

codrone II

코드론 2 로킷 브릭

Rokit Brick

ROKIT
BRICK

for

codrone II



BUILD, CODE, PLAY

codrone II

로켓 브릭으로 코드론 제어하기

프로그램 소개




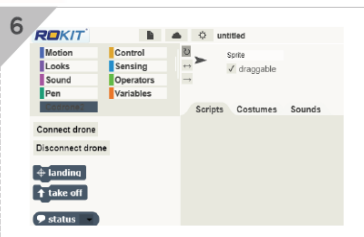
BUILD, CODE, PLAY

1. PC 에서 코딩모드로 진입하기

드론 시뮬레이터 & 로킷brick

<p>1</p>  <p>USB 케이블을 PC(노트북)에 연결합니다.</p>	<p>2</p>  <p>마이크로 5핀 단자를 리모컨의 포트에 연결합니다.</p>
<p>3</p>  <p>정상적으로 인식된 경우 장치관리자 > 포트 항목에서 위와 같이 표시됩니다. (COM포트 번호는 다를 수 있습니다)</p>	<p>4</p>  <p>리모컨이 위 화면처럼 변경된다면, 코딩모드로 변경 완료된 것을 확인할 수 있습니다.</p>

! - windows 10 에서는 리모컨 USB 드라이버가 자동 설치되지만 windows 7과 windows 8 에서는 드라이버를 수동으로 설치해 주셔야 합니다. (수동 설치 방법은 홈페이지 참조)
- 장치 관리자의 리모컨 통신 장치 이름은 'STM32 virtual COMport'로도 표시될 수 있습니다.

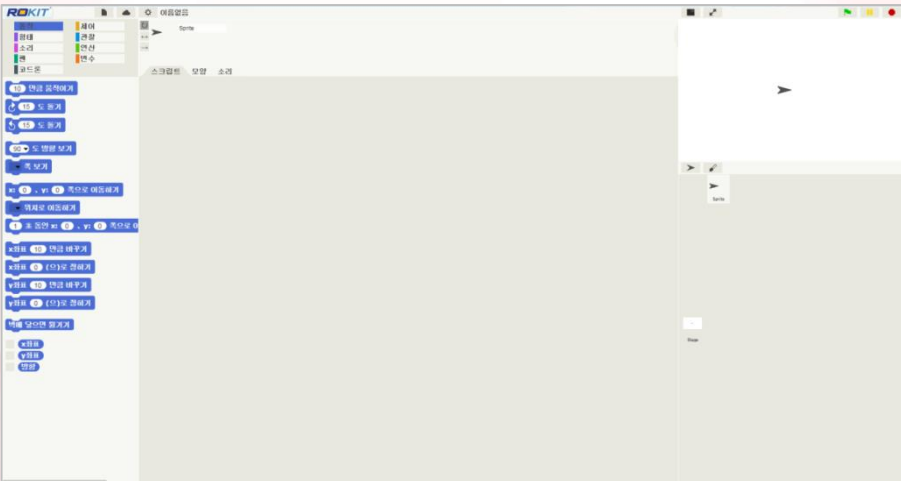
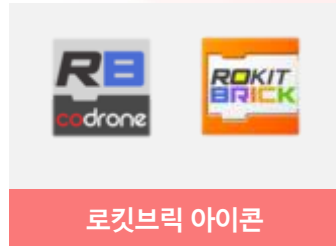
<p>5</p>  <p>드론에 배터리를 넣어 전원을 켜고 리모컨과 드론을 연결합니다.</p>	<p>6</p>  <p>로킷 브릭(Rokit Brick)으로 코딩하여, 드론이 원하는 대로 움직일 수 있도록 해 보세요!</p>
--	--

* Rokitbrick for codrone의 다운로드 및 실행방법은 홈페이지를 참조하세요.

2. Rokit Brick 이란?

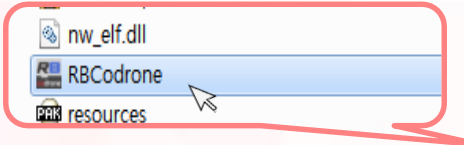
Rokit Brick은 "Snap" 기반으로 기존 Scratch 프로그램에 로봇, 드론 등 다양한 제어기능이 합쳐져 개발된 Scratch 방식의 SW이며, 사용 방법도 Scratch와 거의 동일하다.

로킷 스마트 시리즈와 코드론 같은 HW들을 Scratch 형식으로 제어할 수 있다.



3. Rokit Brick for CoDrone 실행

(1) Rokit Brick for CoDrone2은 프로그램 폴더를 원하는 경로에 위치시키고 RBCodrone2.exe 실행파일을 클릭한다.



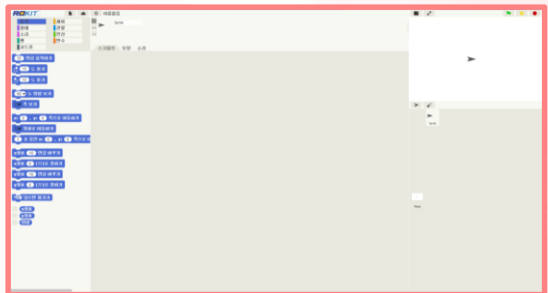
icons	2019-03-27 오전...	파일 폴더	
locales	2019-03-27 오전...	파일 폴더	
swiftshader	2019-03-27 오전...	파일 폴더	
credits	2018-11-20 오전...	Opera Web Docu...	2,069KB
d3dcompiler_47.dll	2018-11-20 오전...	응용 프로그램 확장	4,245KB
ffmpeg.dll	2018-11-20 오전...	응용 프로그램 확장	1,335KB
icudtl.dat	2018-11-20 오전...	DAT 파일	9,979KB
libEGL.dll	2018-11-20 오전...	응용 프로그램 확장	107KB
libGLESv2.dll	2018-11-20 오전...	응용 프로그램 확장	4,896KB
natives_blob.bin	2018-11-20 오전...	BIN 파일	111KB
node.dll	2018-11-20 오전...	응용 프로그램 확장	11,511KB
notification_helper	2018-11-20 오전...	응용 프로그램	561KB
nw.dll	2018-11-20 오전...	응용 프로그램 확장	106,988KB
nw_100_percent	2018-11-20 오전...	ALZip PAK File	1,000KB
nw_200_percent	2018-11-20 오전...	ALZip PAK File	1,312KB
nw_elf.dll	2018-11-20 오전...	응용 프로그램 확장	559KB
RBCodrone	2019-03-12 오후...	응용 프로그램	49,930KB
resources	2018-11-20 오전...	ALZip PAK File	5,163KB
v8_context_snapshot.bin	2018-11-20 오전...	BIN 파일	1,014KB



(2) 프로그램이 실행되면 왼쪽의 < > 버튼으로 Codronell 의 Rokit Brick을 실행한다.



(3) 정상적으로 실행이 되면 오른쪽 이미지와 같은 Rokit Brick이 실행된다.



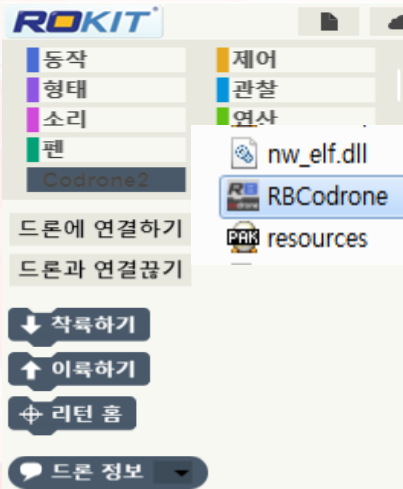
4. Rokit Brick과 BLE보드의 연결 (1)

PC와 리모컨을 연결하였다면 **USB 드라이브 설치**가 잘 되었는지를 확인하시기 바랍니다.

※ [부록] 리모컨 USB 드라이버 설치하기

드라이브 설치 및 COM번호를 확인한 후 **Rokit Brick for Codrone** (RBCodrone2, 로킷 브릭 코드론2)을 실행한다.

왼쪽 하단 팔레트에서 **Codrone2** 을 클릭하면 드론 제어 블록들이 나타나고 윗부분에 두 개의 버튼 (드론과 연결하기/드론과 연결끊기) 이 있습니다.



리모컨을 PC에 연결한 후 '드론에 연결하기' 버튼을 눌러 리모컨이 연결된 포트를 클릭하면 연결이 됩니다.

이 위에 드론에 배터리를 넣으면 컨트롤러와 드론이 연결이 되어 제어가 가능해 집니다.

사용 도중 배터리를 빼서 드론의 전원을 꺼도 드론과의 연결은 끊을 필요가 없으며 다시 드론의 전원을 켜면 자동 연결되어 계속 사용할 수 있습니다.

주의사항 !

•드론에 연결하기 버튼을 누르기 전에 미리 드론과 컨트롤러는 페어링이 되어 있어야 합니다!

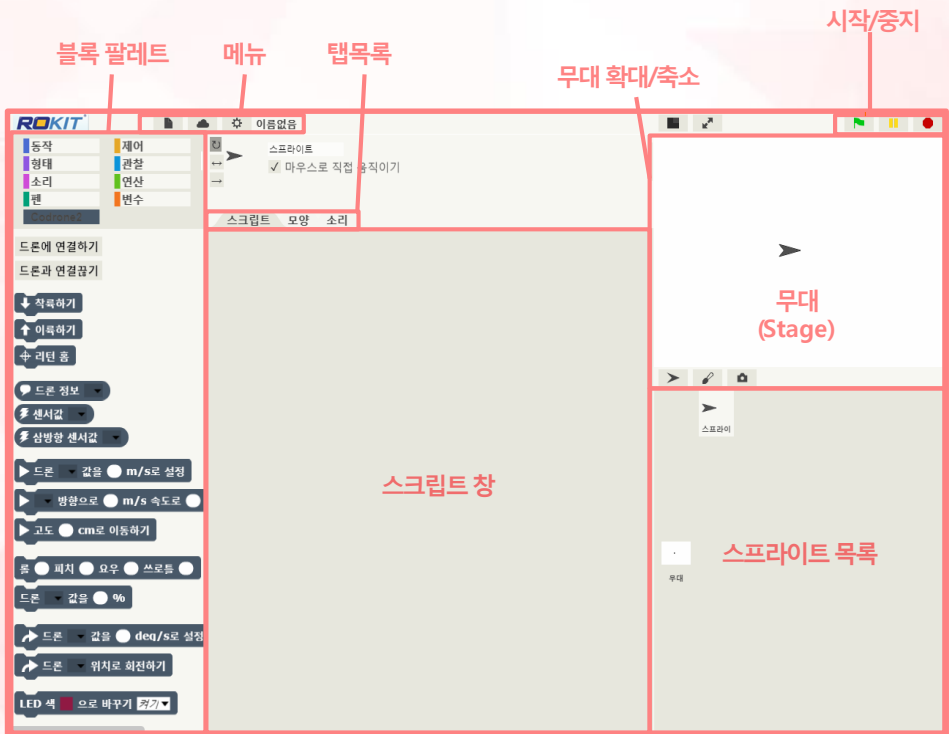
※ [부록] 코드론2 페어링 하기

드론에 연결하기 버튼을 눌렀을 때 여러 개의 포트가 있어서 어떤 포트가 컨트롤러와 연결되어 있는 것인지 알 수 없다면 장치관리자를 열어 확인하세요!

※ [부록] 코딩모드

5. Rokit Brick의 화면 구성

기본적인 메뉴와 화면의 레이아웃은 MIT의 스크래치와 유사하며 아래와 내용이 같습니다.



로켓 브릭으로 코드론 제어하기

기본 스크립트작성

로켓브릭으로 스크립트를 작성/실행 하는 방법

※ 스크래치 경험이 있으신 사용자들은 이번 장을 넘어
가셔도 됩니다.

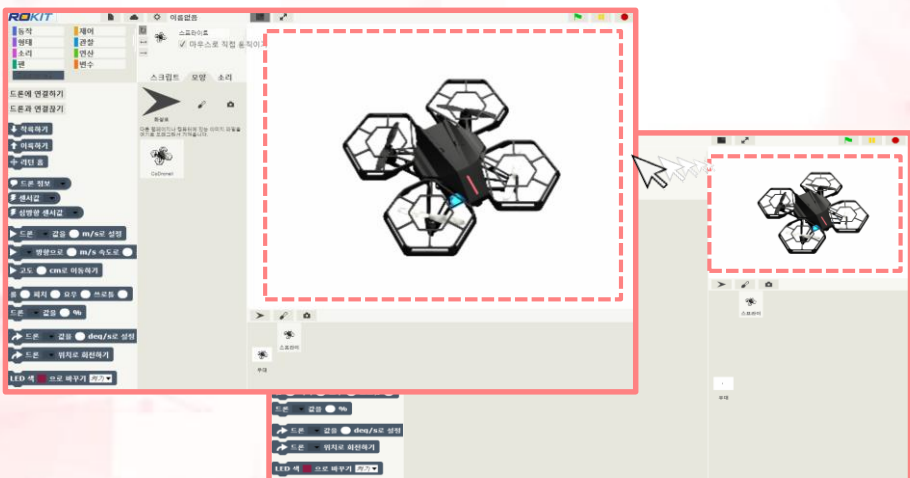


1. 이미지 가져오기

메뉴에서 **모양** 탭을 이용하여 원하는 이미지를 가져올 수 있다.



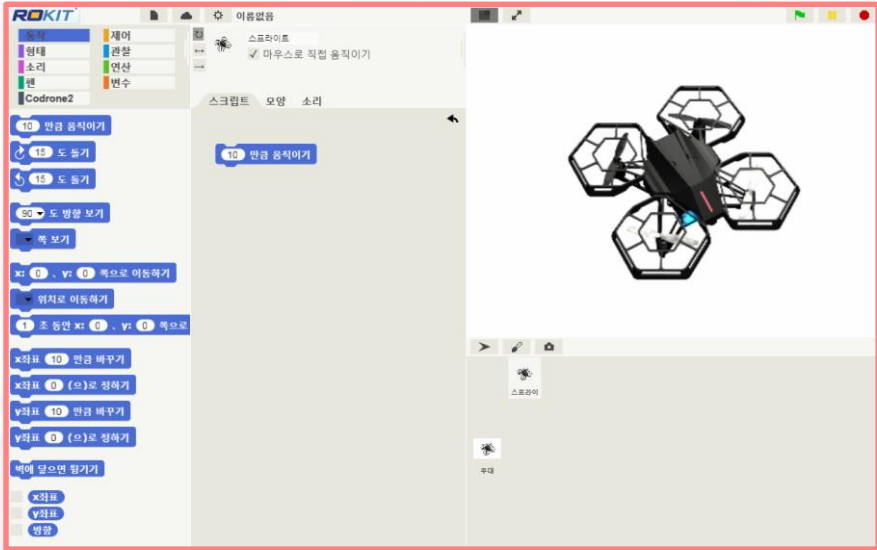
경계선을 마우스로 드래그하여 **무대의 크기를 조정**할 수 있다.



2. 코드론(CoDrone) 기본 구동 예제

1) 드론 이동하기 (코드론 직선 이동하기)

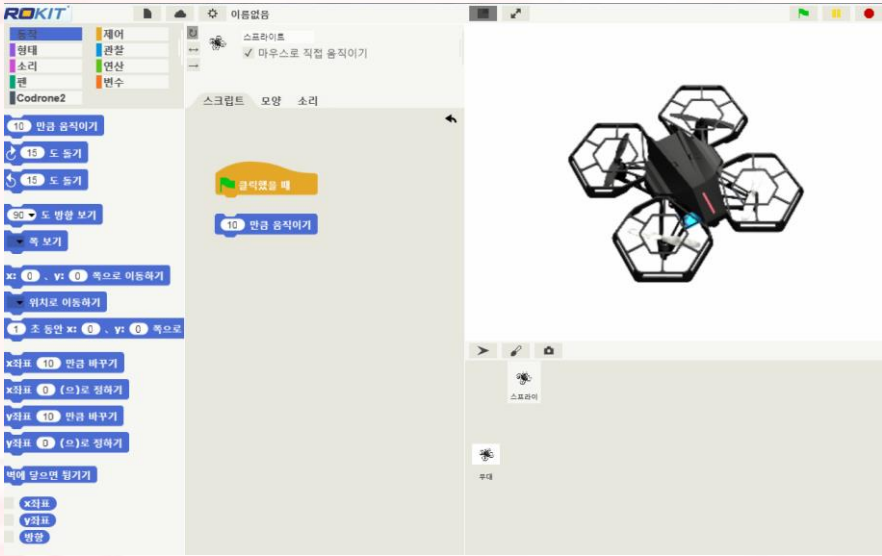
동작 블록군에서 **10** 만큼 움직이기 를 스크립트 화면으로 가져온다.



스크립트 화면에 놓인 블록을 마우스로 계속 클릭할 때마다 드론이 10만큼 앞으로 이동하게 된다.



마우스로 클릭하기로 바로 동작을 확인할 수 있으며, 일반적으로 제어블록군에서 **클릭했을 때** 이나 **스페이스 키를 눌렀을 때** 를 이용하여 구동을 시작하게 된다.



클릭했을 때 블록을 **10 만큼 움직이기** 와 연결하고 녹색깃발을 클릭하면 이동한다.

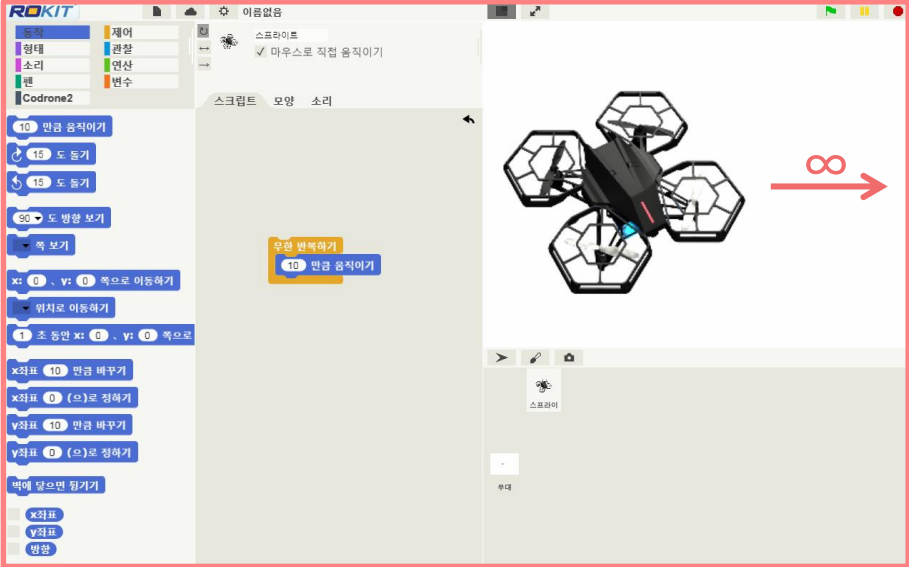


드론 원점 이동

만약 드론이 안 보이는 곳까지 사라지게 되었을 때, **[동작] 형태** 블록의 **[x:0, y:0 쪽으로 이동하기]** **x: 0, y: 0 쪽으로 이동하기** 블록을 클릭하면 원점으로 돌아온다.

2) 드론 이동하기 (무한 반복으로 드론 이동하기)

제어블록군에서 무한 반복하기 블록을 [()만큼 움직이기] [10] 만큼 움직이기 와 연결하여 구동해본다. 드론이 한 번의 클릭으로 보이지 않는 곳까지 이동하게 된다.



드론이 화면 밖으로 사라지지 않고, 벽에 튕기는 동작을 반복하게 하려면 [벽에 닿으면 튕기기] [벽에 닿으면 튕기기] 블록을 이용한다.



3) 드론 이동하기 (이륙 및 이동 착륙하기)

그림1과 같이 드론이 움직이려면 그림2와 같이 좌표 값과 초 기다리기 (Delay)를 이용하여 코딩할 수 있다.

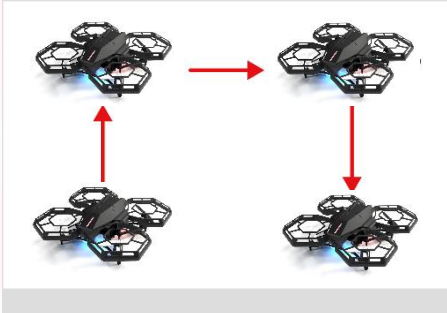


그림1

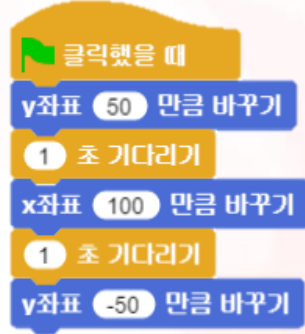


그림2

그림1에서 나타난 드론의 이동거리를 바꿔주기 위해서는 그림3에 나타난 블록들의 x,y 좌표 값을 변경한다. 마찬가지로 기다리는 시간도 수치를 입력하여 원하는 시간으로 변경할 수 있다.

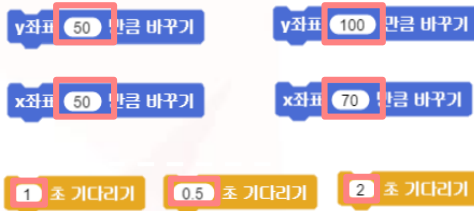
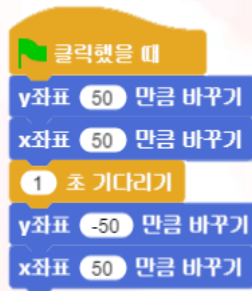
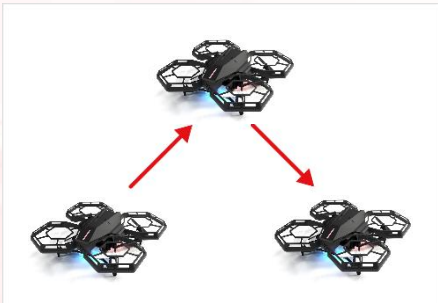


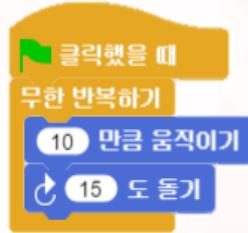
그림3



4) 드론 이동하기 (변수 사용하기)

변수를 이용하여 드론의 동작을 변화시켜 보자.

우선 아래의 그림과 같이 드론이 회전하는 코드를 만들어 보자.



10 만큼 움직이기 블록의 이동 값을 변화시키기 위하여 변수를 만든다. 변수블록군에서 **변수만들기** 버튼을 누르고 A라는 변수를 만든다.



변수란?

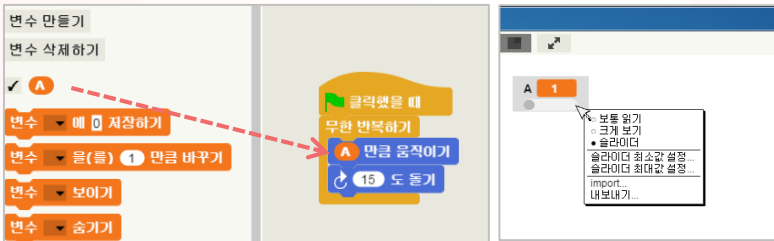
특히 프로그램에서 많이 사용하는 **변수란, 숫자를 저장할 수 있는 공간**을 만들고, 원하는 숫자를 언제든지 넣을 수 있도록 하는 숫자가 아닌 문자를 뜻한다.

예) A = 5, Value = 0

A라는 변수를 만들면 아래와 같이 **변수블록군**이 생기며, 무대에 변수가 생성된다.



왼쪽에 생성된 A 변수 **블록을 변화하고 싶은 부분**으로 아래와 같이 **드래그** 한다. 무대에 나타난 변수에서 **마우스 오른쪽클릭**을 눌러 **슬라이더를 선택**해준다.

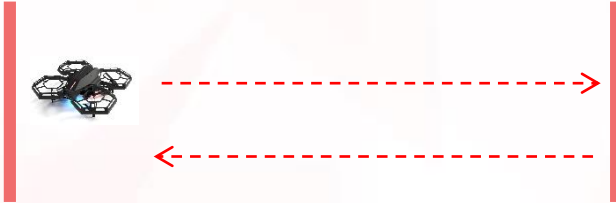


무대를 실행한 후, 마우스로 무대의 A 변수 슬라이더를 움직이면 **실시간으로 변수의 값이 달라지면서** 드론의 회전 반경이 커졌다 작아졌다 하는 것을 볼 수 있다.

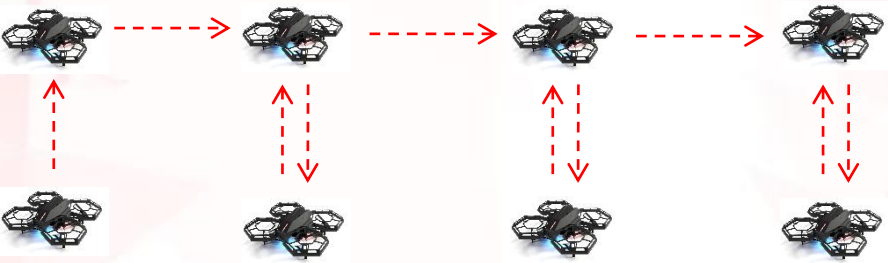


3. 코드론(CoDrone) 구동 연습문제

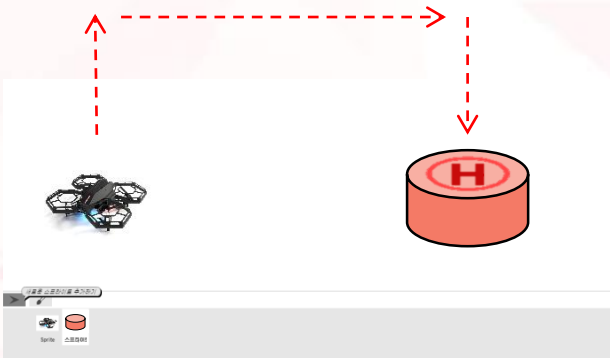
1) 드론이 이동하며 벽을 반사하여 왔다갔다하는 동작을 만들어 보고 변수를 이용하여 움직이는 속도를 조절하도록 한다.



2) 다음과 같은 동작을 무한반복이 아닌 **3회만 반복**하는 프로그램을 만들어보자. 사용하는 **블록 개수를 최소화**하여 시도해본다.



3) 새로운 스프라이트 추가하기를 통해 착륙장을 만든 후, 드론이 착륙장에 착륙하는 동작을 다양한 방법으로 구현해본다.



동작블록으로 코드론 제어하기

코드론 블록설명

동작블록으로 코드론을 제어하는 방법



5. 코드론(CoDrone) 블록 설명 - 속도로 제어하기











삼차원 좌표계를 기준으로 하여 드론을 **좌표계의 방향에 따른 속도로 제어하는 방식**입니다.

오일러 각으로 제어하는 것 보다 좀 더 드론을 정밀하게 제어할 수 있게 해 줍니다.



※ [부록] 좌표계 - 코드론2 좌표계

▶ 드론 상승 (+Z) ▾ 값을 ● m/s로 설정	일정한 각 속도(m/s)로 상승하거나 하강함
▶ 드론 하강 (-Z) ▾ 값을 ● m/s로 설정	
▶ 드론 전진 (+X) ▾ 값을 ● m/s로 설정	일정한 속도(m/s)로 전진하거나 후진함
▶ 드론 후진 (-X) ▾ 값을 ● m/s로 설정	
▶ 드론 좌측 (-Y) ▾ 값을 ● m/s로 설정	일정한 속도 (m/s)로 좌측이나 우측으로 이동함
▶ 드론 우측 (+Y) ▾ 값을 ● m/s로 설정	
▶ 드론 좌회전 ▾ 값을 ● deg/s로 설정	일정한 각 속도(deg/s)로 회전함
▶ 드론 우회전 ▾ 값을 ● deg/s로 설정	

5. 코드론(CoDrone) 블록 설명

 이륙하기	<p>드론을 이륙시킴 드론의 모든 비행 동작은 이 블록부터 시작함</p>
 착륙하기	<p>드론을 착륙시킴</p>
 리턴 홈	<p>드론이 처음 이륙했던 곳으로 되돌아감</p>
 드론 정보 비행 상태 ▾	<p>드론의 비행 상태를 알려줌 (대기/착륙/이륙/비행 등)</p>
 드론 정보 제어 방식 ▾	<p>드론이 속도(위치) 기준으로 제어되는 지 오일러 각 기준으로 제어되는지를 알려줌 (위치/에티튜드)</p>
 드론 정보 동작 상태 ▾	<p>어떤 형태의 비행 동작을 하고 있는지를 알려줌 (대기/호버링/이동중/리턴 홈)</p>
 드론 정보 방향 기준 ▾	<p>드론이 비행할 때의 방향의 기준이 무엇인지를 알려줌 (Headless / Normal)</p>
 드론 정보 센서 방향 ▾	<p>각도 측정 센서를 참조하여 현재 드론의 상태가 어떤지를 알려 줌 (정상 / 뒤집히기 시작 / 뒤집힘)</p>
 드론 정보 배터리 잔량 ▾	<p>드론의 남은 배터리량 %로 표시해줌</p>
 드론 ▾ 위치로 회전하기	<p>드론이 센서 영점을 기준으로 특정 각도의 위치로 회전함 (0도 / 90도 / 180도 / -90도)</p>

5. 코드론(CoDrone) 블록 설명

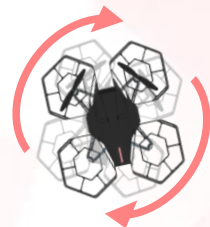
 센서값 롤	드론 기체가 얼마나 좌우로 기울어져 있는지를 각도로 알려줌
 센서값 피치	드론 기체가 얼마나 상하로 기울어져 있는지를 각도로 알려줌
 센서값 요우	드론 기체가 얼마나 좌우로 회전하였는지를 각도로 알려줌
 센서값 고도	드론 기체가 지면을 기준으로 얼마나 떠 있는지를 cm로 알려줌
 센서값 X좌표	호버링 지점으로부터 드론이 이동한 X축의 변위값을 알려줌
 센서값 Y좌표	호버링 지점으로부터 드론이 이동한 Y축의 변위값을 알려줌
 센서값 Z좌표	호버링 지점으로부터 드론이 이동한 Z축의 변위값을 알려줌
 센서값 드론의 온도	드론의 내부 온도를 알려줌
 센서값 기압	드론이 측정한 주위의 공기압을 알려줌



롤



피치

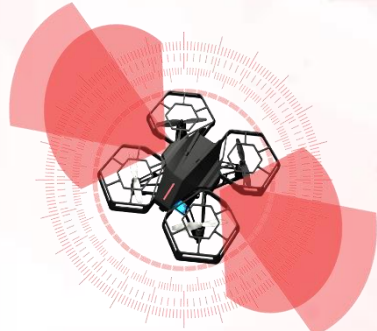


요우

5. 코드론(CoDrone) 블록 설명



정면센서감지



측면센서감지 (옵션)

삼방향 센서값 왼쪽

드론의 삼방향 센서 모듈 중 왼쪽 센서의 거리 측정값

삼방향 센서값 오른쪽

드론의 삼방향 센서 모듈 중 오른쪽 센서의 거리 측정값 (옵션)

삼방향 센서값 전방

드론의 삼방향 센서 모듈 중 전방 센서의 거리 측정값 (옵션)

방향으로 m/s 속도로 초간 비행하기

일정한 좌표의 방향으로 정해진 속도에 정해진 시간동안 비행한 후 진행을 멈추고 호버링함 (방향 옵션은 위의 속도로 제어하기 블록이랑 동일함)

고도 cm로 이동하기

지면을 기준으로 정해진 고도로 드론이 이동하게 됨

LED 색 으로 바꾸기 **켜기**

드론 앞부분의 LED 색을 조절함

로켓 브릭으로 코드론 제어하기

코드론 실습 예제

코드론 실습 전 아래의 사항을 꼭 확인해주세요.

- 1) 리모컨 연결 상태를 확인합니다.
- 2) PC연결 상태를 확인합니다.



1. 코드론2 띄우기 예제

로킷 브릭의 블럭들을 사용하여 다양한 동작들을 만들어 봅니다.
 위쪽 화살표키를 눌렀을 때 잠시 공중에 떴다가 멈추는 예제



이륙부터 5초간 비행 -> 착륙

위쪽 화살표 키를 눌렀을 때

- ↑ 이륙하기
- 5 초 기다리기
- ↓ 착륙하기

스페이스 키를 눌렀을 때

- ↓ 착륙하기

TIP!
 스크립트 작성시에는 비상시를 위해 스페이스 키를 눌렀을 때 착륙할 수 있도록 따로 비상착륙 스크립트를 만들어 놓는 것이 안전합니다!

2. 코드론2 전진 시키기 예제

녹색 깃발 버튼 클릭시 이륙한 후 1m/s의 속도로 2초간 전진하고 착륙하기



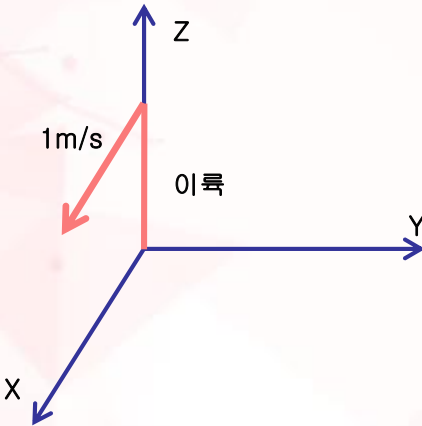
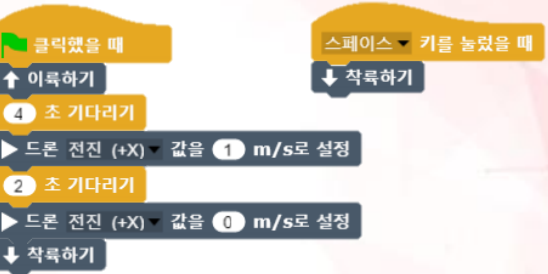
① 이륙



② 4초 대기



⑤ 착륙



-> 이륙부터 4초간 비행 -> 1m/s의 속도로 전진 (X축의 + 방향)
-> 2초 기다린 후 착륙

TIP!

전진 방향 속도를 0으로 만들어 주면서 착륙해야 수직으로 착륙할 수 있습니다.

3. 코드론 2 조합 동작 예제

▶ (녹색 깃발 버튼) 클릭시 2초 동안 우측방향 (Y축의 + 방향)으로 1m/s의 속도로 이동한다음 2초 동안 전진 방향 (X축의 +방향)으로 1m/s로 전진하기

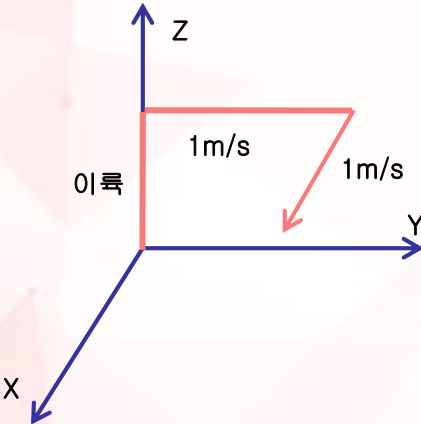


① 이륙

② 우측 이동

③ 전진 이동

④ 착륙



```

클릭했을 때
  ↑ 이륙하기
  4 초 기다리기
  ▶ 드론 우측 (+Y) ▾ 값을 1 m/s로 설정
  2 초 기다리기
  ▶ 드론 모든값 초기화 ▾ 값을 0 m/s로 설정
  1 초 기다리기
  ▶ 드론 전진 (+X) ▾ 값을 1 m/s로 설정
  2 초 기다리기
  ▶ 드론 모든값 초기화 ▾ 값을 0 m/s로 설정
  1 초 기다리기
  ↓ 착륙하기

스페이스 키를 눌렀을 때
  ↓ 착륙하기
  
```

이륙 -> 2초간 1m/s로 우측 이동 -> 2초간 1m/s로 전진 -> 착륙

TIP!

드론의 이동 방향을 바꿀 때 속도와 방향을 클리어(Clear)한후 놓은 후 1초 정도의 시간을 주면 다음 동작에 영향을 줄 수 있는 관성을 없앨 수 있어서 각각의 동작이 뚜렷하게 구분되는 효과를 가져옵니다.

4. 두 가지 성분의 속도를 조합하여 동작하기

🚩 (녹색 깃발 버튼) 클릭시 +X축, +Y축 속도 성분이 각각 1m/s로 하여 1초동안 이동하다가 +X축의 방향으로 1m/s 로 하여 1.5초 동안 전진하기



① 이륙



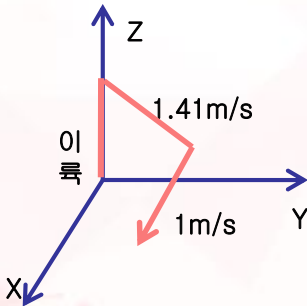
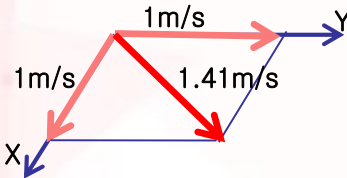
② 대각선 이동



③ 전진 이동



④ 착륙



```

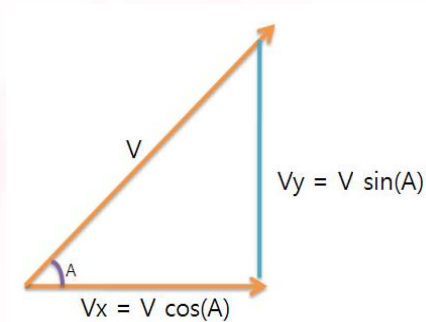
    클릭했을 때
    ↑ 이륙하기
    5 초 기다리기
    ▶ 드론 전진 (+X) 값을 1 m/s로 설정
    ▶ 드론 우측 (+Y) 값을 1 m/s로 설정
    1 초 기다리기
    ▶ 드론 전진 (+X) 값을 0 m/s로 설정
    ▶ 드론 우측 (+Y) 값을 0 m/s로 설정
    1 초 기다리기
    ▶ 드론 전진 (+X) 값을 1 m/s로 설정
    1.5 초 기다리기
    ▶ 드론 전진 (+X) 값을 0 m/s로 설정
    1 초 기다리기
    ↓ 착륙하기
  
```

스페이스 키를 눌렀을 때
↓ 착륙하기

이륙 -> 1초간 비행 (노란 화살표 참조) -> 1.5초간 전진(+X) -> 착륙

TIP!

Q. X 성분의 속도가 1m/s, Y 성분의 속도가 1m/s라면 실제 드론의 진행 속도 (V)는 어떻게 될까요?



여기서 $V_x = 1\text{m/s}$, $V_y = 1\text{m/s}$ 이므로 각도 A는 45도 이고 $V \times \cos(45) = 1$, $V \times \sin(45) = 1$ 입니다. 따라서

$$\begin{aligned} \text{속도} &= \sqrt{V_x^2 + V_y^2} = \sqrt{(V\cos(45))^2 + (V\sin(45))^2} \\ &= \sqrt{1^2 + 1^2} = \sqrt{2} = 1.41\text{m/s} \end{aligned}$$

이므로 속도 성분에 따른 드론의 진행 속도는 1.41m/s가 됩니다.

5. 고도 이동하기

이륙 후 잠시 기다리다가 지면을 기준으로 높이 50cm로 고도를 낮추었다가 다시 1m로 높인 후 1m/s의 속도로 2초간 전진 비행하기



① 이륙



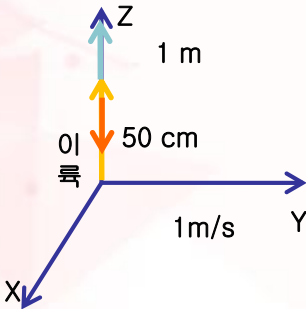
② 우측 이동



③ 전진 이동



④ 착륙



```

    위쪽 회상표 키를 눌렀을 때
    ↑ 이륙하기
    5 초 기다리기
    ▶ 고도 50 cm로 이동하기
    5 초 기다리기
    ▶ 고도 100 cm로 이동하기
    3 초 기다리기
    ▶ 드론 전진 (+X) 값을 1 m/s로 설정
    2 초 기다리기
    ▶ 드론 전진 (+X) 값을 0 m/s로 설정
    1 초 기다리기
    ↓ 착륙하기
  
```

이륙 -> 50cm 높이로 하강 -> 1m 높이로 상승 -> 전진 -> 착륙

TIP!

고도 이동하기 블록을 사용할 때에는 드론이 고도를 이동할 수 있는 충분한 시간을 기다려 주는 것이 좋습니다. 너무 짧은 시간을 주게되면 정해진 고도로 이동하지 못한 상태에서 다른 동작으로 넘어갈 수 있습니다.

6. 드론 LED 제어하기

드론의 LED 가 계속 여러가지 색으로 변화하면서 깜빡이게 하기



다양한 LED 제어 가능

클릭했을 때

무한 반복하기

LED 색 ■ 으로 바꾸기 **켜기** ▼

0.5 초 기다리기

LED 색 ■ 으로 바꾸기 **끄기** ▼

0.5 초 기다리기

LED 색 ■ 으로 바꾸기 **켜기** ▼

0.5 초 기다리기

LED 색 ■ 으로 바꾸기 **끄기** ▼

0.5 초 기다리기

LED 색 ■ 으로 바꾸기 **켜기** ▼

0.5 초 기다리기

LED 색 ■ 으로 바꾸기 **끄기** ▼

0.5 초 기다리기

스페이스 키를 눌렀을 때

↓ 작동하기

모두 ▼ 멈추기

TIP!

LED 제어블록에서 LED 끄기 옵션으로 LED를 끌 때에는 LED 색의 지정은 어느 색이든 상관 없습니다.

7. 드론의 센서값 출력하기

변수블록을 만들어서 자세 제어 값과 배터리 잔량을 캔버스에 표시하기

변수 만들기
변수 삭제하기

- ✓ ROLL
- ✓ PITCH
- ✓ YAW
- ✓ BATTERY

변수 탭 >
변수 만들기 버튼을 클릭 >
4개의 변수 블록 생성

클릭했을 때
무한 반복하기

- 변수 ROLL 에 센서값 롤 저장하기
- 변수 PITCH 에 센서값 피치 저장하기
- 변수 YAW 에 센서값 요우 저장하기
- 변수 BATTERY 에 드론 정보 배터리 잔량 저장하기

스크립트 작성하기 > 드론
연결 > 캔버스 변수 출력값
확인하기

ROLL	10
PITCH	-2
YAW	-2
BATTERY	100



스프라이트 이미지

모양 > 로보링크 카테고리
> codrone2_up.png 파일 선택 >
스프라이트를 코드론2의 모양으로
바꿈

클릭했을 때
무한 반복하기

- 변수 ROLL 에 센서값 롤 저장하기
- 변수 PITCH 에 센서값 피치 저장하기
- 변수 YAW 에 센서값 요우 저장하기
- 변수 BATTERY 에 드론 정보 배터리 잔량 저장하기
- 270 - YAW 도 방향 보기

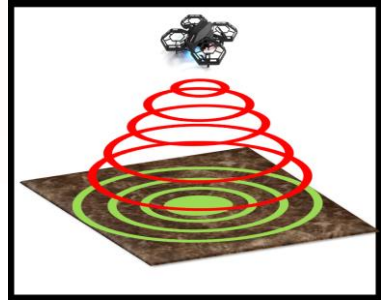
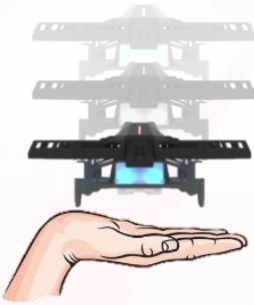
스크립트 추가하고 실행하기
(붉은색 사각형 부분 추가)

TIP!

스프라이트의 캐릭터가 시작시 어느 방향을 기준으로 하고 있는냐에 따라서 각도값의 설정(예제에서는 270도)은 달라질 수 있습니다.

8. 고도 감지 예제

드론이 호버링 하는 중에 드론 아래 부분에 물체나 손이 감지되면 착륙하기



물체 감지

이륙 -> 고도를 약 1m로 유지-> 비행중 센서값 확인 -> 센서값이 20cm이하 -> 착륙

```

    클릭했을 때
    ↑ 이륙하기
    5 초 기다리기
    ▶ 고도 100 cm로 이동하기
    3 초 기다리기
    무한 반복하기
    만약 < 센서값 고도 > < 20 > 라면
    ↓ 착륙하기
    모두 > 멈추기
    
```

```

    스페이스 키를 눌렀을 때
    ↓ 착륙하기

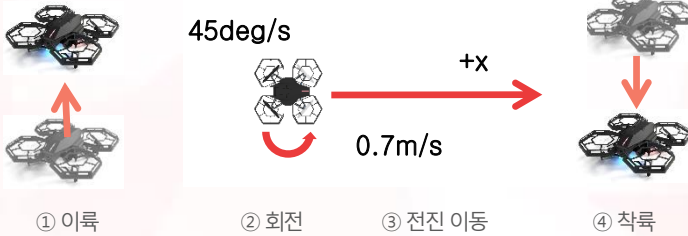
    클릭했을 때
    무한 반복하기
    변수 all 에 < 센서값 고도 > 저장하기
    
```

TIP!

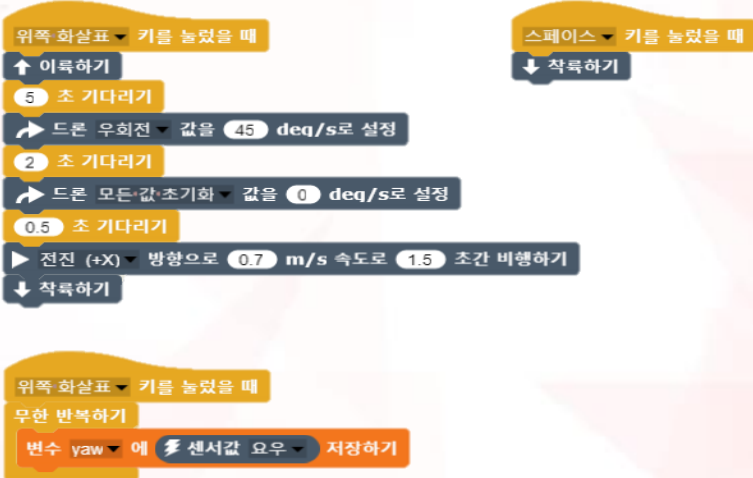
위의 예제에서 처럼 헛 블록은 둘 이상의 스크립트에 사용될 수 있습니다. 이 예제는 깃발 버튼을 클릭했을 때 헛 블록을 두 개의 스크립트에 사용하였습니다. 이 때 깃발 버튼을 누르면 이 블록으로 시작하는 모든 스크립트가 동시에 실행됩니다.

9. 드론이 일정한 각속도로 회전하는 예제

드론이 이륙한 후 원점을 기준으로 오른쪽으로 90도 회전한 후 0.7m/s의 속도로 1.5초 간 비행하기
(회전하는 동안 yaw 값을 관찰 할 수 있도록 변수 블록 사용)



이륙 > 45deg/s 의 속도로 2초간 회전 > 0.7m/s의 속도로 1.5초간 전진 > 착륙



TIP!

위의 예제에서 처럼 헛 블록은 둘 이상의 스크립트에 사용될 수 있습니다. 이 예제는 깃발 버튼을 클릭했을 때 헛 블록을 두 개의 스크립트에 사용하였습니다. 이 때 깃발 버튼을 누르면 이 블록으로 시작하는 모든 스크립트가 동시에 실행됩니다.

10. 전방센서 감지 예제

드론이 이륙한 후 0.3 m/s 로 전진하다가, 전방에 있는 장애물과 50cm 이내로 가까워지면 감지하여 후진했다가 착륙 하기



① 이륙

② 전진 이동

③ 장애물감지

④ 후진하기

⑤ 착륙

이륙 > 0.3m/s 의 속도로 전진 > 센서감지 > 0.5m/s로 후진 > 착륙

클릭했을 때

무한 반복하기

변수 fsensor 에 **삼방향 센서값 FRONT** 저장하기

스페이스 키를 눌렀을 때

↓ 착륙하기

클릭했을 때

↑ 이륙하기

5 초 기다리기

▶ 드론 전진 (+X) - 값을 0.3 m/s로 설정

무한 반복하기

만약 **fsensor < 50** 그리고 **fsensor > 0** 라면

▶ 후진 (-X) - 방향으로 0.5 m/s 속도로 2 초간 비행하기

▶ 드론 모든 값 초기화 - 값을 0 m/s로 설정

2 초 기다리기

↓ 착륙하기

모두 - 멈추기

TIP!

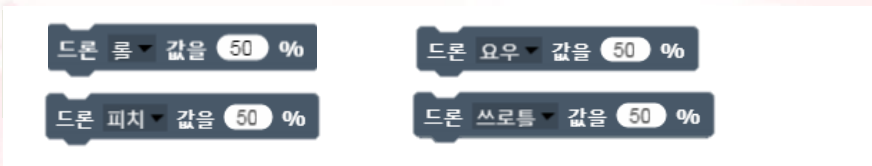
- 전방 센서는 약 2m 정도를 감지합니다. 만약 너무 가까운 거리에 물체가 있거나 2m내에 감지할 물체가 없는 경우 전방 센서 블록은 -1을 출력하게 됩니다. 따라서 감지 거리를 측정하기 전에 센서값이 0보다 큰지를 확인해야 합니다
- 사방이 둘러싸인 벽을 감지하려 할 때는 드론이 발생시킨 바람이 나아갈 곳이 없어 역풍이 심해져 드론의 비행이 불안정해질 수 있습니다. 벽을 감지시킬 때에는 감지 거리를 더욱 여유있게 잡아야 합니다.

11. 오일러 각으로 드론 제어하기

오일러 각도를 직접 변경하여 키보드로 드론을 조종할 수 있는 스크립트 작성하기

이륙 > 45deg/s 의 속도로 2초간 회전 > 0.7m/s의 속도로 1.5초간 전진 > 착륙

Roll / Pitch / Yaw / Throttle 블록은 아래의 그림과 같이 사용하면 된다.
 또한 코드론 구동 시 맨 처음 블록은 **↑ 이륙하기** 혹은 **Throttle** 제어 로 시작하고 가장 마지막은 **↓ 착륙하기** 으로 해야 안정적이고 제대로 된 코드론 제어를 할 수 있다.



그리고 원하는 **지속시간을 추가**하여 더 안정적인 동작을 만들 수 있다.



TIP!
 조종시 드론의 속도와 힘을 조절하는 변수는 POW입니다. 이 값은 각 오일러 각의 출력값으로서 절대값 100이 최대입니다. 예제에서는 최대 출력(속도)의 50%만 사용할 수 있도록 POW값을 50으로만 설정하였습니다.

코드론의 기본 동작

모든 값은 -100부터 100사이의 값을 갖는다.

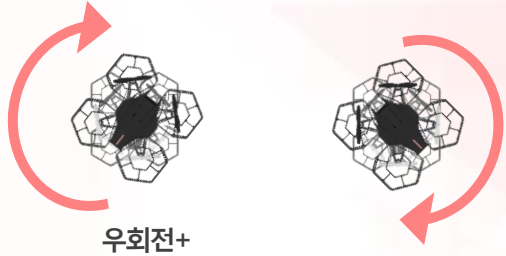
드론 쓰로틀 ▼ 값을 50 %

throttle : 상하 수직 이동



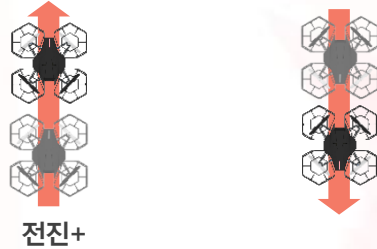
드론 요우 ▼ 값을 50 %

yaw : 좌회전, 우회전 이동



드론 피치 ▼ 값을 50 %

pitch : 전진, 후진 이동



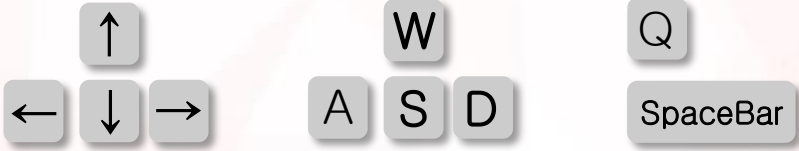
드론 롤 ▼ 값을 50 %

roll : 좌측, 우측 이동



코드론 조종하기

키보드로 드론 조종하기



클릭했을 때

변수 POW에 50 저장하기 → 이동 명령 증가량 지정

변수 pitch에 0 저장하기

변수 roll에 0 저장하기

변수 yaw에 0 저장하기

변수 throttle에 0 저장하기

무한 반복하기

만약 임의의 키를 눌렀는가? 라면

만약 위쪽 화살표 키를 눌렀는가? 라면

변수 pitch에 POW 저장하기 → Pitch에 50 대입

드론 피치 값을 pitch % → Pitch + 50 (전진)

만약 아래쪽 화살표 키를 눌렀는가? 라면

변수 pitch에 0 - POW 저장하기 → Pitch에 -50 대입

드론 피치 값을 pitch % → Pitch에 -50 (후진)

만약 왼쪽 화살표 키를 눌렀는가? 라면

변수 roll에 0 - POW 저장하기 → roll에 +50 대입

드론 롤 값을 roll % → Pitch에 +50 (좌측이동)

만약 오른쪽 화살표 키를 눌렀는가? 라면

변수 roll에 POW 저장하기 → roll에 -50 대입

드론 롤 값을 roll % → Pitch에 -50 (우측이동)

스페이스 키를 눌렀을 때 → 착륙

↓ 착륙하기

q 키를 눌렀을 때 → 이륙

↑ 이륙하기

만약 s 키를 눌렀는가? 라면

- 변수 throttle 에 **- POW** 저장하기 ▶ Throttle 에 - 50 대입
- 드론 쓰로틀 값을 **throttle %** ▶ Throttle 에 - 50 (하강)

만약 w 키를 눌렀는가? 라면

- 변수 throttle 에 **POW** 저장하기 ▶ Throttle 에 + 50 대입
- 드론 쓰로틀 값을 **throttle %** ▶ Throttle 에 + 50 (상승)

만약 a 키를 눌렀는가? 라면

- 변수 yaw 에 **- POW** 저장하기 ▶ Yaw 에 - 50 대입
- 드론 요우 값을 **yaw %** ▶ Yaw 에 - 50 (좌회전)

만약 d 키를 눌렀는가? 라면

- 변수 yaw 에 **POW** 저장하기 ▶ Yaw 에 + 50 대입
- 드론 요우 값을 **yaw %** ▶ Yaw 에 + 50 (우회전)

아니면

- 변수 pitch 에 **0** 저장하기
- 변수 roll 에 **0** 저장하기
- 변수 yaw 에 **0** 저장하기
- 변수 throttle 에 **0** 저장하기

롤 roll 피치 pitch 요우 yaw 쓰로틀 throttle ▶ Pitch, Roll, Yaw, Throttle 에 0대입

로켓 브릭으로 코드론 제어하기

부 록



1. 리모컨과 USB 드라이버 설치하기 (윈도우 7과 윈도우 8에서 사용)

PC(노트북)와 리모컨이 연결된 후 코딩 모드로 전환되기 위해서는 리모컨과 PC가 통신을 할 수 있게 해 주는 드라이버를 설치해야 합니다.

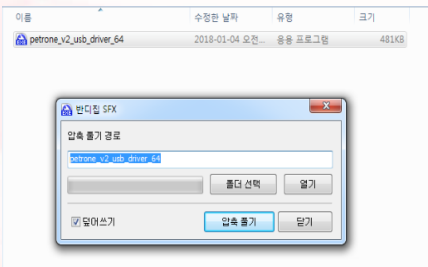
윈도우 10에서는 자동 설치가 되어 따로 드라이버 설치를 할 필요가 없지만 윈도우7과 윈도우8에서는 사용자가 아래와 같이 수동 설치를 해 주셔야 합니다.



마이크로 USB 케이블을 PC(노트북)에 연결합니다.



마이크로 USB 단자를 리모컨의 USB 포트에 연결해주세요

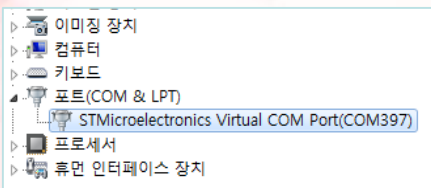


파일의 압축을 해제하고 응용 프로그램을 실행하면 위와 같이 압축 풀기 창이 나옵니다. 이 때 압축 풀기를 클릭하면 드라이버 설치가 시작됩니다.



Download 다운로드
필요한 프로그램을 다운받아주세요!

USB 드라이버를 다운 받습니다 (다운로드위치: 로보링크다운로드) 이 때 사용자의 PC가 32비트 인지 64비트 인지에 따라 다운받을 파일을 선택하세요



설치후 혹은 설치 중에도 장치 관리자 > 포트에서 왼쪽과 같은 장치 이름이 나옵니다. 그러나 설치된 드라이버가 활성화 될 때까지는 몇분 정도 더 걸릴 수 있습니다.

2. 코드론 2 페어링 하기

페어링이란 드론과 리모컨의 통신 설정을 같게 하여 짝을 지어 주는 것을 말합니다. 이렇게 하면 다른 장치들과의 간섭없이 두 장치들끼리만 통신을 주고 받을 수 있게 됩니다.

페어링이 필요한 경우는 사용중에 페어링 상태가 해제 되거나 새로운 드론이나 리모컨을 구입하게 되어 기존의 장치와 함께 사용해야 하는 경우 등이 있습니다.

페어링을 하시려면 먼저 드론의 전원과 리모컨의 전원이 전부 켜져 있어야 합니다. 이 상태에서 아래 그림과 같이 드론의 밑부분의 버튼을 길게(4초 정도) 눌러 주세요.



페어링 모드로 진입하면 드론의 앞 부분의 LED가 녹색에서 노란색으로 점멸하게 됩니다. 이 후 아래와 같이 페어링 버튼을 눌러 줍니다.



페어링이 성공하면 노란색으로 점멸하던 드론의 LED가 녹색으로 켜져 있게 되며 리모컨의 디스플레이에도 더 이상 disconnect 메시지가 뜨지 않게 됩니다. 이후에는 드론을 리모컨으로 제어할 수 있게 됩니다.

주의사항!

- 페어링은 한번만 해 주면 계속 페어링 상태가 유지됩니다. 드론이나 리모컨을 켤때마다 페어링을 하실 필요가 없습니다
- 위의 절차대로 했음에도 페어링이 잘 되지 않은 경우 처음부터 다시 시도해 보시기 바랍니다. 상황에 따라 페어링이 한번에 되지 않고 여러 번 시도해야 하는 경우도 있습니다.

3. 드론의 좌표계

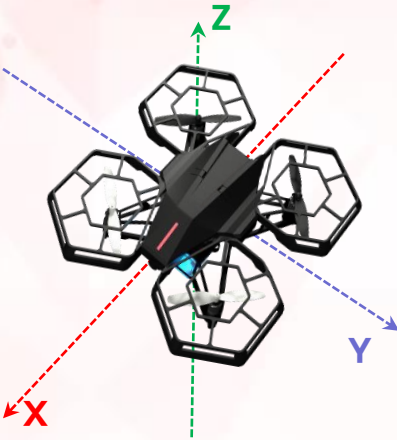


그림1 이동 할때 좌표의 방향

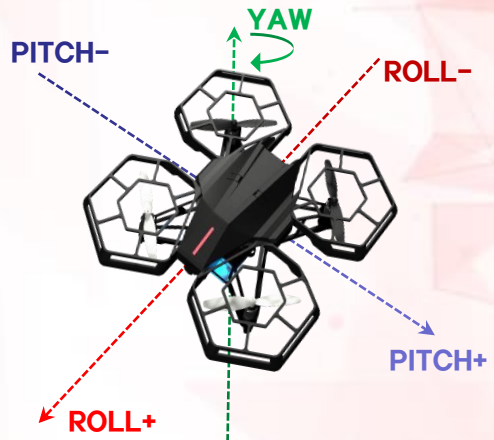


그림2 회전할 때 좌표와 방향

드론의 움직임은 3D 좌표를 기준으로 구분할 수 있습니다. 드론의 전진과 후진시에는 X축을 따라서 움직이고 왼쪽이나 오른쪽으로 평행 이동할 때에는 Y축을 따라서 움직입니다. 그리고 드론이 상승하거나 하강 할 때에는 Z축을 따라 움직이는 것으로 봅니다.

좌표를 기준으로 움직이는 것으로 보면 이동하는 방향에 따라 축을 기준으로 + 방향과 - 방향으로 구분할 수도 있습니다. (왼쪽 표 참조)

드론 기체의 기울어짐(회전)도 3D 좌표를 따라 이해할 수 있습니다. X축을 회전축으로 하여 회전한 각도를 롤(Roll)이라 하고 Y축을 회전축으로 회전한 각도를 피치(pitch)라 합니다. 그리고 Z축을 기준으로 회전한 각도를 요우(yaw)라고 합니다. 이 때 그림2 에서 표시된 회전 방향(반시계)을 + 회전으로 하고 반대 방향을 -회전으로 합니다.



www.RobolinkSW.com

[Youtube.com/CoDrone](https://www.youtube.com/CoDrone)

- 온라인 사이트에서 매뉴얼과 프로그램 다운로드, 교육 자료 등을 확인해주세요..
- 로보링크 SW / Arduino 등이 오픈 소스로 제공 됩니다.

