



# 스크래치로 배우는 비글



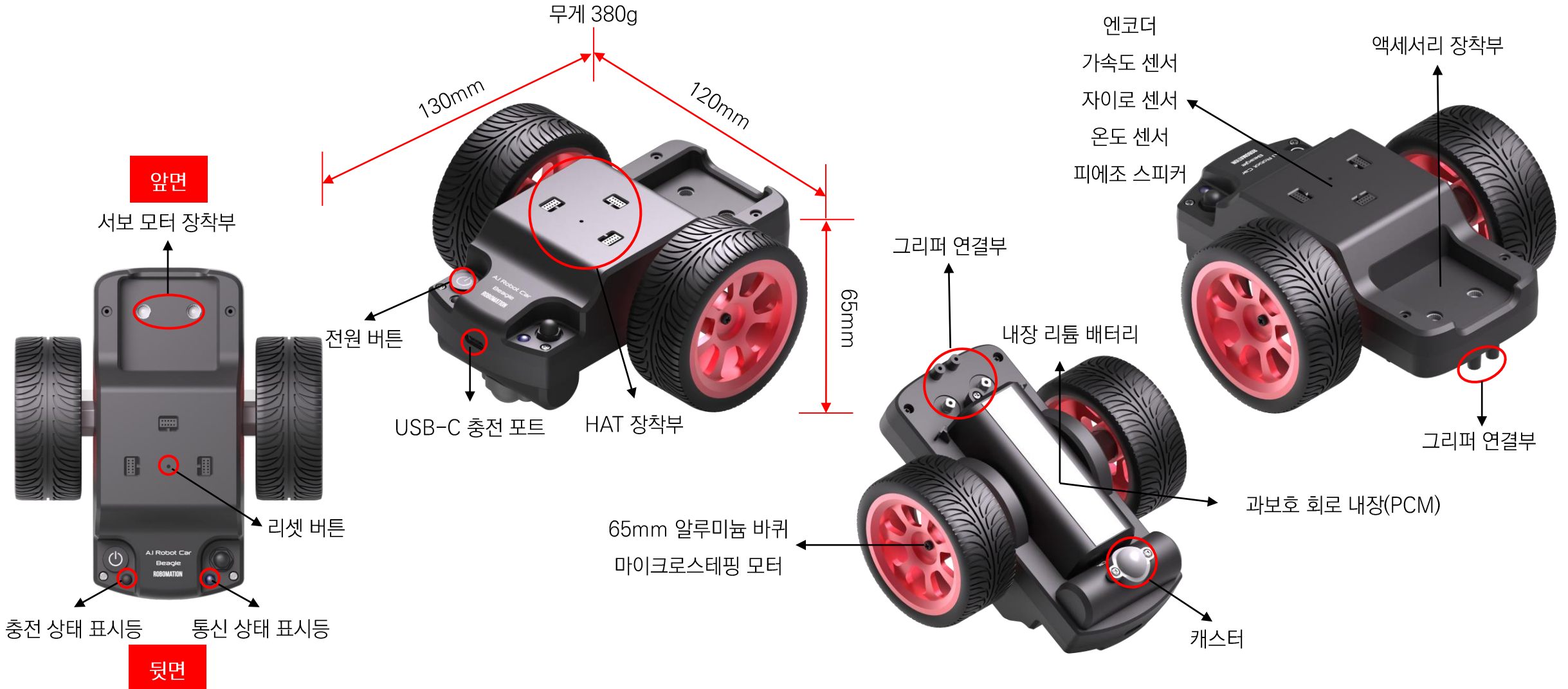
# 목차

1차시	수업 준비	<a href="#">3</a>
2차시	블록 설명	<a href="#">17</a>
3차시	이동 블록 코드	<a href="#">40</a>
4차시	센서 블록 코드	<a href="#">75</a>

1차시

# 수업 준비

- 비글은 교육용 로봇입니다. 다음 그림과 같이 다양한 장치를 포함하고 있습니다.



- 전원이 꺼진 상태에서 전원 버튼을 한 번 누르면 전원 버튼의 LED가 빨간색으로 점등되면서 전원이 켜집니다.
- 전원이 켜진 상태에서 전원 버튼을 1초 이상 누르고 있으면 부저음이 울리고 LED가 꺼지면서 전원이 꺼집니다.
- 배터리가 부족하면 전원 버튼의 LED가 깜빡 거립니다.
- 배터리가 부족한 상태에서는 시스템이 제대로 동작하지 않으므로 반드시 충전해 주시기 바랍니다.



오작동 시 로봇 상단의 리셋 버튼 부분을 핀으로 딸각 하도록 누르면 전원이 꺼지고 시스템이 초기화됩니다.

전원을 켜 LED에 불이 들어온 모습입니다.

충전 중에는 빨간  
불이 들어옵니다.



- 비글은 스마트 폰용 충전기를 사용하여 충전할 수 있습니다. USB-C 단자를 비글의 충전 포트에 연결하면 됩니다.
- USB 케이블을 사용하여 충전할 수도 있습니다. USB 케이블의 USB-C 단자를 비글의 충전 포트에 연결하고, 반대쪽을 컴퓨터의 USB 포트에 연결합니다.
- 7V 이상의 과전압이 공급되면 사이렌 소리로 경고하고 전원을 차단합니다. 이 때는 코드를 뽑고 5V 케이블을 사용해 주시기 바랍니다.
- 충전 중에는 충전 상태 표시등이 빨간색으로 표시되고, 충전이 완료되면 충전 상태 표시등이 꺼집니다.
- 완전히 충전하면 약 70분 동안 사용할 수 있습니다.
- 완충까지는 약 4시간이 걸리며, 전원을 끈 상태에서 충전합니다.



전원을 켜면 파란 불이  
들어옵니다.

※ 비글 로봇은 자사의 타 로봇과 페어링 시 사용하는 블루투스 동글과는 연결할 수 없으므로 꼭 익스프레스 리시버를 사용해 주시기 바랍니다.

- 파란색으로 천천히 깜박임**  
 연결을 기다리는 상태입니다.  
 비글 로봇의 전원을 켜면 통신 상태 표시등이 파란색으로 천천히 깜박입니다.
- 파란색으로 계속 켜져 있음**  
 연결된 상태입니다.  
 익스프레스 리시버와 페어링되면 통신 상태 표시등이 계속 켜져 있습니다.
- 파란색으로 빠르게 깜박임**  
 정상적으로 연결 후 통신을 하고 있는 상태입니다.  
 데이터를 주고 받는 동안에는 통신 상태 표시등이 파란색으로 빠르게 깜박입니다.
- 통신 상태 표시등이 꺼짐**  
 로봇의 전원이 꺼진 상태입니다.



- 다음 그림과 같이 나노 라이더를 연결할 수 있는 확장 포트를 지니고 있습니다.





- 다음 그림과 같이 햄스터 AI 카메라를 거치할 수 있습니다.

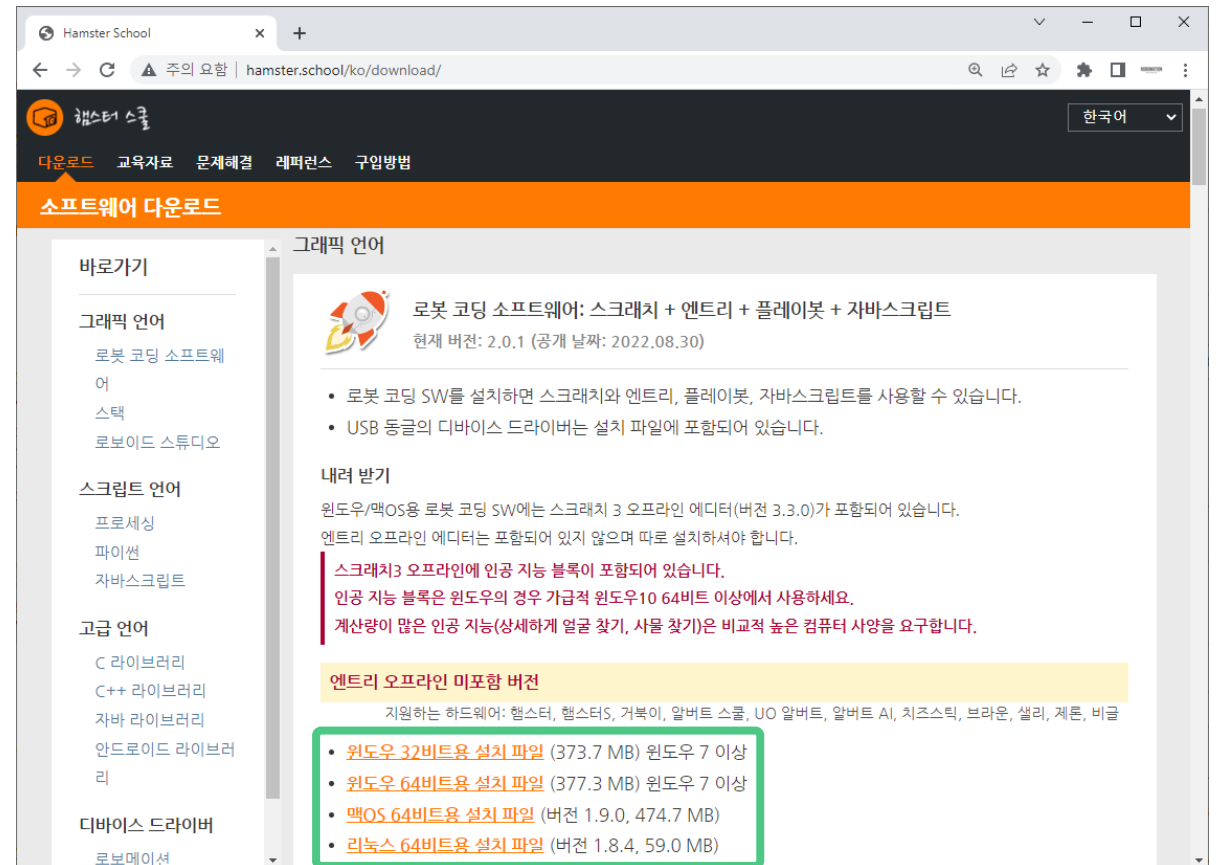
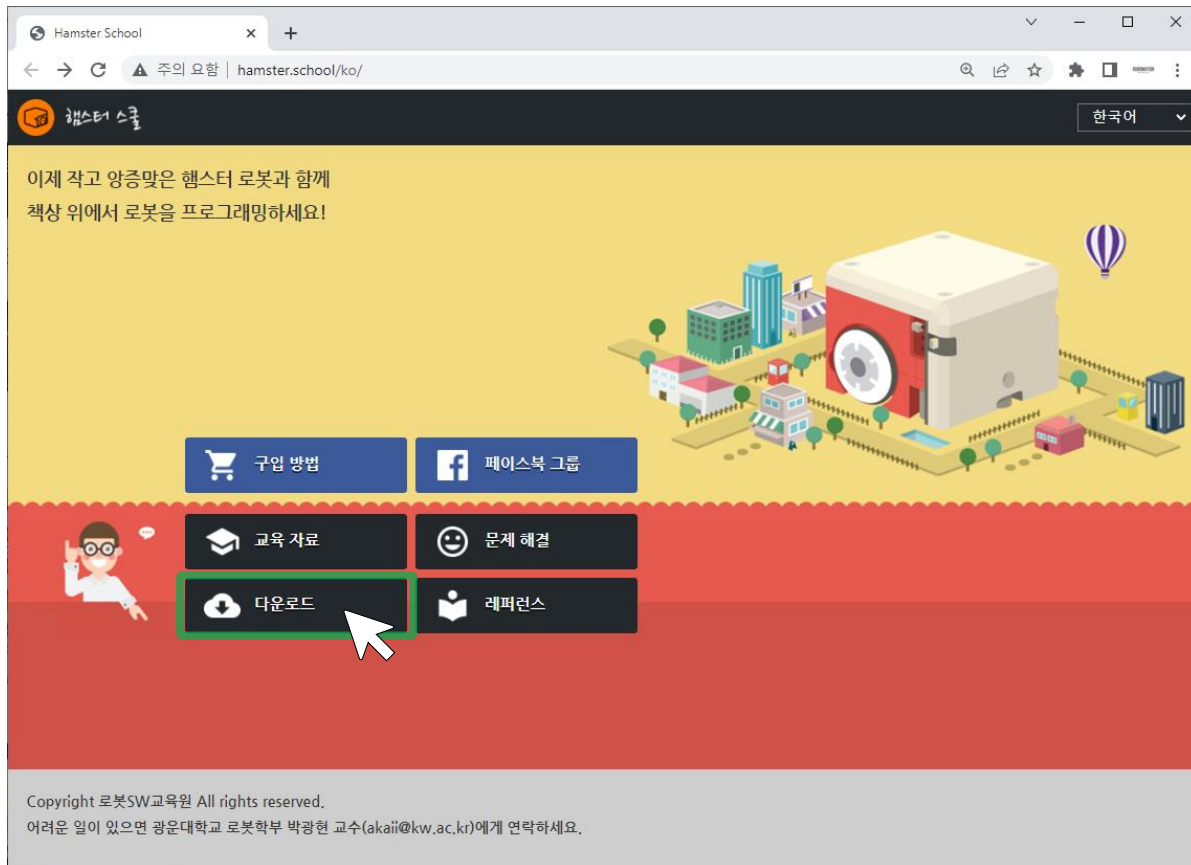


햄스터 AI 카메라 제품에 동봉된 재접착 양면 테이프를 붙여 고정할 수 있습니다.  
비글용 카메라 거치대(별도 판매)를 활용하여 고정할 수도 있습니다.

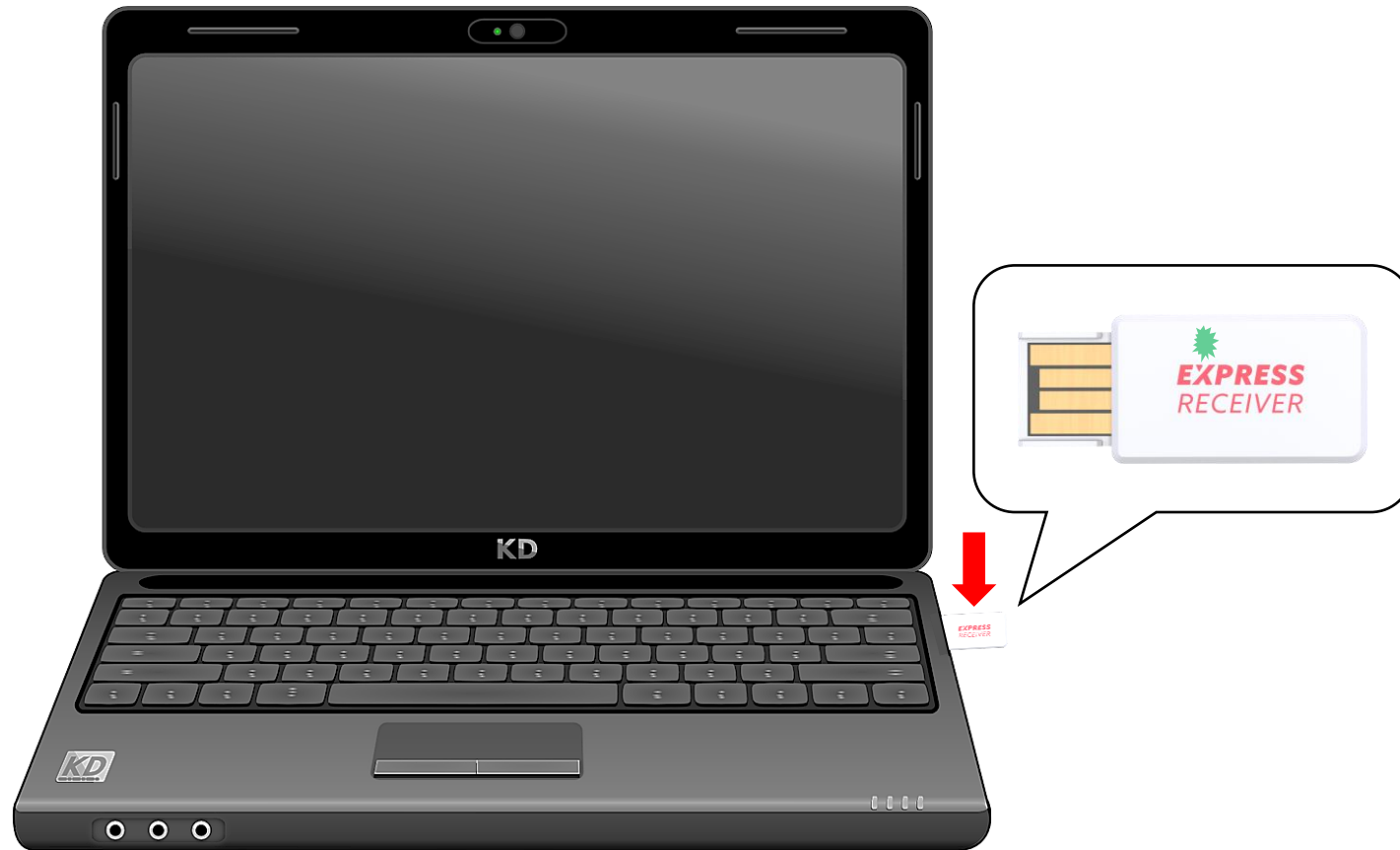
- 나도 라이다와 햄스터 AI 카메라를 동시에 연결해 활용할 수도 있습니다.



- 햄스터스쿨(<https://hamster.school>)에 접속해 다운로드를 클릭합니다.
- 로봇 코딩 소프트웨어에서 본인 pc의 사양에 맞는 버전을 다운받습니다.



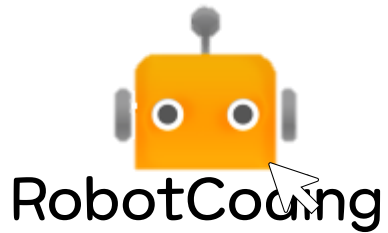
- 비글과의 페어링을 위해 익스프레스 리시버를 PC의 USB 단자에 꽂습니다.
- 익스프레스 리시버의 연결 상태 표시등이 초록색으로 천천히 깜박이면 정상입니다.



- 익스프레스 리시버에 가까이 가져가 비글의 전원 버튼을 눌러 전원을 켭니다.
- 비글에서 뽁 소리가 나고 비글의 통신 상태 표시등과 익스프레스 리시버의 연결 상태 표시등이 계속 켜져 있거나 빠르게 깜빡이면 정상입니다.



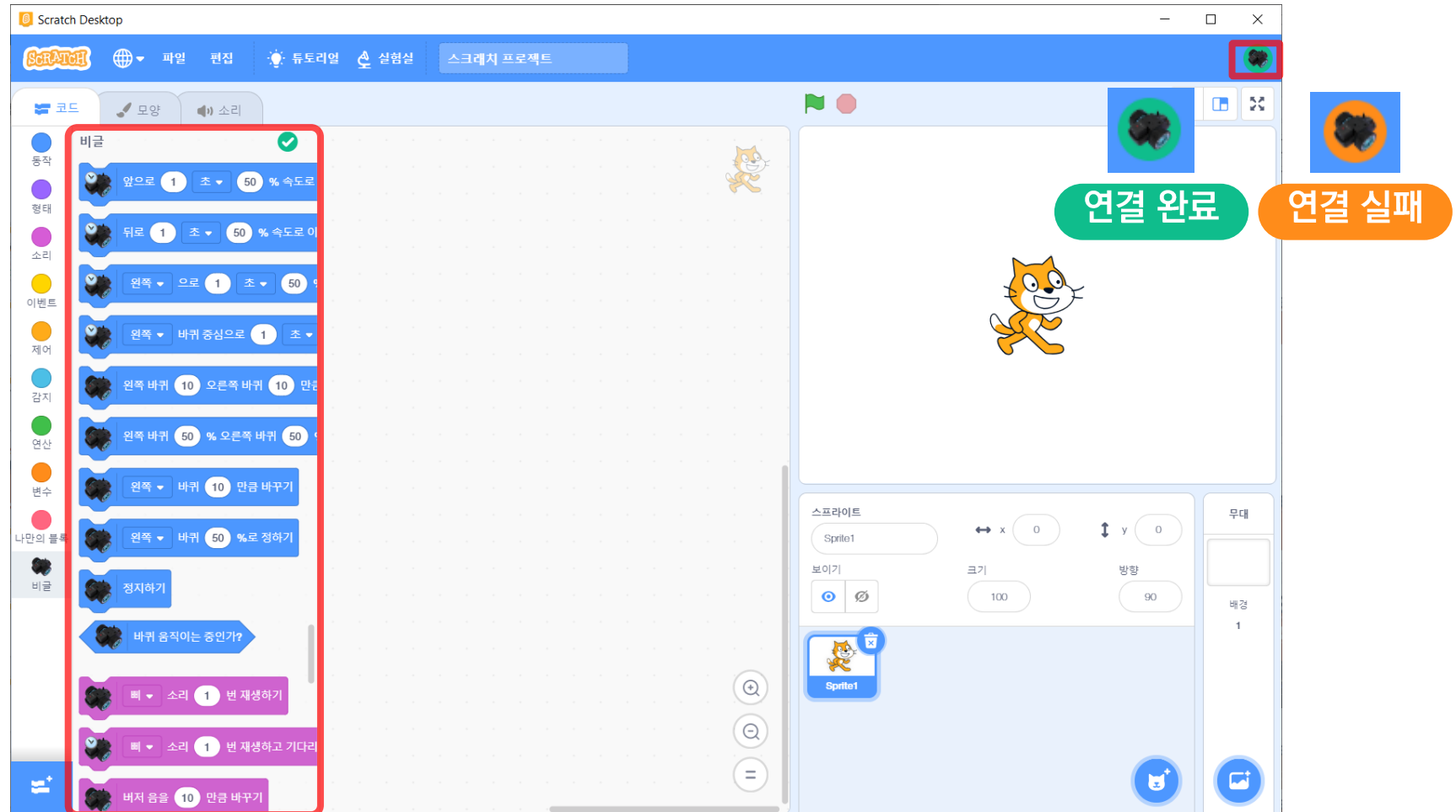
- 로봇 코딩 소프트웨어를 실행합니다.(설치된 로봇 코딩 프로그램의 버전이 2.0.1 이상인지 확인합니다.)
- 연결된 로봇의 정보를 확인합니다.



- 스크래치 3 오프라인을 클릭합니다.
- 단일 로봇을 선택하고 스크래치3 실행을 클릭합니다.



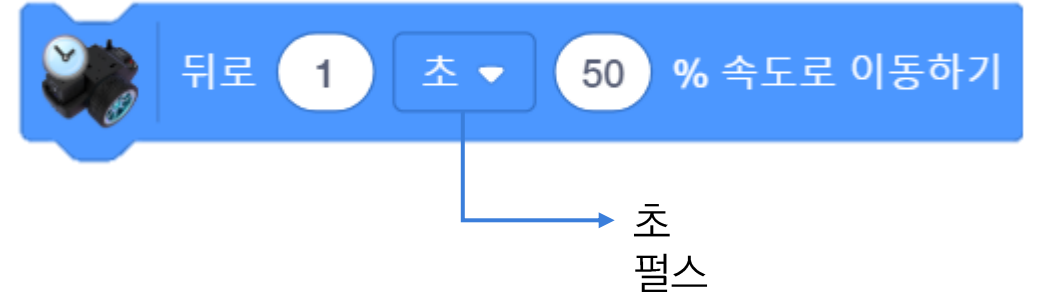
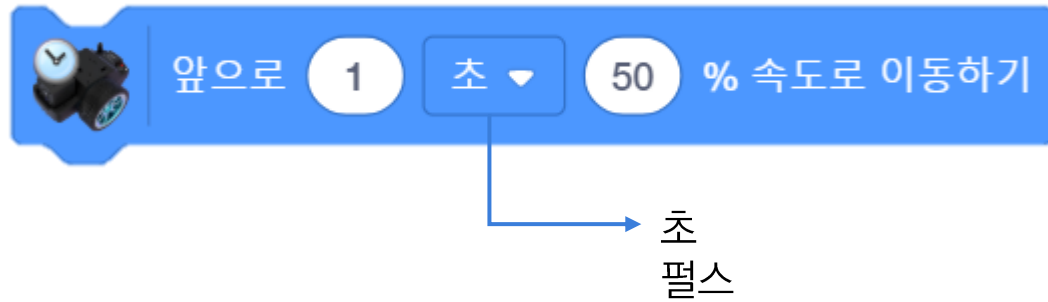
- 연결이 제대로 된 상태로 프로그램이 실행되면 비글 관련 블록이 나타납니다.
- 오른쪽 위 비글 아이콘의 배경 색으로 로봇 연결 상태를 확인할 수 있습니다.





2차시

# 블록 설명



앞으로/뒤로 입력한 시간(초, 펄스) 동안 입력한 속도(%)로 이동합니다.

#### 첫 번째 파라미터

- 이동할 시간 또는 펄스(양수 값)

#### 두 번째 파라미터

- 초: 입력한 시간(초) 동안 이동합니다.
- 펄스: 입력한 펄스만큼 이동합니다.

#### 세 번째 파라미터

- 양쪽 바퀴의 속도 값(-100 ~ 100)



왼쪽/오른쪽으로 입력한 시간(초, 펄스) 동안 입력한 속도(%)로 제자리에서 회전합니다.

#### 첫 번째 파라미터

- 왼쪽: 왼쪽 방향으로 회전합니다.
- 오른쪽: 오른쪽 방향으로 회전합니다.

#### 두 번째 파라미터

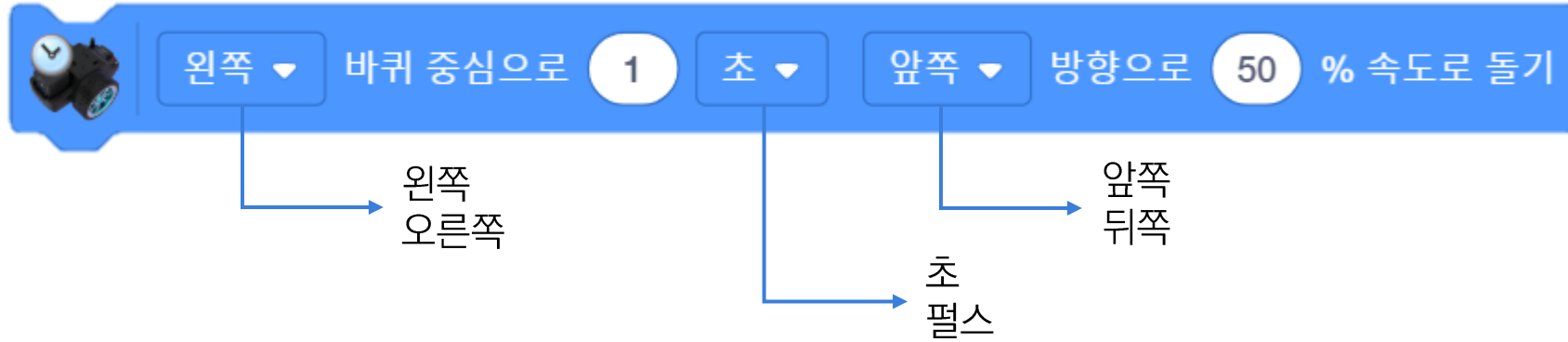
- 이동할 시간 또는 펄스(양수 값)

#### 세 번째 파라미터

- 초: 입력한 시간(초) 동안 이동합니다.
- 펄스: 입력한 펄스만큼 이동합니다.

#### 네 번째 파라미터

- 양쪽 바퀴의 속도 절대값(-100 ~ 100)



왼쪽/오른쪽 바퀴를 중심으로 입력한 시간(초, 펄스) 동안 앞쪽/뒤쪽 방향으로 입력한 속도(%)로 회전합니다.

#### 첫 번째 파라미터

- 왼쪽: 왼쪽 바퀴를 중심으로 회전합니다.
- 오른쪽: 오른쪽 바퀴를 중심으로 회전합니다.

#### 두 번째 파라미터

- 회전할 시간 또는 펄스(양수 값)

#### 세 번째 파라미터

- 초: 입력한 시간(초) 동안 회전합니다.
- 펄스: 입력한 펄스만큼 회전합니다.

#### 네 번째 파라미터

- 앞쪽: 앞쪽 방향으로 회전합니다.
- 뒤쪽: 뒤쪽 방향으로 회전합니다.

#### 다섯 번째 파라미터

- 왼쪽/오른쪽 바퀴의 속도 값(-100 ~ 100)



왼쪽 바퀴 10 오른쪽 바퀴 10 만큼 바꾸기

왼쪽 바퀴와 오른쪽 바퀴의 현재 속도(%)에 입력한 값을 각각 더합니다.

더한 결과가 양수 값이면 바퀴가 앞으로 회전하고, 음수 값이면 뒤로 회전합니다.

첫 번째 파라미터

- 왼쪽 바퀴의 현재 속도 값(-100 ~ 100)에 더할 값

두 번째 파라미터

- 오른쪽 바퀴의 현재 속도 값(-100 ~ 100)에 더할 값



왼쪽 바퀴 50 % 오른쪽 바퀴 50 %로 정하기

왼쪽 바퀴와 오른쪽 바퀴의 속도를 입력한 값으로 각각 설정합니다.

양수 값을 입력하면 바퀴가 앞으로 회전하고, 음수 값을 입력하면 뒤로 회전합니다.

숫자 0을 입력하면 정지합니다.

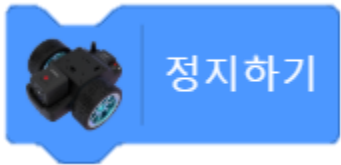
입력하는 값은 최대 속도를 기준으로 한 % 값입니다.

첫 번째 파라미터

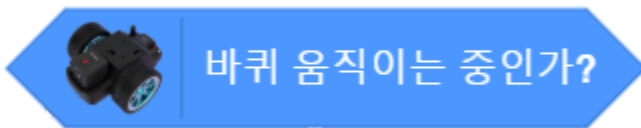
- 왼쪽 바퀴의 속도 값(-100 ~ 100)

두 번째 파라미터

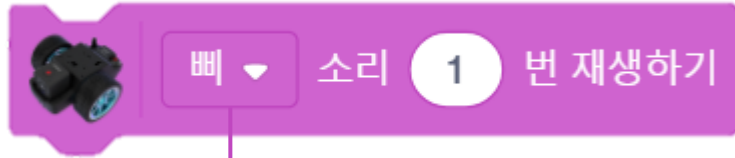
- 오른쪽 바퀴의 속도 값(-100 ~ 100)



양쪽 바퀴를 정지합니다.



바퀴가 움직이고 있으면 '참'으로 판단하고, 아니면 '거짓'으로 판단합니다.



삐	잘 했어요
무작위 삐	행복
지지직	화남
사이렌	슬픔
엔진	즐림
쩍	행진
로봇	생일
디비디비답	

선택한 소리를 입력한 횟수만큼 재생합니다.

-1을 입력하면 계속 반복하여 재생합니다.

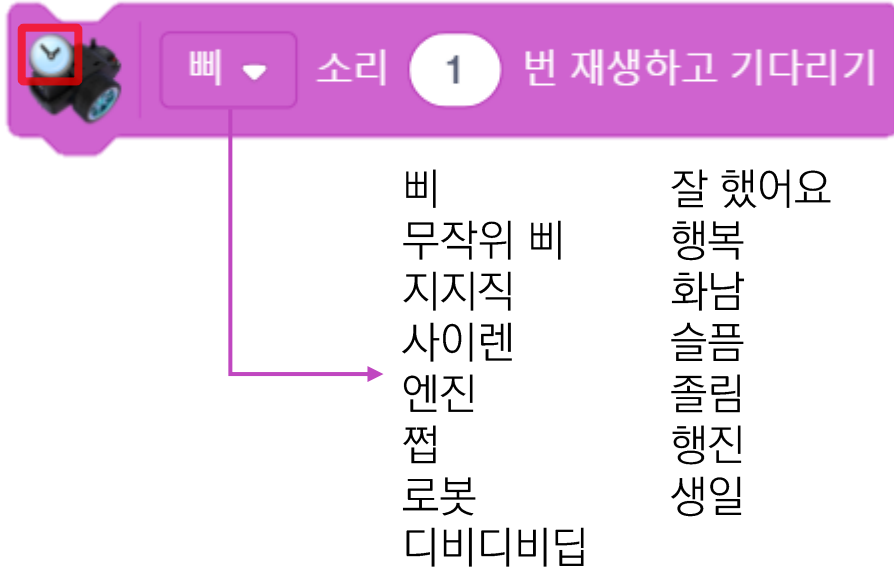
첫 번째 파라미터

- 재생할 소리

두 번째 파라미터

- 재생할 횟수(-1 또는 양수 값)





선택한 소리를 입력한 횟수만큼 재생하고, 재생이 완료될 때까지 기다립니다.

-1을 입력하면 계속 반복하여 재생합니다.

첫 번째 파라미터

- 재생할 소리

두 번째 파라미터

- 재생할 횟수(-1 또는 양수 값)



버저 음을 10 만큼 바꾸기

버저 소리의 현재 음 높이(Hz)에 입력한 값을 더합니다.

0.01 Hz 단위로 주파수를 지정하고 소수점 둘째 자리까지 입력할 수 있습니다.

파라미터

- 버저 소리의 현재 음 높이(Hz)에 더할 값



버저 음을 1000 Hz로 정하기

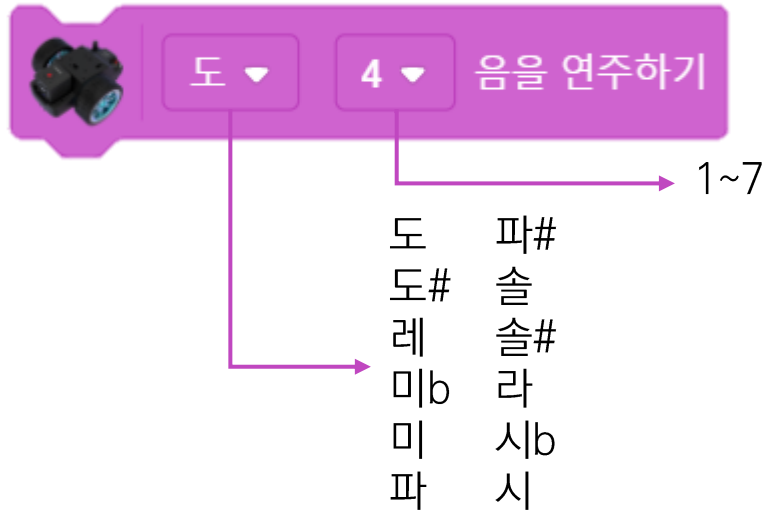
버저 소리의 음 높이를 입력한 값(Hz)으로 설정합니다.

0.01 Hz 단위로 주파수를 지정하고 소수점 둘째 자리까지 입력할 수 있습니다.

숫자 0을 입력하면 버저 소리를 끕니다.

#### 파라미터

- 버저 소리의 음 높이 (1.00Hz ~ 165000Hz)



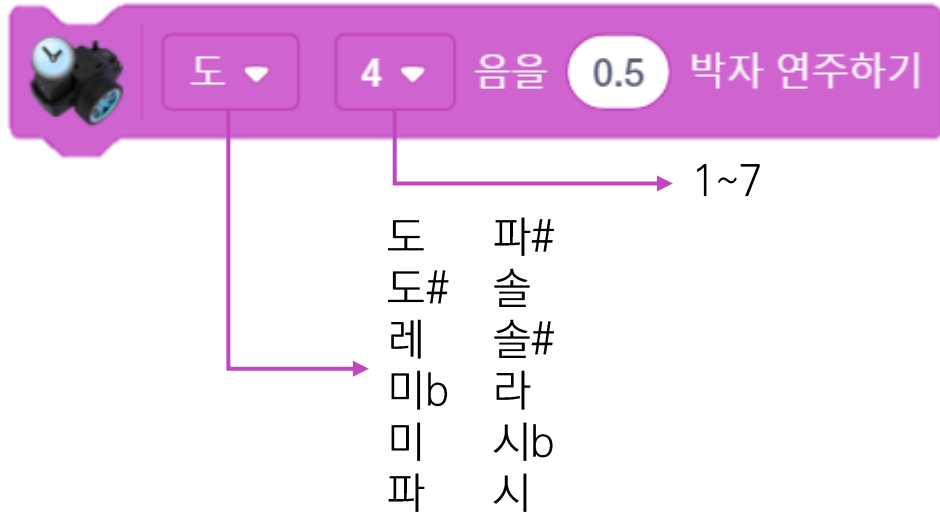
선택한 계이름과 옥타브의 음을 계속 소리 냅니다.

첫 번째 파라미터

- 계이름(84 key)

두 번째 파라미터

- 옥타브



선택한 계이름과 옥타브의 음을 지정한 박자 동안 소리 냅니다.

첫 번째 파라미터

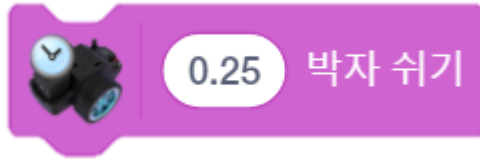
- 계이름(84 key)

두 번째 파라미터

- 옥타브

세 번째 파라미터

- 박자(0 이상)



입력한 박자만큼 쉬니다.

파라미터

- 박자(0 이상)



연주 속도를 20 만큼 바꾸기

연주하거나 쉬는 속도의 현재 BPM(분당 박자 수)에 입력한 값을 더합니다.

#### 파라미터

- 현재 BPM(분당 박자 수)에 더할 값



연주 속도를 60 BPM으로 정하기

연주하거나 쉬는 속도를 입력한 BPM(분당 박자 수)으로 설정합니다.

파라미터

- 연주하거나 쉬는 속도 (BPM, 분당 박자 수)





소리 끄기

버저 소리 또는 재생하는 소리 등 모든 소리를 끕니다.



소리 내는 증인가?

소리를 내고 있으면 ‘참’으로 판단하고, 아니면 ‘거짓’으로 판단합니다.



엔코더 값 초기화하기



왼쪽 엔코더



오른쪽 엔코더

비글이 이동하면서 이동한 step수 만큼 펄스 값이 증가합니다.

카운트 값은 정방향은 증가하고 역방향은 감소합니다.

카운팅을 초기화하기 위해서 엔코더 값 초기화하기 블록을 실행하면 값이 0으로 초기화 됩니다.

SLAM 활동을 할 때 라이다와 함께 사용해 비글의 위치를 확인하는데 활용할 수 있습니다.



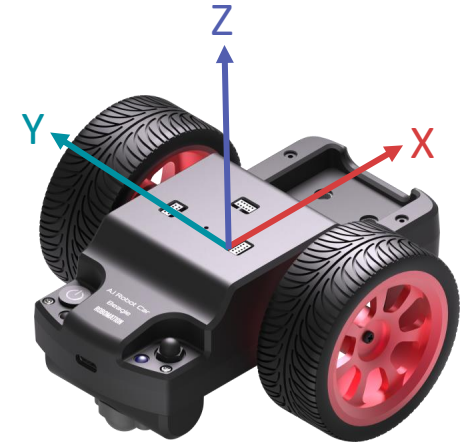
가속도 센서 x (g)



가속도 센서 y (g)



가속도 센서 z (g)



가속도 센서의 동작 상태를 알려줍니다.

가속도 센서의 측정 범위는  $\pm 2g$ (12비트 정밀도)입니다.

X축: 비글을 앞, 뒤 방향으로 기울이는 동작을 할 때 변화합니다.(기본 값: 약 0)

Y축: 비글을 왼쪽, 오른쪽 방향으로 기울이는 동작을 할 때 변화합니다.(기본 값: 약 0)

Z축: 비글을 위, 아래 방향으로 뒤집을 때 변화합니다.(기본 값: -1)



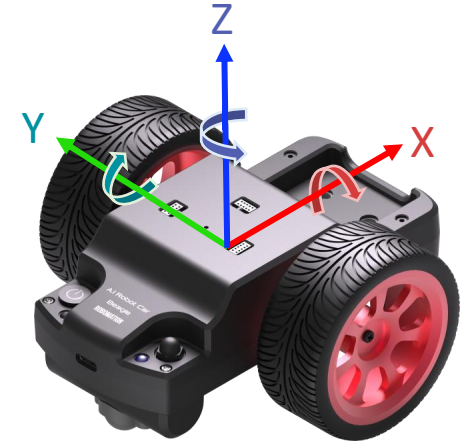
자이로 센서 x (°/s)



자이로 센서 y (°/s)



자이로 센서 z (°/s)



자이로 센서의 동작 상태를 알려줍니다.

자이로 센서의 값은 각 축에 대한 각속도로 초당 회전하는 속도와 방향을 값으로 나타냅니다.

자이로 센서의 측정 범위는  $\pm 250^\circ/\text{s}$ (16비트 정밀도)입니다.

X축: 비글을 왼쪽, 오른쪽 방향으로 기울이는 동작을 할 때 변화합니다.(기본값: 약 0)

Y축: 비글을 앞, 뒤 방향으로 기울이는 동작을 할 때 변화합니다.(기본값: 0)

Z축: 비글을 수평하게 놓고 좌우로 돌릴 때 변화합니다.(기본값:0)



온도 (°C)

비글 로봇 내부의 온도 값을 나타냅니다.  
측정 범위는 섭씨-40도에서+88도입니다.



신호 세기 (dBm)

무선 통신의 신호 세기(-128 ~ 0 dBm)를 나타냅니다.  
거리 및 방향에 따라 변화하고 신호의 세기가 셀수록 값이 커집니다.



앞으로 기울였을 때	오른쪽으로 기울였을 때
뒤로 기울였을 때	거꾸로 뒤집었을 때
왼쪽으로 기울였을 때	기울이지 않았을 때

비글의 기울임 방향의 변화를 가속도 센서로 감지하여 로봇을 움직이는 방향이 해당 상태일 때를 감지합니다.



앞으로 기울임 ?	오른쪽으로 기울임 ?
뒤로 기울임 ?	거꾸로 뒤집음 ?
왼쪽으로 기울임 ?	기울이지 않음 ?

비글의 기울임 방향의 변화를 가속도 센서로 감지하여 로봇을 움직이는 방향이 해당 상태일 때를 '참', 아니면 '거짓'으로 판단합니다.



배터리 상태를 알려주는 블록입니다.

배터리 상태를 고르고 블록을 클릭하면 블록 아래에 true 또는 false가 나타나 현재 배터리 상태를 확인할 수 있습니다.



비글의 충전 상태를 알려주는 블록입니다.

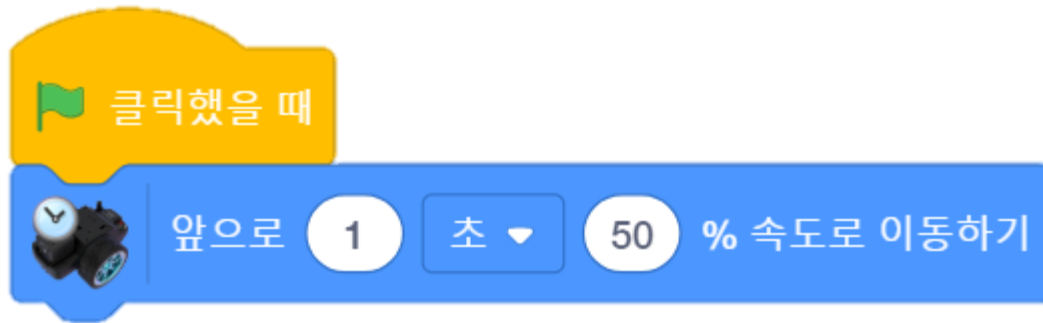
블록을 클릭하면 블록 아래에 true 또는 false가 나타나 충전 중인지 아닌지를 확인할 수 있습니다.

3차시

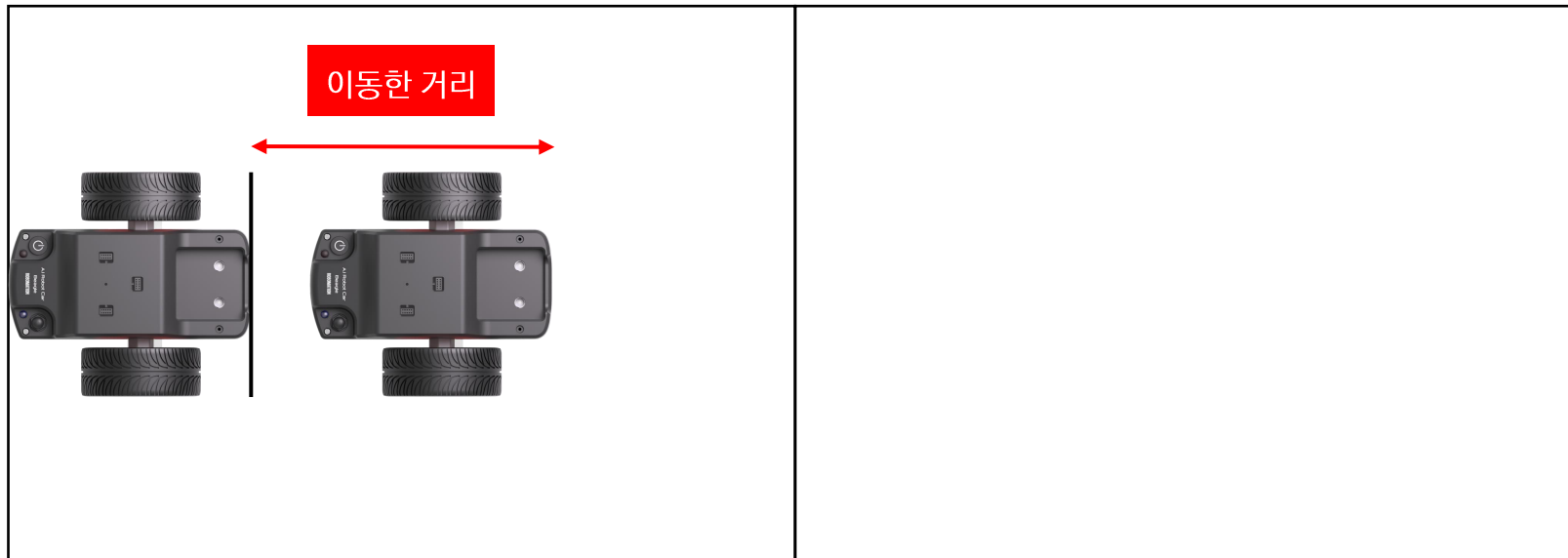
# 이동 블록 코드



- 비글을 움직이기 위한 프로그램을 작성해 봅니다.
- 아래와 같이 블록을 배치하고 스크래치에서 초록색 깃발을 클릭하면 비글이 앞으로 1초 50% 속도로 이동합니다.



- 비글이 이동한 거리를 측정하기 위해 A3 용지 두 개를 이어 붙입니다. 왼쪽 A3용지의 가장 끝 부분에 맞춰 비글을 내려 놓고 아래와 같이 선을 긋습니다.
- 비글이 움직이면 A3 용지가 함께 움직일 수 있으므로 A3 뒷면에 테이프 등을 붙여 움직이지 않도록 고정합니다.
- 코드 실행 후 해당 위치부터 비글이 이동한 거리를 측정합니다.



- 비글이 이동한 거리를 자를 이용하여 측정한 뒤 표에 기록해 봅니다.

측정 횟수	1회	2회	3회	4회	5회
이동한 거리	cm	cm	cm	cm	cm

- 이동한 거리는 측정할 때마다 조금씩 달라질 수 있습니다.
- 앞에서 측정한 거리를 모두 더한 후, 5로 나누어 보면 비글이 1초 동안 평균적으로 얼마나 이동하는지 알 수 있습니다.

이동한 거리의 합	cm
합을 5로 나눈 값	cm

- 이제 비글이 얼마나 빨리 이동하는지, 비글의 속력을 알아보도록 합시다.
- 속력을 좀 더 다양한 방법으로 계산해 보기 위해서 이동하는 시간을 다르게 변경하여 이동한 거리를 측정한 후, 이동한 거리를 이동한 시간으로 나누는 활동을 반복합니다.
- 속력을 구하는 식은 다음과 같습니다.

$$\text{속력(cm/초)} = \frac{\text{이동 거리(cm)}}{\text{이동 시간(초)}}$$

이동한 시간	1초	2초	3초	4초	5초
이동한 거리	cm	cm	cm	cm	cm
속력	cm/초	cm/초	cm/초	cm/초	cm/초

- 속력은 측정할 때마다 조금씩 달라질 수 있으므로 평균 속력이 얼마나 되는지 알아보기 위해 앞에서 계산한 속력을 모두 더한 후 5로 나누어 봅니다.

속력의 합	cm/초
합을 5로 나눈 값	cm/초

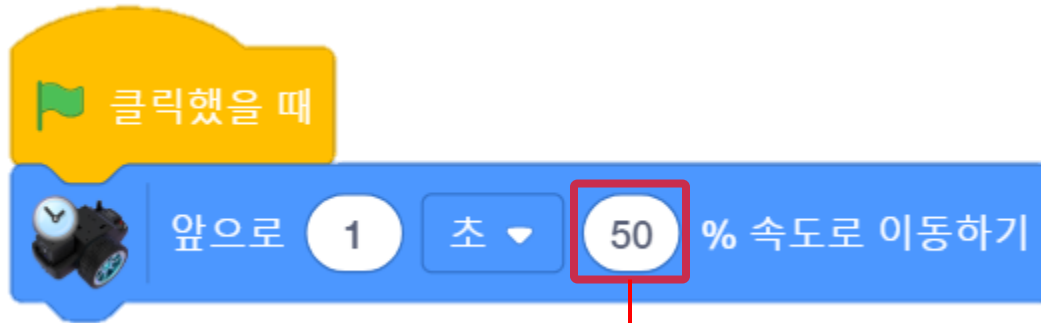
- 이번에는 반대로 일정 거리를 이동하기 위해 몇 초 동안 이동해야 하는지 알아봅니다.
- 비글의 앞부분과 15cm 떨어진 위치에 선을 긋습니다.
- 앞에서 측정한 속력을 이용하여 15cm를 이동하기 위해 필요한 시간을 계산합니다.
- 앞서 배운 속력을 구하는 식을 변형하여 이동 시간을 구하는 식을 만듭니다.

$$\text{이동 시간(초)} = \frac{\text{이동 거리(cm)}}{\text{속력(cm/초)}}$$

- A4 용지의 재질이나 비글의 배터리 상태 등에 따라 이동하는 거리는 조금 달라질 수 있습니다. 앞에서 계산한 값을 입력하여 로봇이 실제 이동한 거리를 관찰한 후, 숫자를 약간씩 변경해 15cm를 이동하도록 합니다.
- 같은 방법으로 20cm, 30cm 등 다양한 거리를 이동하려면 몇 초 동안 이동해야 하는지 계산하고, 프로그램을 수정하여 실행해 봅니다.

이동 거리	15cm	20cm	30cm	45cm
이동 시간	초	초	초	초

- 이번엔 이동하기 블록의 세 번째 파라미터인 양쪽 바퀴의 속도 값을 각자 원하는 값으로 바꿔 봅니다.
- 비글의 속력, 이동한 거리, 이동 시간을 다양하게 측정 및 계산 후 친구들과 결과를 비교합니다.



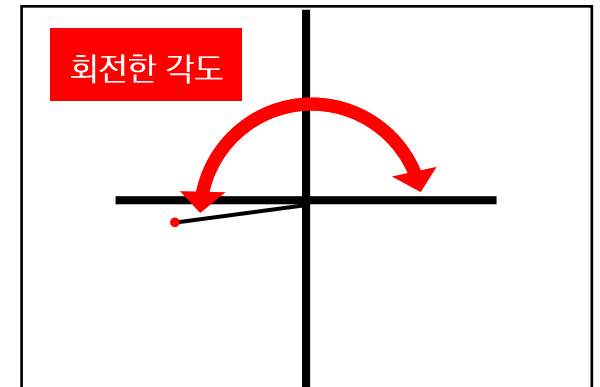
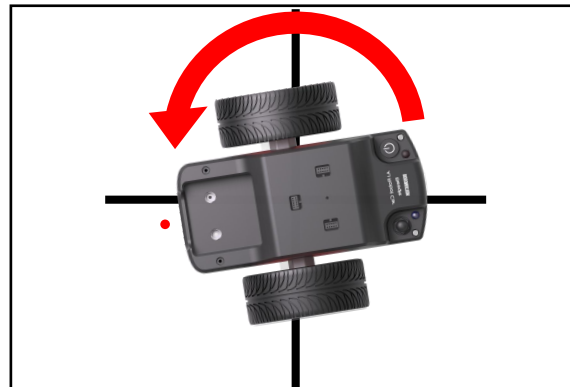
입력할 수 있는 값의 범위는 '-100 ~ 100'이지만 음수 값을 넣으면 바퀴 이동 방향이 반대로 바뀌어 비글이 뒤로 움직이므로 양수 값을 입력합니다.

- 비글이 제자리에서 돌기 위한 프로그램을 작성해 봅니다.
- 아래와 같이 블록을 배치하고 스크래치에서 초록색 깃발을 클릭하면 비글이 왼쪽으로 1초 50% 속도로 제자리에서 돕니다.



## 회전 각도 측정하기

- 비글이 얼마나 회전하였는지 측정하기 위해 A4 용지 위에 + 모양으로 선을 그리고 비글을 중앙에 맞춰 올려놓습니다. (가로선은 비글의 중심에 맞추고, 세로선은 바퀴의 중심에 맞춥니다.)
- 비글이 움직이면 A4 용지가 함께 움직일 수 있으므로 A4 뒷면에 테이프 등을 붙여 움직이지 않도록 고정합니다.
- 아래 코드 실행 후 비글이 회전을 멈추면 전면 중앙 바로 앞에 점을 찍고, 그 점과 + 표시의 중앙을 이으면 햄스터 로봇이 회전한 각도를 측정할 수 있습니다.





- 비글이 회전한 각도를 측정한 뒤 표에 기록해 봅니다.

측정 횟수	1회	2회	3회	4회	5회
회전한 각도	도	도	도	도	도

- 회전한 각도는 측정할 때마다 조금씩 달라질 수 있습니다.
- 앞에서 측정한 각도를 모두 더한 후, 5로 나누어 보면 평균적으로 얼마나 회전하는지 알 수 있습니다.

회전한 각도의 합	도
합을 5로 나눈 값	도

- 이제 비글이 얼마나 빨리 회전하는지 알아보도록 합시다.
- 회전한 각도를 회전한 시간으로 나눈 값은 각속력 또는 회전속력이라고 합니다. 각도(도)를 시간(초)으로 나누었기 때문에 회전속력의 측정 단위는 도/초입니다.
- 회전속력을 구하는 식은 다음과 같습니다.

$$\text{회전속력(도/초)} = \frac{\text{회전 각도(도)}}{\text{회전 시간(초)}}$$

- 회전속력을 좀 더 다양하게 계산해 보기 위해 회전하는 시간을 다르게 변경하여 회전한 각도를 측정한 후, 회전한 각도를 회전한 시간으로 나누어 회전속력을 계산해 봅시다.

회전한 시간	1초	2초	3초	4초	5초
회전한 각도	도	도	도	도	도
회전속력	도/초	도/초	도/초	도/초	도/초

- 회전속력은 측정할 때마다 조금씩 달라질 수 있으므로 평균적인 회전속력이 얼마나 되는지 알아보기 위해 앞에서 계산한 회전속력을 모두 더한 후 5로 나누어 봅시다.

회전속력의 합	도/초
합을 5로 나눈 값	도/초

- 이번에는 반대로 일정 각도를 회전하기 위해 몇 초 동안 회전해야 하는지 알아봅니다.
- 앞에서 측정한 회전속력을 이용하여 180도를 이동하기 위해 필요한 시간을 계산해 볼 수 있습니다.
- 앞서 배운 회전 속력을 구하는 식을 변형하면 회전 시간을 구하는 식을 만들 수 있습니다.

$$\text{회전 시간(초)} = \frac{\text{회전 각도(도)}}{\text{회전속력(도/초)}}$$

- A4 용지의 재질이나 비글의 배터리 상태 등에 따라 회전하는 각도는 조금 달라질 수 있습니다. 앞에서 계산한 값을 입력하여 로봇이 실제 회전한 거리를 관찰한 후, 숫자를 약간씩 변경해 180도를 이동하도록 합니다.
- 같은 방법으로 90도, 180도 등 다양한 각도를 회전하려면 몇 초 동안 회전해야 하는지 계산하고, 프로그램을 수정하여 실행해 봅니다.

회전 각도	90도	180도	270도	360도
회전 시간	초	초	초	초

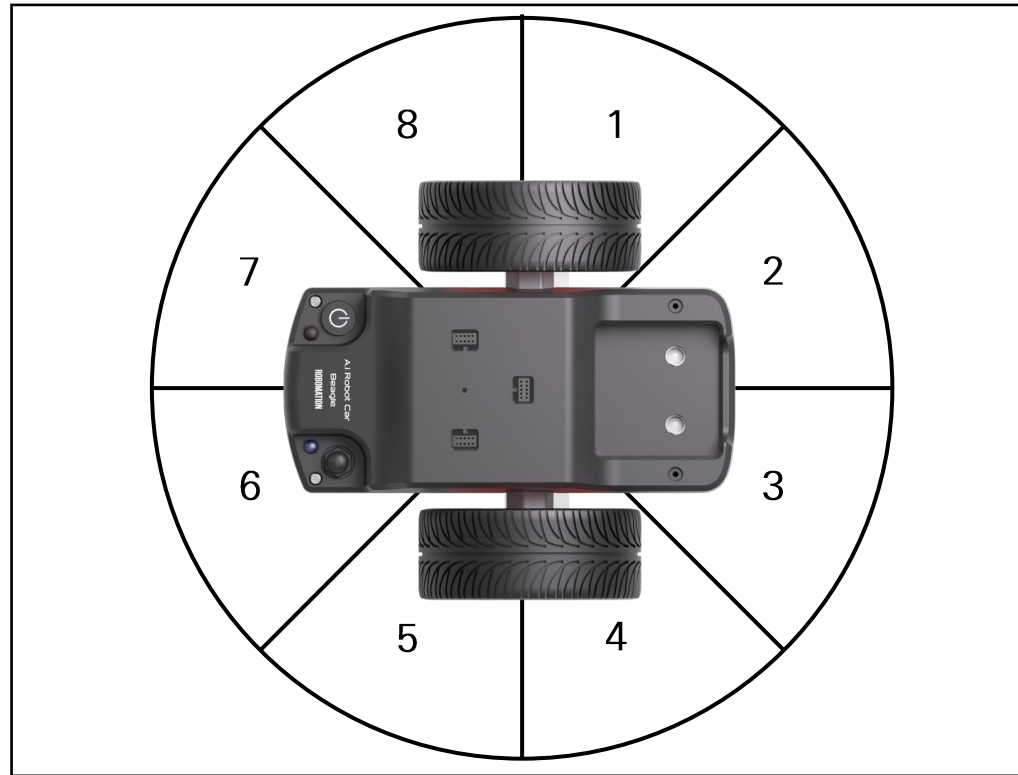
- 이번엔 제자리 돌기 블록의 네 번째 파라미터인 양쪽 바퀴의 속도 값을 각자 원하는 값으로 바꿔 봅니다.
- 비글의 속력, 회전한 각도, 회전 시간을 다양하게 측정 및 계산한 후 친구들과 결과를 비교합니다.



입력할 수 있는 값의 범위는 '-100 ~ 100'이지만 음수 값을 넣으면 바퀴 이동 방향이 반대로 바뀌므로 양수 값을 입력합니다.

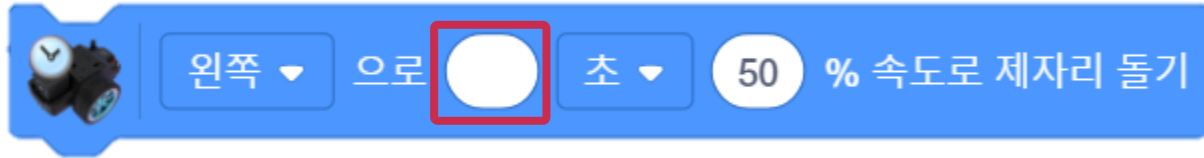
## 회전 각도 측정 하기(게임)

- 앞서 배운 제자리 돌기를 응용해 무작위로 점수를 받아가는 게임을 만들어 봅니다.
- A4 용지에 원을 그리고 직선을 그어 구역을 나누는 후 각 구역마다 다른 숫자를 적고 비글을 원의 중심에 맞춰 놓습니다.



## 회전 각도 측정 하기(게임)

- 두 번째 파라미터에 직접 회전할 시간을 입력하면 얼마나 회전한 뒤 멈출지 예측할 수 있으므로 시간이 무작위로 입력되도록 코드를 작성합니다.



- 위의 빨간 박스 안에 들어갈 변수 값을 연산 블록을 활용해 작성합니다.
- 먼저 1부터 10사이의 난수 블록을 선택합니다. 이 블록은 첫 번째 동그라미에 입력한 숫자, 1부터 두 번째 동그라미에 입력한 숫자 10까지의 자연수 중 하나를 변수 값으로 정하는 블록입니다.

1 부터 10 사이의 난수

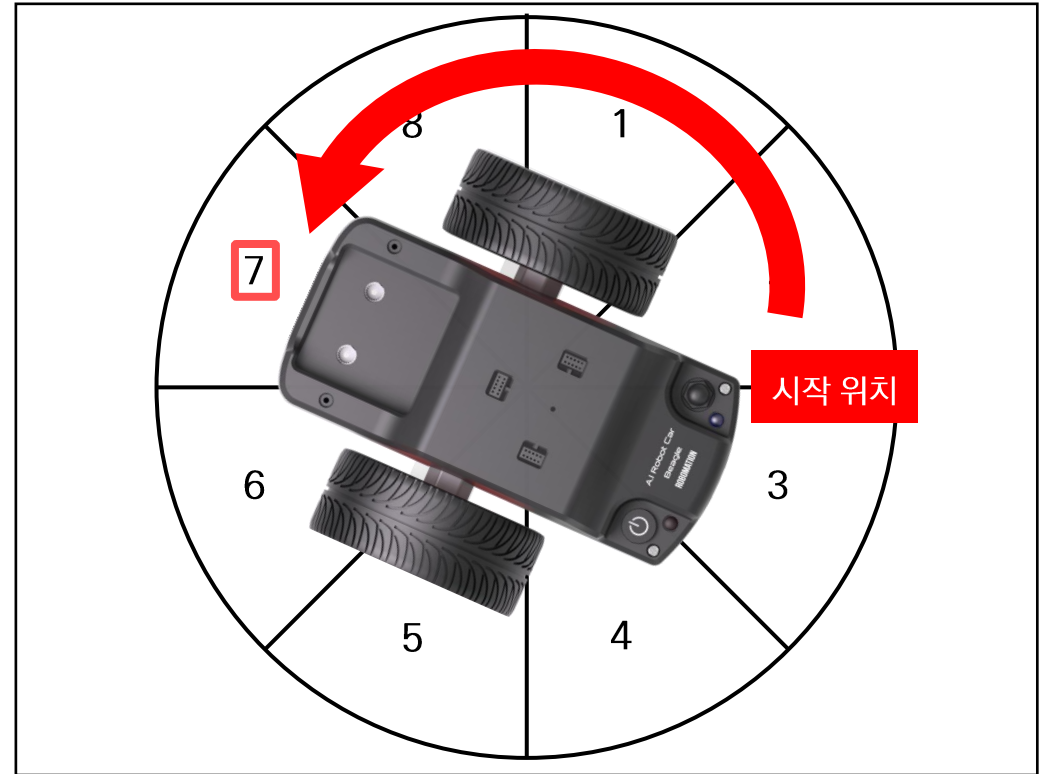
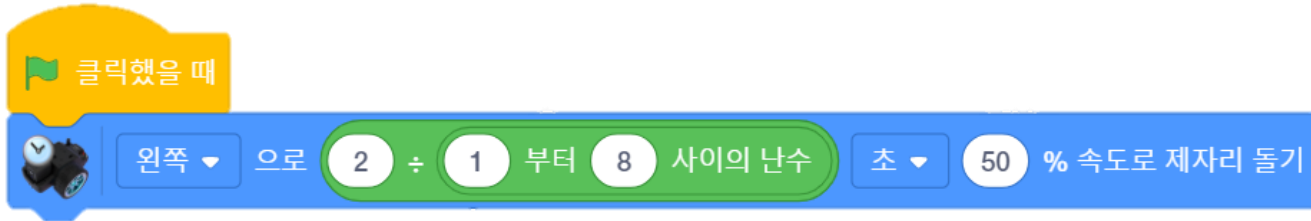
- 속도가 50%일 때 대략 2초 회전하면 비글이 한 바퀴(360도)를 돕니다. 따라서 기준 값을 2로 정하고 원 안의 숫자 범위와 같은 1부터 8 사이의 난수로 나누기 위해 아래와 같이 블록을 구성합니다.

2 ÷ 1 부터 8 사이의 난수

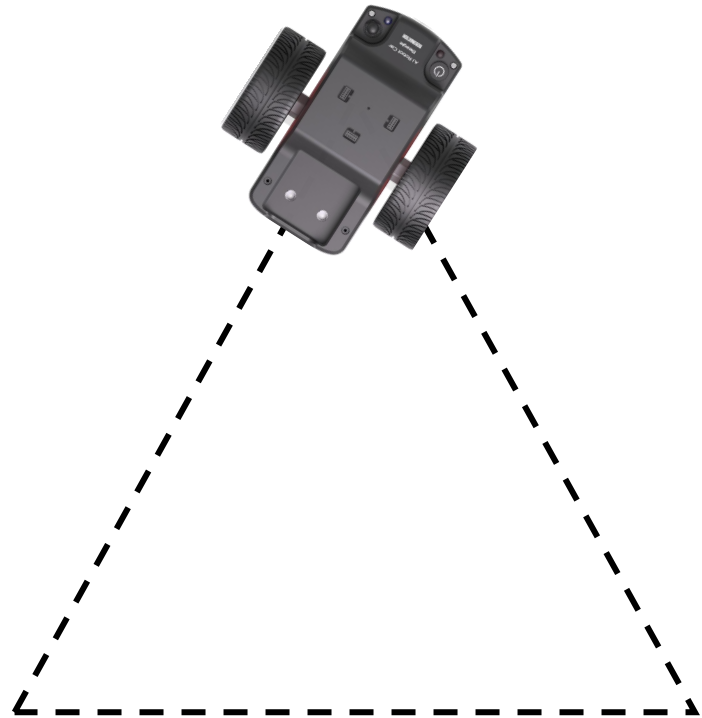
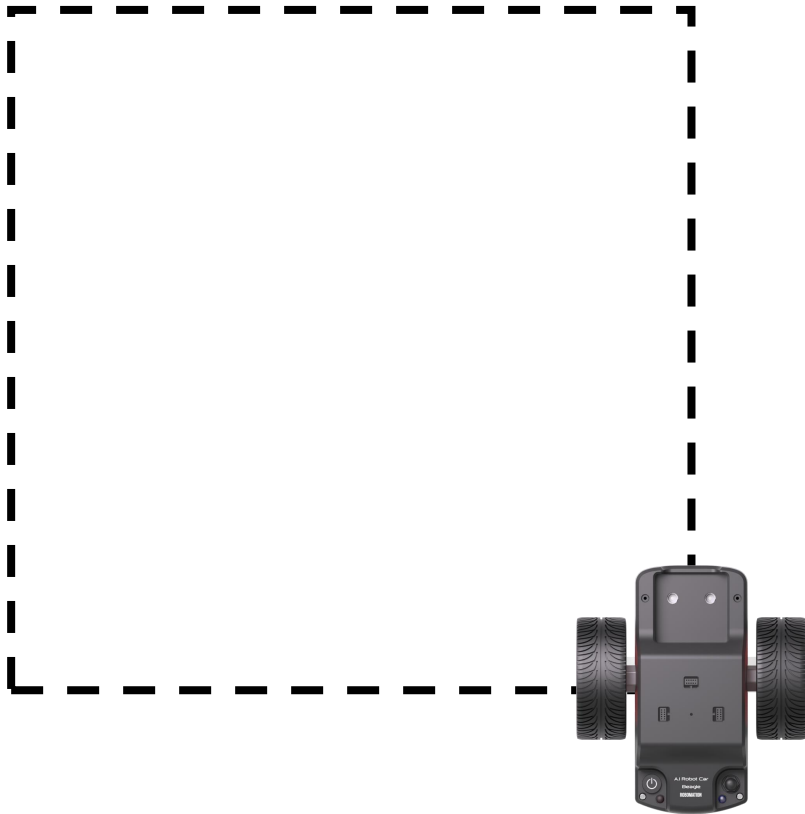
- 위 블록을 두 번째 파라미터 자리에 넣으면 코드가 완성됩니다.

## 회전 각도 측정 하기(게임)

- 친구들과 돌아가며 코드를 실행시켜 보고 누가 가장 큰 점수를 얻을지 겨뤄봅니다.
- 시작 위치와 도는 방향도 자유롭게 바꿔가며 게임을 진행합니다.

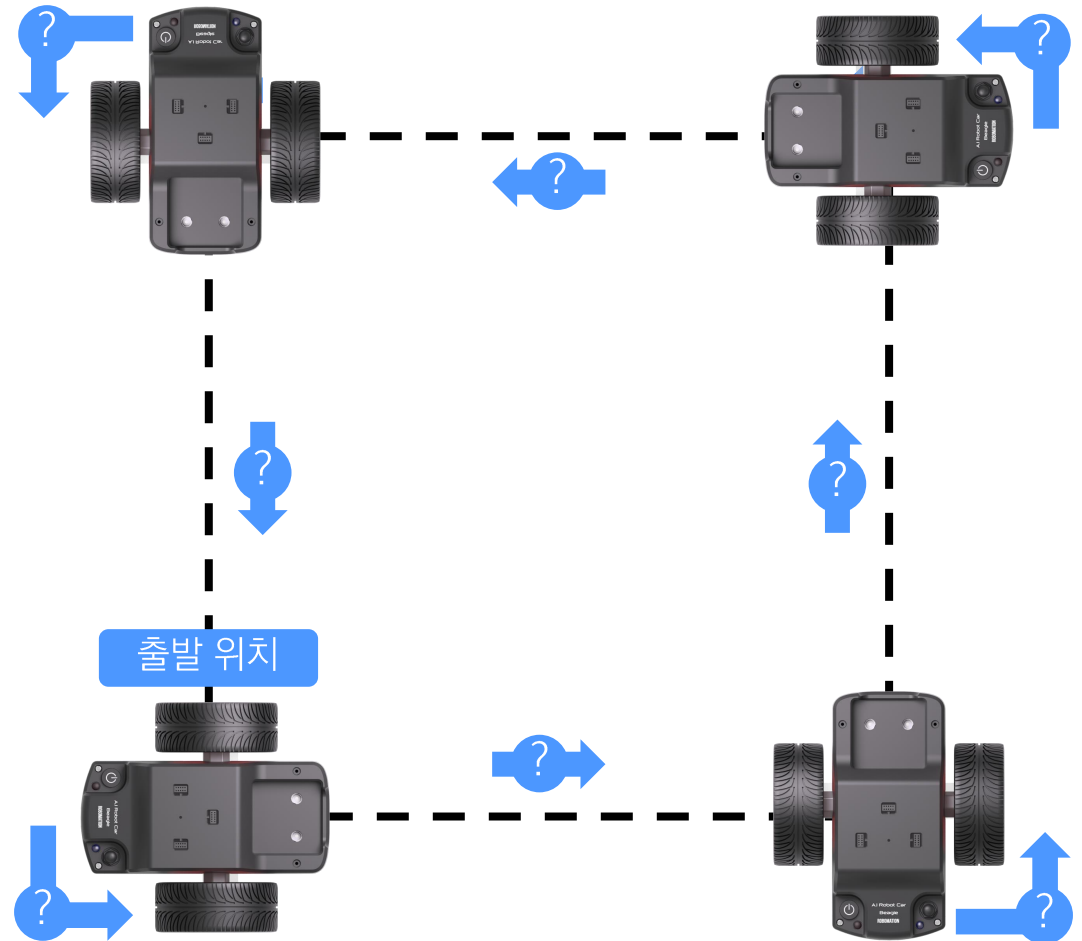


- 앞서 배운 블록을 이어 붙여 여러 줄의 프로그램을 작성해 봅니다.
- 여러가지 도형을 그리기 위해 비글에게 순서대로 명령하는 프로그램을 작성해 봅니다.
- 다음 단계 진행 전 어떤 블록을 어떻게 사용해야 할지 각자 의견을 제시합니다.





- 비글이 정사각형으로 움직이게 하기 위해서는 어떻게 코드를 작성해야 할지 생각해 봅시다.
- 명령은 어떤 순서로 작성해야 할지, 사용할 수 있는 블록은 무엇일지 생각해 봅시다.
- 정사각형을 그리기 위해서는 몇 도로 회전해야 할지 생각해 봅시다.
- 한 바퀴 마지막에는 출발할 때와 같은 방향을 바라보게 합니다.

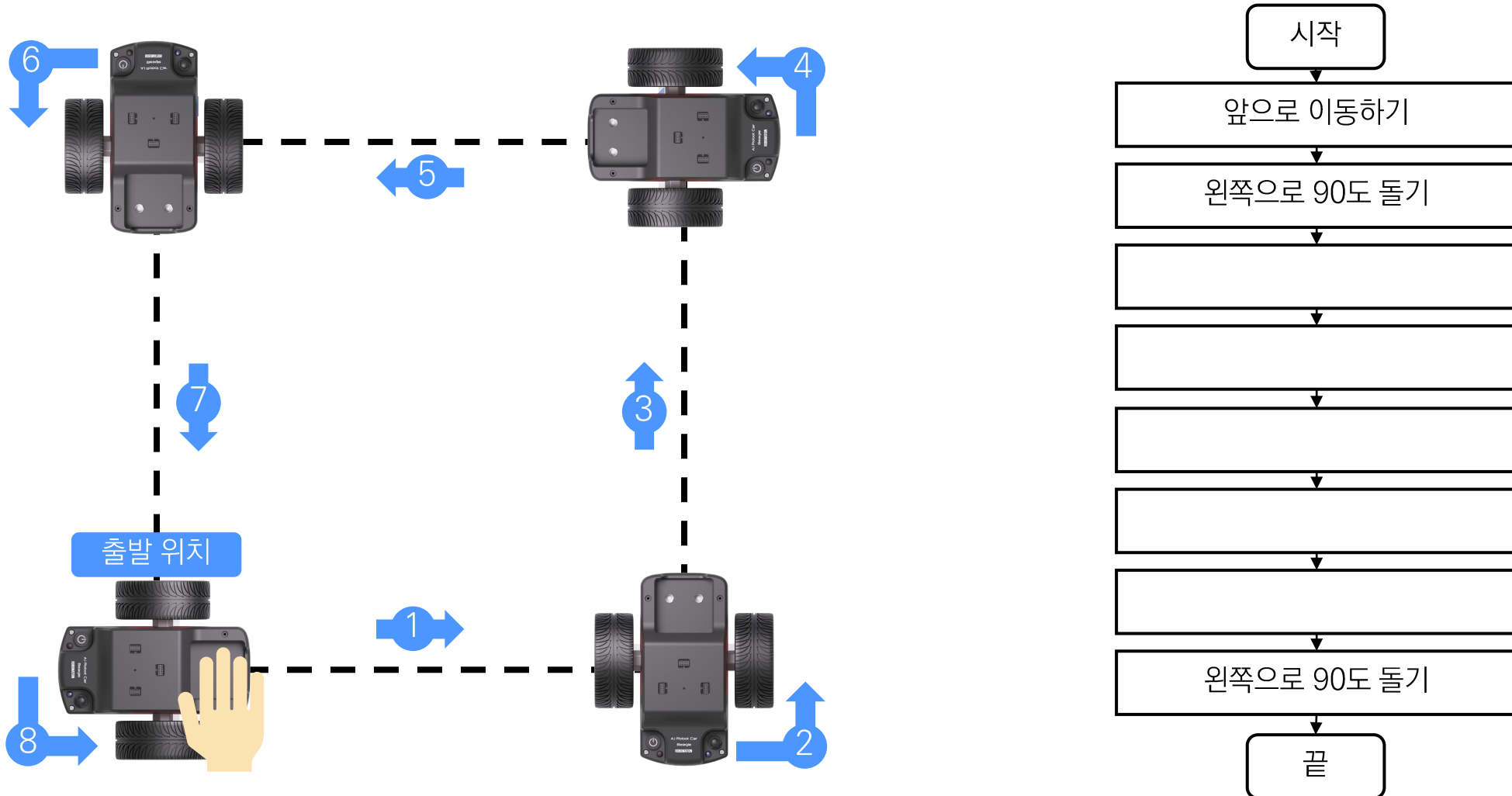


앞으로 가는 블록은 무엇일까요?



왼쪽으로 도는 블록은 무엇일까요?

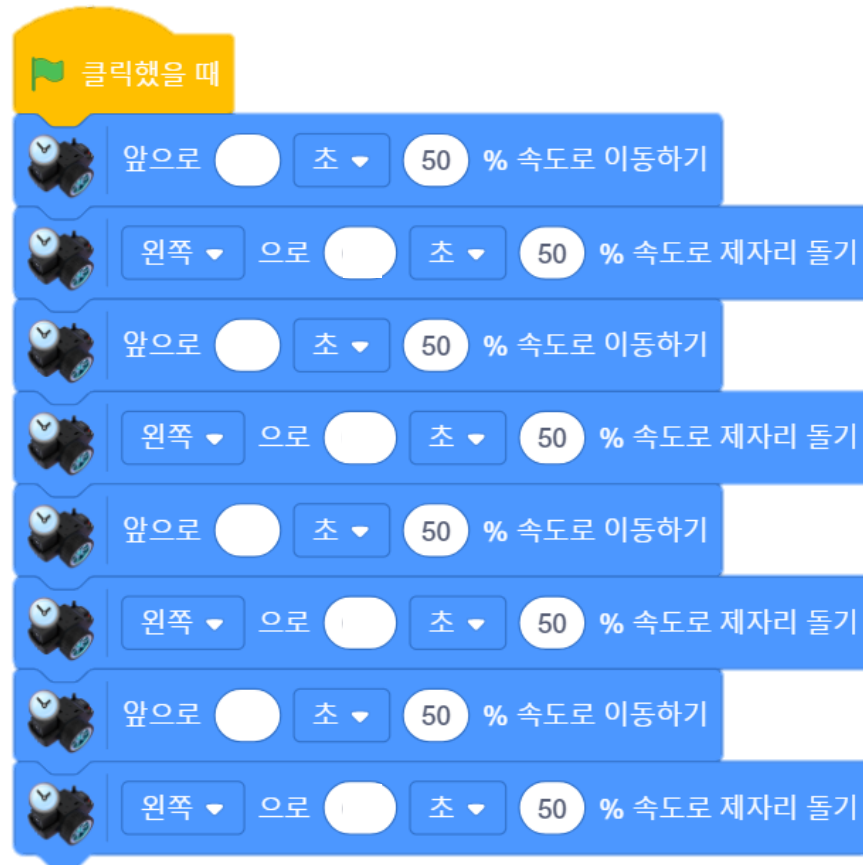
- 먼저 블록을 배치할 순서에 대해서 생각해 보고 빈 칸에 순서대로 명령을 적어 봅시다. 이러한 그림을 순서도라고 합니다.
- 적은 순서에 따라 비글을 손으로 들고 명령대로 움직여 봅시다. 비글이 정사각형 모양으로 잘 이동하면 성공입니다.



- 이동 거리, 회전 각도 측정 시간에 사용했던 블록과 측정했던 값을 참고하여 비글이 앞으로 가고 90도로 돌게 합니다.
- 50% 속도를 기준으로 사각형 한 변의 길이와 비글이 90도를 돌게하는 값을 초 자리에 각각 입력합니다.
- 시간 만으로 움직임을 제어하기 힘들 때에는 속도를 함께 조절하여도 됩니다.



- 블록이 완성되면 앞에서 정리한 순서대로 프로그램을 작성하고 실행하여 동작을 확인해 봅니다.
- A4 용지의 재질이나 비글의 배터리 상태 등에 따라 이동, 회전하는 거리가 변할 수 있기 때문에 입력 값은 상황에 따라 달라질 수 있습니다.



- 같은 동작을 반복하는 것이므로 반복문을 활용해 오른쪽과 같이 표현할 수도 있습니다.



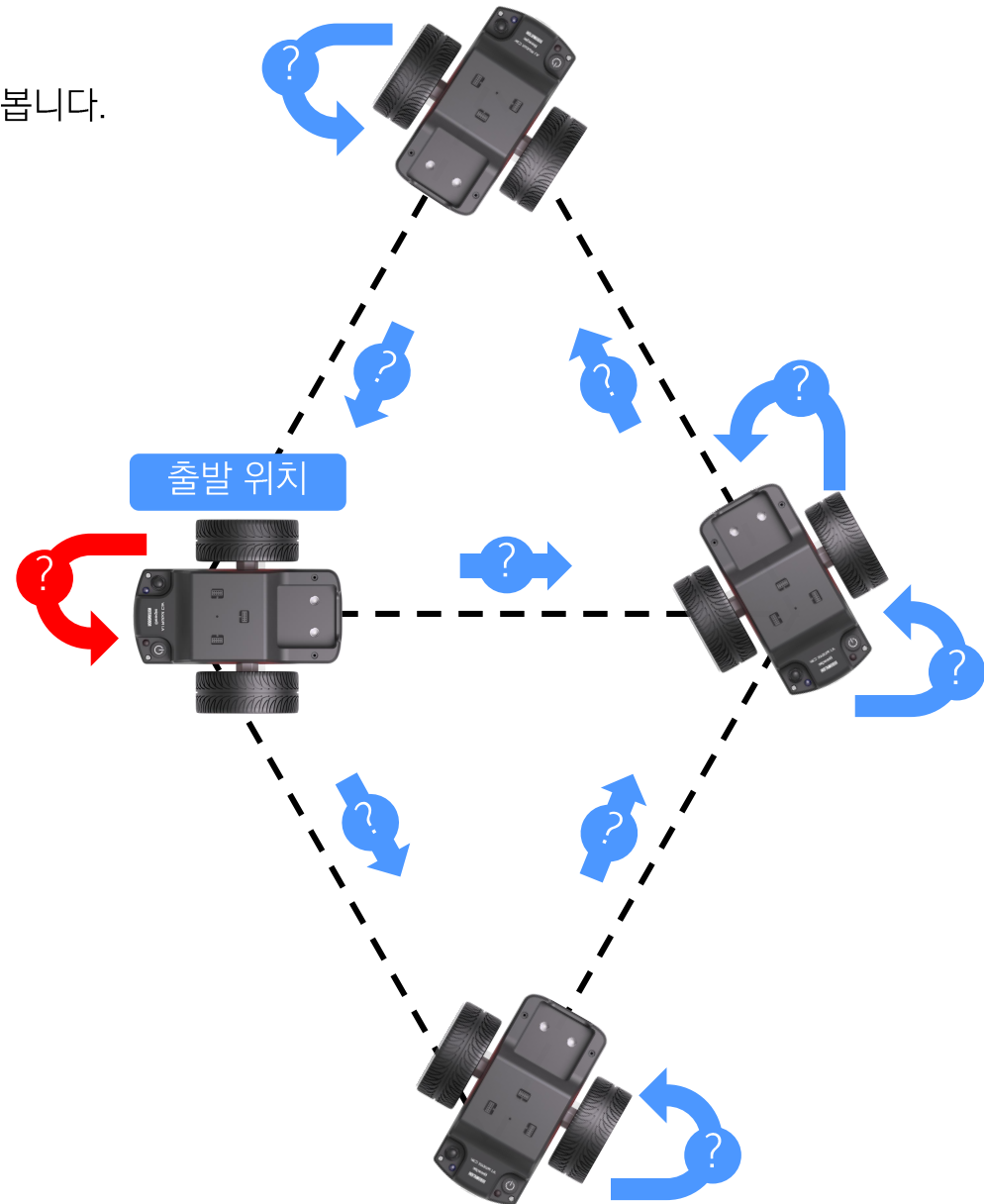
- 비글이 마름모 모양으로 움직이게 하기 위해서는 어떻게 코드를 작성해야 할지 생각해 봅니다.
- 명령은 어떤 순서로 작성해야 할지, 사용할 수 있는 블록은 무엇일지 생각해 봅니다.
- 마름모를 그리기 위해서는 몇 도로 회전해야 할지 생각해 봅니다.
- 이번에는 회전해야 하는 각도가 2가지 이므로 회전하는 명령이 하나 더 필요합니다.
- 한 바퀴 마지막에는 출발할 때와 반대 방향을 바라보게 합니다.



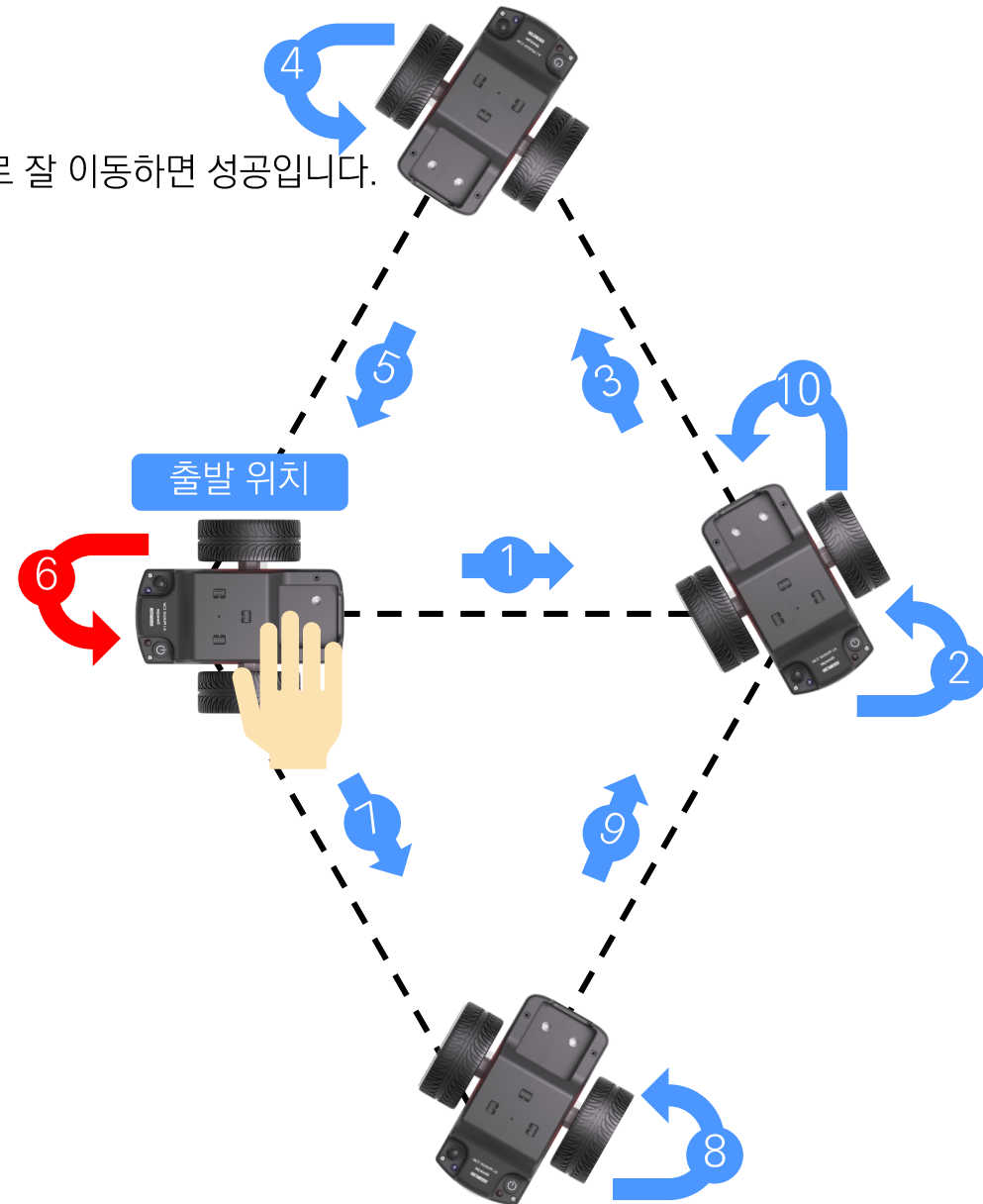
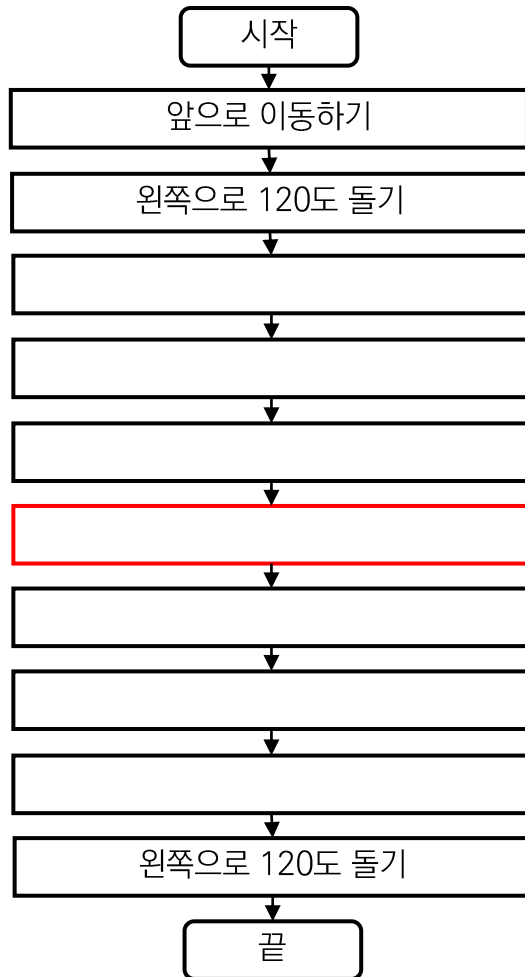
앞으로 가는 블록은 무엇일까요?



왼쪽으로 도는 블록은 무엇일까요?



- 먼저 블록을 배치할 순서에 대해서 생각해 보고 빈 칸에 순서대로 명령을 적어 봅시다.
- 적은 순서에 따라 비글을 손으로 들고 명령대로 움직여 봅시다. 비글이 마음모 모양으로 잘 이동하면 성공입니다.



- 이동 거리, 회전 각도 측정 시간에 사용했던 블록과 측정했던 값을 참고하여 비글이 앞으로 가고 90도로 돌게 합니다.
- 50% 속도를 기준으로 사각형 한 변의 길이와 비글이 120도, 60도 만큼 돌게하는 값을 초 자리에 각각 입력합니다.
- 시간 만으로 움직임을 제어하기 힘들 때에는 속도를 함께 조절하여도 됩니다.

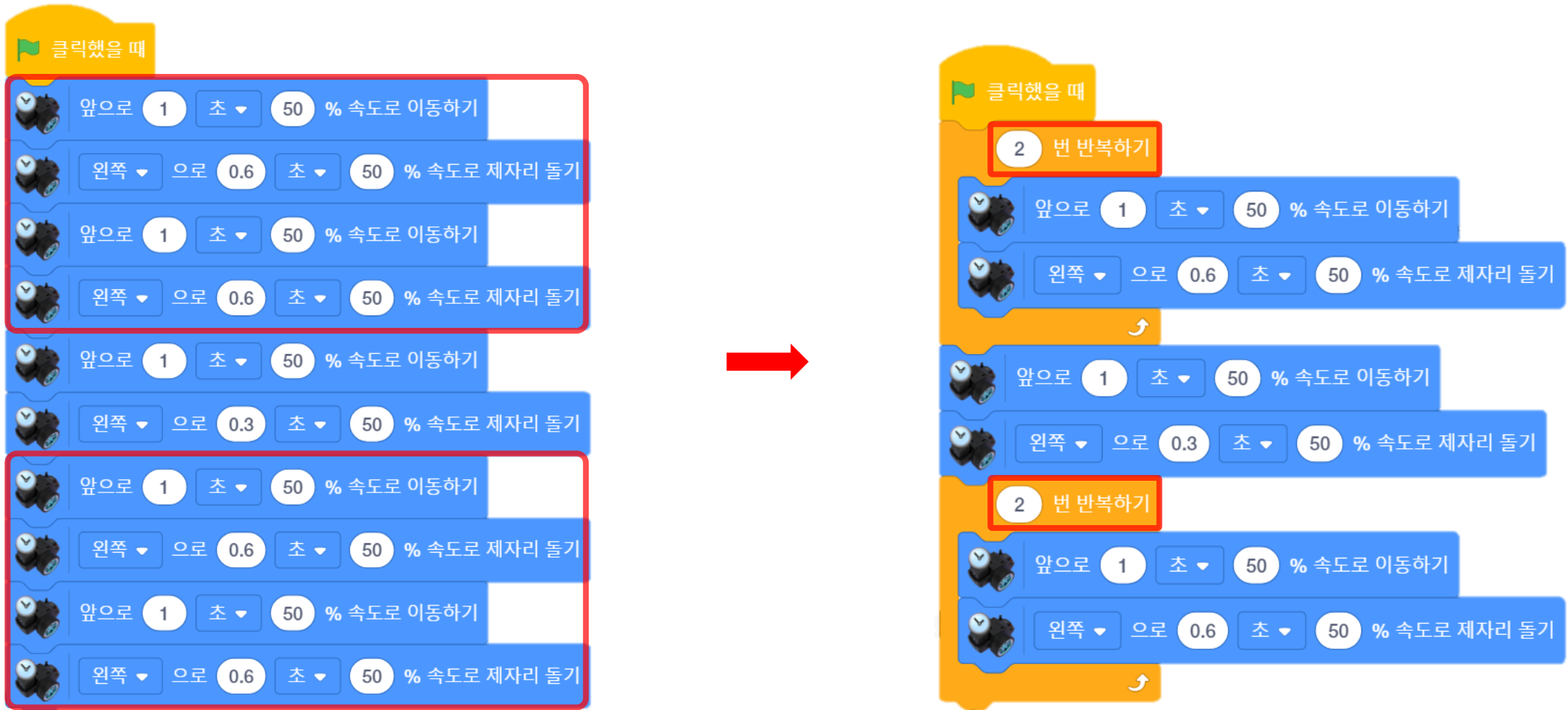




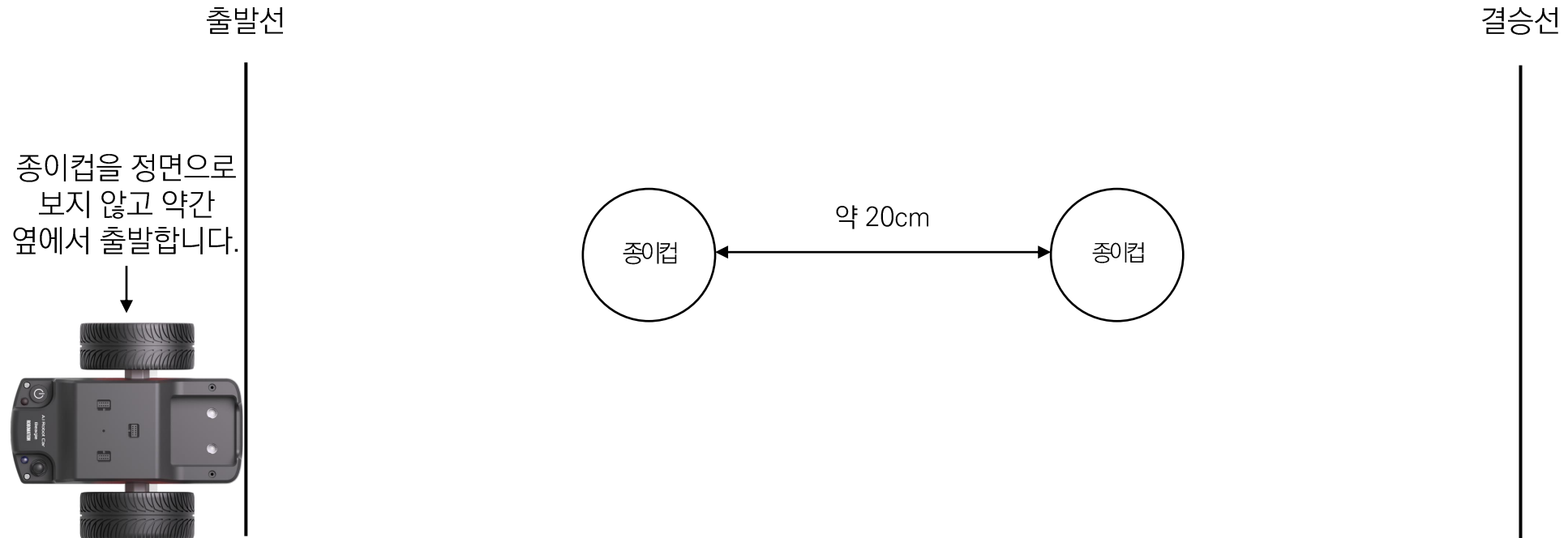
- 블록이 완성되면 앞에서 정리한 순서대로 프로그램을 작성하고 실행하여 동작을 확인해 봅니다.
- A4 용지의 재질이나 비글의 배터리 상태 등에 따라 이동, 회전하는 거리가 변할 수 있기 때문에 입력 값은 상황에 따라 달라질 수 있습니다.



- 같은 동작을 반복하는 것이므로 반복문을 활용해 오른쪽과 같이 표현할 수도 있습니다.
- 마름모는 정삼각형 두 개를 위아래로 이어 붙인 모양이므로 반복문 안의 코드를 3번 반복하면 정삼각형을 그릴 수 있습니다.

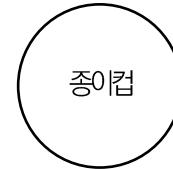


- 이동하기 블록과 돌기 블록을 활용해 슬라럼 경기를 해봅니다.
- 슬라럼은 장애물을 세워 놓고 그 사이를 지그재그로 통과하는 것입니다.
- 대략 20cm 간격으로 놓은 종이컵 사이를 지나 결승선에 도착하도록 합니다. 결승선에 도착 하는 시간이 가장 짧은 사람이 승리합니다.



## 이동하기/돌기 블록 응용하기

- 사용할 수 있는 블록이 아래에 나온 두 블록일 때 비글을 어떤 경로로 이동시킬지 예상하고 종이에 그려 봅니다.
- 원하는 경로로 이동하기 위한 코드를 작성해 봅니다.

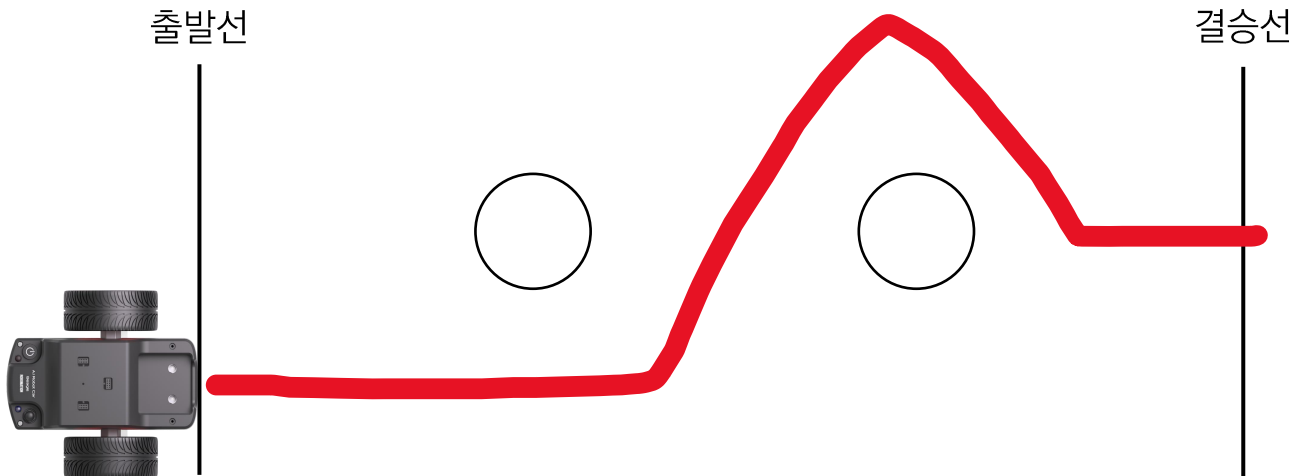


결승선



- A4 용지의 재질이나 비글의 배터리 상태, 컵 사이의 간격 등에 따라 이동, 회전하는 거리가 변할 수 있으므로 입력 값은 상황에 따라 달라질 수 있습니다.
- 누가 가장 빠른 시간에 결승선에 도달하는 경로와 코드를 작성하였는지 시간을 기록해 친구와 비교해 봅니다.

## 예시



클릭했을 때

- 앞으로 2.5 초 50 % 속도로 이동하기
- 왼쪽 0.3 초 50 % 속도로 제자리 돌기
- 앞으로 2 초 50 % 속도로 이동하기
- 오른쪽 0.6 초 50 % 속도로 제자리 돌기
- 앞으로 1.5 초 50 % 속도로 이동하기
- 왼쪽 0.3 초 50 % 속도로 제자리 돌기
- 앞으로 1 초 50 % 속도로 이동하기

- 비글 바퀴 값에 입력하는 숫자를 다르게 하여 비글이 이동하는 방향을 조종해 봅니다.
- 아래 블록의 첫 번째 파라미터와 두 번째 파라미터의 값을 바꿔가며 비글이 어떻게 움직이는지 살펴봅니다. 파라미터의 값은 -100~100 사이의 값을 입력할 수 있습니다.



왼쪽 바퀴 50 % 오른쪽 바퀴 50 %로 정하기



왼쪽 바퀴 -50 % 오른쪽 바퀴 -50 %로 정하기

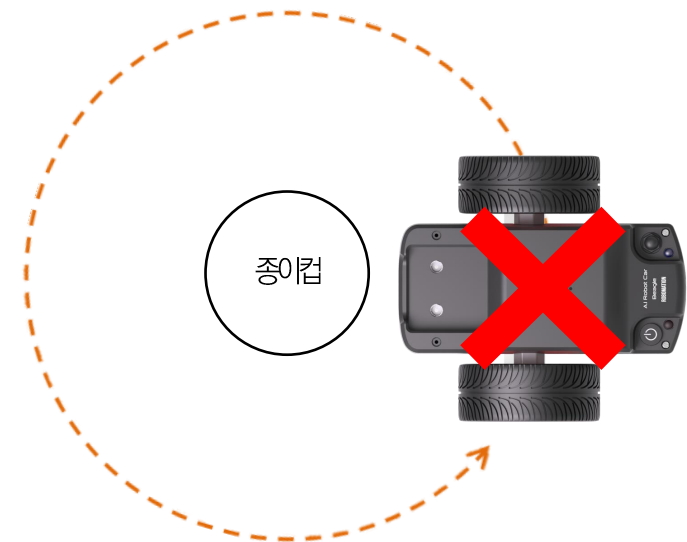


왼쪽 바퀴 60 % 오른쪽 바퀴 40 %로 정하기



왼쪽 바퀴 50 % 오른쪽 바퀴 -50 %로 정하기

- 바퀴 값에 따라 달라지는 비글의 움직임을 확인했다면 이번엔 종이컵을 놓고 비글이 그 주변을 돌도록 코드를 작성해 봅니다.
- 비글을 종이컵 옆에 왼쪽과 같이 내려놓습니다. 오른쪽 그림처럼 비글의 앞부분이 종이컵을 바라보도록 내려 놓으면 종이컵 주위를 돌기 어렵습니다.



## 바퀴 값으로 움직이기

- 종이컵이 비글의 왼쪽에 놓여 있으므로 비글이 왼쪽 방향으로 돌도록 코드를 작성해야 합니다.
- 아래 블록과 같이 오른쪽 바퀴 속도를 왼쪽 바퀴 속도 보다 크게 하면 비글이 왼쪽으로 원을 그리며 돕니다.
- 두 바퀴 값 크기의 차이가 달라지면 비글이 그리는 원의 크기도 달라집니다.
- 다양한 값을 넣어 종이컵을 따라 도는데 가장 적합한 값을 찾습니다.



클릭했을 때



왼쪽 바퀴 20 % 오른쪽 바퀴 60 %로 정하기





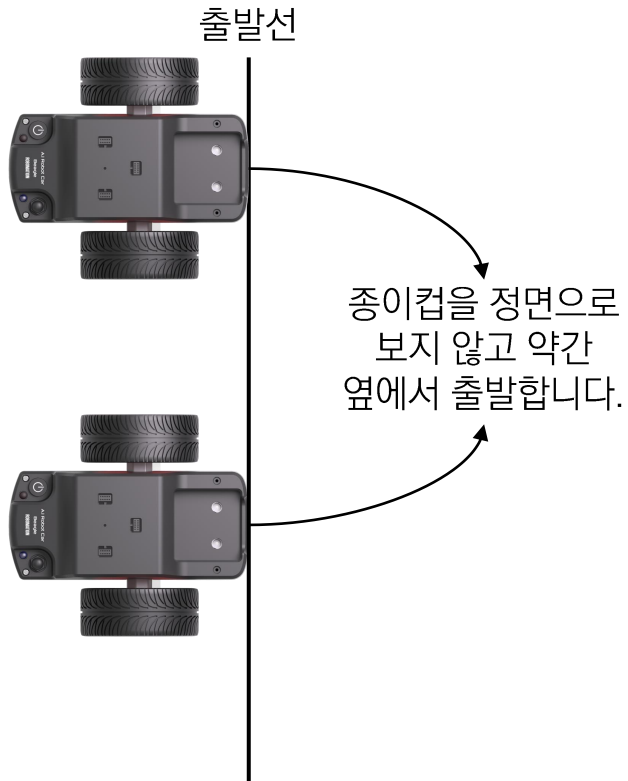
## 바퀴 값으로 움직이기

- 이번엔 연속해서 2개의 종이컵을 통과할 수 있는 프로그램을 만들어 봅니다.
- 두 종이컵은 비글의 가로 넓이 간격으로 세워둡니다.
- 파라미터 값으로 다양한 숫자를 입력하여 바퀴의 움직임을 조작합니다. 오른쪽 코드는 예시로 다양한 방식으로 코드를 작성할 수 있습니다.



## 바퀴 값으로 움직이기

- 출발선에서 30cm 정도 떨어진 적당한 거리에 종이컵을 둡니다.
- 출발선에서 동시에 출발하여 종이컵을 돌아 반환점으로 먼저 돌아오는 사람이 승리하는 경기입니다.
- 앞에서 공부했던 회전 방법에 대해 생각해 보고 예시 코드를 참고하여 코드를 작성합니다.



예시

클릭했을 때

앞으로  초  % 속도로 이동하기

왼쪽 바퀴  % 오른쪽 바퀴  %로 정하기

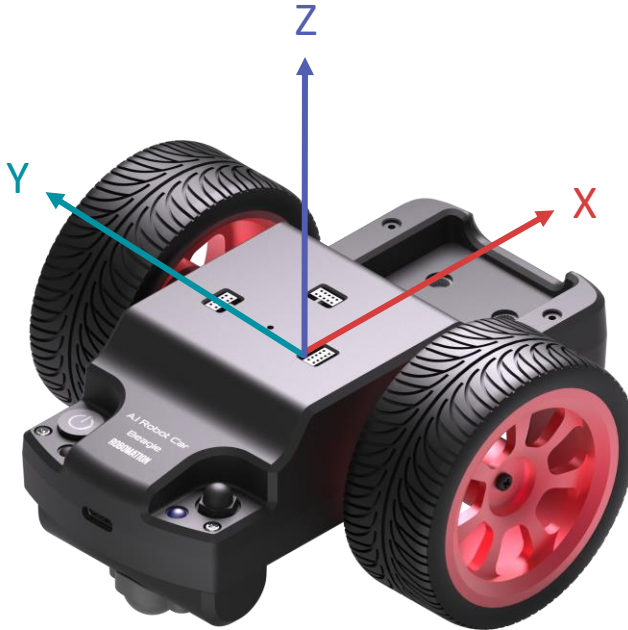
초 기다리기

앞으로  초  % 속도로 이동하기

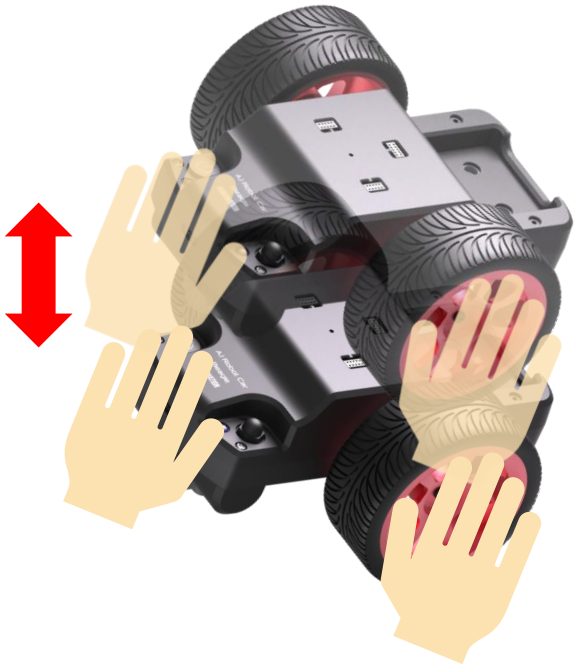
4차시

# 센서 블록 코드

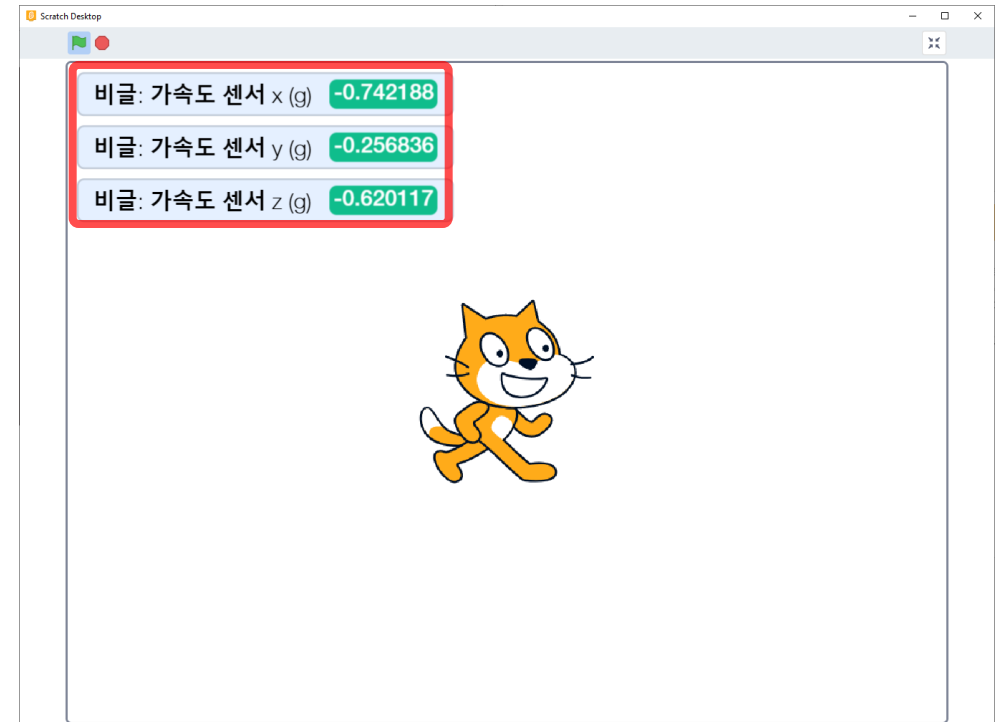
- 비글에는 3축 가속도 센서가 내장되어 있습니다.
- 가속도 센서의 X축은 로봇의 정면 방향이 양수값이고 뒷면 방향이 음수 값입니다. Y축은 왼쪽 방향이 양수 값, 오른쪽 방향이 음수 값이며, Z축은 위쪽 방향이 양수 값, 아래쪽 방향이 음수 값입니다.



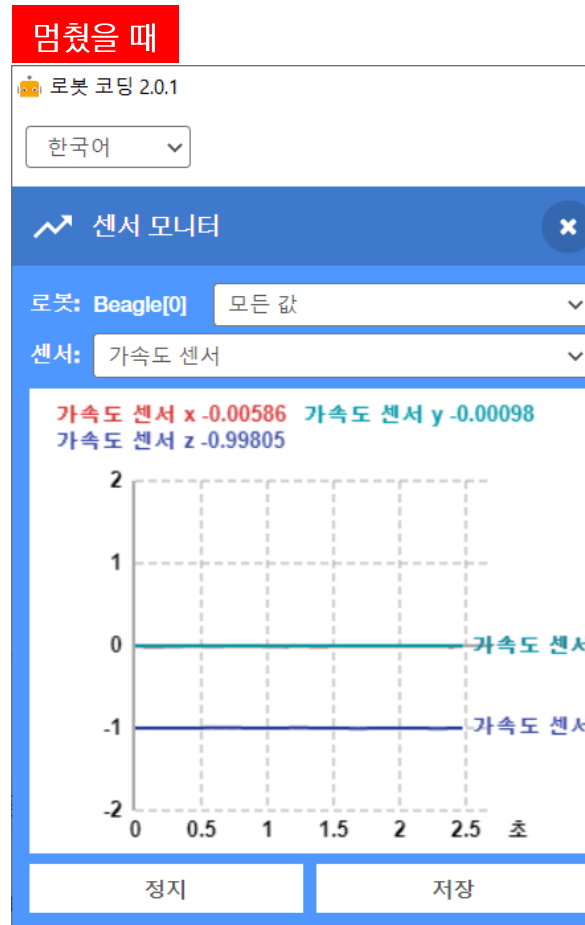
- 각 축에 대한 가속도 값은 가속도 센서 x (g), 가속도 센서 y (g), 가속도 센서 z (g) 블록을 사용하여 알 수 있습니다. 각 블록 앞의 체크 박스를 클릭하면 무대에서 각 축의 가속도 값을 실시간으로 확인할 수 있습니다.
- 프로그램을 구현하기 전 우선 비글을 들고 움직이며 어떤 방식으로 움직일 때 각 축의 값이 변화하는지 확인합니다.



- 가속도 센서 x (g)
- 가속도 센서 y (g)
- 가속도 센서 z (g)

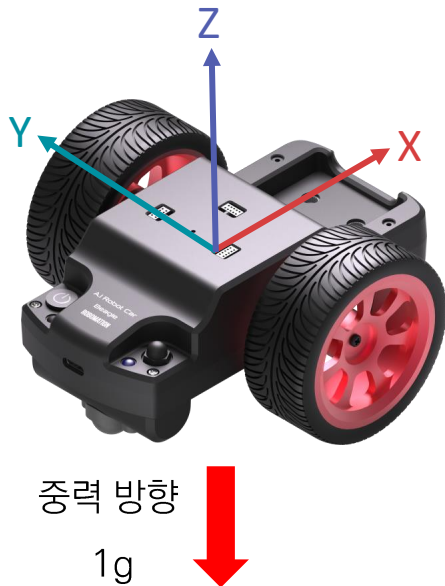


- 아래와 같이 로봇 코딩의 센서 모니터 기능으로도 가속도 센서의 값을 확인해 볼 수 있습니다.
- 각 축의 가속도 값의 변화가 그래프 형태로 나타납니다.



## 가속도 센서 알아보기

- 비글을 가만히 두면 앞뒤나 좌우로 가해지는 힘이 없으므로 X축 가속도 값과 Y축 가속도 값은 0에 가까운 값을 가집니다.
- Z축의 가속도 방향은 위쪽을 향하지만 비글에 항상 작용하고 있는 중력의 방향은 땅을 향합니다. 그러므로 움직임이 없을 때에도 비글의 Z축 가속도 값은 음수 값을 가지게 되고, 값의 크기는 중력의 값을 1로 하여 -1이 됩니다.



비글: 가속도 센서 x (g) **-0.006836**

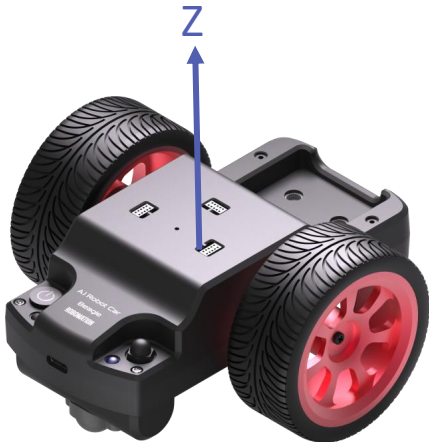
비글: 가속도 센서 y (g) **0**

비글: 가속도 센서 z (g) **-1**

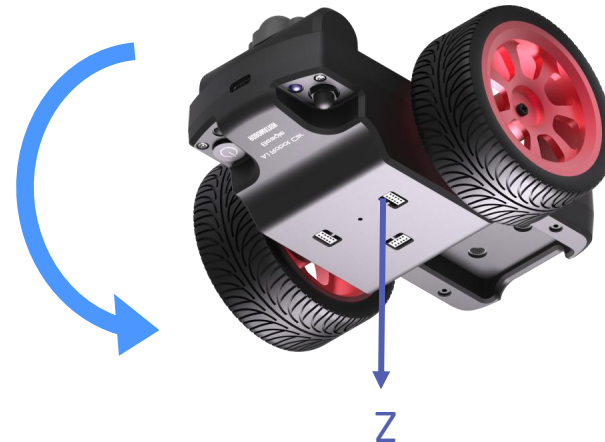
## 가속도 센서z 값 활용하기

- 이제 가속도 센서의 Z축 값을 활용하여 비글이 노래하는 코드를 작성해 봅니다.
- 비글을 뒤집으면 Z축 가속도 값이 양수 값을 가지며 값의 크기가 1과 같거나 커지는 것을 확인할 수 있습니다.

비글: 가속도 센서 z (g) **-1**

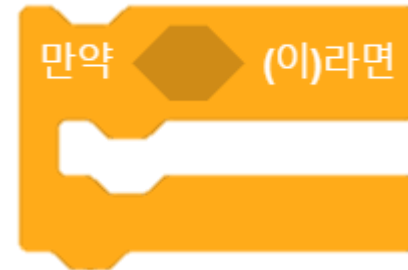
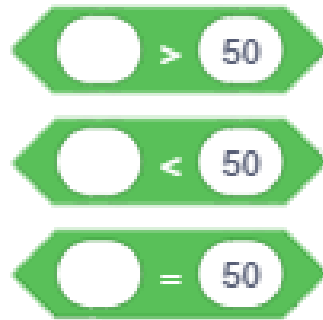


비글: 가속도 센서 z (g) **1.008789**







- 앞서 살펴 본 값을 조건문에 활용하면 비글이 뒤집어진 것을 감지할 수 있습니다.
- 가속도 센서 z (g)블록과 변수 값의 크기를 비교하는데 사용할 수 있는 연산 블록, 조건의 참과 거짓을 확인할 수 있는 블록을 조합해 Z축 값의 변화를 감지하는 코드를 작성합니다.





- 아래와 같이 코드를 작성하면 비글이 뒤집어져 있을 때를 감지할 수 있습니다. 비글의 상태를 계속해서 감지할 수 있도록 무한 반복하기 블록을 사용했습니다.
- 거꾸로 뒤집었을 때 블록도 같은 역할을 합니다.
- 이제 조건문 안이나 거꾸로 뒤집었을 때 블록 아래에 원하는 동작 블록을 삽입하면 비글이 뒤집어졌을 때 해당 동작을 수행합니다.
- 연주하기와 박자 쉬기 블록을 활용하여 비글이 뒤집어 졌을 때 노래를 연주하는 코드를 작성해 봅니다.

무한 반복하기

만약  가속도 센서 z (g) > 1 또는  가속도 센서 z (g) = 1 (아)라면


코드 작성



 거꾸로 뒤집었을 때

코드 작성

 도 ▾ 4 ▾ 음을 0.5 박자 연주하기

 0.25 박자 쉬기

- 각자 원하는 노래를 코드로 작성한 후 비글을 뒤집으면 노래를 부릅니다.

## 예시

클릭했을 때

무한 반복하기

만약  $\text{가속도 센서 z (g)} > 1$  또는  $\text{가속도 센서 z (g)} = 1$  (이)라면

도	4	음을 0.5	박자 연주하기
도	4	음을 0.5	박자 연주하기
솔	4	음을 0.5	박자 연주하기
솔	4	음을 0.5	박자 연주하기
라	4	음을 0.5	박자 연주하기
라	4	음을 0.5	박자 연주하기
솔	4	음을 1	박자 연주하기
파	4	음을 0.5	박자 연주하기
파	4	음을 0.5	박자 연주하기
미	4	음을 0.5	박자 연주하기
미	4	음을 0.5	박자 연주하기
레	4	음을 0.5	박자 연주하기
레	4	음을 0.5	박자 연주하기
도	4	음을 1	박자 연주하기



## 03

## 가속도 센서x, y 값 활용하기

- 이번엔 가속도 센서x, y의 값을 활용한 코드를 만들어 봅니다.
- 비글의 앞이나 뒤로 힘을 가해서 로봇이 움직이면 X축 방향으로 가속도가 작용해 값이 변합니다.

비글: 가속도 센서 x (g) **-0.005859**



비글: 가속도 센서 x (g) **0.317383**



- 비글의 왼쪽이나 오른쪽으로 힘이 가해져 로봇이 움직이면 Y축 방향으로 가속도가 작용해 Y축 가속도 값이 변화합니다.

비글: 가속도 센서 y (g) 0.000977

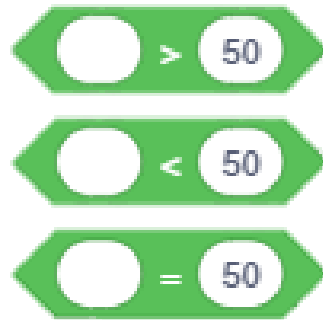


비글: 가속도 센서 y (g) 0.180664

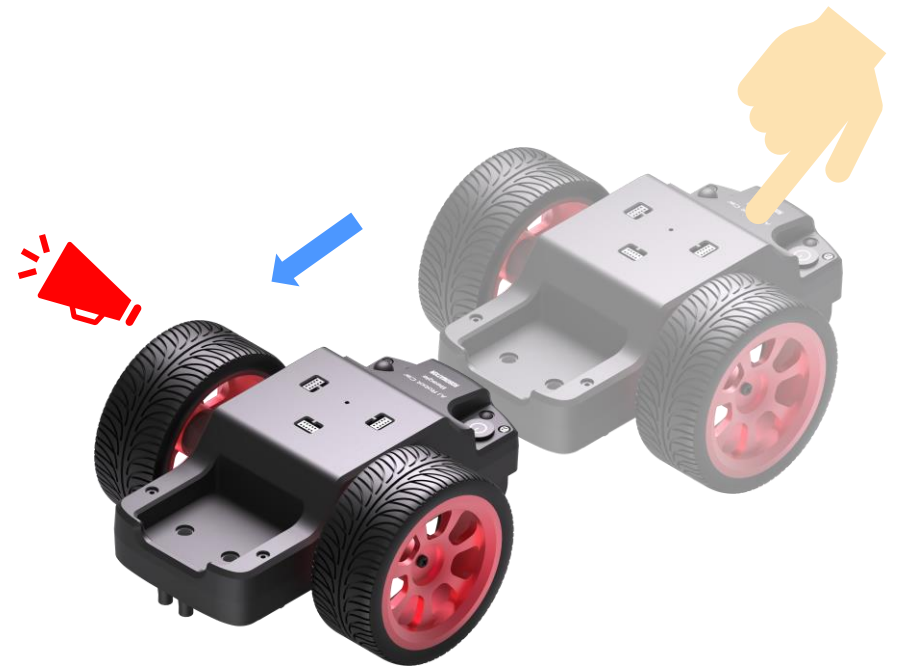


## 가속도 센서 x, y 값 활용하기

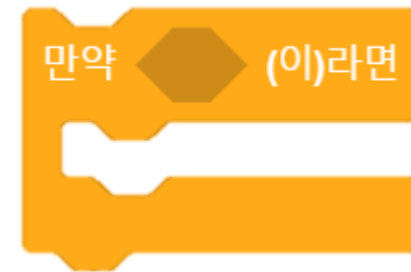
- 먼저 가속도 센서 x 값의 변화를 감지하여 삐 소리를 내도록 합니다.
- 가속도 센서 x (g) 블록과 변수 값의 크기를 비교하는 데 사용할 수 있는 연산 블록, 조건의 참과 거짓을 확인할 수 있는 블록을 조합해 x축 값의 크기 변화를 감지하는 코드를 작성합니다.



- 아래와 같이 코드를 작성하고 비글의 후면을 손가락으로 툅 치면 이를 감지하여 삐 소리를 냅니다.



- 이번엔 어느 방향으로든 로봇을 툭 치면 소리를 내도록 코드를 작성해 봅니다.
- 힘이 작용하는 방향에 관계없이 값이 바뀌기만 해도 삐 소리가 날 수 있도록 해야 하므로 숫자의 부호에 관계 없이 값을 판단할 수 있는 절댓값 블록을 활용합니다.





- 아래와 같이 코드를 작성하고 비글을 손가락으로 툭 치면 어느 방향에서 비글에 힘을 가하더라도 이를 감지하여 삐 소리를 냅니다.

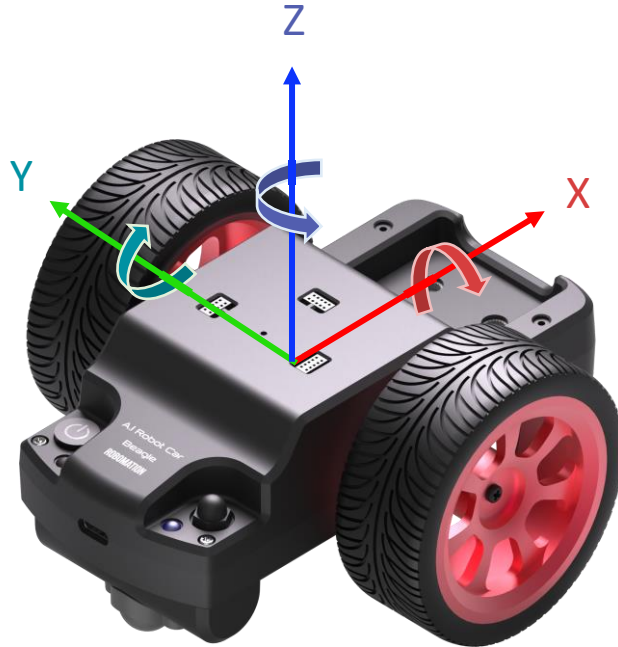
클릭했을 때

무한 반복하기

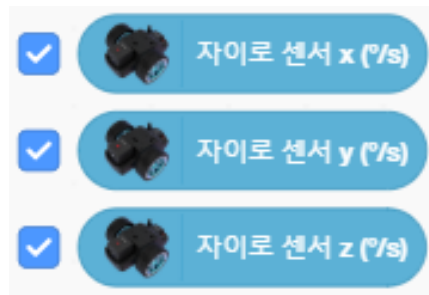
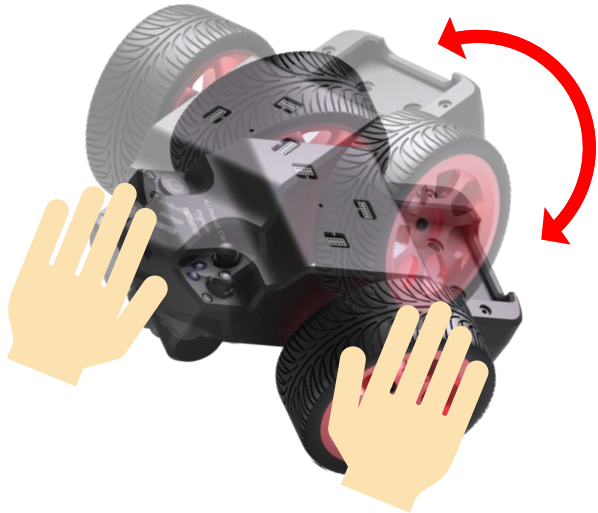
만약  $\left( \begin{array}{c} \text{절댓값} \downarrow \\ \text{가속도 센서 x (g)} \end{array} \right) > 0.1$  또는  $\left( \begin{array}{c} \text{절댓값} \downarrow \\ \text{가속도 센서 y (g)} \end{array} \right) > 0.1$  (아)라면

삐  $\downarrow$  소리 1 번 재생하기

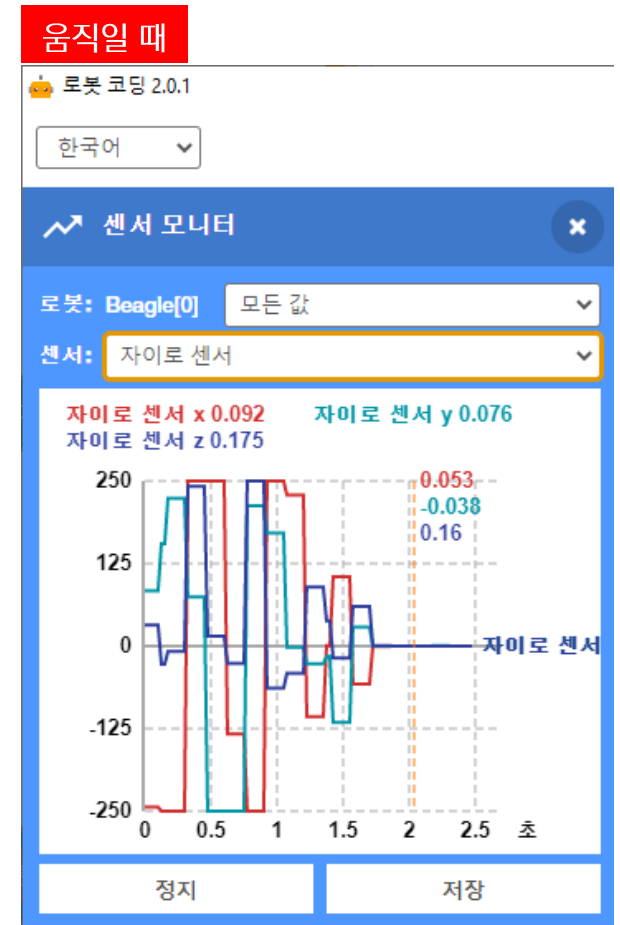
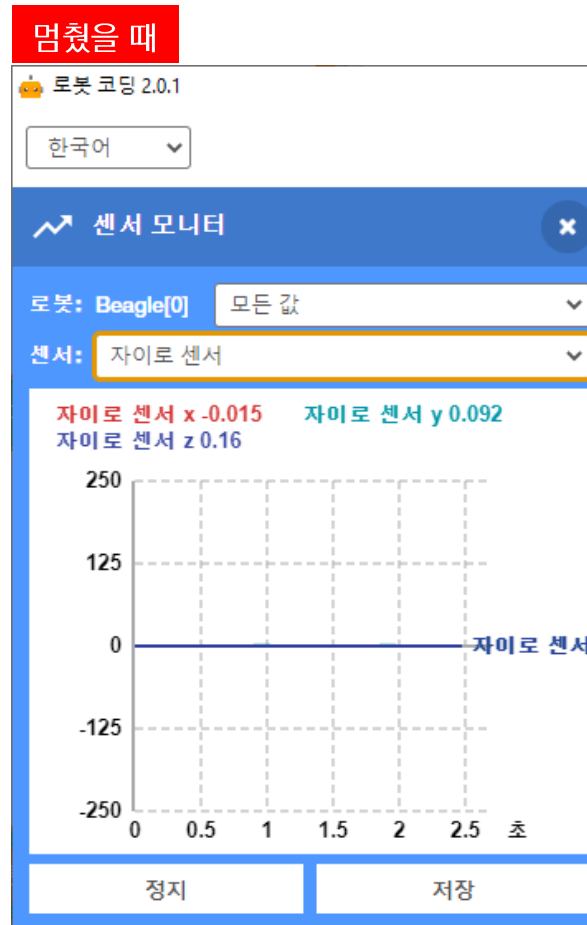
- 비글에는 각 축에 대한 각속도의 값을 감지하는 자이로 센서가 내장되어 있습니다.
- X축은 비글을 오른쪽으로 기울일 때 양수 값, 왼쪽으로 기울일 때 음수값입니다. Y축은 앞으로 기울일 때 양수 값, 뒤로 기울일 때 음수 값입니다. Z축은 비글을 평행하게 놓고 반시계 방향으로 돌리면 양수 값, 시계 방향으로 돌리면 음수 값입니다.



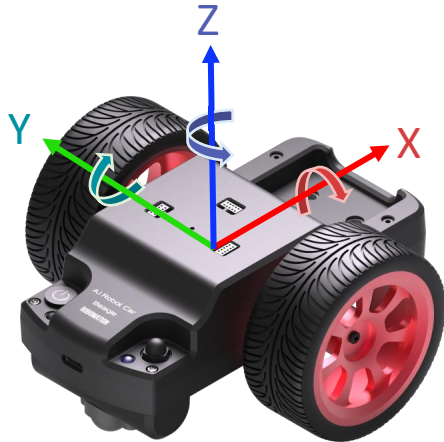
- 각 축에 대한 각속도 값은 자이로 센서 x (°/s), 자이로 센서 y (°/s), 자이로 센서 z (°/s) 블록을 사용하여 알 수 있습니다. 각 블록 앞의 체크 박스를 클릭하면 무대에서 각 축의 각속도 값을 실시간으로 확인할 수 있습니다.
- 프로그램을 구현하기 전에 우선 비글을 들고 움직이며 어떤 방식으로 움직일 때 각 축의 값이 변화하는지 확인합니다.



- 아래와 같이 로봇 코딩의 센서 모니터 기능으로도 자이로 센서의 값을 확인해 볼 수 있습니다.
- 각 축의 각속도 값의 변화가 그래프 형태로 나타납니다.



- 비글을 가만히 두면 움직임으로 인한 각속도의 변화가 없으므로 X축, Y축, Z축의 각속도 값은 0에 가까운 값을 가집니다.



비글: 자이로 센서 x ( $^{\circ}/s$ )

0.022888

비글: 자이로 센서 y ( $^{\circ}/s$ )

-0.190735

비글: 자이로 센서 z ( $^{\circ}/s$ )

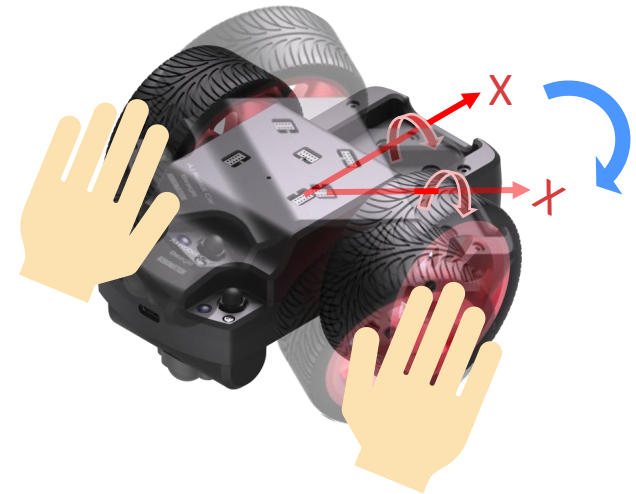
0.183105

- 비글을 왼쪽이나 오른쪽으로 기울이는 순간 X축의 각속도 값이 변화합니다.

비글: 자이로 센서 x (°/s) **-0.068665**



비글: 자이로 센서 x (°/s) **249.992371**

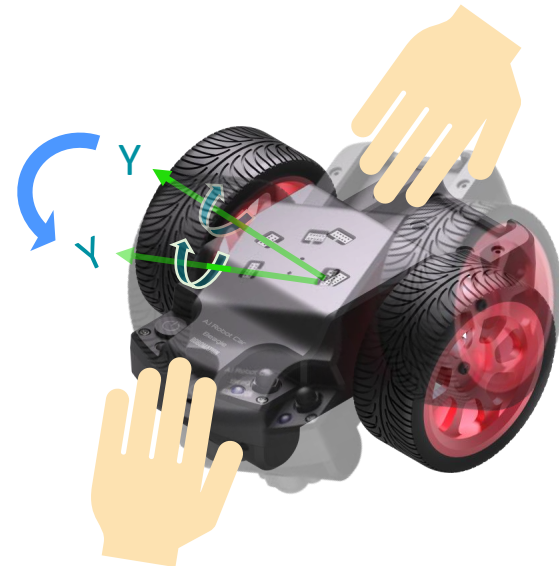


- 비글을 앞쪽이나 뒤쪽으로 기울이는 순간 Y축의 각속도 값이 변화합니다.

비글: 자이로 센서 y (°/s) **0.106812**



비글: 자이로 센서 y (°/s) **-156.12793**



- 비글을 평평한 곳에 올려놓고 시계 방향이나 반시계 방향으로 돌리는 순간 Z축의 각속도 값이 변화합니다.

비글: 자이로 센서 z (°/s) **0.228882**



비글: 자이로 센서 z (°/s) **249.504089**

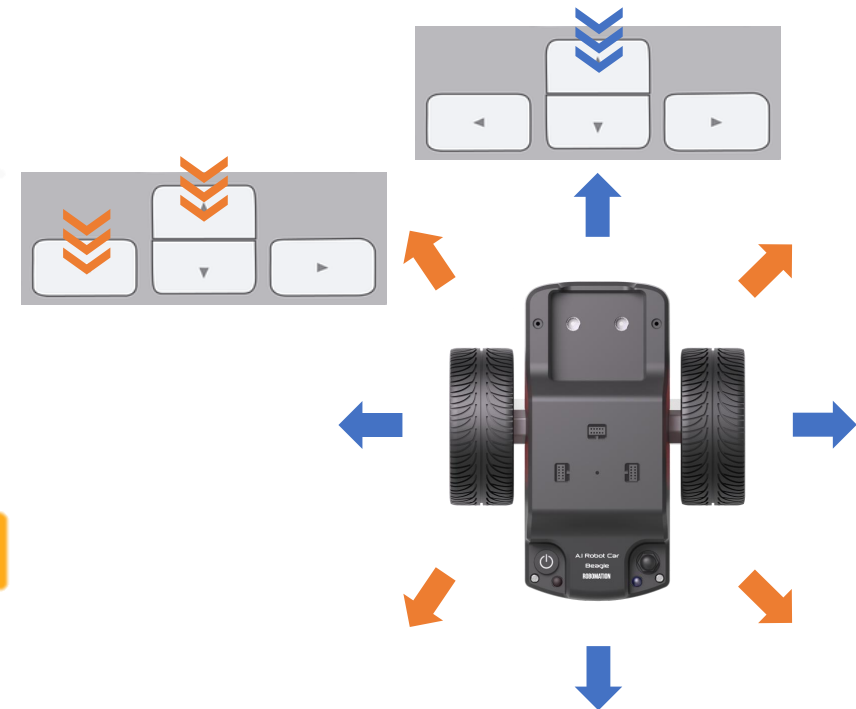
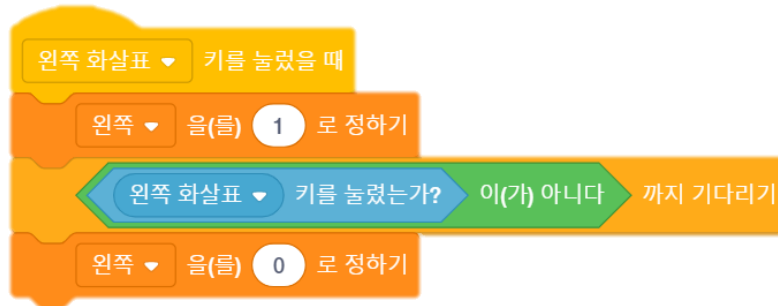




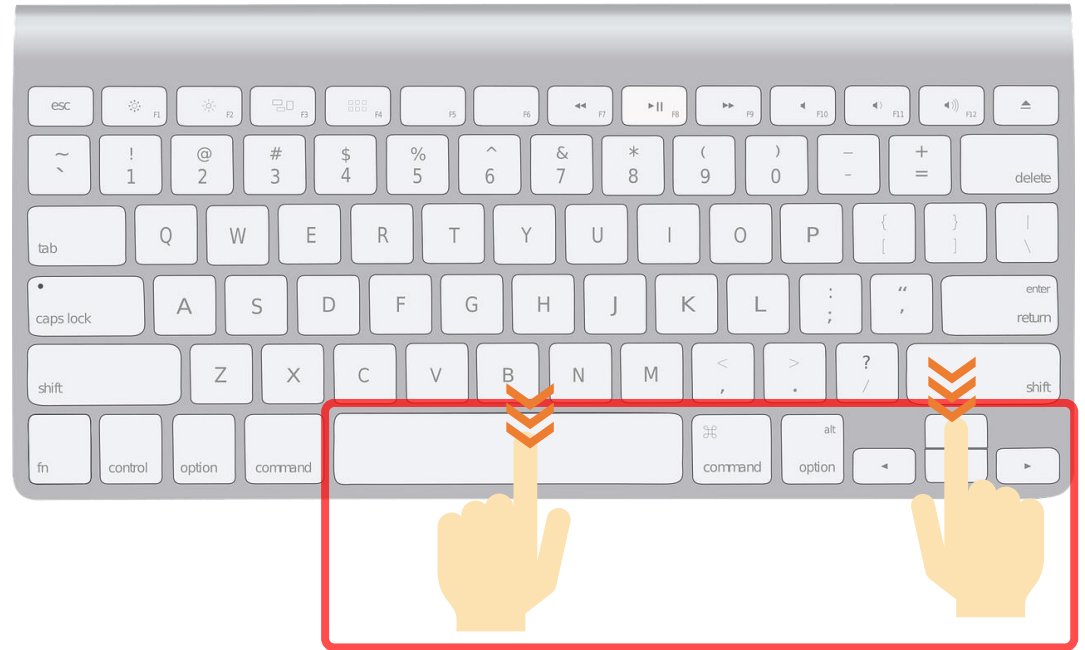


스크래치 코드  
다운로드

- 이제 자이로 센서로 비글의 난폭 운전을 감지하는 코드를 작성해 봅니다.
- 먼저 오른쪽의 스크래치 코드 다운로드를 클릭해 키보드로 비글을 조종하는 코드를 다운받습니다.
- 방향키를 누르고 있으면 해당 방향으로 비글이 움직입니다. 주황색 화살표처럼 대각선으로 이동하기 위해서는 두 방향의 키를 동시에 누르면 됩니다.



- 방향키로 비글을 조종하면서 스페이스 키를 동시에 누르면 부스터 모드가 되어 비글의 속도가 빨라집니다.



- 코드의 바퀴 값을 바꾸면 속도를 더욱 빠르거나 느리게 할 수 있습니다.

The image shows a Scratch script for a robot. It starts with a 'Clicked when green flag clicked' event. A 'Repeat forever' loop contains the following blocks:

- Condition: `위 + 아래 와(과) 왼쪽 + 오른쪽 결합하기 > 0 (이)라면`
- Condition: `부스터 = 1 (이)라면`
- Condition: `위 = 1 (이)라면`
- Action: `왼쪽 바퀴 70 % 오른쪽 바퀴 70 %로 정하기`
- Condition: `아래 = 1 (이)라면`
- Action: `왼쪽 바퀴 -70 % 오른쪽 바퀴 -70 %로 정하기`
- Condition: `오른쪽 = 1 (이)라면`
- Action: `왼쪽 바퀴 70 % 오른쪽 바퀴 -70 %로 정하기`

The values 70 and -70 in the motor power blocks are highlighted with red boxes.

- 자이로 센서  $y$  ( $^{\circ}/s$ )값의 변화를 감지하여 삐 소리를 2번 내도록 합니다.
- 자이로 센서  $y$  ( $^{\circ}/s$ )블록과 변수 값의 크기를 비교하는데 사용할 수 있는 연산 블록, 조건의 참과 거짓을 확인할 수 있는 블록을 조합해  $y$ 축 각속도 변화를 감지하는 코드를 작성합니다.
- 힘이 작용하는 방향에 관계없이 값이 바뀌기만 해도 소리가 나야 하므로 숫자의 부호에 관계 없이 판단할 수 있는 절댓값 블록을 활용합니다.

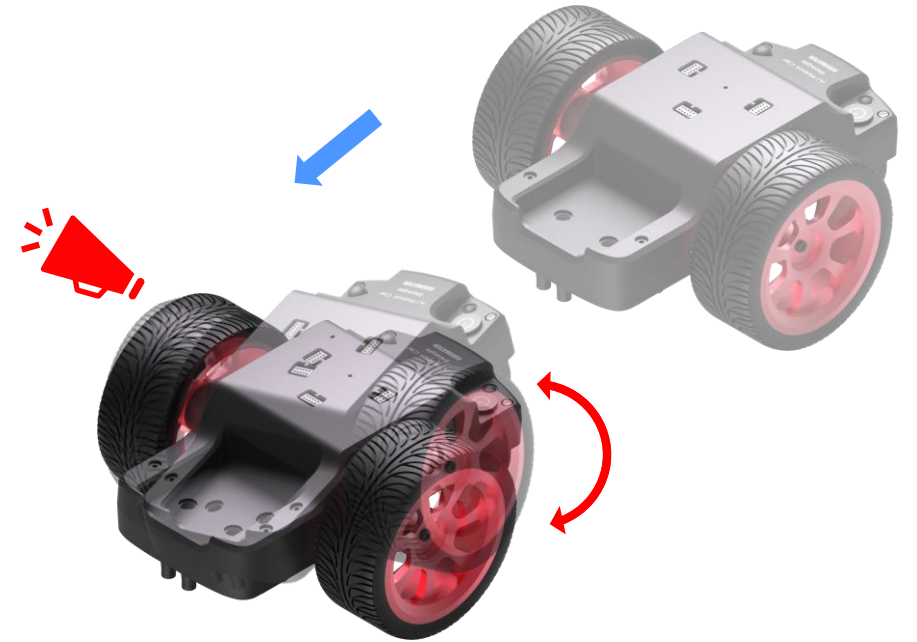


- 아래와 같이 코드를 작성하고 비글을 앞쪽이나 뒤쪽으로 기울이면 이를 감지하여 삐 소리를 냅니다.
- 이 코드로 비글이 빠르게 이동하다 급정거할 때 뒷부분이 살짝 들리는 것을 감지합니다.

무한 반복하기

만약  $\left( \text{절댓값} \left( \text{자이로 센서 } y \text{ (}^\circ/\text{s)} \right) > 25 \right)$  (이)라면

삐 소리 2 번 재생하기



- 자이로 센서  $z$  ( $^{\circ}/s$ )값의 변화를 감지하여 사이렌 소리를 내도록 합니다.
- 자이로 센서  $z$  ( $^{\circ}/s$ )블록과 변수 값의 크기를 비교하는데 사용할 수 있는 연산 블록, 조건의 참과 거짓을 확인할 수 있는 블록을 조합해  $z$ 축 각속도 변화를 감지하는 코드를 작성합니다.
- 힘이 작용하는 방향에 관계없이 값이 바뀌기만 해도 소리가 나야 하므로 숫자의 부호에 관계 없이 판단할 수 있는 절댓값 블록을 활용합니다.

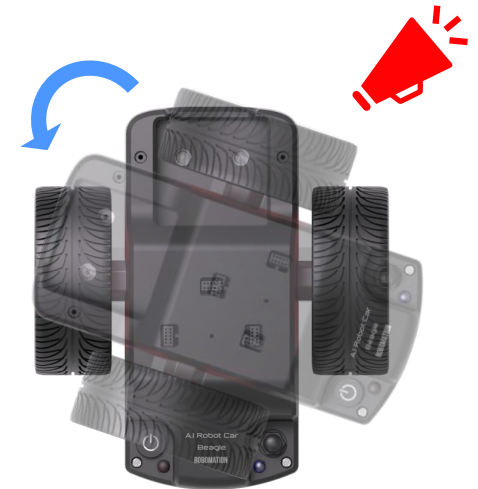


- 아래와 같이 코드를 작성하고 비글을 빠른 속도로 왼쪽이나 오른쪽 돌리면 이를 감지하여 사이렌 소리를 냅니다.
- 이 코드로 비글이 빠른 속도로 좌우로 회전하는 것을 감지합니다.

무한 반복하기

만약 절댓값 ( 자이로 센서 z (°/s) ) > 200 (이)라면

사이렌 소리 1 번 재생하기



- 키보드로 조종하는 코드에 아래 코드를 더한 후 비글을 빠른 속도로 움직이도록 조종하면 비글이 난폭하게 운전하는지를 감지할 수 있습니다.
- 비글을 조종할 때 변화하는 자이로 센서 값을 관찰하고 각자가 정한 규칙과 상황에 맞게 입력 값을 바꿔 봅니다.

The image shows a Scratch script starting with a 'Clicked when green flag clicked' event. It then enters an 'Infinite loop' block. Inside the loop, there are two conditional 'if' blocks. The first 'if' block checks if the absolute value of the 'Gyro sensor y (°/s)' is greater than 25. If true, it plays the 'Bee' sound 2 times. The second 'if' block checks if the absolute value of the 'Gyro sensor z (°/s)' is greater than 200. If true, it plays the 'Siren' sound 1 time. The script ends with a 'Repeat from top' arrow.

```
클릭했을 때
무한 반복하기
만약 절댓값 ( 자이로 센서 y (°/s) ) > 25 (이)라면
  삐 소리 2 번 재생하기
만약 절댓값 ( 자이로 센서 z (°/s) ) > 200 (이)라면
  사이렌 소리 1 번 재생하기
↶
```



감사합니다.

