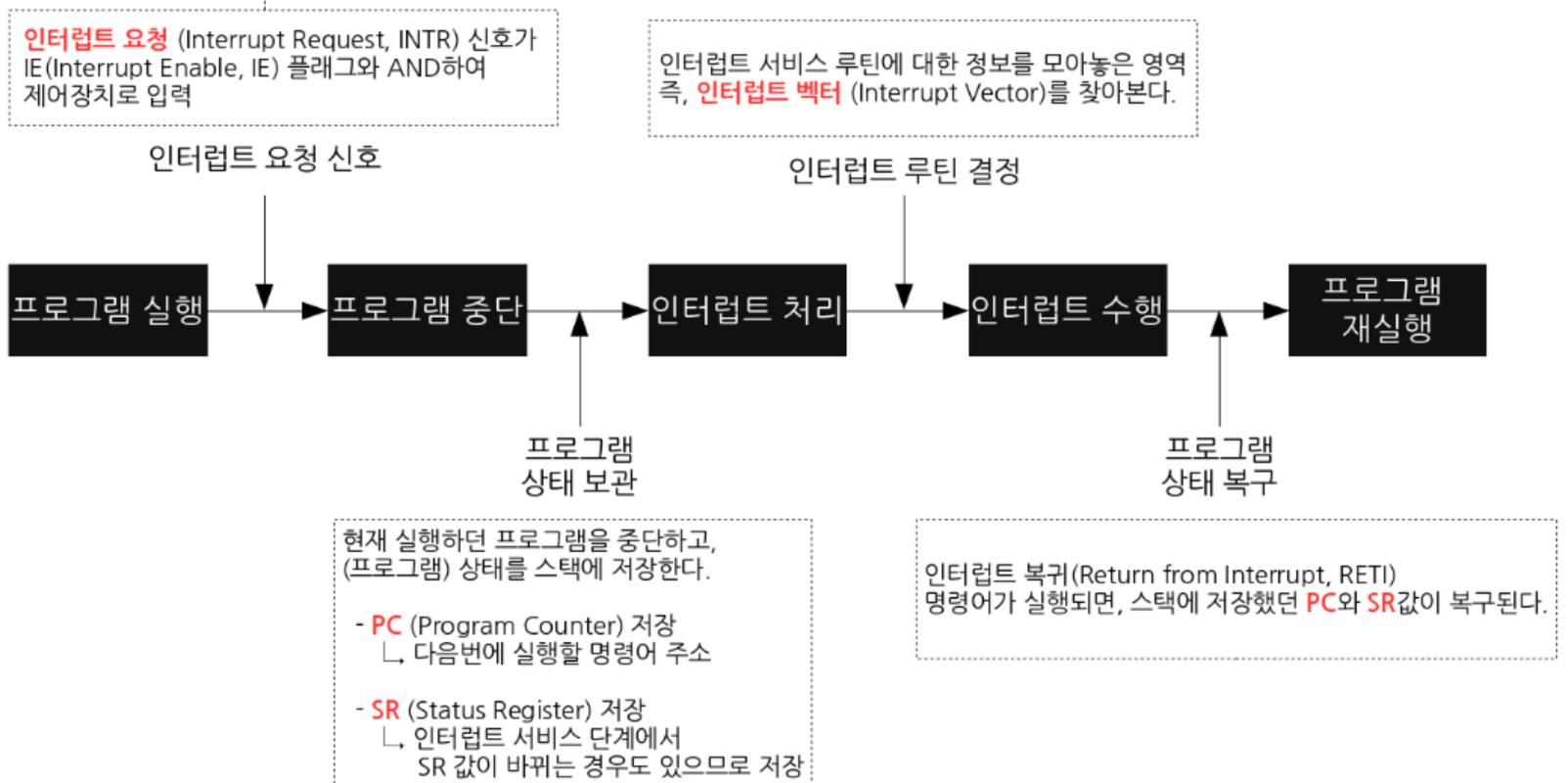
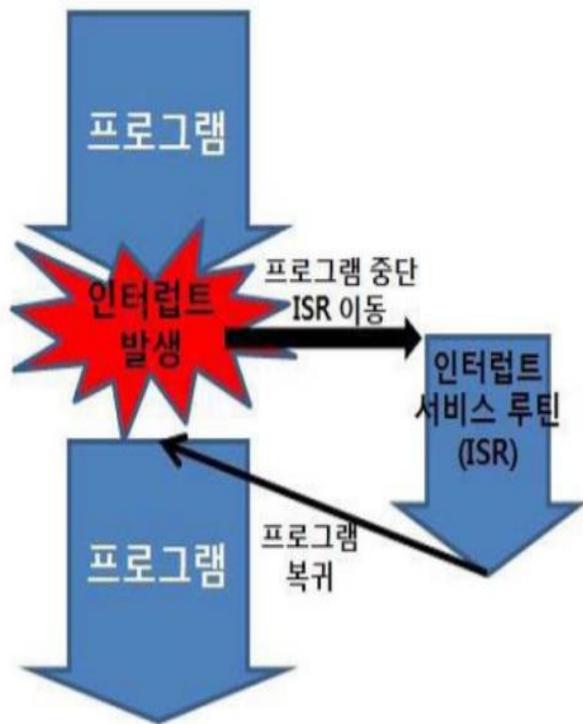
FC와 바인딩 시작

```
}// setup()함수 끝
```

## Timer 인터럽트

마이크로프로세서(CPU)가 프로그램을 실행하고 있을 때, 입출력 하드웨어 등 의 장치나 내, 외부의 어떤 변화에 의한 예외 상황이 발생하여 처리가 필요할 경우에 마이크로프로세서에게 알려 프로그램의 실행을 정지하고 변화에 대응 하는 다른 프로그램을 처리할 수 있도록 하는 것을 말한다.

- 타이머 인터럽트 : 타이머가 일정한 시간 간격으로 중앙처리장치에게 인터럽트를 요청
- 입출력 인터럽트 : 속도가 느린 입출력장치가 입출력 준비가 완료되었음을 알리기 위해 인터럽트를 요청



```
void handler_10ms(void) {
```

```
  G_COUNT++;
  if(G_COUNT > 100)
  {
    G_COUNT = 0;
    ledstate = !ledstate;
    if ( startFlag == true )
      digitalWrite(ledPin, ledstate);
  }
```

스마트 폰과 BLE 모듈간이 연결이 되었다면 led를 blink 시킨다.

```
  if(ble.BleRX)
    FailCount=0;
  else
    FailCount++;

  int throttle =0;
```

스마트 폰과 BLE 모듈간의 통신이 끊겼는지 확인한다.2초 이상 끊긴 경우 착륙한다.

```
  if((ble.blebutton == 1)&&(startFlag ))//DOWN, 고도 PID ON
  {
```

```
    myPID.Compute(); //고도 PID
    throttle= (int)Output; //고도 , Throttle
    if(throttle <200) throttle = 200;
    if(throttle >900) throttle = 900;
```

만약, 고도 유지 버튼이 눌러졌다면 고도 유지 (Setpoint)을 유지하기 위해 PID 제어를 output을 throttle로 사용하며, 그 값은 200~900 사이로 제한된다. 반면, 고도 유지 버튼이 눌러지지 않았다면 throttle 값은 스마트 폰에서 받은 비행 조종 값을 그대로 사용한다.

```
  }else
  {
    throttle= ChannelValue[3]; // 조종값 바이패스
  }
```

```
  if( FailCount > 200)
  {
    throttle = 0;
  }
```

```
  if ( (startFlag == true) && (ble.error_count == 0))
    Flight.Tx(ChannelValue[0],ChannelValue[1],ChannelValue[2],throttle);
```

```
  ble.BleRX = false;
} // handler_10ms() 함수 끝
```

FC로 비행 조정 데이터를 보내는 함수 Flight.Tx(ROLL,PITCH,YAW,THROTTLE)

```
int ChannelValue[4]; int G_COUNT = 0; boolean ledstate =0; int FailCount = 0;
```

```
void loop() {  
  int ret = Flight.Event();  
  if(ret==1)digitalWrite(LED_BUILTIN,LOW);  
  if(ret>0)Serial.println(Flight.retString);  
  ble_process();  
  //-----적외선 센서값 입력-----//  
  //Moving average  
  total = total - readings[readIndex];  
  readings[readIndex] =IRsensor();  
  total = total + readings[readIndex];  
  readIndex += 1;  
  if(readIndex >= numReadings) { readIndex = 0; }  
  average = total/numReadings;  
  Input = average;  
  //-----적외선 센서값 입력-----//  
}
```

Flight.Event() 함수  
FC로 부터 데이터를 수신 했을 경우 1, 그렇지 않은 경우 0을 반환 한다. 다음은 이벤트에 따라 반환되는 값이다.  
0 : No event  
1 : Binding event

ble\_process() 함수  
BLE 모듈을 이용하여  
스마트 폰으로 부터  
데이터를 수신하는 함수

\*적외선 센서 값을 이동 평균을 사용하여 획득  
다음 페이지에 이동 평균에 대한 설명됨

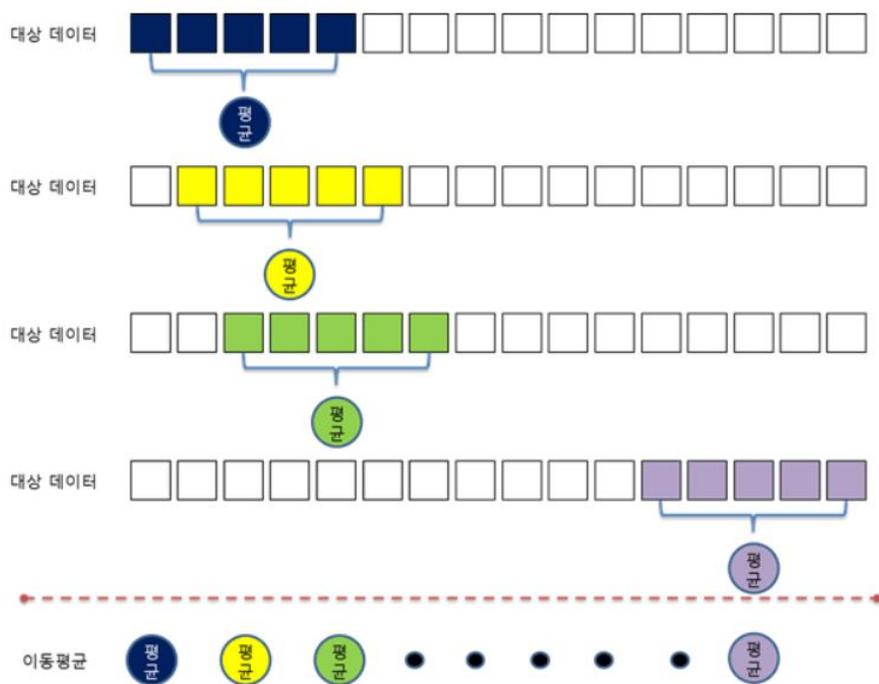
Input : PID 제어기의 입력 값으로 사용됨

## 이동 평균(Moving Average)

보통 이동 평균이라고 하면 단순이동평균(Simple moving average)을 의미한다. 단순이동평균은 최근 n개 데이터의 평균값을 의미한다.

$$SMA = \frac{A_1 + A_2 + \dots + A_n}{n}$$

위 수식의 의미는 t시점을 기준으로 이전 n개 데이터의 평균을 계산하는 것이다.



//-----적외선 센서 측정 함수-----//

```
Int Count = 10;  
Float Distance, SumDistance, AverageDistance = 0;  
Float Voltage = 0;
```

```
Float Irsensor(void){  
  for(int i = 0; i < Count; i++) {  
    Voltage = analogRead(IR_Sensor)*0.0008056640625;  
  
    Distance = 13*pow(Voltage, -1);  
  
    SumDistance = SumDistance + Distance;  
  }  
  AverageDistance = SumDistance/Count * 10;  
  SumDistance = 0;  
  if(AverageDistance > 300) AverageDistance = 300;  
  
  return AverageDistance;  
}
```

SMT32은 12bit의 ADC를 사용하기 때문에 (0~4,095) 값을 갖는다.  
이 값을 전압(0~3.3V)로 변환하기 위한 식은 다음과 같다.  
 $Voltage = analog\ 값 * (3.3/4,096)$

10개의 샘플을  
구한 후 평균값을  
사용한다.

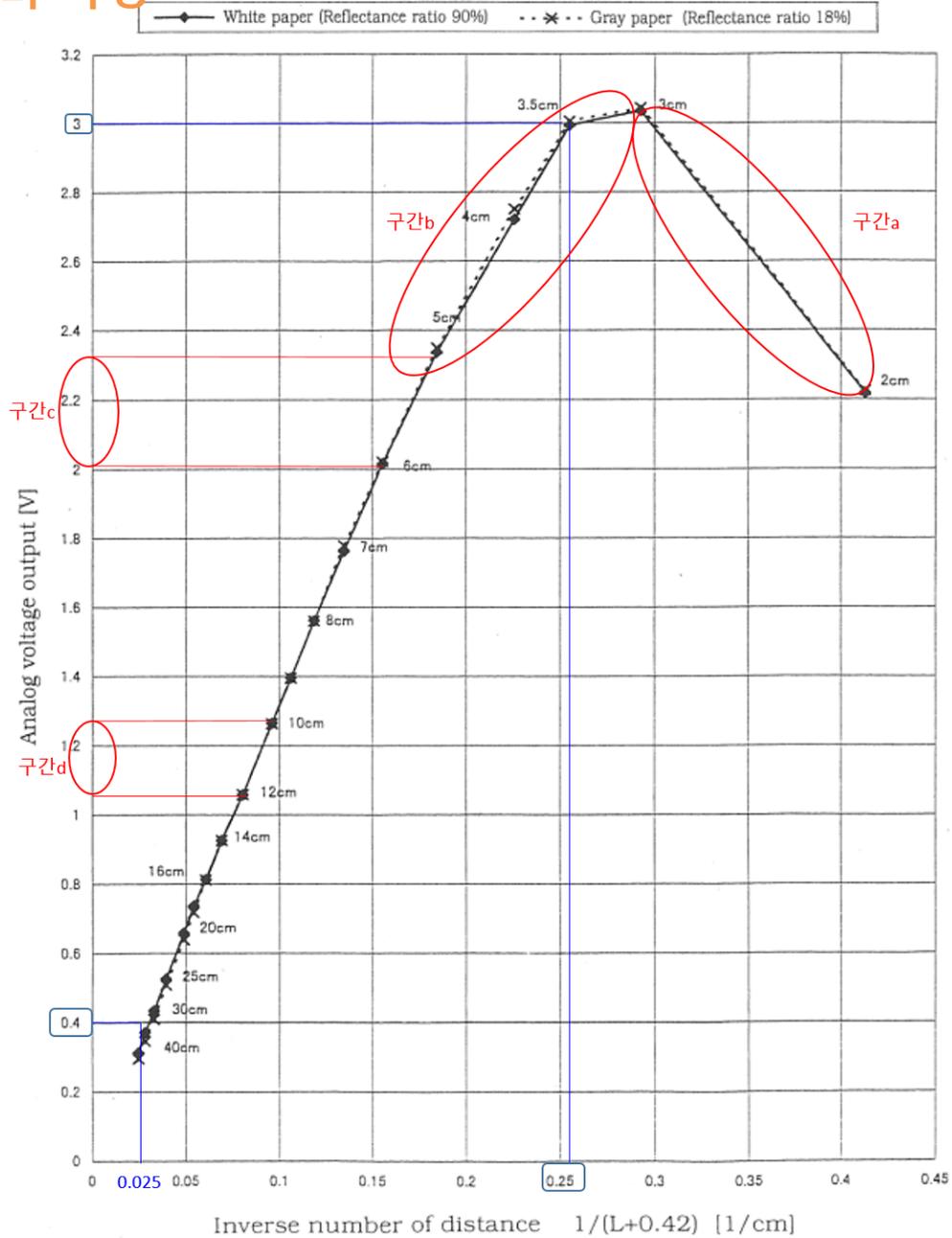
$$\therefore Distance = 13 \cdot (Voltage)^{-1}$$

\*이 수식은 다음 페이지에 설명되어 있음

적외선 센서의 최대 거리 측정 값 300mm이상인 경우 300mm로  
제한한다.

적외선 센서로 측정한 거리 값을 반환한다.

# 적외선 센서 거리 측정



$$Voltage = \frac{0.25 - 0.05 [1/cm]}{3 - 0.4 [V]} \times (Distance)^{-1 [cm]}$$

$$Distance = \frac{3 - 0.4 [V]}{0.25 - 0.05 [1/cm]} \times (Voltage)^{-1 [V]}$$

$$= \frac{2.6 [V]}{0.2 [1/cm]} \times \frac{1}{Voltage [V]}$$

$$Distance [cm] = 13 [cm] \times \frac{1}{voltage}$$

$$\therefore Distance = 13 \cdot (Voltage)^{-1}$$

```
//----- ble 처리 함수-----//
```

```
void ble_process(void){  
    int event1 = ble.Event();  
    if(event1==1){  
        blink_led(); //Binding OK  
        startFlag = true;  
        ble.error_count = 0;  
    }  
    else if(event1==2){  
        for(int i = 0 ; i < 6; i++)  
        {  
            ChannelValue[i]=ble.bleValue[i];  
            //debug  
            //Serial.print(ChannelValue[i]);  
            //Serial.print(",");  
        }  
        //Serial.println("");  
    }  
    else if(event1==3){  
        //debug  
    }  
}
```

Binding event  
스마트 폰 과  
바인딩을 성공한  
경우 LED을  
Blink 시킨다.

ble.Event() 함수

BLE 모듈이 스마트 폰으로 부터 데이터를 수신했을 경우 1,2 또는 3을 반환 한다. 하지만 받은 데이터가 없으면 0을 반환 한다. 다음은 이벤트에 따라 반환되는 값이다.

0 : No event

1 : Binding event

2 : Flight Data event

3 : Wrong Data event

Flight Data event

스마트 폰으로부터 비행 조정 데이터를 수신한 경우 비행 데이터를 ChannelValue[]에 저장한다. 이 값은 타이머 인터럽트 함수 handler\_10ms(void)에서 사용된다.

# 연습문제

- Setpoint 200 -> 250으로 바꿔보자
- P값을 조정해보자.
- D값을 조정해보자.

실행예제 유튜브

<https://youtu.be/f8FOOMSvzow>