

# ROKIT® Smart

로봇을 좋아하는 학생부터 선생님까지 누구나 쉽고 재미있게 배우는  
로킷 스마트 SW 에듀케이션 키트

컨트롤러에 내장된 센서로  
즐기는 다양한  
엔터테인먼트  
컨텐츠



간편한 시각적 UI로 누구나  
쉽고 재미있게  
배우는 코딩  
에듀케이션

만든 로봇에  
프로그램을  
넣어 나만의  
움직이는 발명로봇을 만들어보자!



코딩은 주어진  
문제를 해결할  
능력과 컴퓨팅  
사고 능력을 증진시켜준다



BUILD, CODE, PLAY

# 목 차

## 1. 로켓 브릭(Rokit Brick) 사용하기

0장. 프로그램 소개	3
1장. 블록, 스크립트, 스프라이트	9
2장. 스크립트 만들기	20
3장. 변수와 반복 그리고 조건	31
4장. 그림 그리기와 스프라이트 조종하기	48

## 2. 디바이스 제어하기

1장. 인벤터 내장보드 시뮬레이션 - 전방 센서값 점검하기	60
2장. 장치 연동 게임 시뮬레이션 - 로켓 시뮬레이터 게임	75
3장. 로봇 구동 시뮬레이션 - 로켓 시뮬레이터 로봇 시뮬레이션	90
4장. 자신만의 시뮬레이터 만들기- 로켓 시뮬레이터 응용 시뮬레이터	107

## 3. 로켓 기본 모델 구동하기

0장. 로봇이란?	116
1장. 간단히 로봇 조립 방법 익히기 / 지능 로봇 조립	120
2장. 어보이더	141
3장. 바닥감지로봇	148
4장. 라인트레이서	155

## 1. Rokit Brick 사용하기

# 0장.

# 프로그램 소개

## 01. 스크래치 란?



# SCRATCH



S4A : Scratch for Arduino

Java와 C++같은 대부분의 프로그래밍 언어는 텍스트 기반으로 구성되어 있기 때문에 초보자들이 접근하기에 많은 어려움이 있습니다.

하지만 미국 MIT 대학의 미디어랩 (Media Lab)에서 개발한 스크래치(Scratch)는 시각화된 프로그래밍 언어로서 보다 쉽고 재미있게 프로그래밍을 배울 수 있도록 하였습니다.

본래 스크래치는 아이들을 위한 교육용으로 제작되어졌으며 컴퓨터 프로그래밍에 관한 풍부한 경험을 쌓을 수 있도록 하는데 그 목적을 두고 있습니다.

하지만 컴퓨터 프로그래밍에 관심이 있는 일반 입문자에게도 적극 권장합니다.

Arduino와 같은 Text 기반의 프로그램도 있지만, 문법적인 기초지식이 필요하여 다루기 어렵기 때문에 최근에 Scratch방식으로 제어되는 Scratch for Arduino, Snap for Arduino 등이 만들어 지고 있습니다.

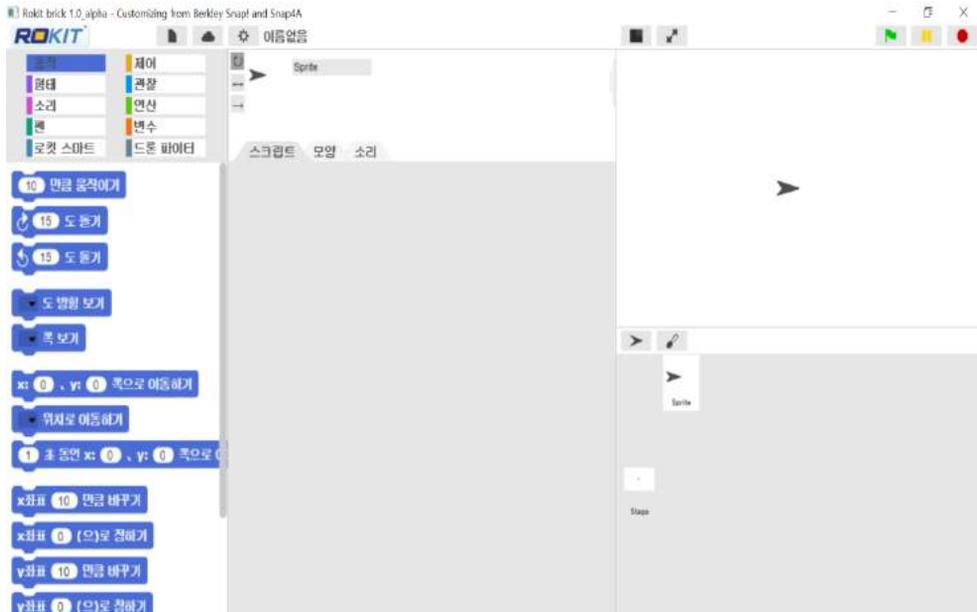
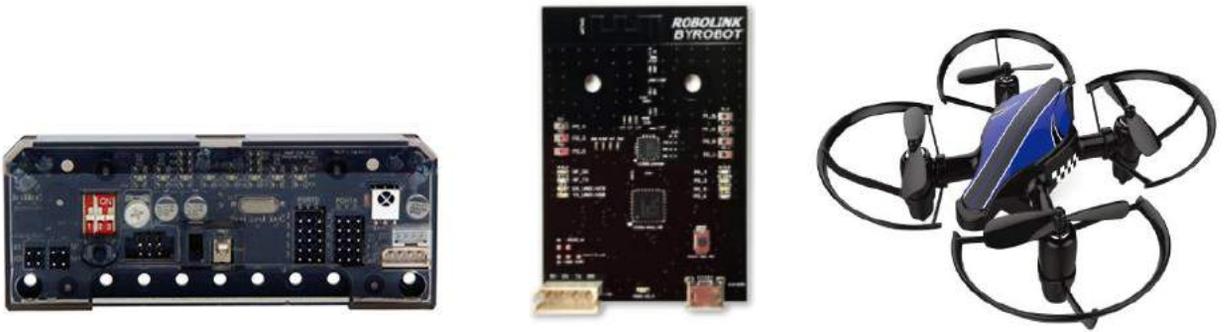
이러한 프로그램도 모두 Scratch를 기반으로 프로그래밍 하게하는 유사 프로그램입니다.

## 02. Rokit Brick 이란?



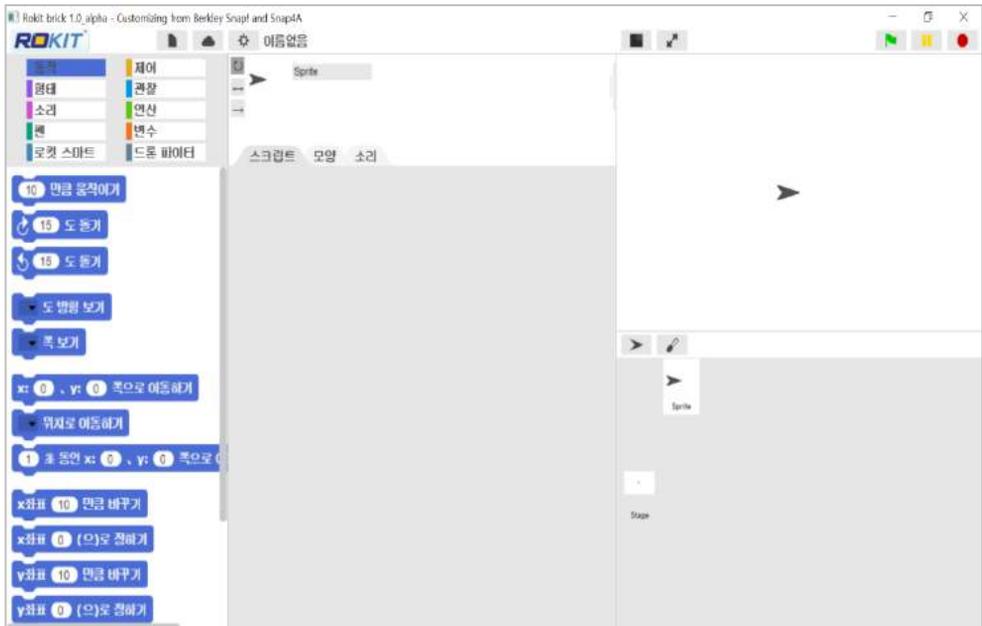
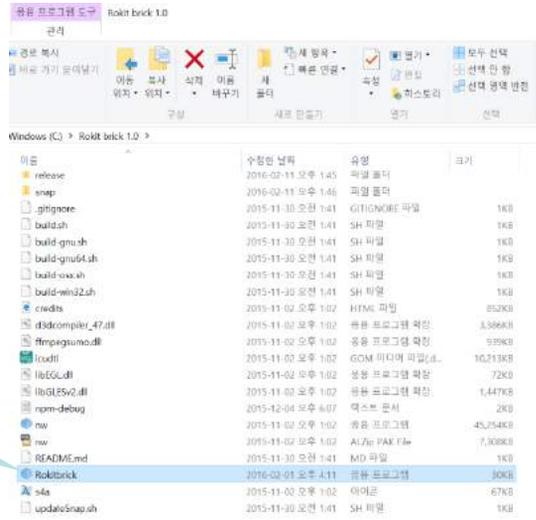
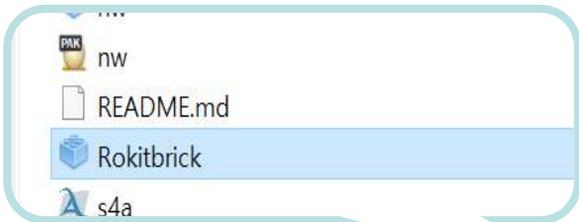
Rokit Brick은 Snap4Arduino를 기반으로 기존 Scratch 프로그램에 로봇, 드론 등 다양한 제어 기능 등이 합쳐져 개발된 Scratch 방식의 SW이며, 사용방법은 Scratch와 동일합니다.

로킷 스마트와 드론 파이터 등의 버튼들을 통해 HW를 실시간으로 Scratch 형식으로 제어할 수 있게 됩니다.



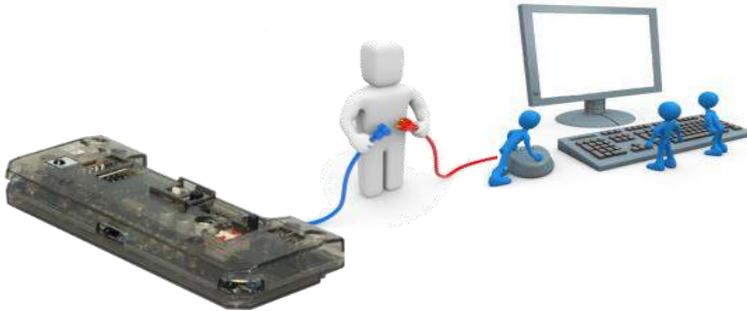
### 03. Rokit Brick 실시

Rokit Brick은 간단히 프로그램 폴더를 원하는 경로에 위치시키고 **Rokitbrick.exe** 실행파일을 클릭하면 됩니다.



## 04. Rokit Brick 로봇제어를 위한 인벤터 보드와 PC연결

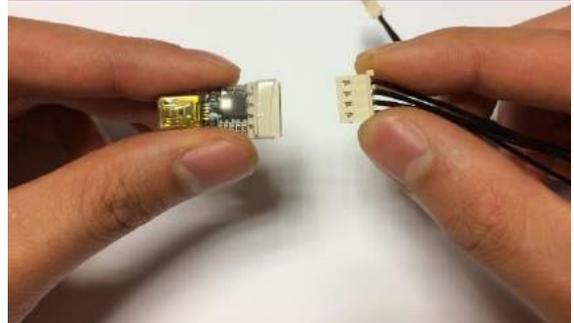
Rokit Brick 프로그램으로 인벤터 보드를 제어하기 위해서는 다음과 같이 연결이 필요합니다.



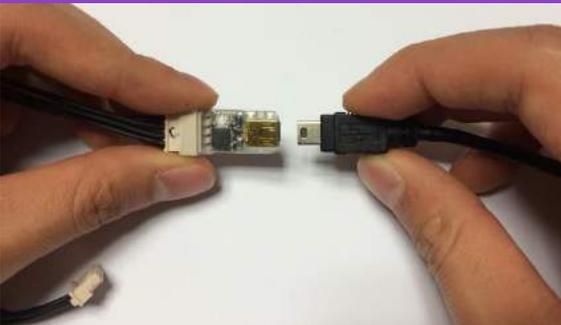
1. 인벤터 보드의 DIP스위치를 모두 올립니다.



2. Downloader와 Board Cable을 연결합니다.



3. Downloader와 NEXI Cable을 연결합니다.



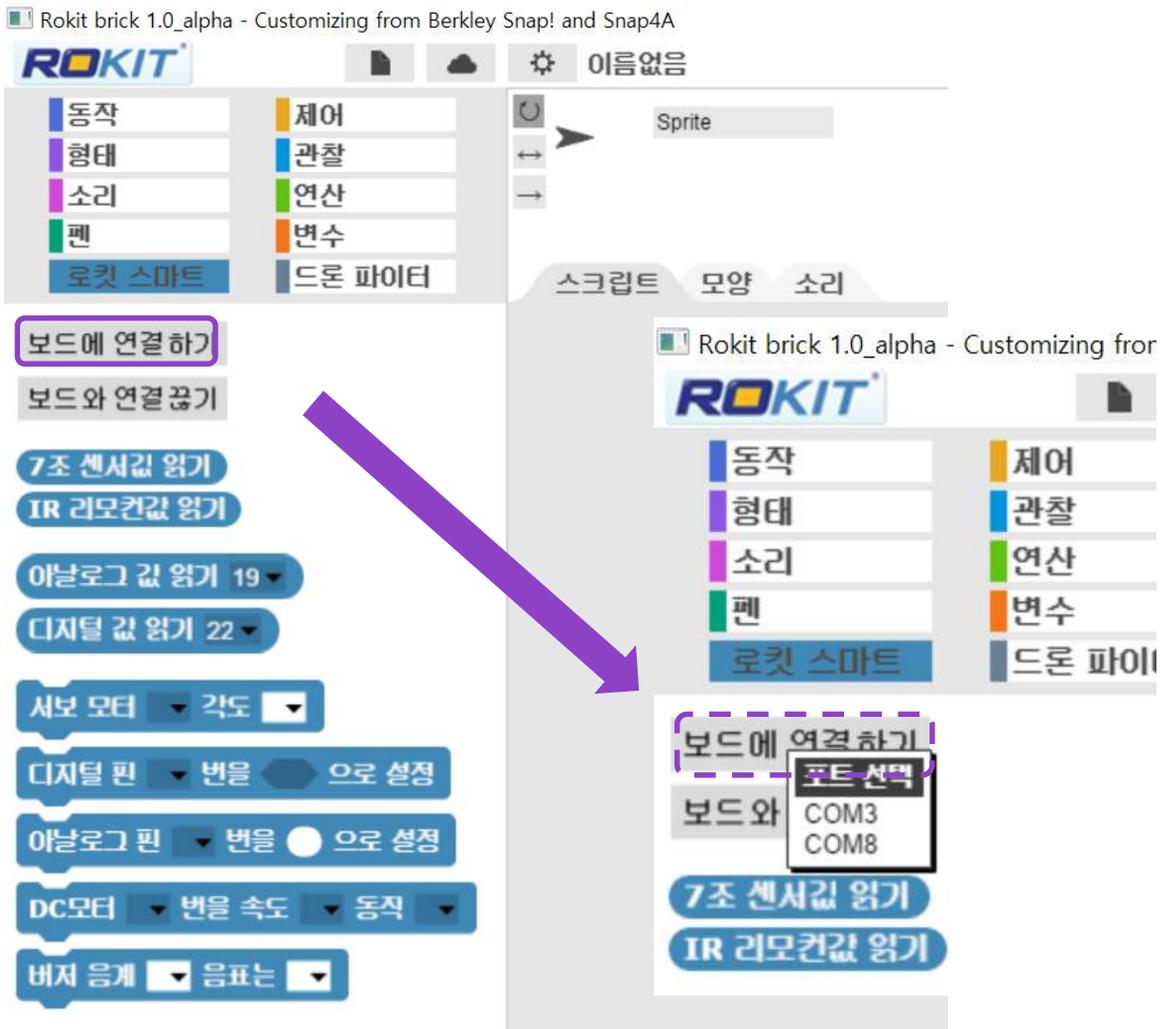
4. 인벤터 보드와 컴퓨터를 연결합니다.



## 05. 로킷 브릭(Rokit Brick)과 인벤터 보드 SW Connect

PC와 인벤터 보드를 연결한 후 제일 먼저 USB 케이블 드라이브 설치를 확인하여야 합니다.  
([www.rokitarduino.org](http://www.rokitarduino.org) 참조)

드라이브 설치 및 COM번호를 확인한 후 로킷 브릭(Rokit Brick)의 로킷 스마트 블록 카테고리  
가면 **보드에 연결하기**가 나옵니다.



## 1. Rokit Brick 사용하기

### 1장.

블록, 스크립트,

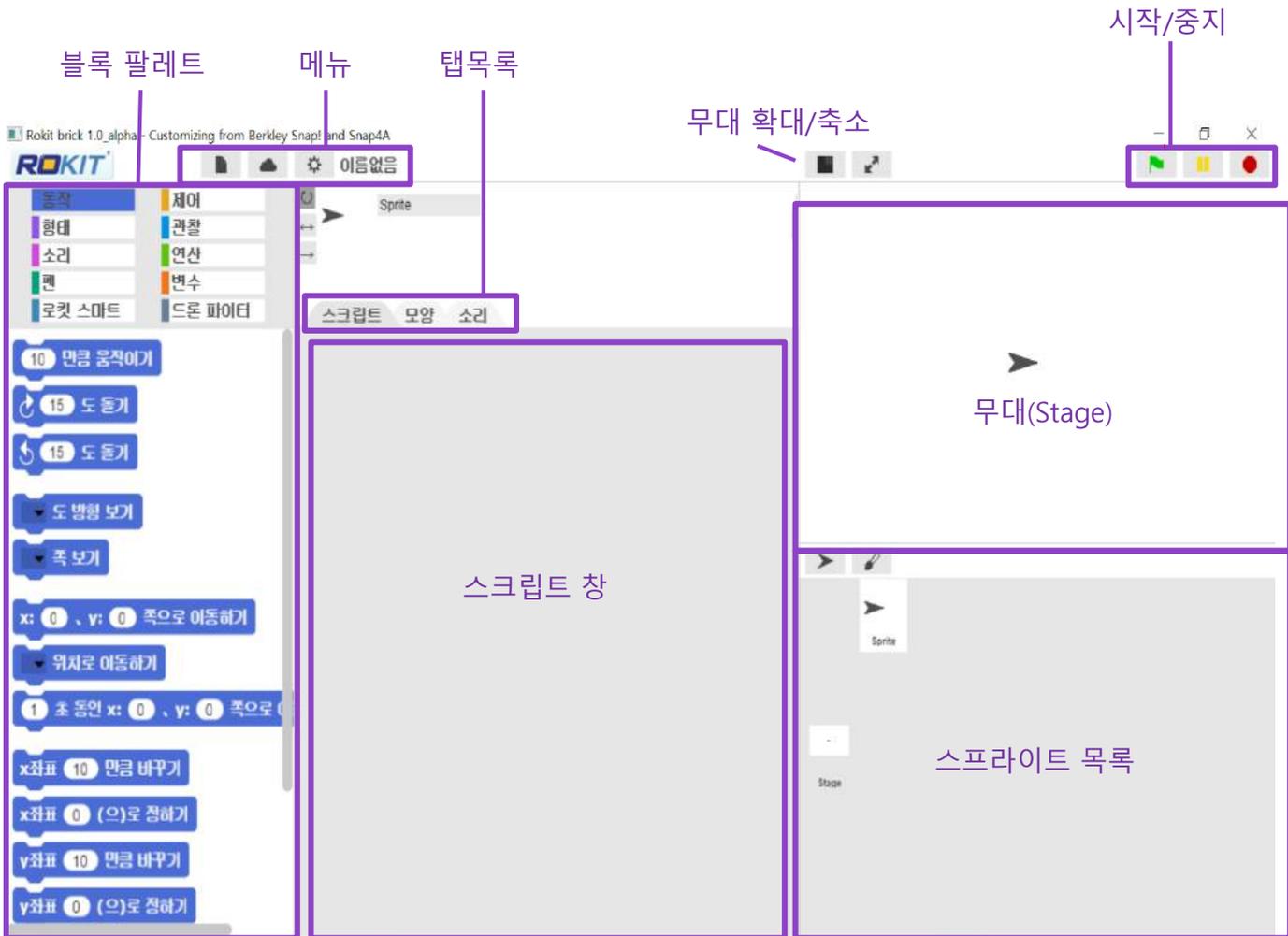
스프라이트

# 1.1 Rokit Brick 구성

## - 화면 구성

기본적인 모든 메뉴와 화면구성은 Scratch와 유사합니다.  
 비교하여 동작해보면 보다 쉽게 빨리 적응할 수 있습니다.

Scratch 유저라면 바로 구동해 보십시오.



## - 메뉴/설정

메뉴를 통해 파일을 읽고 쓰기하면 됩니다.

열기, 저장하기는 PC 내부로 저장되며, 외부로 파일을 공유하려면 가져오기, 블록내보내기 등을 통해 .XML 파일형식으로 활용합니다. 언어는 원하는 형식으로 선택할 수 있습니다.



## 1.2 블록, 스크립트, 스프라이트란?

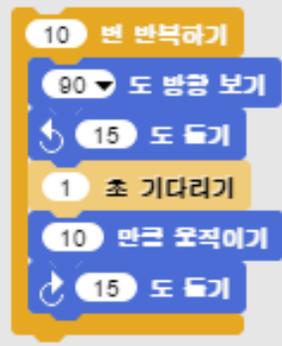
Rokit Brick에는 프로그래밍을 할 때 쓰이는 기본 단위가 있습니다.  
블록과 스크립트, 그리고 스프라이트에 대해 알아보시다.

### 블록



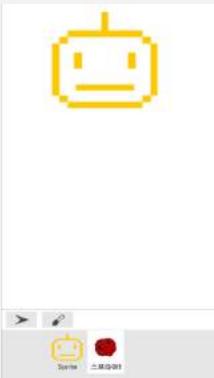
하나 하나 낱개의 단위를 '블록'이라 합니다.  
블록마다 고유색상을 가지고 있어, 색상을 기억하고  
있으면 프로그래밍하기가 수월해집니다.

### 스크립트



각각의 블록을 스크립트 창으로 드래그하여 두 개  
이상의 블록이 합쳐진 것을 '스크립트'라고 부릅니다.

### 스프라이트



스프라이트란 Rokit Brick의 핵심 개체이며 여러가지의  
스프라이트가 모여 하나의 프로젝트를 구성합니다.  
모형부터 소리까지 전반적인 역할을 주관하게 됩니다.

## 2.1 햇 블록 사용하기

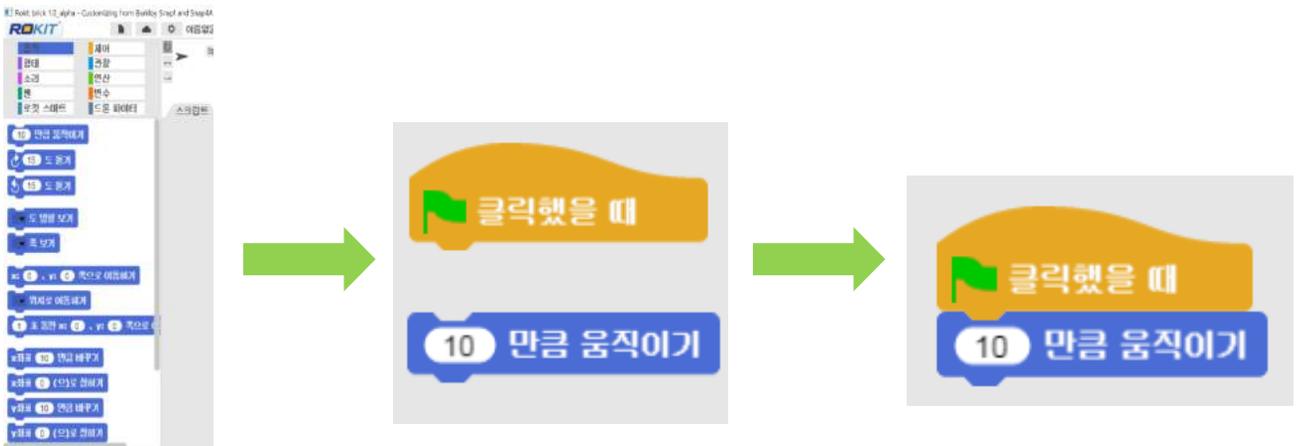
모자처럼 생긴 햇(Hat) 블록은 명령을 시작하게 하는 시작 블록입니다.



마우스를 이용하여 왼쪽 클릭 후 드래그 하여 스크립트 창에 원하는 위치에 놓으면 됩니다.



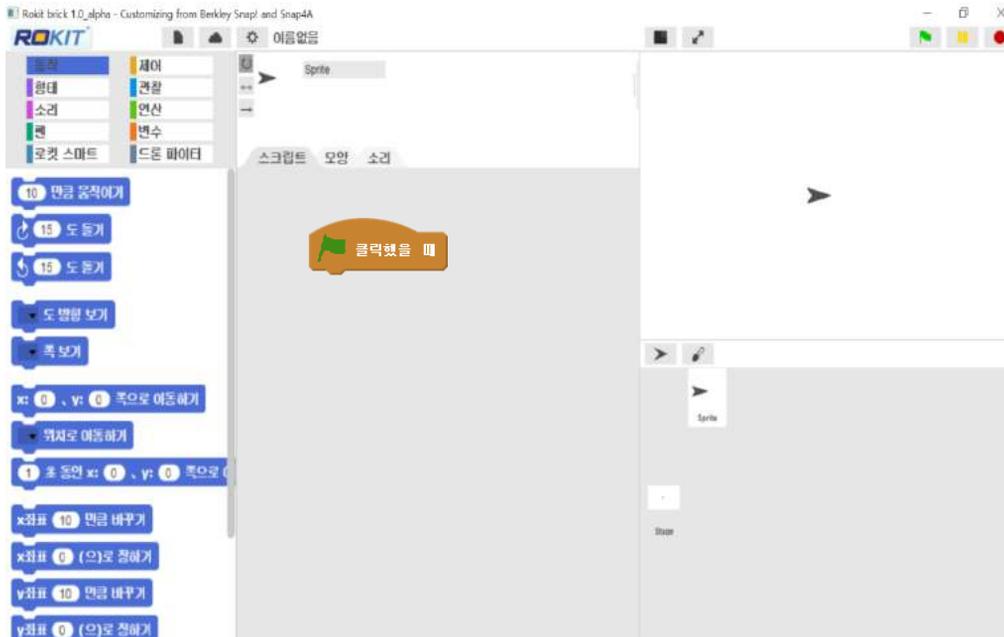
동작 블록 중에서 다른 하나의 블록을 같은 방법으로 햇 블록 아래에 붙여봅니다. 이렇게 모여지면 하나의 스크립트가 됩니다.



## 2.2 동작 블록 사용하기

실제 스프라이트의 이미지를 움직이게 하는 블록을 동작블록이라 하며, 헛 블록과 함께 사용하여야 동작이 가능합니다.

### 10 만큼 움직이기



헛 블록과 같이 동작 블록을 연결하고 어떻게 동작하는지 실행하여 봅시다.



## 2.3 무한 반복 블록 사용하기

무한 반복 블록은 안에 들어 있는 블록들을 무한 반복시키거나 무한반복 블록 안에 있는 조건블록을 지속적으로 사용하게 해주는 블록입니다.

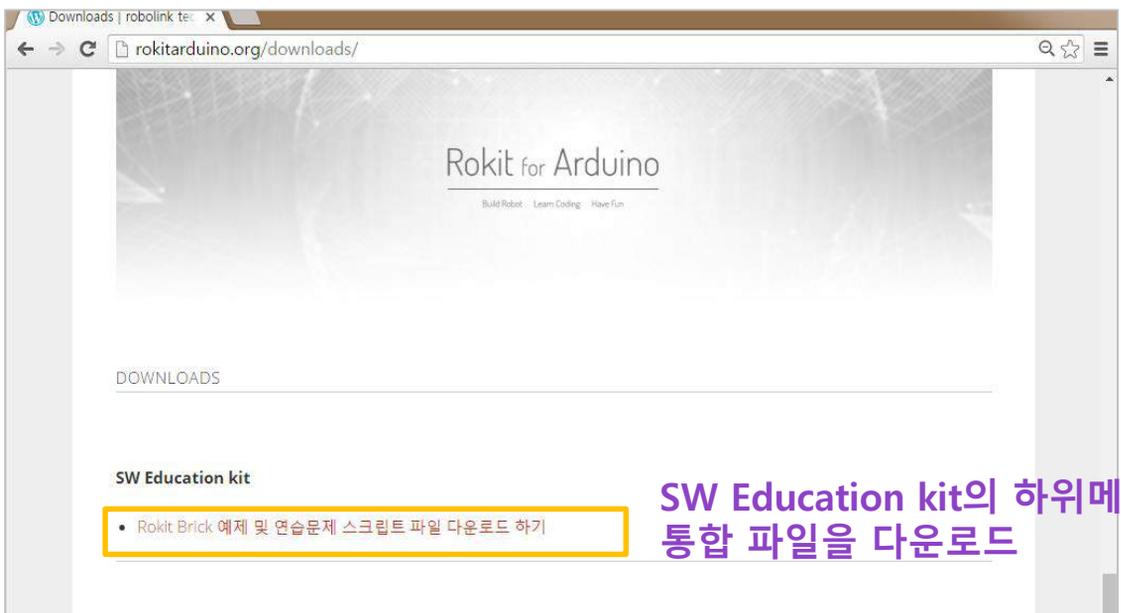
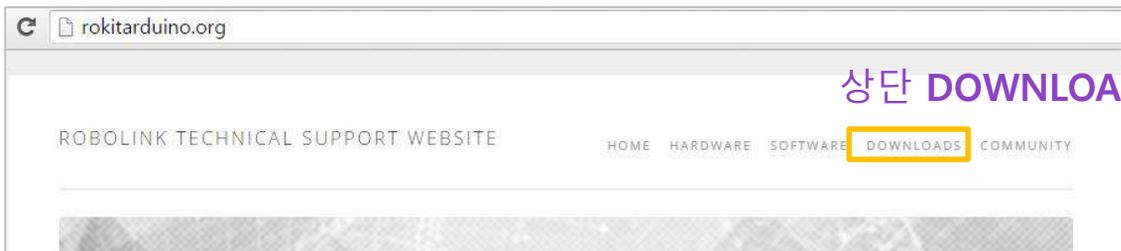


무한 반복 블록 같이 동작 블록을 연결하고 어떻게 동작하는지 실행하여 봅시다.



### 3.0 예제 및 연습문제 다운로드

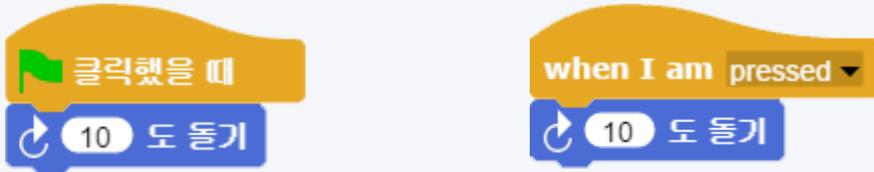
교재에서 요구하는 예제와 연습문제는 로보링크 기술지원 홈페이지 [rokitarduino.org](http://rokitarduino.org)로 접속하셔서 오른쪽 상단의 **DOWNLOADS** (다운로드) 메뉴를 클릭, **SW Education kit**의 **Rokit Brick 예제 및 연습문제 스크립트 파일 다운로드 하기** 에서 통합 파일을 다운로드 받으시면 됩니다.



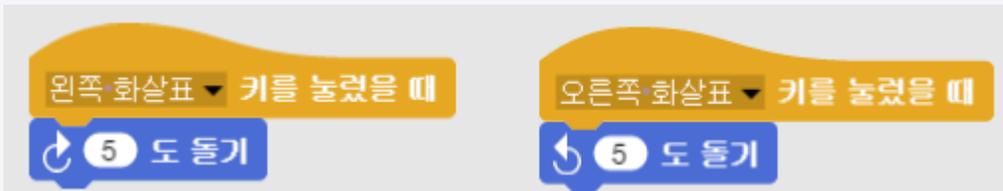
### 3.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 교재 16페이지▶

**예제 1 :** 아래 두 가지 스크립트는 클릭할 때마다 스프라이트가 회전할 수 있도록 해 줍니다.  
 두 스크립트를 실행해 보고 어떤 차이가 있는지 확인해 보세요.



**예제 2 :** 키보드의 왼쪽 화살표 키를 누르면 스프라이트가 왼쪽으로 회전하고 오른쪽 화살표 키를 누르면 오른쪽으로 회전할 수 있도록 해 봅시다.



**예제 3 :** 키보드의 왼쪽 화살표 키를 누르면 스프라이트가 왼쪽으로 이동하고 오른쪽 화살표 키를 누르면 오른쪽으로 이동하게 해 봅시다. 이 때 스프라이트가 이동하다가 스테이지의 벽을 만나면 더 이상 이동하지 않고 멈추게 합시다.



**예제 4 :** '새로운 스프라이트 추가하기' 버튼을 눌러 두 개의 스프라이트를 스테이지에 나타나게 해 봅시다. 스페이스 키를 눌렀을 때 첫 번째 스프라이트는 회전을 하고 두 번째 스프라이트는 좌우로(혹은 아래 위로) 반복해서 움직이게 해 봅시다.

#### 첫 번째 스프라이트



#### 두 번째 스프라이트



### 3.2 연습문제

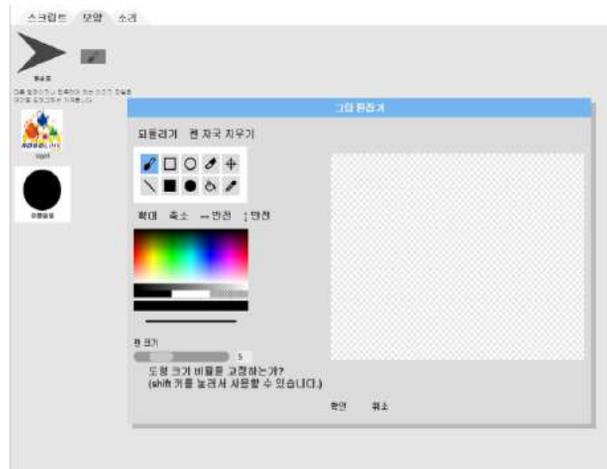
예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 교재 16페이지▶

1. 옆의 그림은 동작 카테고리에서 발견할 수 있는 블록들입니다. 두 블록이 갖고 있는 공통점과 차이점은 무엇인지 조사해 보도록 합니다.



2. 스프라이트를 키보드의 화살표 키에 따라 움직일 수 있게 해 봅시다. 위쪽 화살표를 누르면 위쪽으로 이동하고 아래쪽 화살표를 누르면 아래쪽으로 이동하며 왼쪽 화살표를 누르면 왼쪽으로 이동하고 오른쪽 화살표를 누르면 오른쪽으로 이동하게 스크립트를 작성해 봅시다.

3. 옆의 그림처럼 '모양' 탭을 클릭한 후 붓 모양 버튼을 클릭하여 스프라이트 편집기에서 동그란 원을 그려서 공모양의 새로운 스프라이트를 추가합니다. 그 다음 이 스프라이트가 스테이지 이곳 저곳을 다닐 수 있도록 합니다. 벽에 닿을 때에는 반대편 방향으로 튕겨야 합니다.



4. 스프라이트를 2개 만들어 줍니다. 키보드를 아무거나 누르면 두 개의 스프라이트가 동시에 움직이기 시작하며 자유롭게 스테이지 이 곳 저 곳을 다니게 하여 봅시다. 첫 번째 스프라이트의 움직임은 느리게 하고 두 번째 스프라이트의 속도는 좀 더 빠르게 해 줍니다. 두 스프라이트 모두 벽을 만나면 다른 방향으로 튕겨서 움직여야 합니다.
  
5. 정지해 있던 스프라이트에 마우스 포인터를 갖다 대면 계속 회전할 수 있도록 해 봅시다. 이 때 '관찰' 카테고리의 '마우스의 X좌표' 블록을 사용하여 스프라이트에 마우스 포인터가 가까워지면 회전하는 속도가 느려지고 멀어질 수록 빠르게 돌게 하여 봅시다.

## 1. Rokit Brick 사용하기

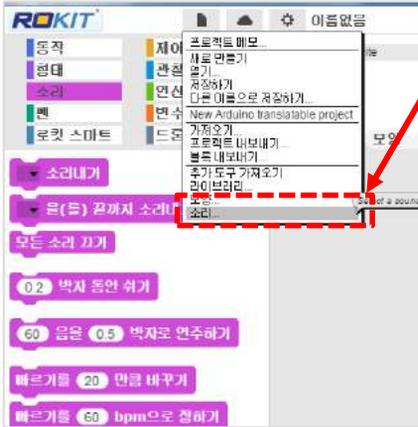
# 2장.

# 스크립트 만들기

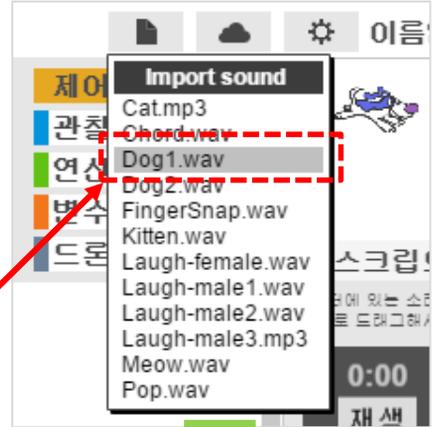
# 1.1 소리와 모양 파일 가져오기

play sound

소리 파일을 특정한 시점에 재생되도록 하면, 스크립트와 스프라이트의 효과를 더할 수 있습니다.



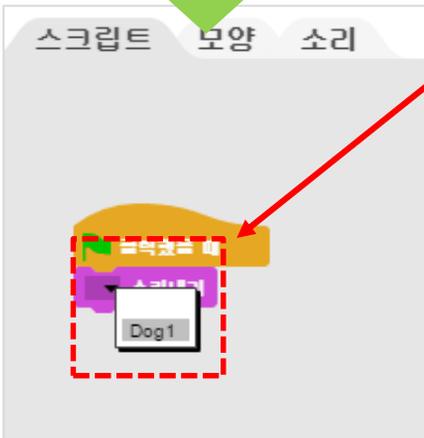
파일에서 소리를 선택합니다.



소리 파일을 선택합니다.



클릭하면 Dog1의 소리가 재생합니다.



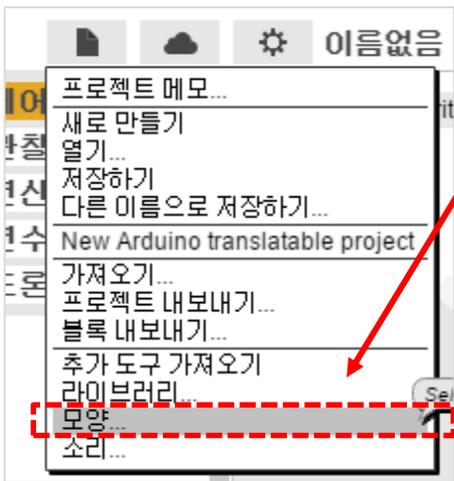
소리내기 블록을 넣은 후 빈칸에 해당 파일은 선택합니다.



클릭하여 실행 해 보세요.

## 1.1 소리와 모양 파일 가져오기

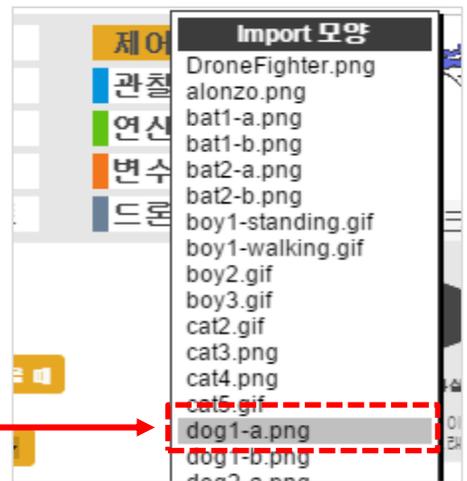
모양 파일은 스프라이트에 자신이 원하는 이미지를 불러와 제어할 수 있으며, 자신이 직접 만든 파일도 불러올 수도 있습니다. 이번 챕터에서는 제공되는 이미지를 선택해 봅니다.



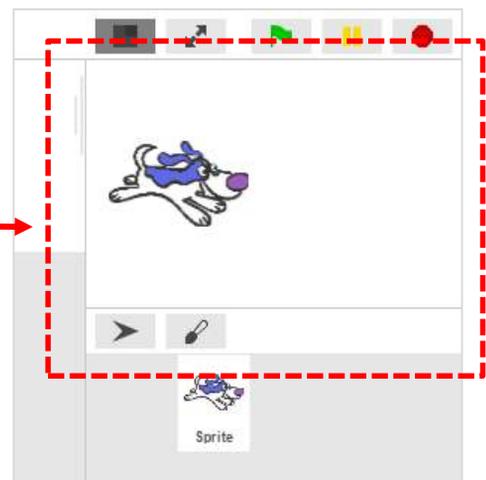
파일에서 모양을 선택합니다.



모양에서 dog1을 선택합니다.



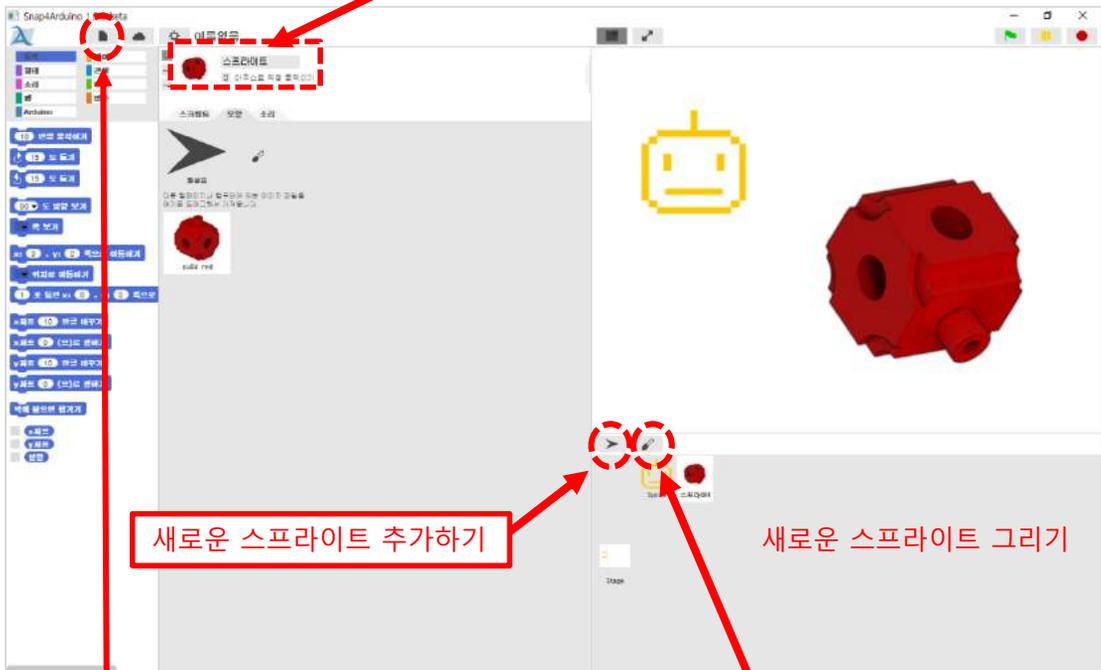
무대에 dog1의 이미지(스프라이트)가 나타났습니다.



## 1.2 스프라이트 새로 추가하기

자신이 직접 만들거나 원하는 파일을 불러올 수도 있습니다. 이번에는 자신이 원하는 이미지를 가져오거나, 또 직접 그린 것도 추가해보도록 하겠습니다.

스프라이트의 이름도 유저가 변경 가능합니다.



파일 - 가져오기 - 원하는 이  
이미지 클릭 하시면 유저가 원하  
는 이미지(스프라이트)로 변경  
이 가능합니다.



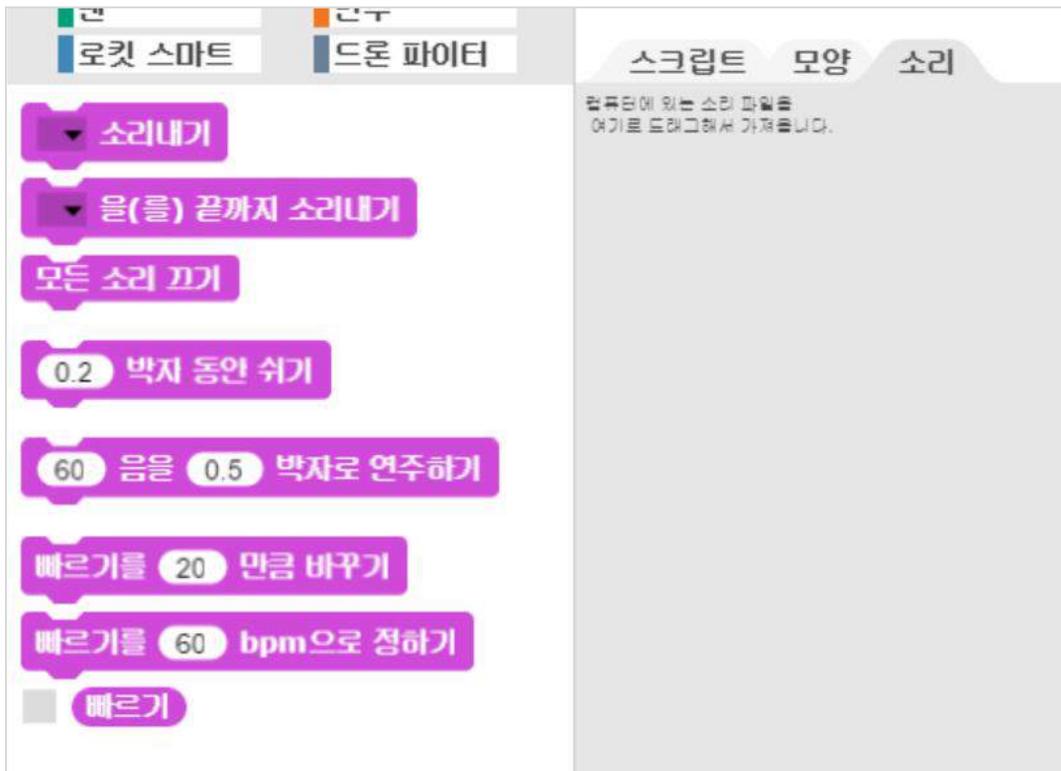
## 2.1 소리내기 블록 사용하기

예제의 소리 파일을 가지고 와서 주인공이 낼 수 있는 효과를 줄 수 있습니다.



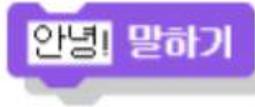
클릭하면 강아지가 “멍멍” 소리를 내며 10만큼 앞으로 이동합니다.

직접 녹음한 소리나 원하는 음악 등의 파일을 가져와 활용할 수 있습니다.



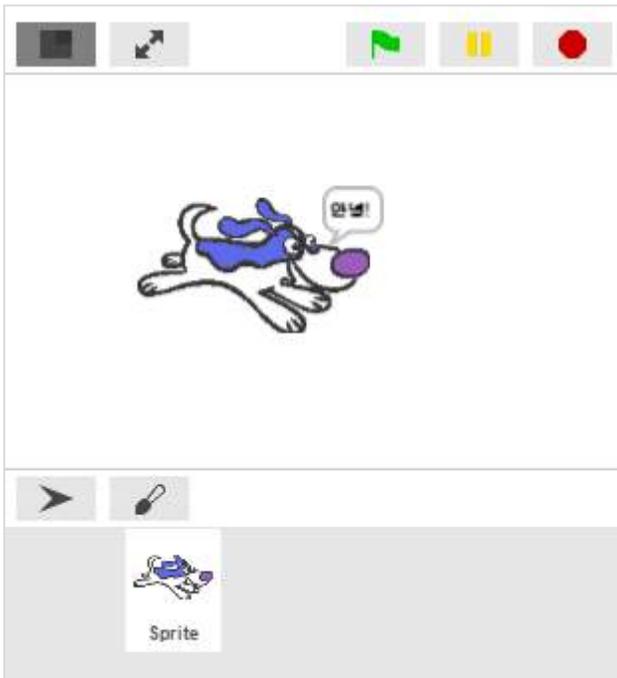
## 2.2 말풍선 블록 사용하기

**말풍선** 블록은 소리 블록과 다르게 눈으로 볼 수 있는 말풍선이 스프라이트에 추가되는 블록이며, 말풍선 모양이 시간과 함께 나타났다가 사라지는 자막 같은 특별한 효과를 줄 수 있습니다.

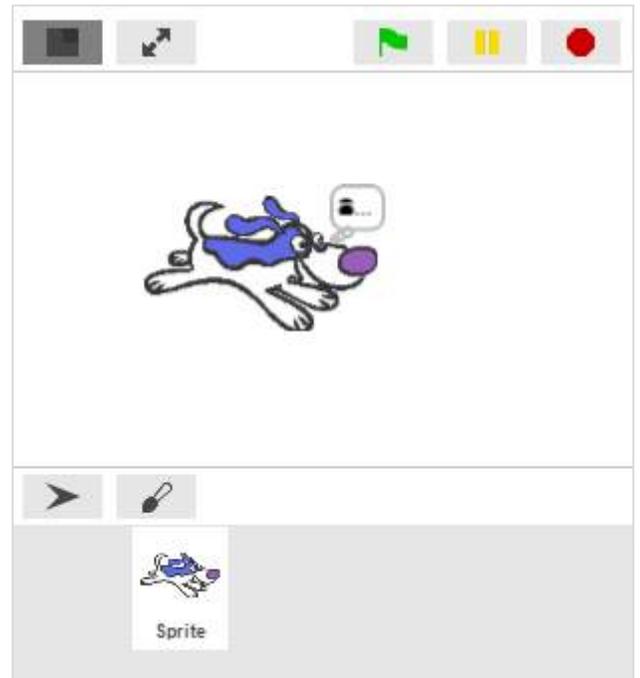


안녕! (극) 2 초 동안 말하기	안녕! 에 입력한 단어가 2초 동안 말풍선에 나타납니다.
안녕! 말하기	안녕! 에 입력한 단어를 말풍선에 나타냅니다.
ㅎㅎㅎ (극) 2 초 동안 생각하기	ㅎㅎㅎ 에 입력한 단어가 2초 동안 생각풍선에 나타납니다.
ㅎㅎㅎ 생각하기	ㅎㅎㅎ 에 입력한 말이 생각풍선에 나타납니다.

아래의 이미지는 말풍선과 생각풍선을 사용한 예입니다.



말풍선



생각풍선

### 3.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트 **DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

**예제 1** : 세 개의 스프라이트를 만들어 줍니다. 각각의 스프라이트는 스테이지 이곳저곳을 자유롭게 다닐 수 있도록 합니다. 이 때 '방송하기' 블록을 사용하여 첫 번째 스프라이트가 움직이기 시작할 때 다른 스프라이트들도 동시에 움직임을 시작할 수 있도록 해 봅시다.

첫 번째 스프라이트



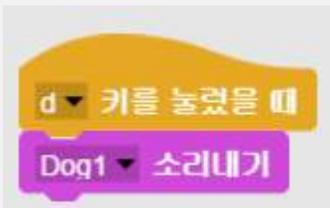
두 번째 스프라이트



세 번째 스프라이트



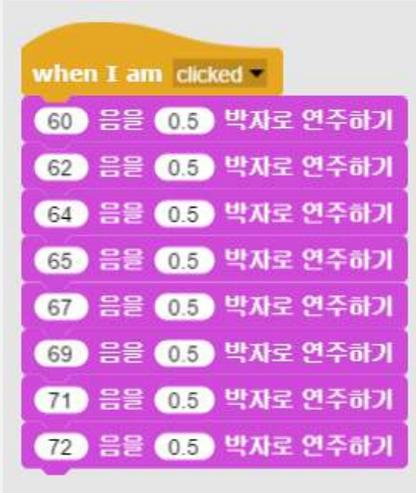
**예제 2** : 프로젝트 메뉴의 '소리' 목차를 클릭하면 옆의 그림처럼 가져올 수 있는 소리 목록을 볼 수 있습니다. 여기서 dog1.wav 파일을 클릭하여 불러온 후에 키보드의 'd'키를 누르면 이 소리 파일이 재생될 수 있도록 합니다.



도움말 : 소리 탭을 클릭하여 현재 불러온 소리 파일 목록에 사용자의 다른 폴더에 있는 \*.wav 파일을 드래그하여 소리 파일을 추가할 수 있습니다.



**예제 3 :** '소리' 카테고리의 음 연주 블록을 사용하여 스프라이트를 클릭했을 때 '도레미파솔 라시도' 연주가 시작되도록 합니다.

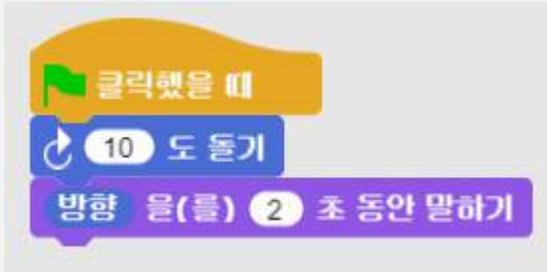


도움말 : 61, 63, 66, 68 음은 반음입니다. 따라서 이 예제에는 사용되지 않았습니다. 음계의 숫자가 커질수록 음이 높아지고 작아질수록 음이 낮아 집니다.

**예제 4 :** 키보드의 방향키 중 왼쪽 화살표를 누르면 스프라이트가 왼쪽으로 이동하면서 'LEFT!'라는 메시지가 담긴 말 풍선이 스프라이트에 나타나고 오른쪽 화살표 키를 누르면 스프라이트가 오른쪽으로 이동하면서 'RIGHT!'라는 메시지가 담긴 말풍선이 스프라이트에 나타나도록 해 보세요.

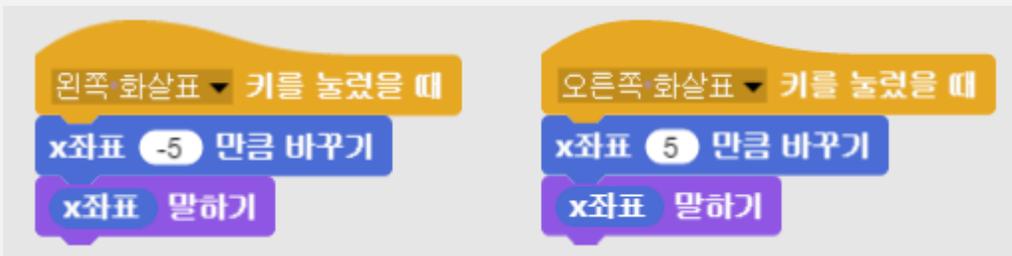


**예제 5 :** 스크립트를 클릭할 때마다 10도씩 시계 방향으로 스프라이트가 회전하도록 합니다. 이 때 말풍선으로 스프라이트가 자신의 회전 각도를 알려주도록 스크립트를 작성해 봅시다.



도움말 : 보라색 말풍선 만들기 블록에 동작 카테고리에 있는 '방향' 리포터 블록을 결합하였습니다. 이렇게 결합시키면 말 풍 선에는 스프라이트의 회전 방향이 각도로 표시됩니다.

**예제 6 :** 키보드의 왼쪽과 오른쪽 방향키로 스프라이트를 왼쪽과 오른쪽으로 이동시킵니다. 이 때 스프라이트에 말풍선으로 스프라이트의 X좌표 값을 항상 나타낼 수 있도록 스크립트를 작성해 봅시다.



도움말 : 동작 카테고리에 있는 'x좌표' 블록을 말하기 블록에 결합하였습니다. 스프라이트를 이동시키면서 말풍선에 나타난 값을 관찰해 보면 5씩 증가하거나 감소하는 것을 볼 수 있습니다.

## 3.2 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

1. 소리 카테고리에 있는 음계 연주 블록을 사용해서 간단한 동요나 음악을 연주할 수 있도록 스크립트를 작성해 봅시다. 이 때 아래 블록을 이용하여 빠르기를 조절해 연주하여 보고, 느리게도 연주하여 봅시다.

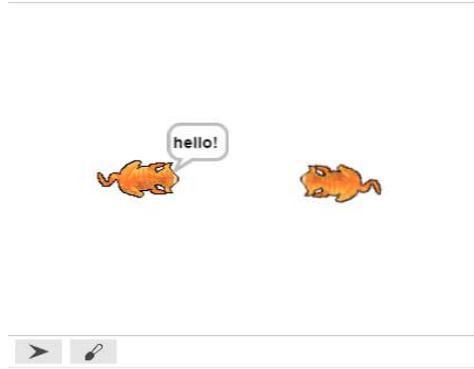


2. 스크립트를 클릭하면 도레미파솔라시도가 계속해서 연주됩니다. 이 때 'q'키를 누르면 연주속도가 느려지고 'w'키를 누르면 원래의 연주속도가 되고 'e'키를 누르면 연주속도가 빨라지게 해 봅시다.
3. 키보드의 방향키로 스프라이트를 위, 아래, 오른쪽, 왼쪽으로 조종하도록 스크립트를 작성합니다. 이 때 말 풍선으로 이동 중인 스프라이트의 좌표를 표시합니다. X좌표(가로) 방향으로 움직일 때에는 X좌표 값이 말 풍선에 표시되고 Y좌표(세로) 방향으로 움직일 때에는 Y좌표 값이 표시되게 합니다.

4. **고양이들의 대화 - 말 풍선 넣기** :옆 그림처럼 스테이지에 스프라이트 2개를 놓습니다.(cat2.gif 사용) 왼쪽과 오른쪽 고양이는 각각 다른 스프라이트 입니다.

왼쪽의 고양이가 말을 걸면 잠시 후에 오른쪽의 고양이가 대답을 합니다. 제어 카테고리에 있는 '방송하기'블록을 이용하여 두 스프라이트가 말을 주고 받을 수 있도록 합니다.

w,a,s,d 키를 누르면 각각 키에 따라 왼쪽 고양이가 다른 말을 하고 이 말에 따라 오른쪽 고양이도 각각 다른 대답을 할 수 있도록 스크립트를 만들어 봅니다.



5. **고양이들의 대화 - 소리 내기** : 고양이들의 대화가 완성되었다면 말 풍선이 생길 때마다 고양이 울음이 나오도록 합니다. 프로젝트 메뉴의 '소리' 목차를 클릭하여 왼쪽 스프라이트에는 Kitten.wav 파일을 로드하여 울음 소리로 사용하고 오른쪽 스프라이트에는 Meow.wav 파일을 로드하여 울음 소리로 사용합니다.

## 1. Rokit Brick 사용하기

### 3장.

변수와 반복

그리고 조건

## 1.1 변수란?

변수란 값이 변하는 수를 말합니다.

말 그대로 지금의 값을 다른 값으로 마음대로 바꿀 수 있다는 것이 변수입니다.

변수를 사용하려면 먼저 이름을 지어야 하며 이런 이름들을 명칭이라고 합니다. 스크래치 자체에서도 변수 블록을 생성할 수 있으며 명칭을 직접 만들어 사용 할 수 있습니다.

### 변수를 왜 사용해야 하나요?

이미지의 동작도 숫자의 변화에 의해 변화되며, 로봇, 드론의 모든 값들도 숫자의 값으로 움직입니다.

그리고 센서의 값들도 감지된 정도에 따라 변화하는 속성을 가지고 있습니다. 이러한 모든 것을 위해서 변수를 사용하여야 합니다.

Sum이라는 변수를 만들고 변수의 값을 바꾸는 예

The image displays the Scratch 'Variables' menu on the left and a code editor on the right. In the menu, 'sum' is selected as the variable name, and '0' is set as the initial value. The code editor shows two blocks: a 'when green flag clicked' block with a 'set sum to 1' block, and a 'when key w is pressed' block with an 'increase sum by 1' block.

## 1.2 변수 블록 만드는 법

### 변수 만들기

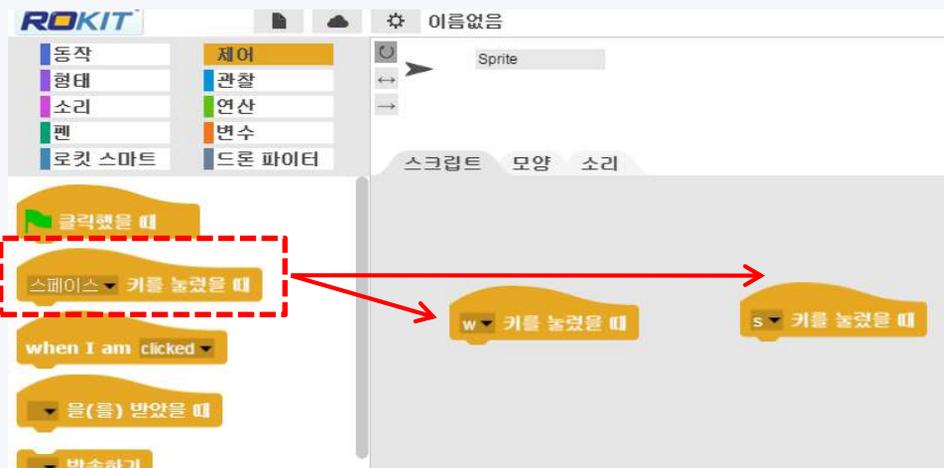
변수 블록 탭을 클릭하여 변수 만들기 버튼을 누르고 원하는 변수 이름을 적어줍니다.  
여기서는 SUM이라는 변수를 만들어 보겠습니다.



다음 따라해보기를 통해 변수의 활용을 잘 익혀보세요.

### 따라해보기

#### ▶ 키보드로 숫자 변수 증감시키기



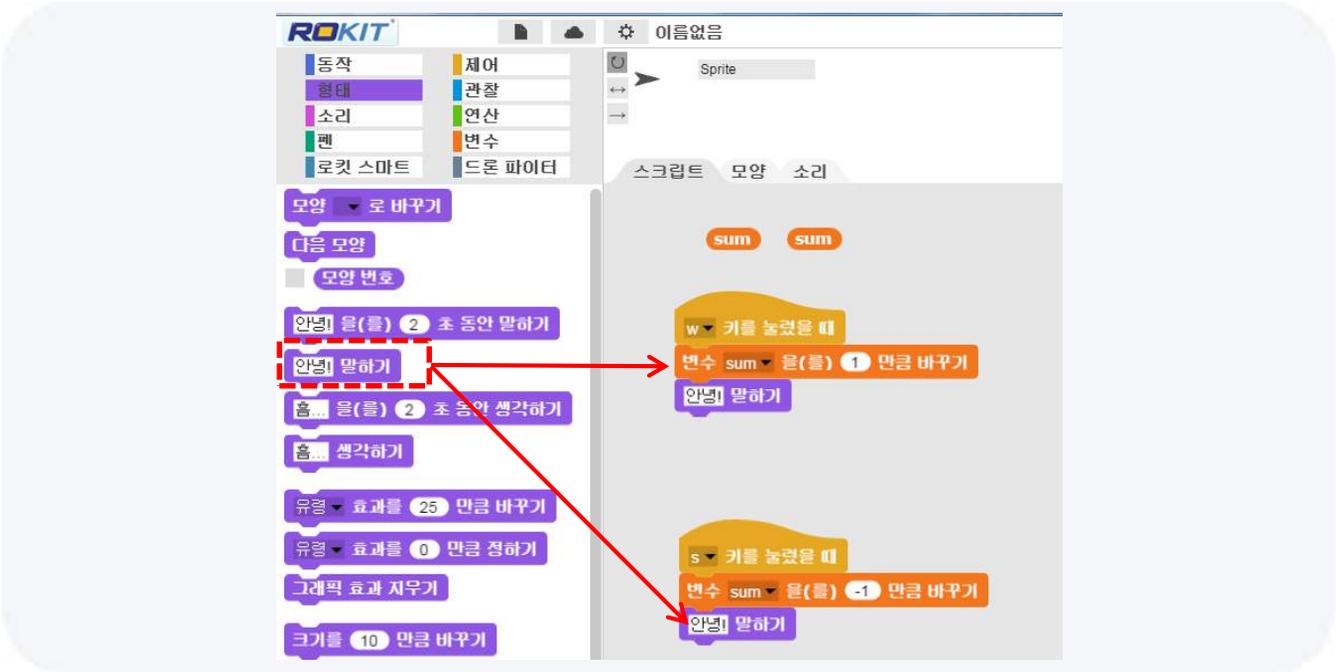
변수 블록으로 들어가 해당 블록을 놓아준 후 1,-1을 각각 넣어주세요.



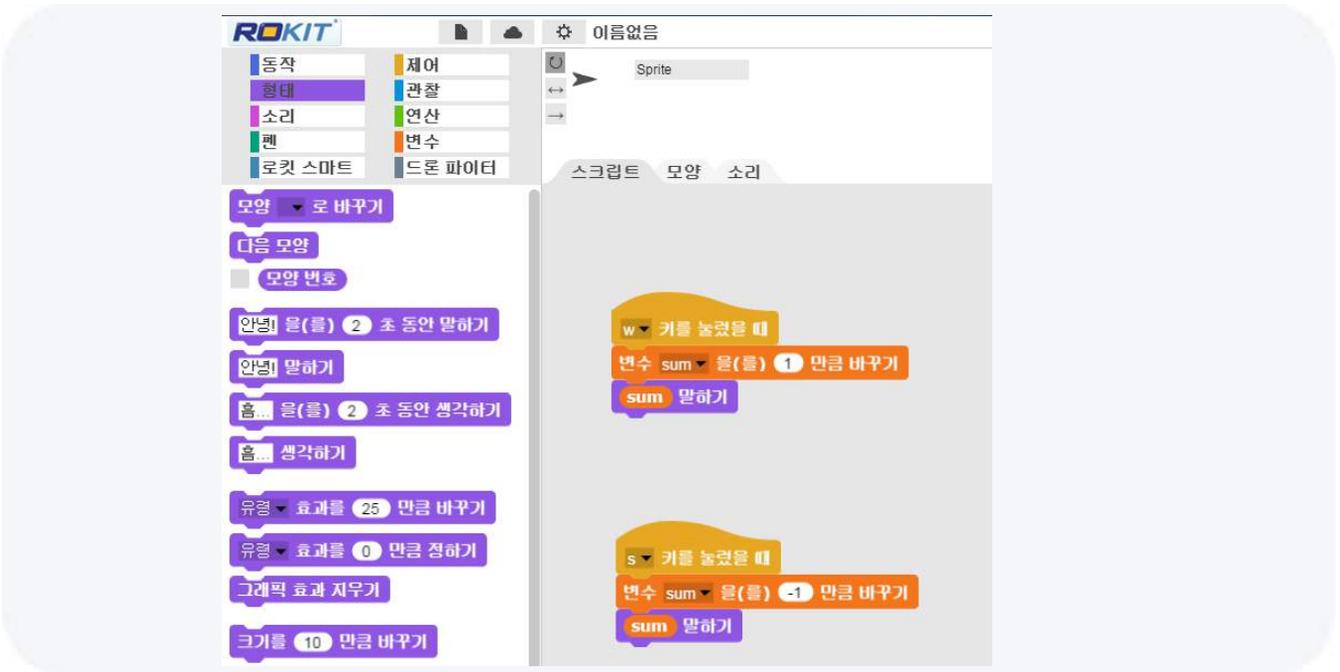
만들어진 SUM이라는 변수를 가져오세요.

도움말 : 값에 대한 정확한 모니터링(monitring)은 스테이지의 변수 값 컨트롤 보다 말풍선으로 확인하는 것이 좋습니다. 값을 줄일 때에는 부호를 붙여야 합니다.

형태 블록에서 아래와 같이 아래 놓아줍니다.



4번에 준비했던 SUM 변수를 “안녕”안에 넣어줍니다.



제어 블록에서 아래와 같이 놓아주세요.



## 실행해보기

이제 키를 눌러서 숫자를 증가 및 감소시켜 보세요.



## 2.1 유한(조건) 반복 블록 사용하기

무한 반복과 유사하지만, 조건이 있기 때문에 조건을 만족하지 못하면 반복문 밖으로 나가버리는 차이가 있습니다.

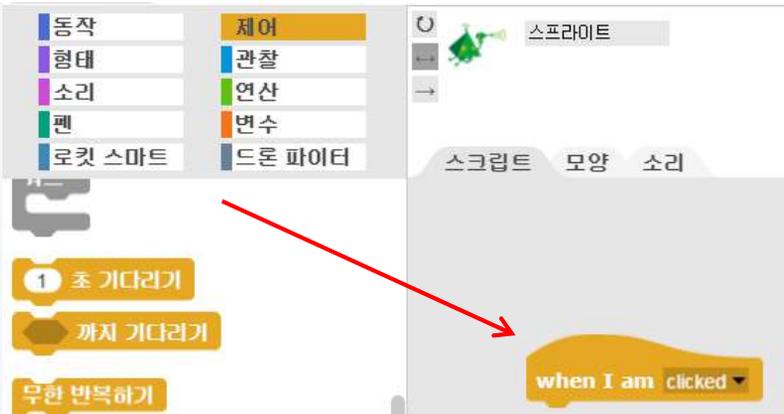
다음 **따라해보기**를 통해 유한 반복 블록의 용도를 알아보시다.

### 따라해보기

#### ▶ 반복 블록을 사용하여 왕복운동하기

스프라이트를 클릭했을 때 유한반복을 사용하여 앞으로 뒤로 왕복 운동하는 스프라이트를 만들어 봅시다.

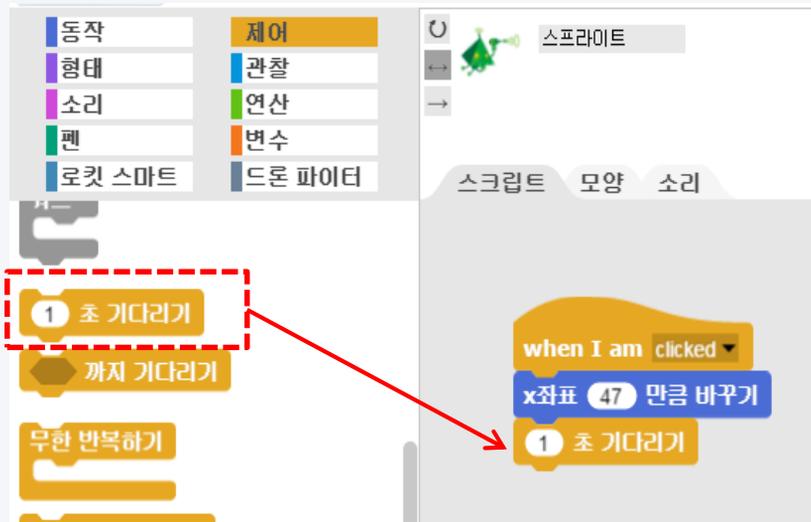
제어 블록에서 아래와 같이 놓아주세요.



동작으로 들어가 동작 블록을 놓아줍니다.



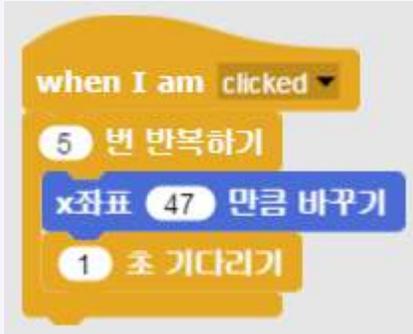
형태 블록에서 아래와 같이 아래 놓아줍니다.



대기 할 수 있는 기다리기 블록을 놓아줍니다.



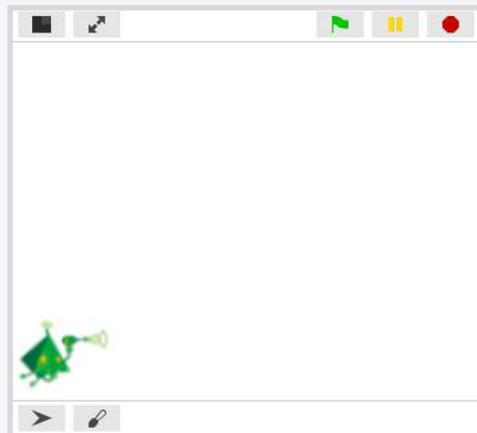
유한 반복 블록에 넣어준 후 90도 회전과 1초 기다리기 블록을 뒤에 놓아줍니다.



## 실행해보기



스프라이트를 클릭하여 실행시켜 보세요.



## 2.2 조건(IF) 블록 사용하기

조건 블록은 프로그램에서 가장 중요한 판단을 하는 기능을 가진 블록입니다.  
추후 지능 로봇이나 판단하는 프로그램 작성을 위해 필요합니다.

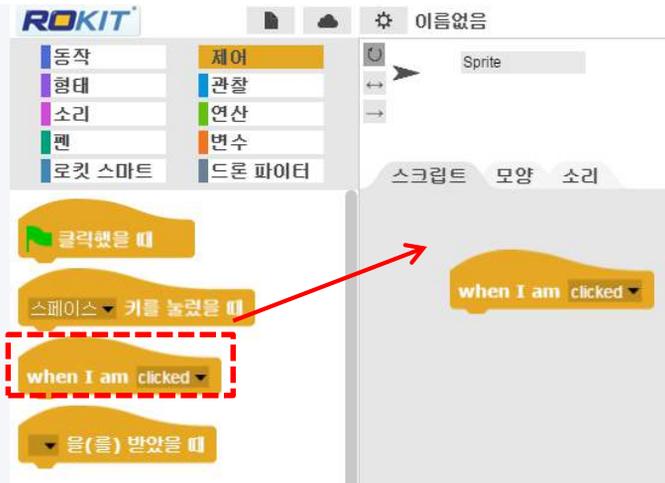
다음 **따라해보기**를 통해 조건 블록을 익혀보십시오.

### 따라해보기

#### ▶ 스프라이트 크기 증가 후 돌아오기

스프라이트를 클릭했을 때마다 스프라이트의 크기가 조금씩 커지도록 스크립트를 작성해 봅시다. 만약 점점 커지다가 크기가 2배가 넘게 되면 원상태로 줄어들어야 합니다.  
(처음 스프라이트를 불러온 경우의 크기는 100% 입니다.)

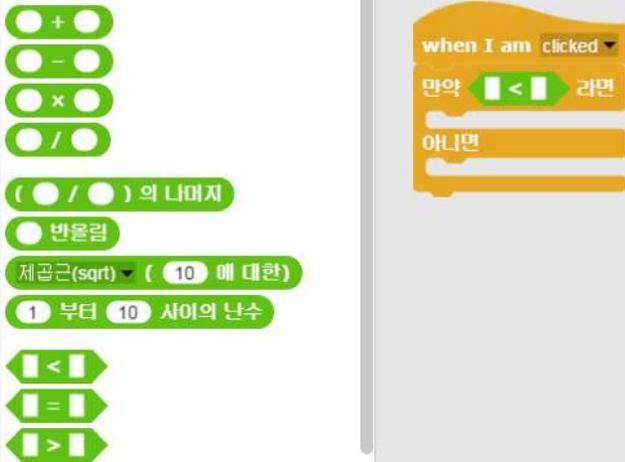
제어 블록에서 아래와 같이 놓아주세요.



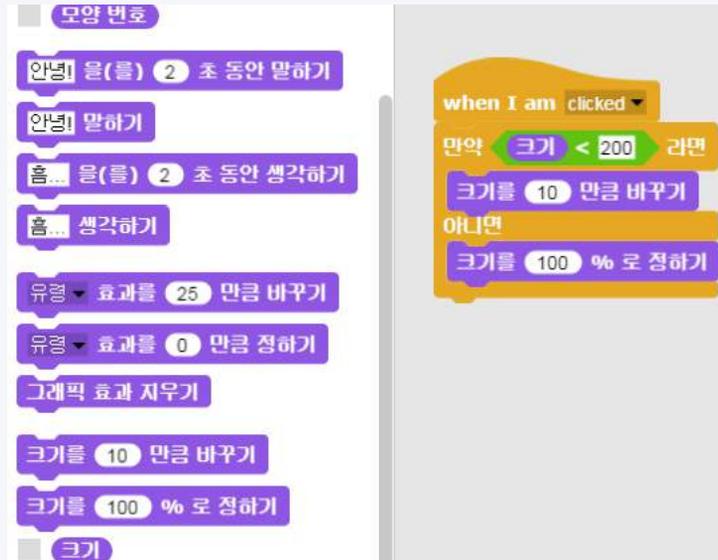
만약, 아니면 이라는 조건 블록을 붙여넣습니다.



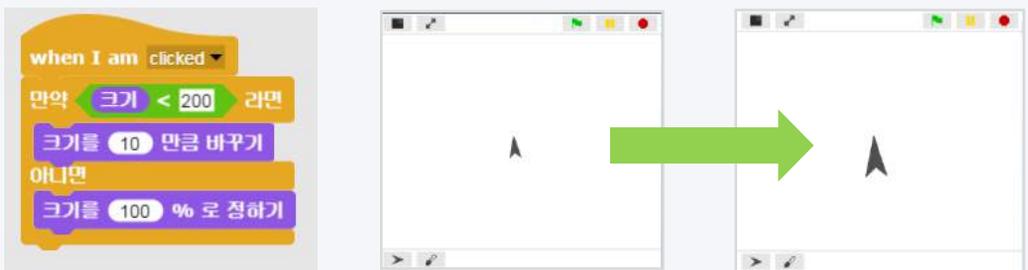
연산으로 들어가 해당 부등호 연산을 넣어 줍니다.



형태로 들어가서 아래의 블록을 넣어주세요.



이제 화살표를 눌러 화살표의 크기를 키워보세요.



도움말 : 크기가 200%(두배) 이상 넘는 것을 막기 위해서 연산 카테고리의 부등호 블록을 사용하였습니다. 현재 스프라이트의 크기를 알려주는 '크기' 리포터 블록은 형태 카테고리에서 발견할 수 있습니다. 스크립트를 보면 스프라이트를 클릭할 때마다 10%씩 커짐을 알 수 있습니다.

## 2.3 연산 블록(부등호와 반올림) 사용하기

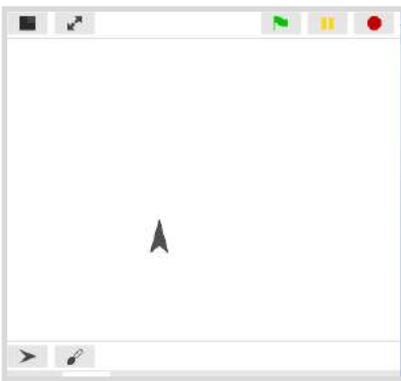
스프라이트를 클릭했을 때마다 스프라이트의 크기가 조금씩 커지는 범위를 부등호에서 바꿔보고 반올림 블록에 대하여 알아봅시다.

### 따라해보기

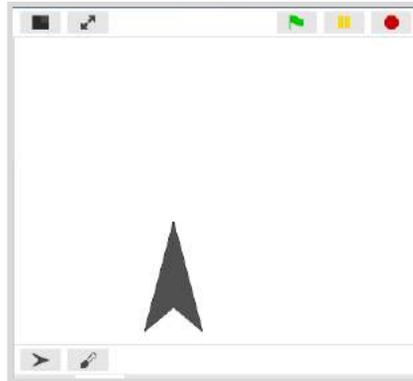
#### ▶ 부등호의 범위 지정 및 반올림 사용



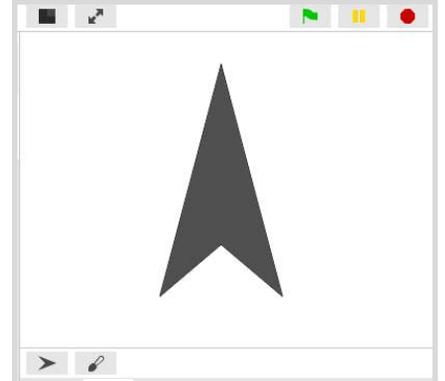
부등호의 범위에 따라 화살표의 크기의 범위가 점점 늘어나는 것을 알 수 있습니다.



크기 < 200



크기 < 400



크기 < 800

반올림은 안에 숫자가 소수점 일 때 반올림해주는 기능의 블록을 말합니다.

4.78 반올림 5

### 3.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 [교재 16페이지](#)▶

**예제 1** : 변수 카테고리에서 sum이라는 변수를 만듭니다. 키보드의 'w' 키를 누르면 sum이 1씩 증가하고 's'키를 누르면 1씩 감소하게 해 봅시다. 이 때 변수 값은 스프라이트의 말풍선으로 표시해 주세요.



도움말 : 값에 대한 정확한 모니터링(monitring)은 스테이지의 변수 값 컨트롤 보다 말풍선으로 확인하는 것이 좋습니다. 값을 줄일 때에는 -부호를 붙여야 합니다.

**예제 2** : 프로젝트 메뉴의 '모양'을 클릭해서 twinArrow.png 를 스프라이트로 사용합니다. 이 스프라이트를 왼쪽 화살표 키를 눌렀을 때에는 왼쪽으로 서서히 90도만큼 회전시키고 오른쪽 화살표 키를 눌렀을 때에는 오른쪽으로 서서히 90도 회전할 수 있도록 합니다.



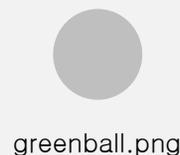
도움말 : 예제에서는 5도씩 회전하므로 18번 반복해서 90도 만큼 회전하였습니다. 반복 횟수와 회전 각도를 조절하면 좀 더 빠르게 돌거나 느리게 돌게 할 수 있을 것입니다.

**예제 3 :** 스프라이트를 클릭했을 때마다 스프라이트의 크기가 조금씩 커지도록 스크립트를 작성해 봅시다. 만약 점점 커지다가 크기가 2배가 넘게 되면 원상태로 줄어들어야 합니다. (처음 스프라이트를 불러온 경우의 크기는 100% 입니다.)



도움말 : 크기가 200%(두 배) 이상 넘는 것을 막기 위해서 연산 카테고리 부등호 블록을 사용하였습니다. 현재 스프라이트의 크기를 알려주는 '크기' 리포터 블록은 형태 카테고리에서 발견할 수 있습니다. 스크립트를 보면 스프라이트를 클릭할 때마다 10%씩 커짐을 알 수 있습니다.

**예제 4 :** 공 모양의 스프라이트가 스테이지 이곳저곳을 빠르게 왔다 갔다 합니다. 이 때 마우스 포인터로 빠르게 움직이는 공을 잡을 수 있도록 합니다. 마우스 포인터와 공 모양의 스프라이트가 만나면 잠시 멈추게 함으로 공을 잡은 효과를 낼 수 있도록 합니다.



도움말 : 조건블록에 관찰 카테고리의 '~ 에 닿았는가' 블록을 결합하였습니다. 마우스 포인터와 스프라이트가 만났다면 1초 동안 공의 움직임을 멈추게 해 줍니다.

**예제 5** : 마우스로 스테이지 이곳저곳을 클릭하면 X좌표와 Y좌표 값이 스테이지 변수 컨트롤에 나타내게 하여 봅시다.



도움말 : 스테이지 좌표 값을 얻기 위해서는 먼저 X와 Y좌표 값을 저장할 변수를 만들어 줍니다. 이후 다음과 같은 단계가 필요합니다.

- 1) 이 변수 값들을 0으로 초기화 합니다.
- 2) 조건 문에 '마우스를 클릭했는가?'블록을 결합하여 마우스를 클릭 했는지를 검사 하고
- 3) 클릭이 되었으면 마우스의 좌표 값을 변수에 넣어줍니다.  
이 때 마우스의 좌표 값은 소수점 자리까지 계산되어 나오므로 연산 카테고리의 반 올림 블록을 이용하여 정수로 나타내 주었습니다.

## 3.2 연습문제

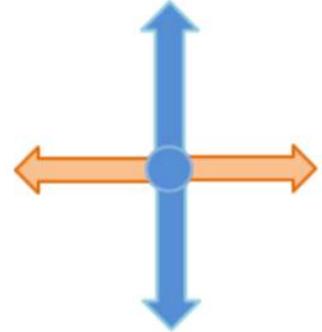
예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

- 아래 리포터 블록들은 어떤 특정한 값들을 변수에 저장해 주는 역할을 합니다. 블록들마다 어떤 값을 저장하고 있는 지 조사하여 보고 이 값들을 정수형으로 표현하고자 할 때 반올림 블록에 결합할 필요가 없는 블록은 어떤 것들이 있는지 확인해 보세요.

리포터 블록 종류	값의 의미	반올림 블록 필요 유무
<ul style="list-style-type: none"> <li>마우스의 x좌표</li> <li>마우스의 y좌표</li> <li>방향</li> <li>x좌표</li> <li>1 부터 10 사이의 난수</li> <li>빠르기</li> <li>( 2 / 10 ) 의 나머지</li> <li>y좌표</li> </ul>		

- 스프라이트가 마우스 포인터를 따라 이동하도록 스크립트를 작성합니다. 이 때 마우스를 클릭하면 스프라이트가 50%정도 부풀어 오르고 그렇지 않은 경우에는 원 상태로 돌아오도록 합니다. 마우스 포인터가 스테이지의 범위를 넘어가면 스프라이트는 스테이지의 벽 근처에 머물러 있고 더 이상 따라가거나 사라져서는 안됩니다.

3. 각각 두 개의 스프라이트를 만든 후 twinArrow.png와 twinArrow2.png파일을 사용하여 옆 그림과 같이 스프라이트를 배치해 줍니다. 파란색 화살표는 첫 번째 스프라이트이고 주황색 화살표는 두 번째 스프라이트입니다.



스페이스 키를 한번 누를 때마다 파란색 화살표가 360도 회전하고 나면 주황색 화살표가 6도씩 회전하여 시계의 분침과 초침 역할을 하게 스크립트를 작성합니다.

4. 스테이지 안에서 스프라이트가 제자리에서 회전을 합니다. 회전 방향은 동일합니다. 키보드의 왼쪽 방향 키를 누르면 회전 속도가 느려지고 오른쪽 방향 키를 누르면 회전 속도가 빨라지도록 해 봅시다. 헛 블록 대신 조건 블록을 사용하도록 하며 회전 각도의 크기로 속도를 조절할 경우 각도 값이 0보다 작아지지 않도록 유의합니다.

5. 키보드의 방향 키를 사용하여 스프라이트를 스테이지 내에서 위아래와 왼쪽 오른쪽으로 자유롭게 조종할 수 있도록 합니다. 헛 블록은 사용할 수 없으며 조건문을 사용하여 스크립트를 작성해야 합니다. 스크립트를 실행하고 난 후에 헛 블록을 사용했을 때와 조건문을 사용했을 때 어떤 차이가 있는 지 이야기해 봅시다.

## 1. Rokit Brick 사용하기

### 4장.

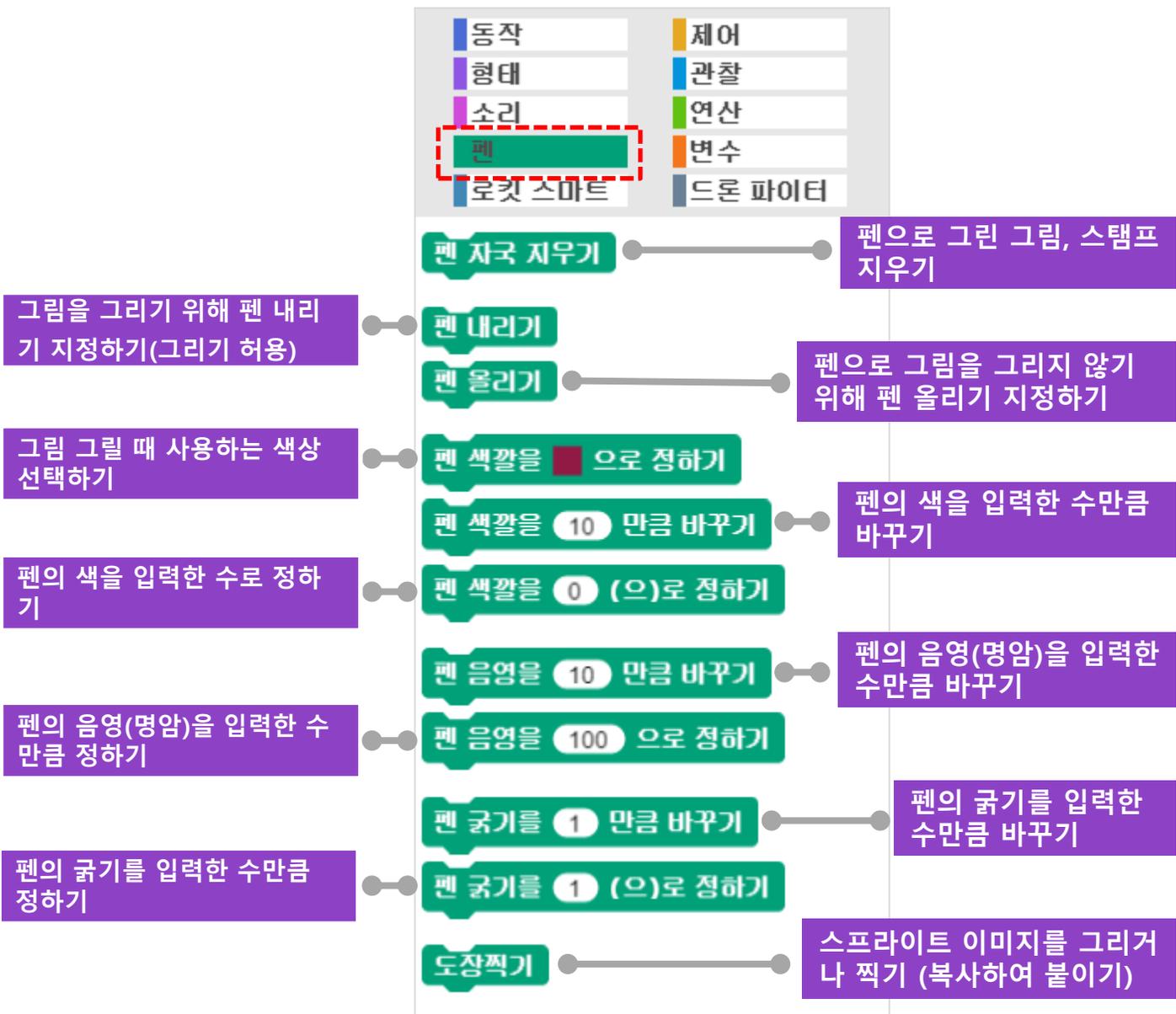
### 그림그리기와

### 스프라이트 조종하기

# 1.1 펜 블록 사용하기

펜 블록은 선이나 도형 등의 그래픽을 그릴 수 있는 11개의 코드 블록으로 구성되어 있습니다.

Rokit Brick 프로그램에서 그림을 그릴 때는 그림판을 이용하거나 펜 블록을 이용하는 두 가지 방법이 있습니다. 그림판에서 그린 그림은 스프라이트나 배경 그림을 그릴 때 사용하지만, 펜 블록을 이용한 그림은 무대에서 스프라이트가 그림을 그리게 하는 방법입니다.



## 1.2 펜의 색과 두께 정하기

### 펜의 색상 정하기

펜 색깔을  으로 정하기	펜 색깔을 선택합니다.
펜 색깔을  만큼 바꾸기	펜 색깔을 지정한 수만큼 바꿉니다.
펜 색깔을  (으)로 정하기	펜 색깔을 지정한 수의 값으로 정합니다.

펜 색깔 정하기 블록에서  부분을 클릭하면 무지개 색상의 팔레트가 나타납니다. 마우스 포인터로 색상 팔레트를 클릭하여 색을 선택할 수 있습니다.



스크립트를 만들어 무대 위에서 스프라이트가 선을 그리며 이동하는지 확인해 봅시다.



## 펜의 두께 정하기

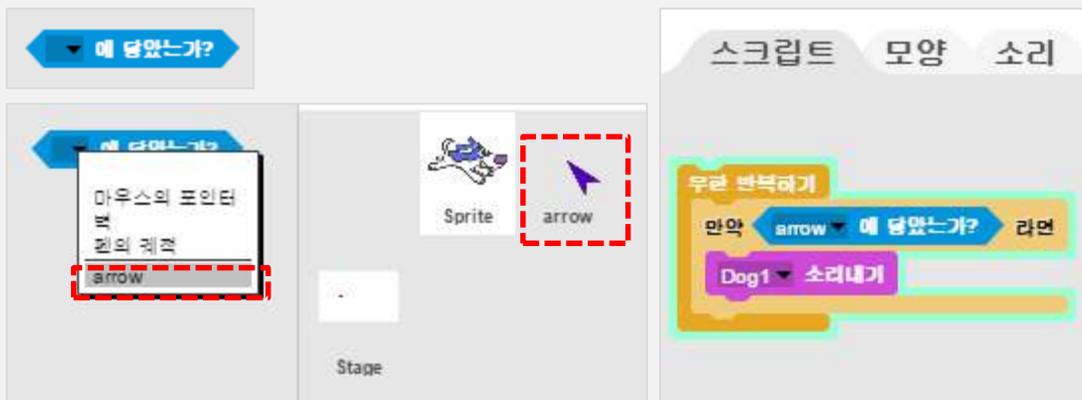
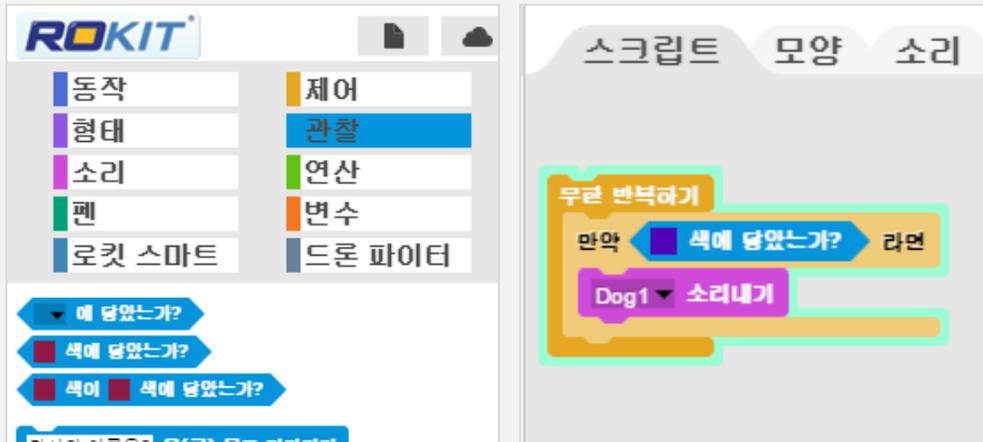
<b>펜 굵기를 1 만큼 바꾸기</b>	<b>1</b> 안에 지정한 수만큼 펜의 굵기를 바꿉니다.
<b>펜 굵기를 1 (으)로 정하기</b>	<b>1</b> 안에 지정한 수로 펜의 굵기를 정합니다.

아래 이미지는 펜의 굵기를 5, 10 그리고 30으로 지정하였을 때 그려지는 펜의 두께를 보여줍니다.

<p>스크립트 모양 소리</p> <p>클릭했을 때</p> <p>펜 내리기</p> <p>펜 굵기를 5 (으)로 정하기</p> <p>펜 굵기를 5 만큼 바꾸기</p> <p>펜 색깔을 ■ 으로 정하기</p> <p>10 만큼 움직이기</p>	
<p>스크립트 모양 소리</p> <p>클릭했을 때</p> <p>펜 내리기</p> <p>펜 굵기를 10 (으)로 정하기</p> <p>펜 굵기를 10 만큼 바꾸기</p> <p>펜 색깔을 ■ 으로 정하기</p> <p>10 만큼 움직이기</p>	
<p>스크립트 모양 소리</p> <p>클릭했을 때</p> <p>펜 내리기</p> <p>펜 굵기를 30 (으)로 정하기</p> <p>펜 굵기를 30 만큼 바꾸기</p> <p>펜 색깔을 ■ 으로 정하기</p> <p>10 만큼 움직이기</p>	

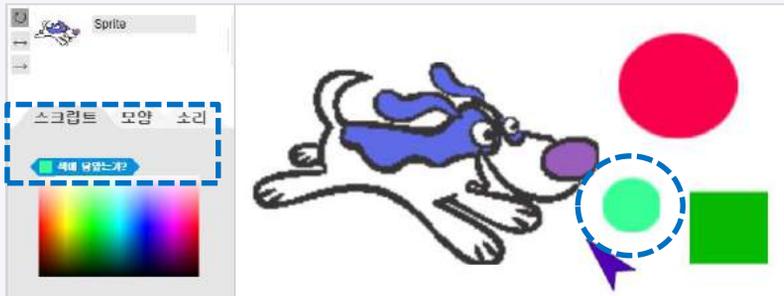
## 2.1 관찰 블록 사용하여 스프라이트 충돌 판단하기

실제 애니메이션에서 사물의 충돌이나, 여러가지 시점으로 상황을 확인할 때 유용한 블록이 바로 관찰 블록입니다.

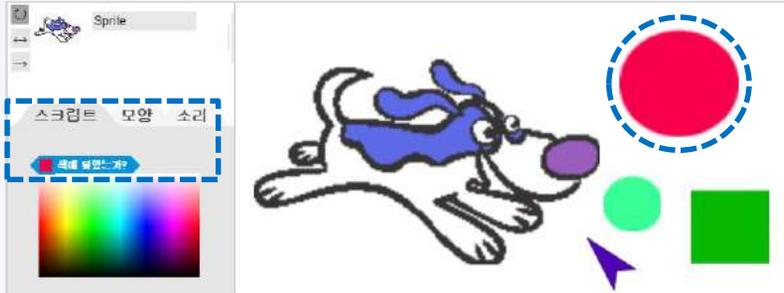


## 2.2 관찰 블록 사용하여 색 감지 하기

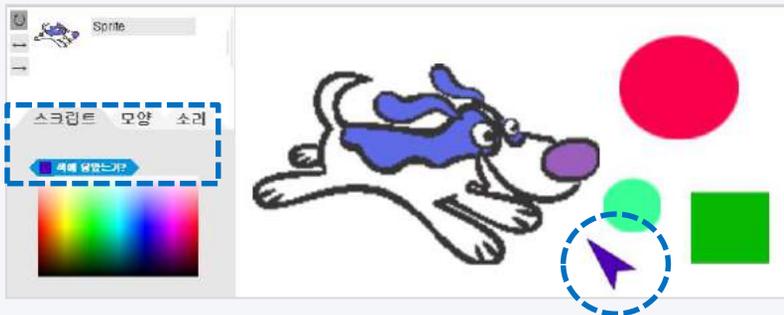
자신이 정의한 색깔이 스크립트에서 작용하도록 색을 감지할 수 있습니다  
 마우스 포인터를 각 색깔 공에 가져다 놓으면 스크립트에서 해당 색을 감지하게 됩니다.



라임 색 공을 클릭하였을 때



빨간 색 공을 클릭하였을 때



화살표를 클릭하였을 때

### 3.1 스테이지 그림 가져오기

자신이 원하는 배경을 무대에 두고 실제 환경처럼 활용할 수 있습니다.

1. Stage를 클릭합니다.

2. 파일을 클릭합니다.

3. Backgrounds를 클릭합니다.

4. Bedroom1을 클릭합니다.

무대에 배경 이미지가 나타납니다.

## 4.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

**예제 1** : 펜 카테고리 에 있는 블록을 사용하여 스프라이트의 펜을 내려서 네모를 그려봅시다. 선의 색이 스테이지의 바탕색(하얀색)과 같으면 잘 보이지 않을 수 있으니 바탕과 잘 구분될 수 있는 색을 사용합시다.

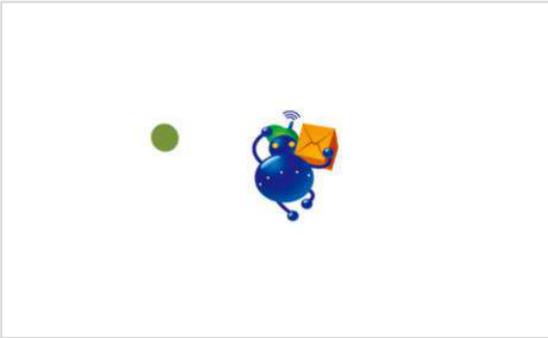


도움말 : 왼쪽과 오른쪽 스크립트 모두 네모를 그릴 수 있지만 오른쪽처럼 반복문을 사용하여 작성하는 것이 좋습니다. 먼저 펜을 내려야만 선이 그려질 수 있음을 잊지 마세요!

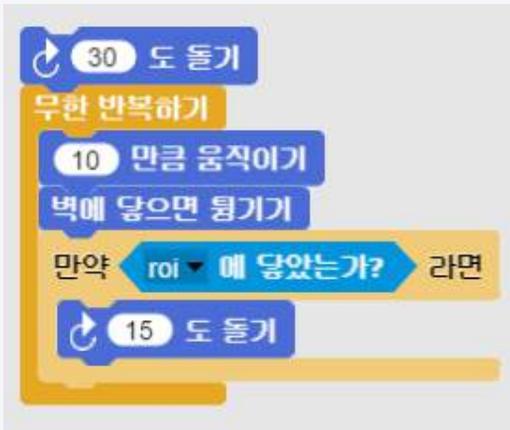
**예제 2** : 마우스 포인터가 스테이지에서 움직이면 마우스 포인터에 따라 선이 그려질 수 있도록 스크립트를 작성해 봅시다.



**예제 3 :** 스프라이트 만들기 버튼을 눌러 스프라이트를 한 개 더 추가합니다. 아래 그림처럼 스프라이트를 구성합니다. 첫 번째 스프라이트는 공 모양이고 두 번째는 캐릭터 형태입니다. 두 번째 스프라이트는 이름을 첫 번째와 구분하기 위해서 이름을 바꾸어 줍니다. 공 모양의 스프라이트는 스테이지를 자유롭게 움직이지만 스테이지의 벽 너머로 나가서는 안 되며 두 번째 스프라이트에 부딪칠 때에는 반사되어 다른 곳으로 튀어나가야 합니다. 이 동작을 계속할 수 있도록 스크립트를 작성해 봅니다.



첫 번째 스프라이트



도움말 : 첫 번째 스프라이트만 스크립트를 넣습니다. 두 번째 스프라이트는 그냥 멈추어 있기만 하면 되므로 따로 스프라이트를 작성하지 않아도 됩니다. 관찰 카테고리에서 스프라이트가 서로 충돌하는 지 여부를 알려주는 '~에 닿았는가'블록을 사용하여 스프라이트 끼리 충돌이 일어났는지 검사합니다. 이 때 블록에 있는 팝업 목록에서 스프라이트 이름을 선택해서 넣어 주어야 합니다.



두 번째 스프라이트 이름 바꾸기  
(스프라이트 → roi)

**예제 4 :** 무대 버튼을 클릭하고 아래와 같은 배경 그림(room2.gif)을 가져옵니다. 스프라이트는 마우스 포인터처럼 마우스의 움직임에 따라 스테이지 이곳 저곳을 이동합니다. 이때 스프라이트가 배경에 있는 특정 물체에 머무르면 말풍선으로 아래와 같은 메시지들이 나타나도록 스크립트를 작성해 봅시다.



도움말 : 실제 그림에 있는 물체를 식별하는 것이 아니라 그림에 있는 물체들의 색이 무엇인지 판단해서 메시지를 띄웁니다. 따라서 다른 물체들이라 할지라도 색이 같다면 같은 메시지가 나타날 수 있습니다.

**무한 반복하기**

x: 마우스의 x좌표 , y: 마우스의 y좌표 쪽으로 이동하기

만약  색에 닿았는가? 라면  
 this-is-bed 말하기

만약  색에 닿았는가? 라면  
 this-is-window 말하기

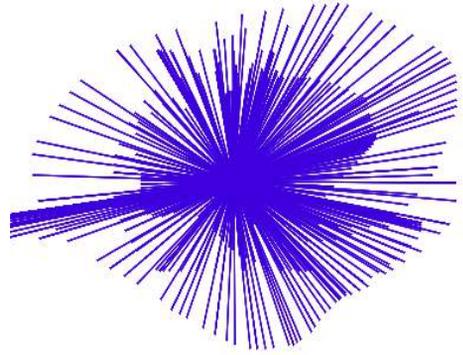
만약  색에 닿았는가? 라면  
 this-is-furniture 말하기

만약  색에 닿았는가? 라면  
 this-is-floor 말하기

## 4.2 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 교재 16페이지▶

1. 옆의 그림처럼 마우스의 움직임에 따라 선이 그려지는 스크립트를 작성하여 봅시다. 마우스가 여기저기 움직이면 중심으로부터 마우스 좌표까지 계속해서 선이 그려지도록 합니다.



2. 테니스 게임 - 스프라이트 만들기 버튼을 눌러서 3개의 스프라이트를 만든 후 첫 번째 스프라이트부터 각각 아래 그림 파일들을 불러옵니다. 검은 선 모양의 스프라이트는 스테이지 화면 맨 아래에 위치하고 공을 막아내는 블로커(blocker) 스프라이트는 마우스의 움직임을 따라 스테이지 하단에서 좌우로 움직입니다. 또한 검은 선 모양의 스프라이트는 스테이지 하단에 위치하며 공이 이 선을 만나면 게임이 끝나게 됩니다. 공은 벽이나 블로커를 만나면 튕기게 합니다.



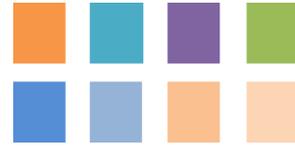
redball.png



blocker.png

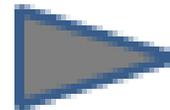
blackbar.png

3. 색깔 건반 – 스프라이트 만들기 버튼을 클릭하여 2개의 스프라이트를 만든 후 둘 중 하나의 스프라이트에는 다양한 색의 사각형들이 그려져 있는 옆의 파일을 불러옵니다. 이 후 마우스 포인터를 이 사각형들에 갖다 댄 후 마우스를 클릭하면 각 색깔마다 다른 음을 연주할 수 있도록 합니다.



colors.png

4. 한 붓 그리기 – 키보드의 화살표 키를 사용하여 스프라이트를 이동 시키면 스프라이트가 움직인 경로를 따라 선이 그려지도록 스크립트를 작성해 봅시다. 이 때 스프라이트는 아래 그림처럼 삼각형 모양의 화살표를 사용하도록 하고 이 화살표의 방향은 화살표 키의 방향과 일치할 수 있도록 합니다.



angle.png

## 2. 디바이스 제어하기

### 1장.

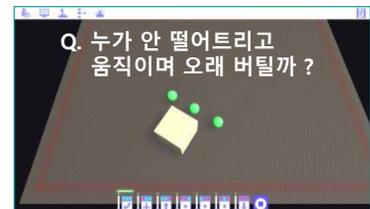
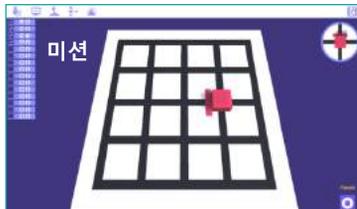
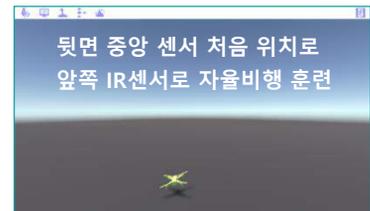
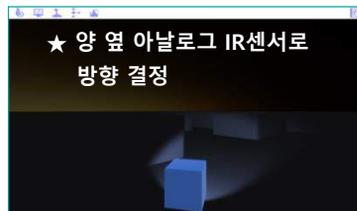
### 전방 센서값

### 점검하기

# Rokit Simulator

## 1.1 Rokit Simulator 란?

인벤터 보드와 PC간의 다양한 콘텐츠를 통해 로봇, 드론, 게임 등을 유저가 원하는 대로 제어 하고 소프트웨어를 재미있게 이해할 수 있는 프로그램



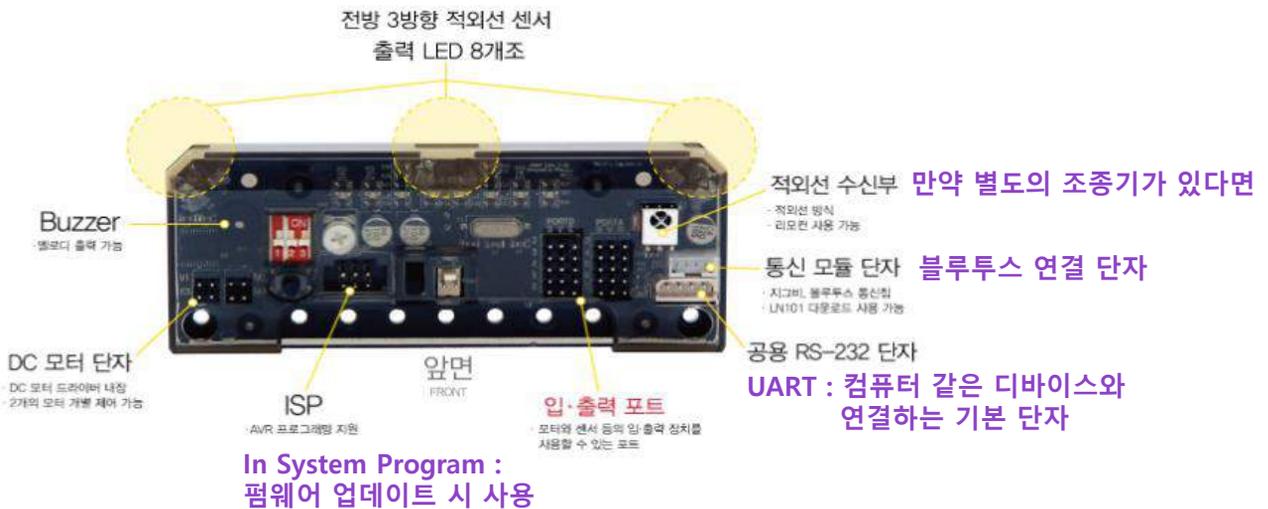
# Rokit Simulator

## 1.2 인벤터 보드 (Inventor Board)

인벤터 보드는 IR 센서, Buzzer(버저), LED, 모터와 통신 제어 모듈 등이 모두 내장되어 있어 다양한 응용이 가능한 하드웨어(HW)입니다.

스크래치, 아두이노를 비롯하여 자체 제작한 보드 전용 프로그램과 접목하여 다양한 기능을 수행해내는 전자 장치입니다.

아날로그 방식의 IR 센서 3개 (0~1024) : 센서인식 대략 10cm

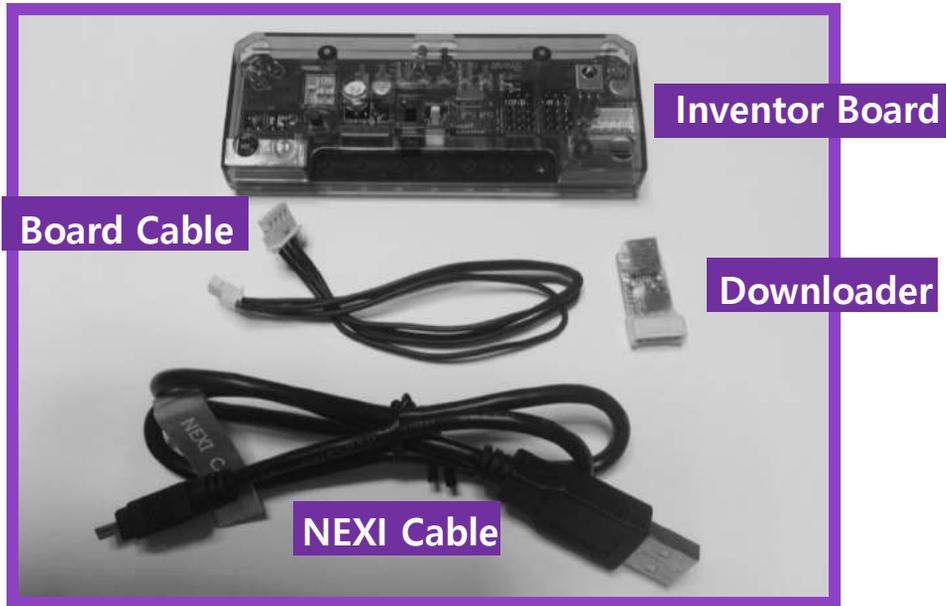


디지털 방식의 IR 센서 7개 : 값은 오직 1과 0뿐

# Rokit Simulator

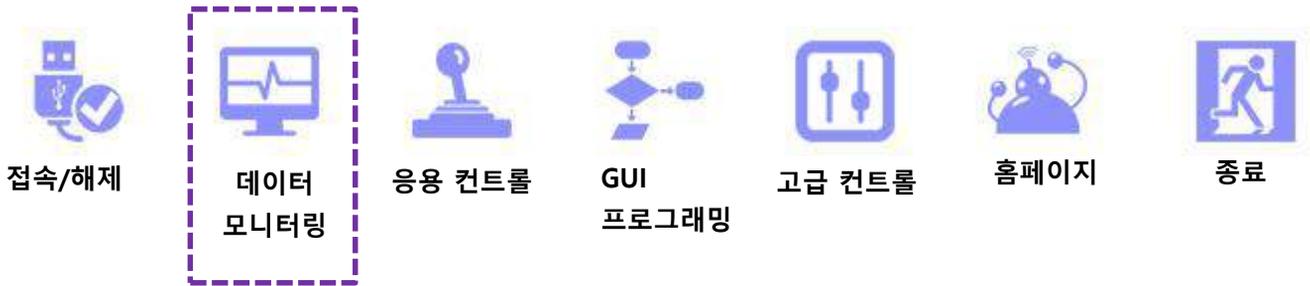
## 1.3 소프트웨어(SW)와 하드웨어(HW)연결

PC와 인벤터 보드는 아래 이미지처럼 통신 케이블로 연결되어야 합니다. 하드웨어를 그림처럼 연결시킨 후, 드라이버를 설치하여 소프트웨어의 통신 상태를 확인합니다.

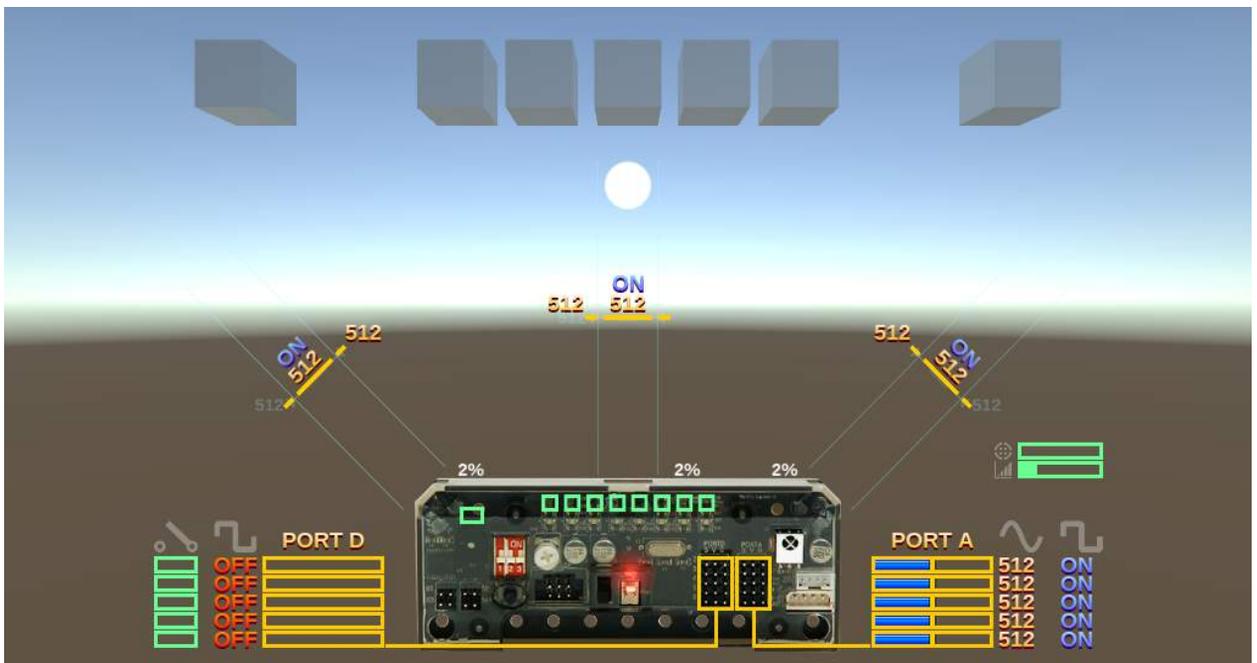


# Rokit Simulator

## 1.4 데이터 모니터링



인벤터 보드에는 수많은 센서와 LED 모듈들이 내장되어 있습니다.  
이 값을 통해 로봇 제어 보드의 원리와 용도를 정확히 이해하고 활용할 수 있습니다.



# Rokit Simulator

## 1.5 데이터 모니터링



### 디지털 IR 센서 수치 표시

#### 디지털이란?

0 또는 1과 같은 연속되지 않는 두개의 숫자로 정보를 생성하는 것을 뜻합니다.



보드 뒷면의 7개 센서는 디지털 IR 센서 입니다.

감지 시에 앞면의 8개의 LED로 확인할 수 있으며 뒷면 가운데 센서가 앞면의 3,4번 LED 두 개와 연결되어 있는 것이 특징입니다.



실습!!

# Rokit Simulator

## 1.6 데이터 모니터링 (아날로그 IR 센서 입력)

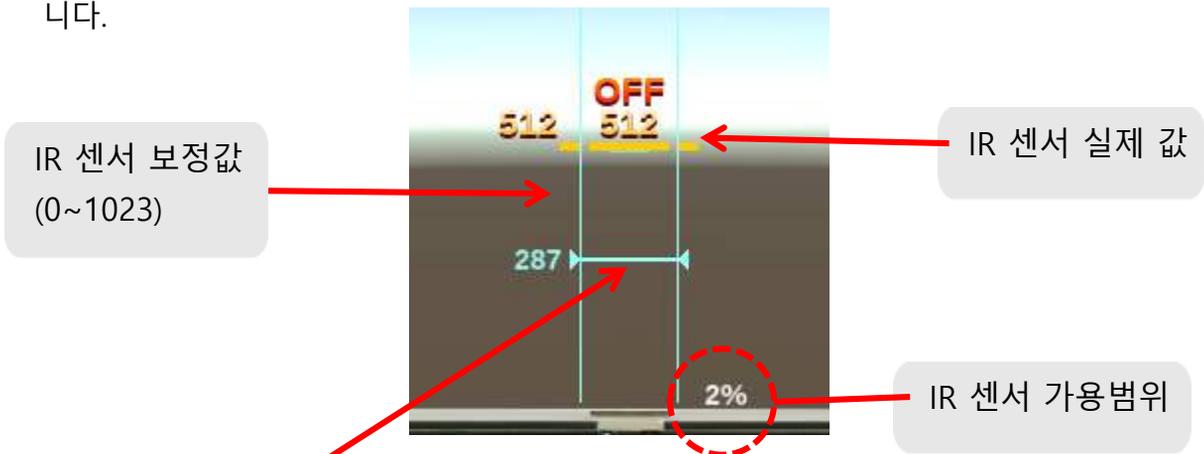


### 아날로그 IR 센서 수치 표시

#### 아날로그란?

끊김 없이 계속적으로 이어지는 숫자시스템으로, 보통 전자에서는 **0~1023**의 값을 가지고 있으며 이러한 특징은 디지털이 0과 1로만 이어져 있는 것과 차이가 있습니다.

IR 센서 이외에도 가변저항이나 초음파 센서 등 입력을 아날로그로 받는 것들이 모두 해당됩니다.



디지털로 사용 시 1: On/ 0 : Off 경계값 미리보기 조절 바  
마우스로 드래그하여 원하는 경계값을 확인할 수 있습니다



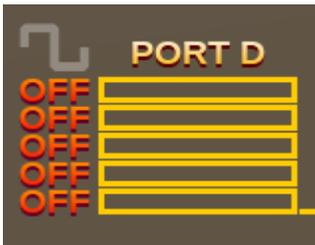
# Rokit Simulator

## 1.7 데이터 모니터링 (LED, 출력)



### 출력 기능

인벤터 보드에 내장된 LED와 추가로 외부에 연결된 출력모듈을 제어할 수 있습니다.



외부 LED 및 출력 모듈을 연결하고 ON/OFF 제어



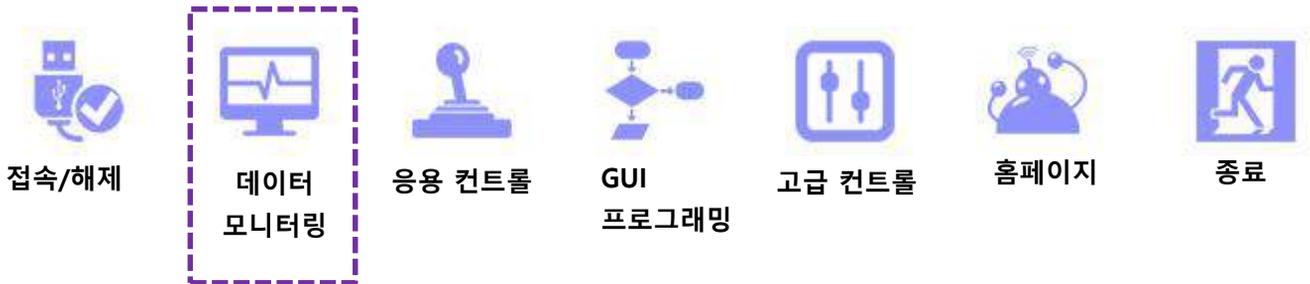
7개의 LED ON/OFF 제어

소리나 LED, 7-Segment와 같은 출력 모듈을 제어 가능합니다.



# Rokit Simulator

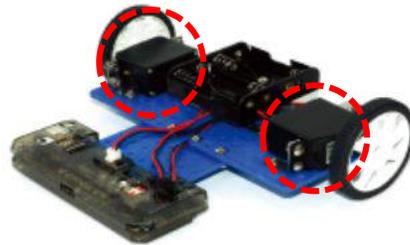
## 1.8 데이터 모니터링 (DC모터)



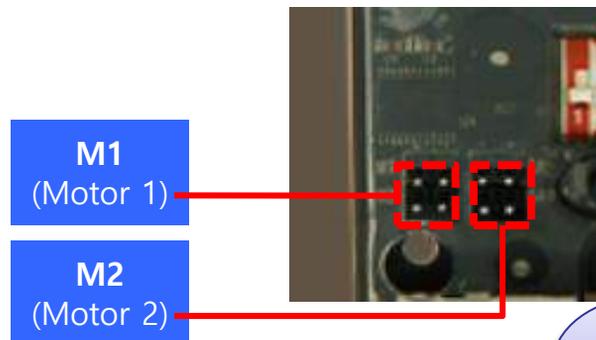
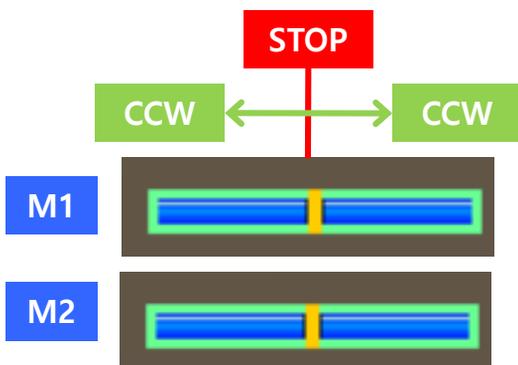
### 모터 제어 기능

#### DC 모터란?

바퀴와 같이 회전하는 모터로, 속도값과 방향값에 변화를 주면 다양한 구동이 가능합니다.



슬라이더 바의 눈금에 따라 속도와 방향이 결정됩니다.

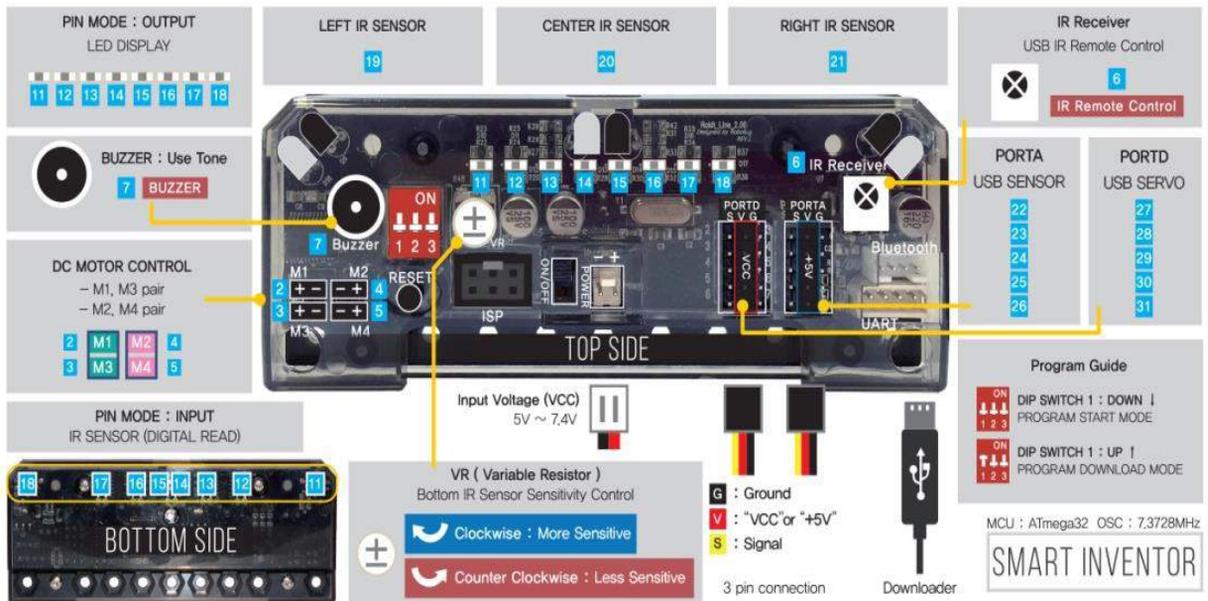


실습 68

## 2.1 핀맵(pin Map)이란?

제어기에는 여러가지 장치들이 있습니다. 이 장치들은 모두 마이크로프로세서에 있는 핀들 중 하나와 연결되어 있어서 마이크로프로세서가 이 핀들을 통해 장치들을 제어할 수 있게 해 줍니다.

따라서 제어기에 연결된 어떤 장치들을 제어하고 싶다면 먼저 어떤 장치가 몇 번 핀에 연결되어 있는지 확인할 수 있어야 합니다. 이러한 정보를 담은 것을 **핀맵**이라고 합니다. 스마트 인벤터 보드의 핀맵은 아래와 같습니다.



### 핀맵이란?

'지도'처럼 제어기에 담긴 핀에 대한 정보를 한 눈에 알아볼 수 있도록, 그림이나 문서로 작성한 것들을 의미합니다. 핀맵은 특히 소프트웨어를 코딩할 때에 큰 도움이 됩니다.

## 2.2 핀맵(pin Map) 사용하기

예를 들어 아래의 그림처럼 11번부터 18번은 각각 LED에 연결되어 있습니다. 만약 보드 맨 왼쪽의 LED를 켜거나 끄고 싶다면 아래와 같이 11번 핀에 명령을 내리면 됩니다.

디지털 핀 11 ▾ 번을 참 으로 설정

: 11번 LED 켜기

디지털 핀 11 ▾ 번을 거짓 으로 설정

: 11번 LED 끄기

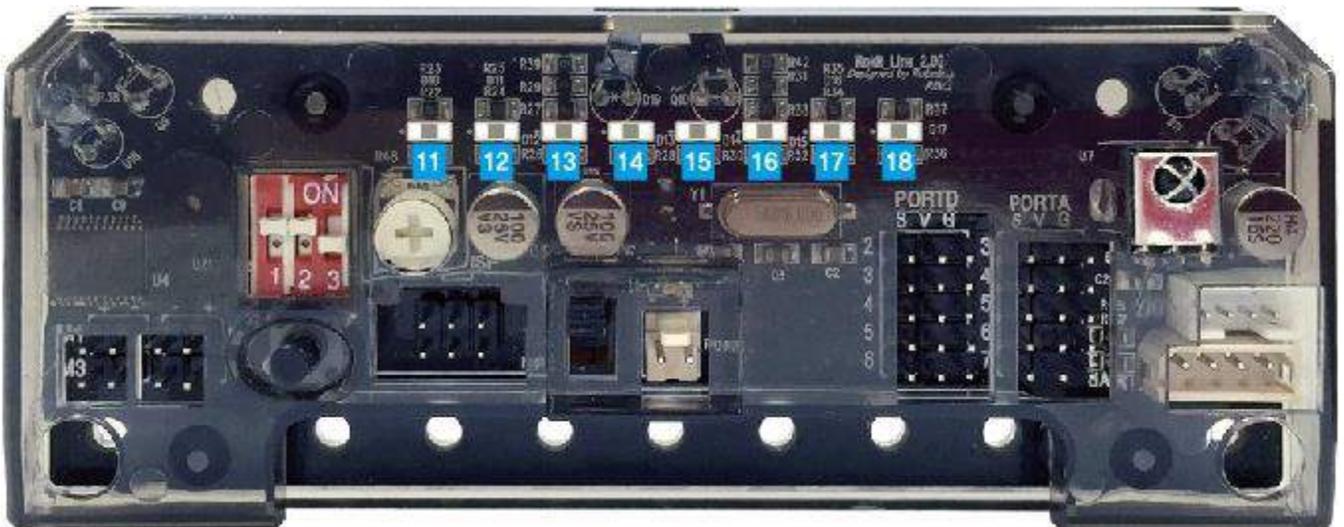
핀의 번호

PIN MODE : OUTPUT

LED DISPLAY

핀에 연결된  
장치

11 12 13 14 15 16 17 18



## 2.3 아날로그(analog) 블록 사용하기

스마트 인벤터 보드의 앞면 3개의 센서에 물체나 손을 감지하면 제어기는 감지 상태에 따라 0에서 부터 1023 사이의 값으로 표현합니다. 물체가 센서에 가까이 있으면 0에 가까운 값을 출력하고, 물체가 멀리 있으면 1023에 가까운 값을 출력합니다. 이처럼 센서의 감지 상태에 따라 제어기가 연속적인 값을 출력하는 것을 아날로그 값 출력이라 합니다.

아날로그 출력이 가능한 핀은 19번부터 26번까지이며, 보드 전방의 세 개 센서는 아래 그림처럼 19번부터 21번 핀에 연결되어 있습니다.



아래와 같이 아날로그 값 블록의 핀 값을 20번으로 한 후 단순히 블록을 클릭해 주면 말풍선으로 센서값이 출력됩니다. (제어기와 로켓 브릭이 먼저 연결되어 있어야 합니다)



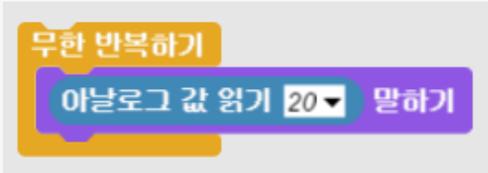
또한 아래와 같이 스크립팅하는 방법도 있습니다. 변수를 만든 후 아날로그 값을 저장한 다음 말하기 블록을 사용하여 변수값을 나타내 줍니다. 이렇게 하면 계속해서 센서의 값이 스프라이트 말풍선으로 출력됩니다.



### 3.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

**예제 1** : 스마트 인벤토리 보드 앞부분에 있는 가운데 센서(20번) 값을 계속해서 스프라이트의 말풍선으로 나타낼 수 있도록 스크립트를 작성합니다. 작성한 스크립트를 실행한 후 이 센서에 손이나 물체를 감지시켜 봅시다.



도움말 : '아날로그 값 읽기' 블록을 말하기 블록과 결합하여 사용하였습니다. 무한 반복을 해야 계속 현재 값이 보여질 수 있습니다. 이 예제에서는 변수 블록을 사용하지 않았다는 것에 유의하세요.

**예제 2** : 스마트 인벤토리 보드 앞부분에 있는 센서 3개 값을 스테이지에 표시하고 3개의 센서들에 손이나 물체를 감지시켜 봅시다.



도움말: 먼저 3개의 센서값을 저장할 수 있는 변수 3개를 만들어 줍니다. 이 3개의 변수에 무한 반복 블록을 사용하여 계속 센서값들을 읽어들입니다.

**예제 3** : 스마트 인벤터 보드의 앞부분 가운데 센서를 손으로 감지할 때 마다 스테이지에 있는 캐릭터가 튀어오르게 해 보세요. 손을 떼면 다시 원래 자리로 돌아와야 합니다.



도움말: 센서값이 100보다 작아지면 스프라이트가 움직이는 것으로 하였습니다. 이 값은 주위 환경의 밝기에 따라 달라질 수 있으므로 직접 값을 검사해보는 것이 좋습니다. 센서값을 검사하는 가장 편한 방법으로는 다른 블록과 결합되지 않은 '아날로그 값 읽기' 블록을 클릭해 보는 것입니다. 이렇게 하면 작은 말풍선으로 클릭한 순간의 센서값이 표시됩니다.



### 3.2 연습문제

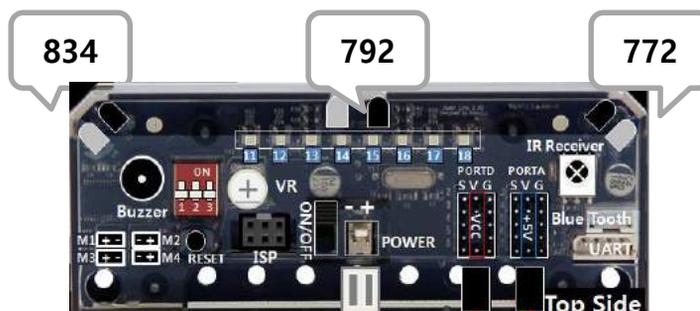
예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트 **DOWNLOADS**를 확인해주세요. 자세한 내용은 교재 16페이지▶

1. 스마트 인벤터 보드의 앞부분 가운데 센서를 감지하면 미사일이 발사되듯이 스프라이트가 스테이지의 아래쪽에서 위로 빠르게 올라갈 수 있도록 스크립트를 작성해 봅시다.
2. 스테이지에 캐릭터 스프라이트를 불러옵니다. 스마트 인벤터 보드의 앞면 왼쪽 센서를 감지하면 스프라이트가 왼쪽으로 이동하고 오른쪽 센서를 감지하면 오른쪽으로 이동하며 가운데 센서를 감지하면 점프하는 효과를 만들 수 있도록 스크립트를 작성해봅시다.

3. 스마트 인벤터 보드의 앞쪽에 있는 세 개의 센서 중 가운데 센서값을 감지하면 스테이지에 센서값을 막대 그래프의 길이로 나타내어주는 스크립트를 작성해 보세요.



4. 스테이지에 아래 그림처럼 인벤터 보드의 앞부분에 있는 3개의 센서마다 말풍선으로 계속 센서값이 출력되도록 합시다(smartInventor.png 사용). 만약 변수를 사용할 경우 사용자의 혼동이 없도록 스테이지의 변수 컨트롤은 보이지 않게 해 주세요.



## 2. 디바이스 제어하기

# 2장.

# 로킷 시뮬레이터

# 게임

# Rokit Simulator

## 1.1 응용 컨트롤



### 시뮬레이션 게임 기능

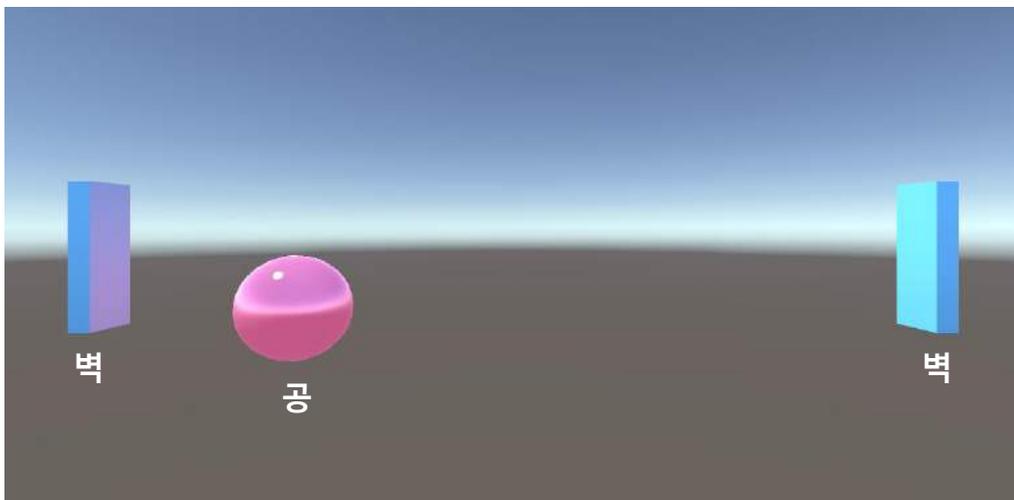
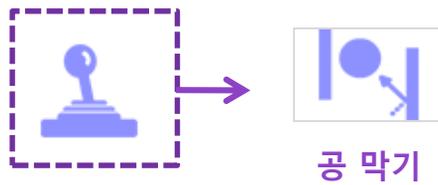
인벤터 보드를 이용한 시뮬레이션 게임을 통해 센서와 보드의 기능을 자연스럽게 이해할 수 있습니다.



해당 버튼을 클릭하면 아래로 풀다운 메뉴가 나타나며 해당 컨트롤 화면으로 이동할 수 있습니다.

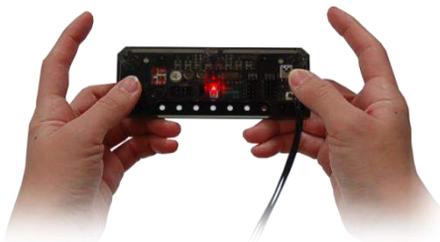
# Rokit Simulator

## 1.2 응용 컨트롤

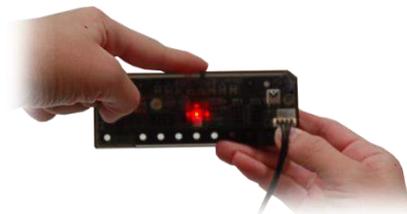


인벤터 보드의 좌우 IR 센서를 감지하여 아날로그 방식으로 화면 양쪽 벽을 움직여줍니다. 벽을 자유롭게 움직여 공이 화면 밖으로 나가지 못하게 합니다.

IR 센서를 감지하는 거리에 따라 공을 막는 벽의 높이가 달라집니다.

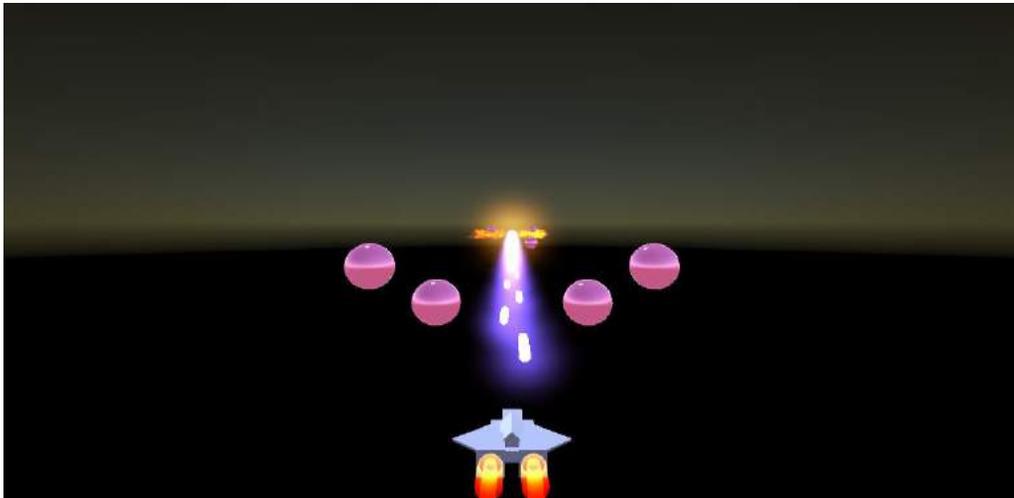
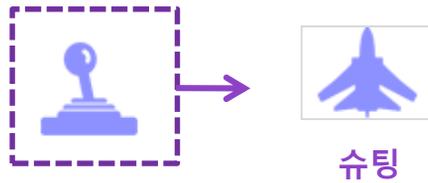


중앙 IR 센서를 약 2초 정도 감지하면 공의 상태가 리셋됩니다.



# Rokit Simulator

## 1.3 응용 컨트롤



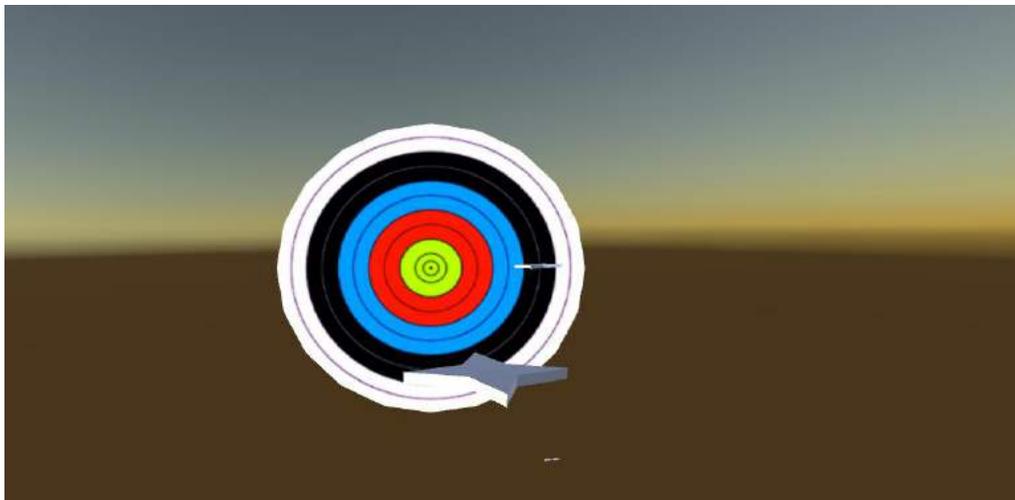
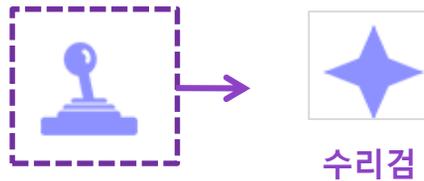
인벤터 보드의 좌우 IR 센서를 감지하여 비행기를 좌우로 움직입니다 (디지털 방식).  
비행기에서 발사되는 미사일로 날아오는 공을 파괴합니다.  
비행기는 좌우 5단계의 정해진 위치로 이동하게 됩니다.

아날로그 센서지만 일정 범위를 기준으로  
디지털 ON/OFF 로 작동합니다.



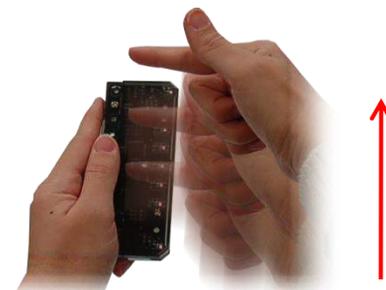
# Rokit Simulator

## 1.4 응용 컨트롤



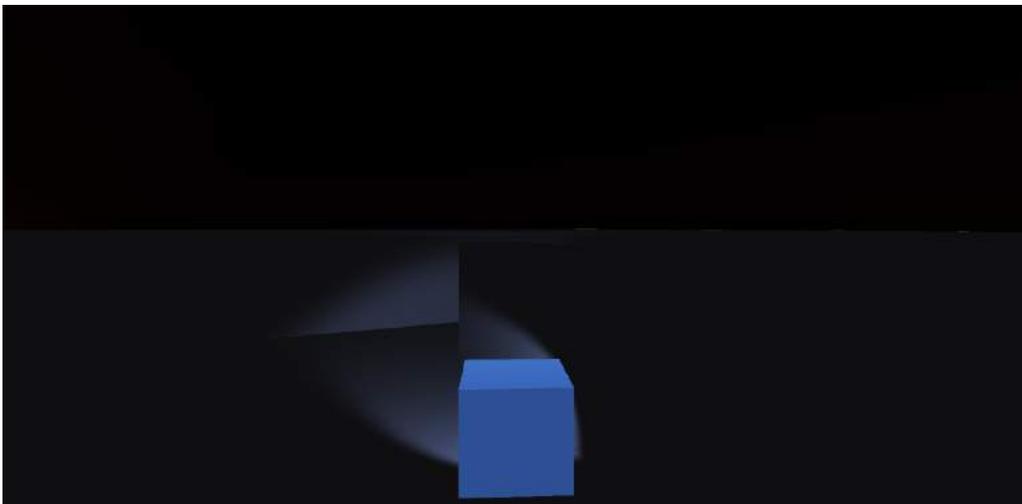
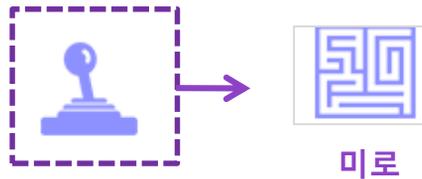
인벤터 보드의 하단 중앙에 위치한 IR센서 5개(양 끝단 센서는 사용하지 않음)를 이용하여 표창을 던집니다. 손가락을 움직여 5개의 센서들을 순차적으로 감지시키면, 각 센서 사이에 입력되는 시간을 속도로 바꾸어 수리검의 속도를 계산합니다.

보드를 뒤집어 잡고  
적당한 속도로 수리검을 던지듯  
손가락을 밀어냅니다.



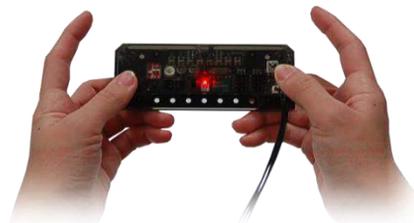
# Rokit Simulator

## 1.5 응용 컨트롤



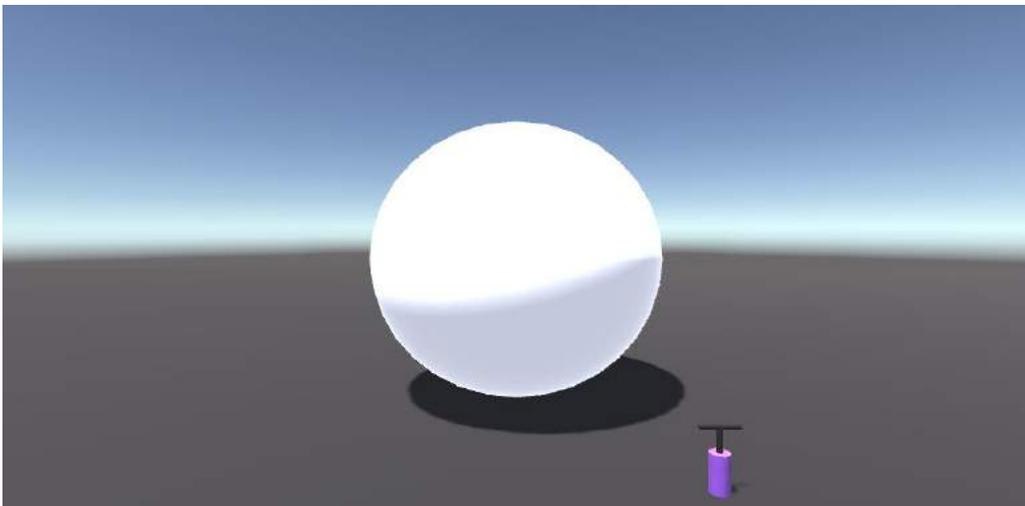
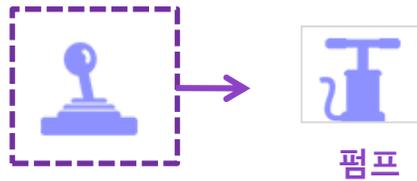
인벤터 보드의 좌우 IR 센서만으로 로봇을 움직여 미로를 탐험합니다. 화면의 로봇은 두 개의 IR센서를 이용하여 총 4가지 동작을 할 수 있습니다.

센서를 모두 가리지 않으면 전진  
양쪽 두 센서를 모두 가리면 후진  
왼쪽 센서를 가리면 좌회전  
오른쪽 센서를 가리면 우회전



# Rokit Simulator

## 1.6 응용 컨트롤



인벤터 보드의 하단 중앙에 위치한 디지털 IR 센서 5개(양 끝단 센서는 사용하지 않음)를 이용하여 펌프를 움직여줍니다. 손가락을 이용하여 센서들을 순차적으로 왕복하여 감지시키면, 각 센서 사이에 입력되는 위치에 따라 펌프도 같은 위치로 움직입니다.

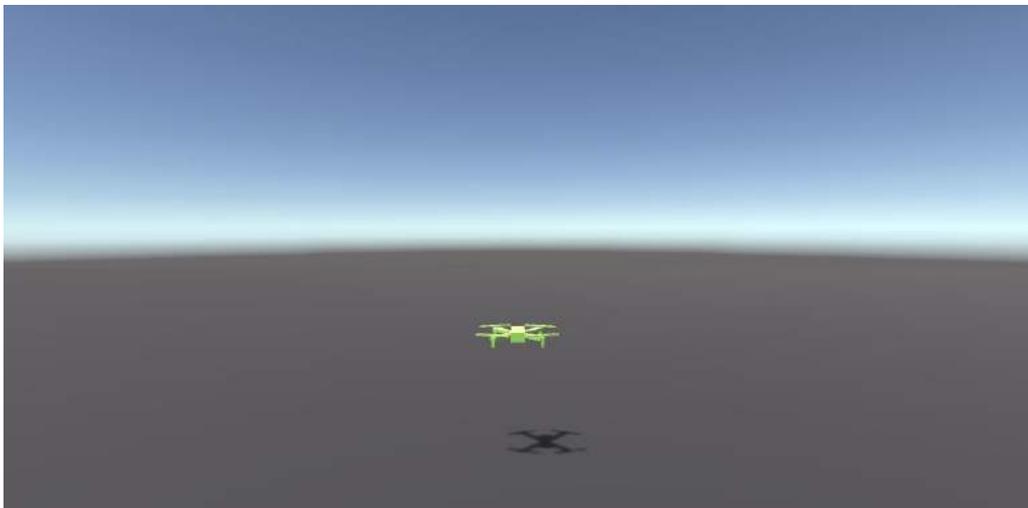
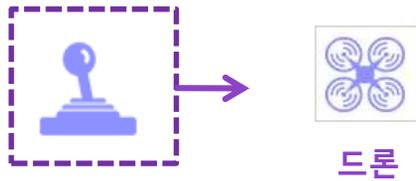
왼쪽 손가락을 가로로 누혀  
디지털 센서에 대고 오른손으로  
보드를 수직으로 잡아  
펌프질 하듯 서로 상하운동 합니다.

왼쪽 손가락은 위치를 고정시켜줍니다.

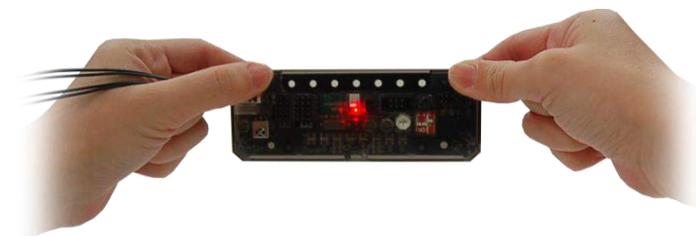


# Rokit Simulator

## 1.7 응용 컨트롤



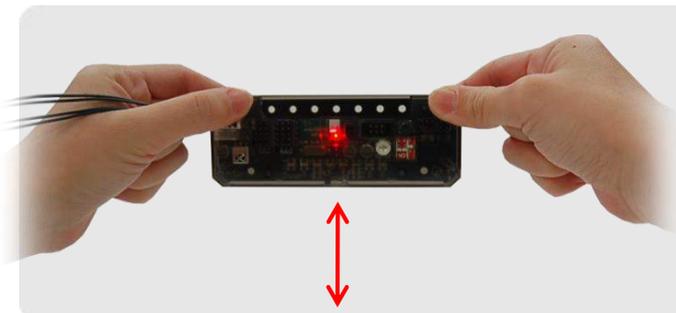
인벤터 보드의 좌우, 중앙 IR 센서와 양쪽 끝 두 개의 디지털 센서를 사용하여 화면의 드론을 조종합니다. 다음 페이지의 **드론 정밀 제어 방법** 을 확인하여 주세요.



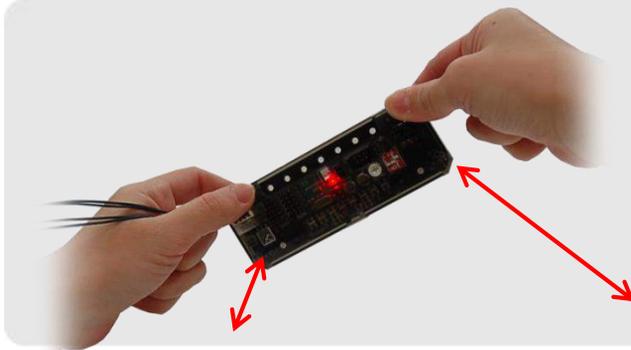
# Rokit Simulator

## 1.7 응용 컨트롤

### 드론 정밀 제어 방법



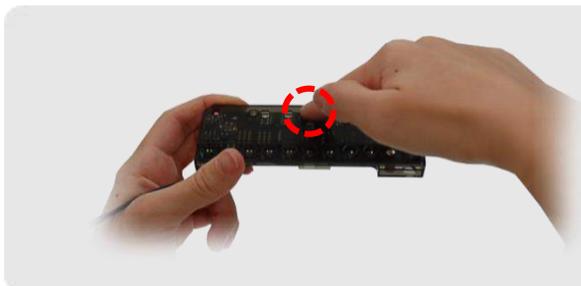
a. 보드의 중앙 IR 센서가 아래쪽으로 향하게 잡은 후, 바닥과의 보드의 거리를 변화시키며 화면 속 드론의 상승과 하강을 컨트롤 합니다.



b. "a."처럼 보드를 잡은 후, 보드를 좌우로 기울여서 양쪽 IR 센서가 바닥과의 거리를 인식해 드론이 좌우로 움직일 수 있도록 해줍니다.



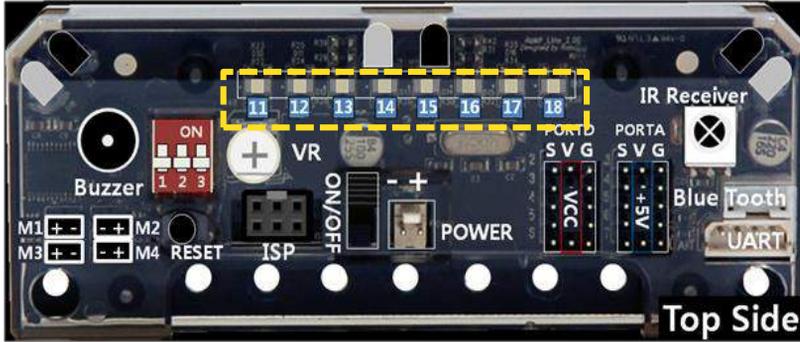
c. 뒷면의 양쪽 끝 두 개의 디지털 센서는 각각 드론의 전진 후진을 컨트롤 합니다.



d. 중앙의 디지털 센서는 멀리 이동한 드론을 다시 제자리로 돌아오게 하는 역할을 합니다.

## 2.1 디지털 출력이란?

디지털 출력이란 한 마디로 On(켜짐), Off(꺼짐) 이렇게 두 가지 신호만으로 표현하는 것을 의미합니다. 스마트 인벤터 보드에서 디지털 출력을 잘 보여주는 장치는 전면 8개의 LED 디스플레이입니다. 아래 그림처럼 11번부터 18번까지의 핀이 LED 디스플레이를 제어하는 데 사용됩니다.



## 2.2 디지털 핀 블록 사용하기

로켓 브릭은 디지털 핀 블록을 사용하여 핀의 상태를 '참'의 상태로 설정했을 때 출력이 On 되면서 LED가 켜집니다. 아래 블록을 실행하면 11번 핀에 연결된 LED를 켜 줍니다.

디지털 핀 11번을 참으로 설정

: 11번 LED 켜기

또한 핀의 상태를 '거짓'으로 설정했을 때에는 출력이 Off가 되면서 LED가 꺼집니다.

디지털 핀 11번을 거짓으로 설정

: 11번 LED 끄기

동시에 여러 개의 LED를 켜거나 끌 때에는 연속적으로 블록을 결합시켜 사용하면 됩니다. 아래의 예제는 1초 동안 3개의 LED를 동시에 켜 뒤, 꺼질 때에도 함께 꺼지는 예제입니다.

디지털 핀 11번을 참으로 설정  
 디지털 핀 12번을 참으로 설정  
 디지털 핀 13번을 참으로 설정  
 1 초 기다리기  
 디지털 핀 11번을 거짓으로 설정  
 디지털 핀 12번을 거짓으로 설정  
 디지털 핀 13번을 거짓으로 설정

### 3.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 교재 16페이지▶

**예제 1** : 스마트 인벤터 보드의 LED 장치들 중 11번과 18번에 동시에 불이 들어와서 1초 간격으로 깜빡일 수 있도록 스크립트를 작성해 봅시다.

무한 반복하기

디지털 핀 11 ▾ 번을 **켜** 으로 설정

디지털 핀 18 ▾ 번을 **켜** 으로 설정

1 초 기다리기

디지털 핀 11 ▾ 번을 **꺼** 으로 설정

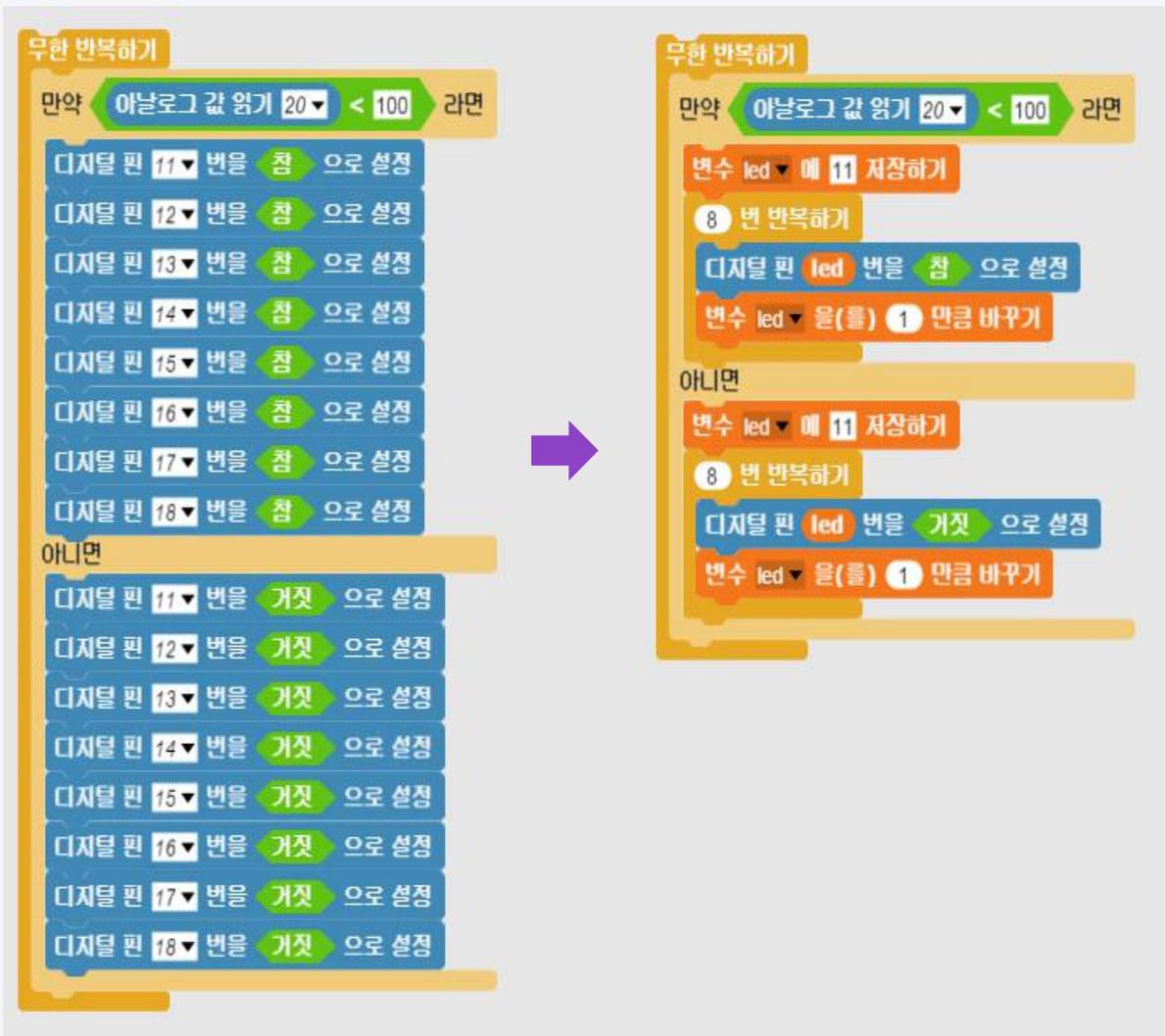
디지털 핀 18 ▾ 번을 **꺼** 으로 설정

1 초 기다리기



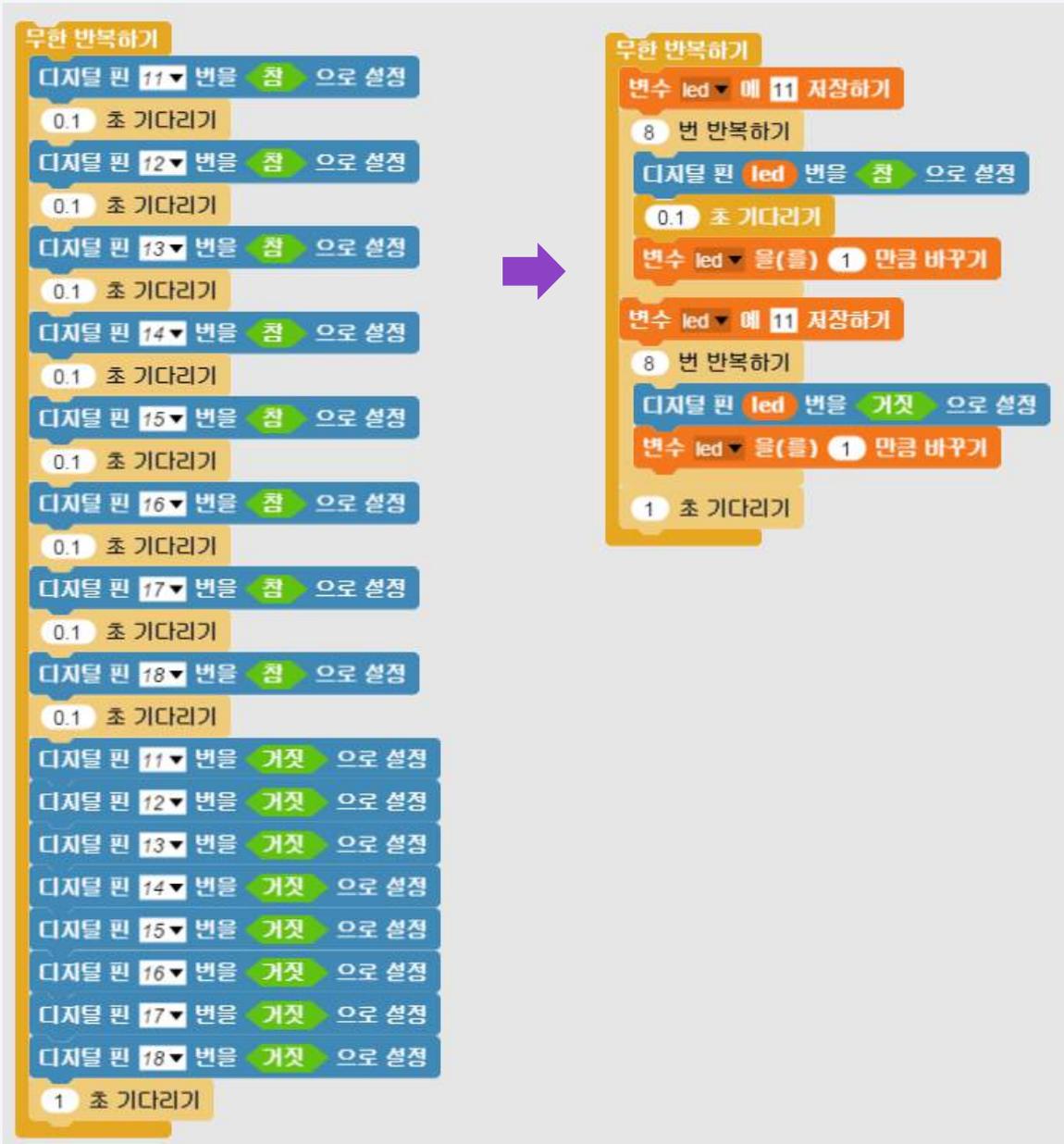
도움말: '~초 기다리기'블록의 값을 조절하면 깜빡이는 주기가 변화하게 됩니다.  
 시간을 변화시켜서 다양한 주기를 만들어 봅시다.

**예제 2 :** 스마트 인벤터 보드의 앞부분에 있는 가운데 센서를 감지했을 때 8개의 LED가 모두 켜지게 해 봅시다. 감지되지 않은 경우에는 8개의 LED가 전부 꺼지게 합니다.



도움말 : 조건 블록을 사용해서 센서가 감지되었는지 알아봅니다. <조건>센서값이 100보다 작아지면 LED를 모두 켭니다. 이 때 왼쪽 스크립트처럼 반복문을 사용하면 더 짧은 스크립트로 만들 수 있습니다. 단, 디지털 핀의 번호에도 변수를 사용하여 바꿔줘야 하는 것에 유의합니다.

**예제 3 :** 스마트 인벤터 보드의 8개 LED를 왼쪽부터 0.1초 간격을 주어 차례대로 불을 켭니다. 8개의 LED가 전부 켜지면, 그 즉시 모두 꺼지고 다시 처음과 같은 동작을 반복하게 하여 봅시다.

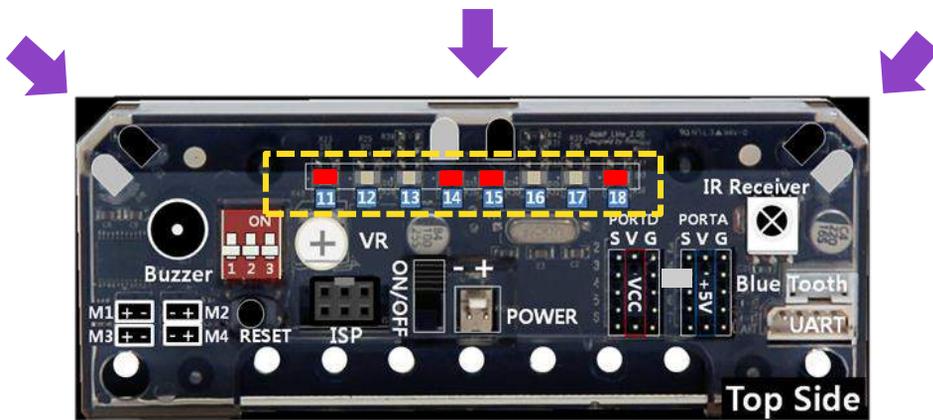


도움말 : 왼쪽과 오른쪽은 서로 동일한 결과를 만들지만 반복문과 변수를 사용하여 스크립트를 좀 더 짧게 만들 수 있습니다. 왼쪽에 있는 스크립트에서는 핀 번호를 직접 입력하지 않고 'led'라는 변수 블록을 만들어 디지털 핀 블록에 결합하였습니다. 그리고 이 변수 블록의 값을 1씩 증가시켜 LED를 제어하였습니다.

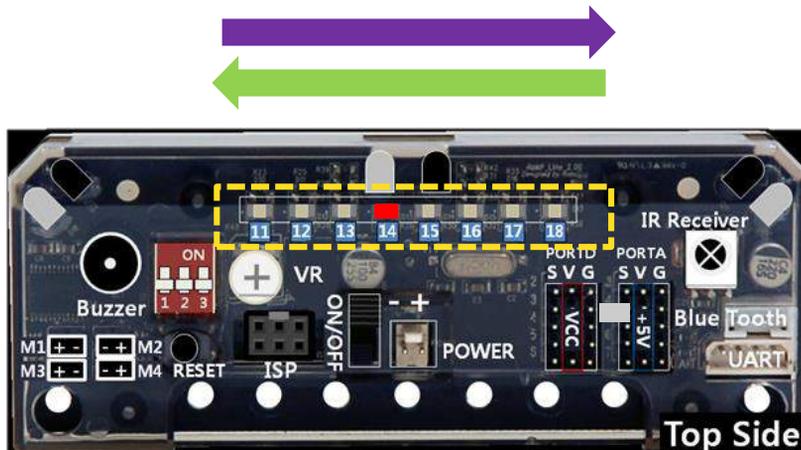
### 3.2 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트 **DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

- 스마트 인벤터 보드의 앞부분에 있는 세 개의 센서를 사용합니다. 왼쪽 센서를 감지하면 가장 왼쪽에 있는 LED가 켜지고 가운데 센서를 감지하면 가운데 있는 두 개의 LED가 켜지며 오른쪽 센서를 감지하면 가장 오른쪽에 있는 LED가 켜질 수 있게 해 봅시다.



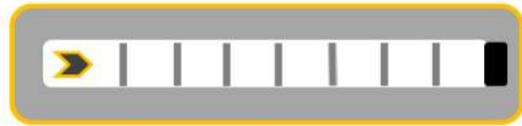
- LED 애니메이션** : 스마트 인벤터 보드의 LED 불빛이 왼쪽부터 오른쪽으로 순서대로 켜진 후, 다시 오른쪽부터 왼쪽으로 켜지는 동작을 계속 반복할 수 있게 스크립트를 작성해 보세요.



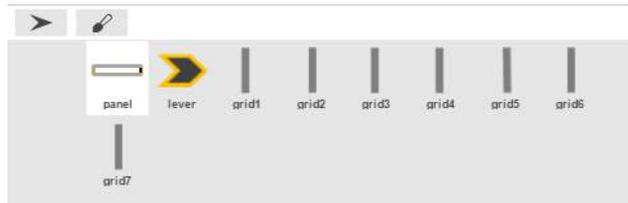
3. **스테이지 버튼으로 LED 제어** : 스테이지에 아래와 같은 세 개의 사각형이 있습니다. 사각형들을 마우스로 클릭하면 사각형의 색이 파랗게 변하는 동시에 스마트 인벤터 보드의 LED가 켜졌다가 마우스 버튼을 떼면 LED가 다시 꺼지게 해 봅시다. 맨 왼쪽 사각형을 클릭하면 맨 왼쪽 LED가 켜지고 가운데 사각형을 클릭하면 가운데 두 개의 LED가 꺼지고 오른쪽 사각형을 클릭하면 맨 오른쪽 LED가 켜지게 해 봅시다.



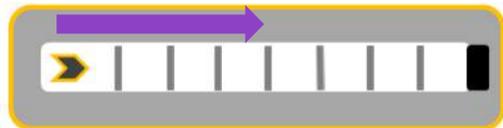
4. **LED로 레벨 표시** : 옆의 그림과 같이 스테이지에 슬라이더를 구성합니다. 슬라이더의 구성은 아래와 같습니다.



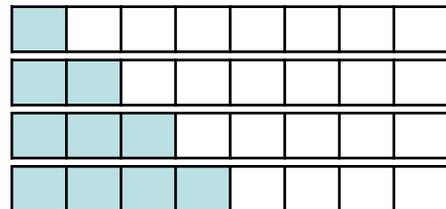
1) 가운데 레버는 마우스를 따라 움직입니다. 그러나 레버가 양끝에 위치하면 마우스를 따라 움직이지 않고 멈추어 있어야 합니다.



2) 옆 그림처럼 그림처럼 레버가 오른쪽으로 움직이면 움직인 만큼 LED 불빛이 늘어납니다. 다시 레버가 왼쪽으로 움직이면 움직인 만큼 LED 불빛이 줄어들게 됩니다.



3) 레벨 미터에 눈금이 7개가 표시되어 있습니다. 각 눈금과 레버가 만났을 때 LED 불빛의 변화가 있습니다.



마우스의 움직임에 따라 레버와 LED가 변화할 수 있게 스크립트를 작성해 봅시다.



Control1.png



Control2.png



Control3.png

## 2. 디바이스 제어하기

### 3장.

로킷 시뮬레이터

로봇 시뮬레이션

# Rokit Simulator

## 1.1 GUI 프로그래밍



### GUI 프로그래밍 기능

▶ 간단한 클릭만으로 로봇의 움직임 프로그래밍 가능

지능적으로 움직이는 로봇의 원리를 프로그래밍하여 시뮬레이팅 해볼 수 있으며, 실제 로봇이 없어도 로봇의 기본 개념을 학습할 수 있습니다.



미로탐험로봇

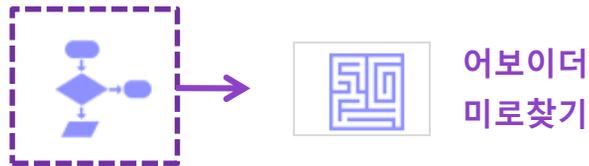
라인트레이서

추락방지로봇

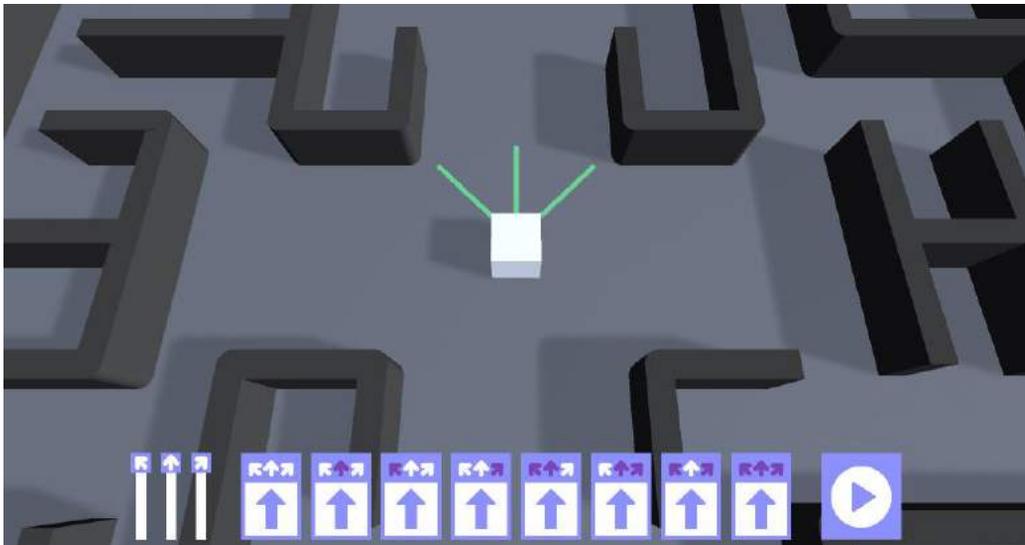
해당 버튼을 클릭하면 아래로 풀다운 메뉴가 나타나며 해당 컨트롤 화면으로 이동할 수 있습니다.

# Rokit Simulator

## 1.2 어보이더 미로찾기



전방에 위치한 2개 이상의 물체 감지 센서를 통해 물체를 감지합니다. 동시에 미로에 부딪히지 않고 벽을 피해서 길을 찾는 기능을 손쉽게 프로그래밍하고 원리를 학습할 수 있습니다.



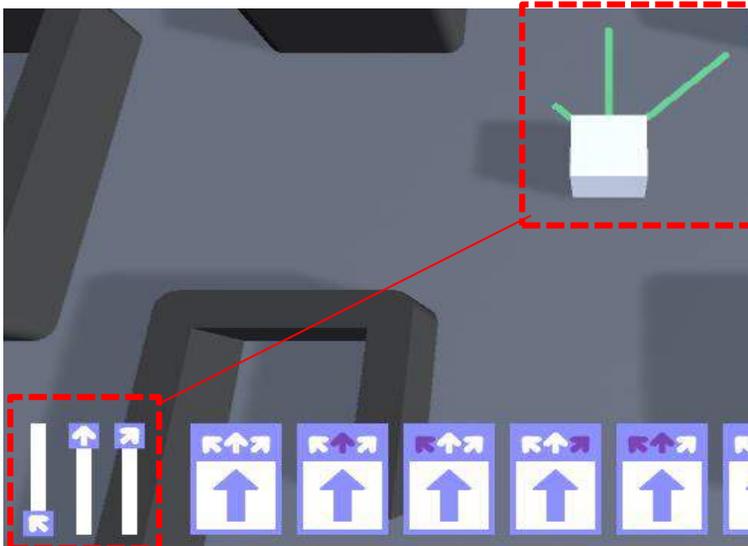
장애물 감지 센서의 감지 상황에 따라 해당 동작을 지정하여 로봇이 미로를 막힘없이 돌아다닐 수 있도록 프로그래밍 합니다.

# Rokit Simulator

## 1.2 어보이더 미로찾기

세 개의 슬라이더 바를 조절하여 물체 감지 센서의 감지 거리를 변경 할 수 있습니다.

벽 감지 센서 감도 조절 슬라이더



상황별 동작 지정 버튼

동작 버튼

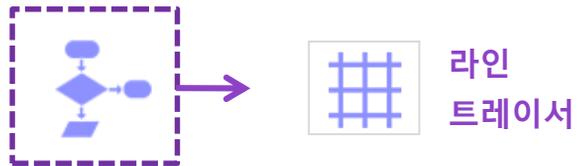
시작 버튼



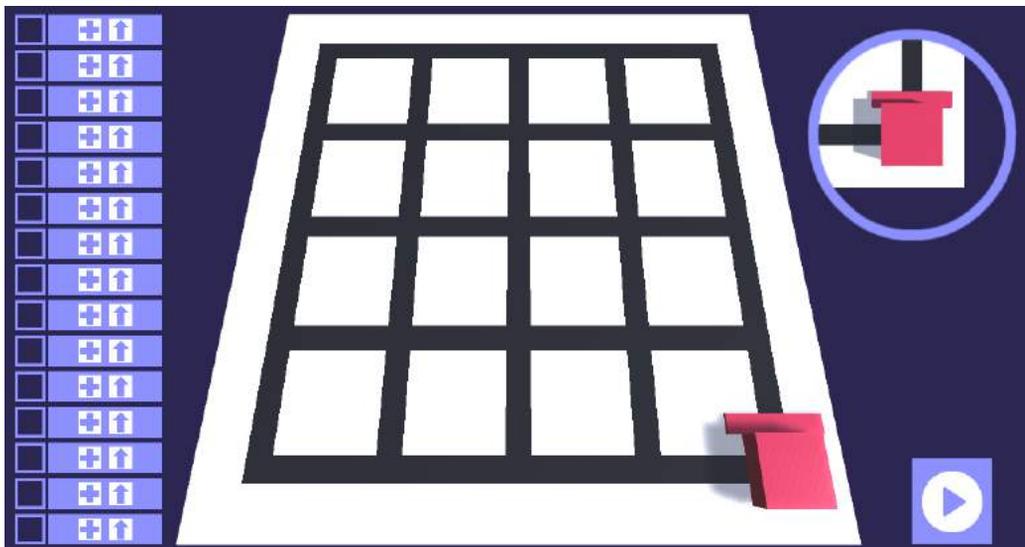
각 센서를 감지한 상황에 따라 지정된 동작을 입력할 수 있습니다. 동작 버튼을 클릭하면 입력한 동작을 변경할 수 있으며, 우측의 시작버튼 ▶을 누르면 각 상황에 맞는 동작을 실행합니다.

# Rokit Simulator

## 1.3 라인트레이서



전방에 위치한 2개 이상의 물체 감지 센서를 통해 물체를 감지합니다. 동시에 라인을 따라 움직이는 로봇을 손쉽게 프로그래밍하고 원리를 학습할 수 있습니다.



로봇이 처해있는 교차로 모양과 그에 따른 동작을 순차적으로 지정하여 원하는 곳으로 로봇을 이동시킵니다.

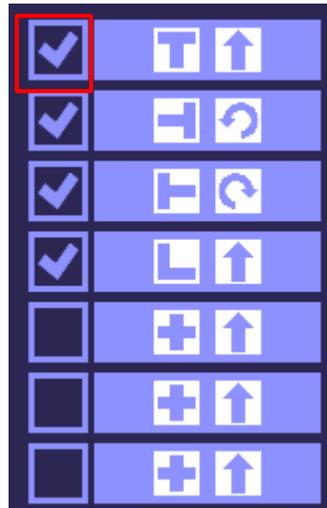
# Rokit Simulator

## 1.3 라인트레이서

상황별 교차로의 모양과 그에 따른 로봇의 동작을 순차적으로 작성합니다. 현재 로봇이 처한 라인의 상황과 동작패턴을 잘 생각해서 프로그램을 작성해보도록 합니다.

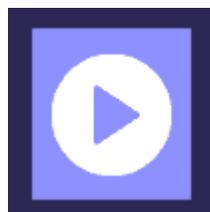
프로그래밍 셀의 좌측 실행 항목 부분에 체크  하여 로봇 동작을 실행시킬 셀을 활성화합니다.

### 진행 순서 프로그래밍 셀



교차로 모양 / 로봇의 동작

시작 버튼을 눌러 로봇을 작동시켜줍니다.

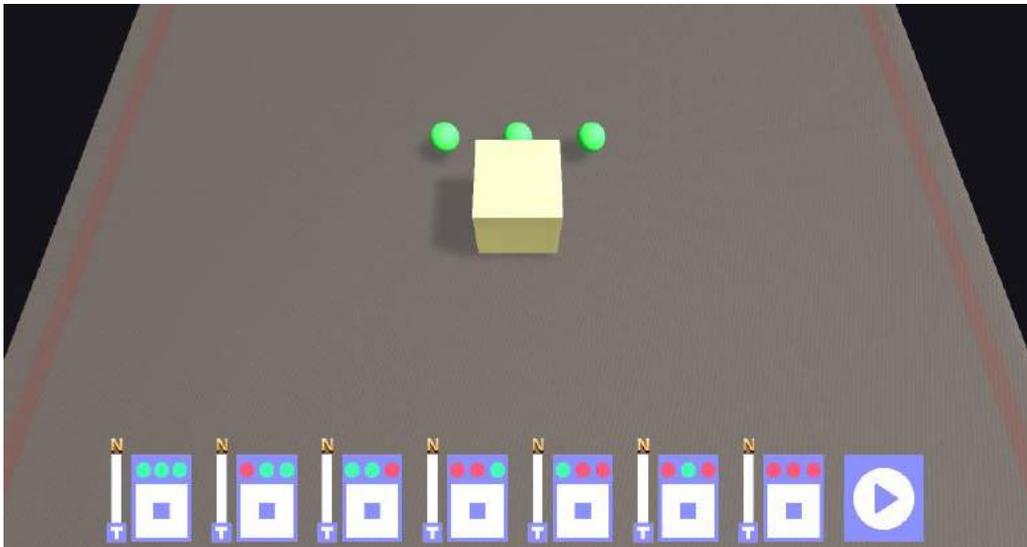


# Rokit Simulator

## 1.4 추락방지 로봇



뒷면 세 개의 센서를 이용하여 낭떠러지를 감지하고 회피하는 로봇입니다. 하얀 바닥 위에서 구동되어야 하는 추락방지 로봇의 원리는 청소기 로봇에도 그대로 적용할 수 있습니다.



낭떠러지 감지 센서의 감지 상황에 따라 해당 동작을 지정하여 로봇이 화면 속의 경기장 밖으로 떨어지지 않고 자유롭게 움직일 수 있도록 프로그래밍 합니다.

# Rokit Simulator

## 1.4 추락방지 로봇



각 센서를 감지한 상황에 따라 지정된 동작을 입력할 수 있습니다. 동작 버튼을 클릭하면 입력된 동작을 변경할 수 있으며, 각 버튼 좌측의 슬라이더를 사용하여 해당 동작의 지속 시간을 변경할 수 있습니다.

(0 - 1.0초)

우측의 시작버튼을 누르면 각 상황에 맞는 동작을 실행합니다.



항상 기본 설정을 모두 마친 다음 시작 버튼을 눌러 로봇을 작동시키고, 동작을 멈춘 후에 설정을 바꿀 수 있습니다.

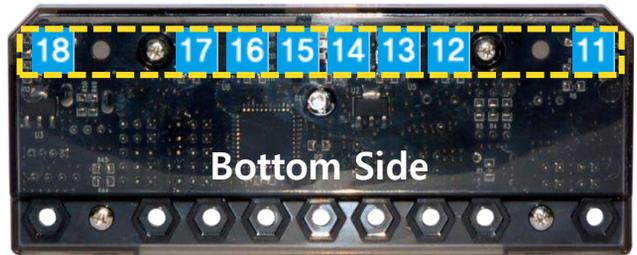
## 2.1 디지털 핀 입력이란?

디지털 입력이란 제어기에 연결된 어떤 장치나 센서의 상태를 오직 On(켜짐), Off(꺼짐) 두 가지 값으로만 표현하여 입력되는 것을 의미합니다. 어떤 장치나 센서의 상태가 0에서부터 1023까지의 숫자의 값으로 표현되는 아날로그 입력값과는 달리 오직 켜짐과 꺼짐, 0과 1, 혹은 참과 거짓으로만 표현됩니다.

스마트 인벤터 보드에서 디지털 핀 입력을 가장 잘 보여줄 수 있는 장치는 보드 뒷면의 7조 센서와 보드 전면에 부착된 3개의 DIP스위치(dip switch)입니다.



DIP스위치



7조 센서

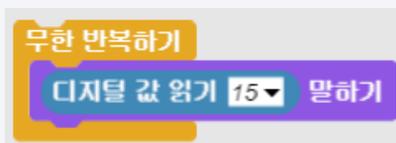
로켓 브릭에서 접근할 수 있는 디지털 핀은 DIP스위치와 연결된 8번 ~ 10번 핀, 그리고 7조 센서와 연결된 11번 ~ 18번 핀 등입니다. 이 핀들은 모두 각각 핀에 연결된 장치들의 상태를 '참' 혹은 '거짓'으로 알려줍니다.

## 2.2 디지털 읽기 블록 사용하기

디지털 핀의 상태를 알기 위해서는 로켓 카테고리에서 아래와 같은 블록을 사용합니다. 블록에서 알고 싶은 핀 번호를 선택한 후 블록을 클릭만 하더라도 말풍선으로 핀의 상태를 알려줍니다.

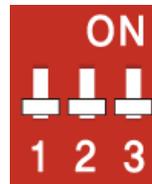


아래와 같이 스크립트 하면 계속해서 지정한 핀 값의 상태를 스프라이트의 말풍선으로 알려줍니다.



### 3.1 DIP 스위치 값 읽어오기

DIP 스위치의 1번은 8번 핀, 2번은 9번 핀, 3번은 10번 핀과 연결되어 있습니다. 만약 3번 스위치를 내리면 10번 핀의 상태는 'On'이 되고 이 때 **디지털 값 읽기 블록**을 사용하면 '참'으로 핀의 값이 표현됩니다. 반대로 스위치를 올리면 핀의 상태는 'Off'가 되고 핀의 값은 '거짓'으로 표현됩니다.

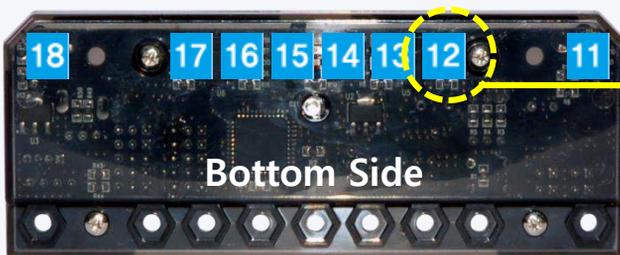


또한 아래와 같이 스크립트를 작성하면 3개 DIP 스위치의 변경되는 값을 스테이지에서 계속 확인할 수 있습니다.



### 3.2 7조 센서 값 읽어오기(디지털 핀 읽기 블록을 사용하기)

7조 센서값을 읽는 방법은 두 가지가 있습니다. 첫 번째 방법은 DIP 스위치처럼 **디지털 값 읽기** 블록을 사용하여 7조 센서를 이루는 각 센서의 상태를 읽어오는 것이며, 두 번째 방법은 7조 센서 값 읽기 블록을 사용하여 7개의 센서값을 한 번에 읽어오는 방식입니다. 아래 그림처럼 12번 핀 센서를 감지한 상태에서 **디지털 값 읽기** 블록을 실행하면 '참'으로 핀의 상태를 알려줍니다. 감지되지 않은 상태에서는 **디지털 값 읽기** 블록은 핀의 상태를 '거짓'으로 알려줍니다.



Bottom Side



## 4.1 2진수와 10진수

7조 센서값 전체를 한 번에 읽어야 할 때 편리하게 사용할 수 있습니다. 다만 이 블록은 7조 센서값을 8자리 2진수로 계산한 후에 다시 10진수로 변환된 값을 알려주기 때문에 2진수에 대한 지식이 필요합니다.

2진수는 0과 1로만 나타낸 수입니다. 어떤 숫자의 값이 1보다 큰 경우에는 자리 올림을 합니다. 예를 들어 2를 2진수로 표현하면 10이 되며 3을 2진수로 표현하면 11이 됩니다. 만약 1100111과 같은 큰 자릿수의 2진수인 경우에는 아래와 같이 계산하면 편리합니다.

$2^7=$	$2^6=$	$2^5=$	$2^4=$	$2^3=$	$2^2=$	$2^1=$	$2^0=$
128	64	32	16	8	4	2	1
	1	1	0	0	1	1	1

$$64 + 32 + 0 + 0 + 4 + 2 + 1 = 103$$

7조 센서값 역시 이처럼 2진수의 형태로 계산됩니다. 예를 들어 7개 중 18번 센서와 16번 센서가 감지된 경우 아래와 같이 2진수로 나타내면 10100000이며 이것을 10진수로 나타내면 160이 됩니다.

18	17	16	15	14	13	12	11
1	0	1	0	0	0	0	0

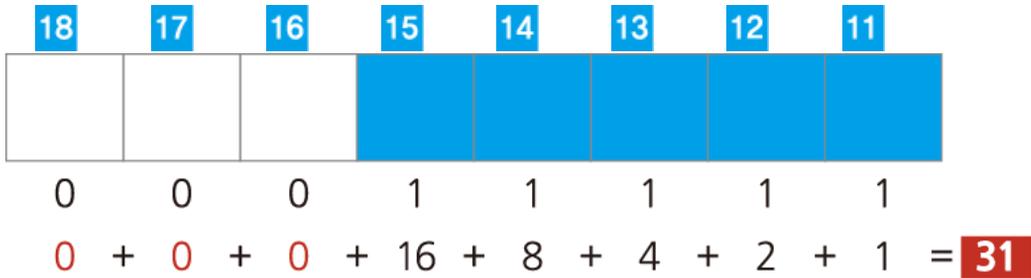
$$128 + 0 + 32 + 0 + 0 + 0 + 0 + 0 = 160$$

7조 센서 중 가운데 센서만 감지된 경우는 14번과 15번 핀이 동시에 1(참)로 상태가 변화합니다. 14번과 15번 두 개의 핀이 가운데에 위치한 하나의 센서에 연결되어 있기 때문입니다. 따라서 가운데 센서가 감지되었을 때의 값은 아래와 같습니다.

18	17	16	15	14	13	12	11
0	0	0	1	1	0	0	0

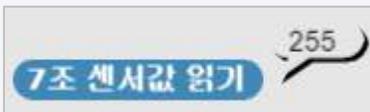
$$0 + 0 + 0 + 16 + 8 + 0 + 0 + 0 = 24$$

7조 센서 중 가운데 센서와 13번, 12번, 11번 센서가 감지된 경우를 이진수로 나타내면 11111이며 아래와 같이 십진수로 계산하면 31이 됩니다.

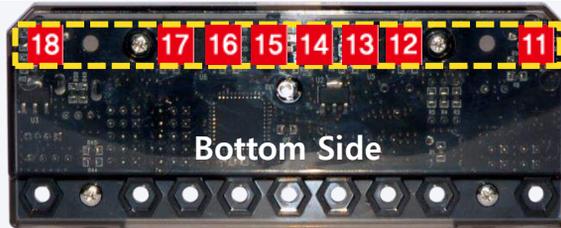


## 4.2 7조 센서 값 읽기(7조 센서값 읽기 블록 사용)

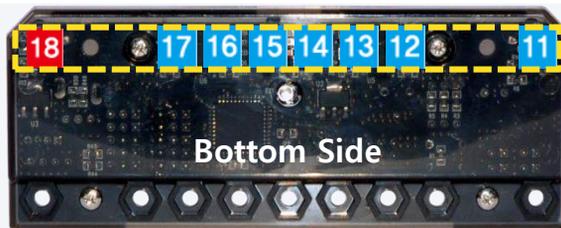
7조 센서에 물체나 손을 감지시킨 후 7조 센서값 읽기 블록을 클릭하여 값을 알아봅시다.



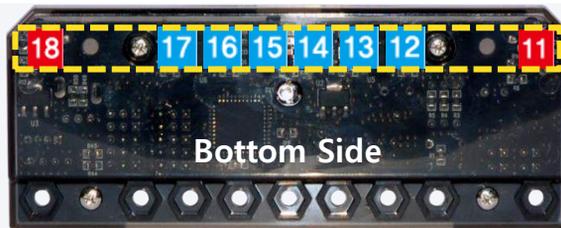
모두 감지한 경우 : **255**



18번만 감지한 경우 : **128**



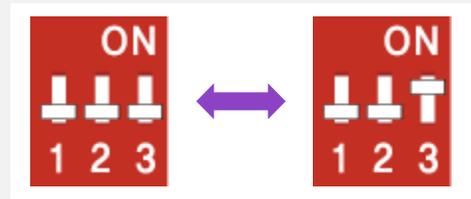
18번과 11번을 감지한 경우 : **129**



## 5.1 예제

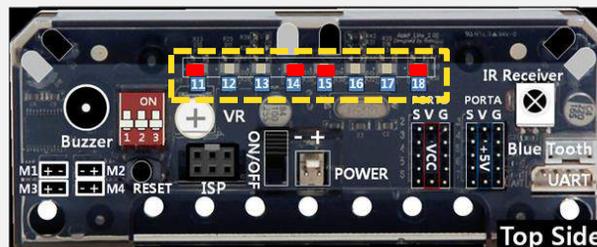
예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

**예제 1** : DIP스위치 3번을 올리면 스프라이트의 말풍선에 'Switch On!'이라는 메시지가 나오고  
 DIP스위치를 내리면 'Switch Off' 라는 메시지가 나오게 스크립트를 작성해 봅시다.



도움말 : DIP스위치 3번에 해당하는 핀 번호는 10번 입니다. 10번 값이 '거짓' 이면 스위치가 아래로 내려간 상태이고 반대로 '참'이면 스위치가 위로 올라간 상태입니다.

**예제 2** : DIP스위치 1번을 올리면 아래 그림과 같이 맨 왼쪽(11번) LED가 켜지고 스위치를 내리면 LED가 꺼지게 해 봅시다. 마찬가지로 DIP스위치 2번을 올리거나 내리면 가운데 LED(14, 15번) 2개가 켜지거나 꺼지고 DIP스위치 3번을 올리거나 내리면 맨 오른쪽(18번) LED가 켜지거나 꺼지도록 스크립트를 작성해 봅시다.



무한 반복하기

만약 **디지털 값 읽기 8** = 거짓 라면

디지털 핀 11 변수 참 으로 설정

아니면

디지털 핀 11 변수 거짓 으로 설정

만약 **디지털 값 읽기 9** = 거짓 라면

디지털 핀 14 변수 참 으로 설정

디지털 핀 15 변수 참 으로 설정

아니면

디지털 핀 14 변수 거짓 으로 설정

디지털 핀 15 변수 거짓 으로 설정

만약 **디지털 값 읽기 10** = 거짓 라면

디지털 핀 18 변수 참 으로 설정

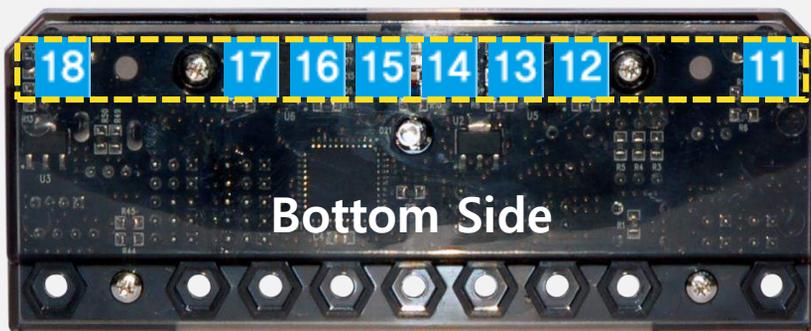
아니면

디지털 핀 18 변수 거짓 으로 설정

도움말 : 스위치의 상태는 계속 고정되므로 스위치를 위로 올릴 경우 디지털 값은 '거짓'이 되고 조건 블록의 '만약' 아래의 스크립트 부분이 실행 됩니다.

딥스위치를 아래로 내리면 디지털 값 읽기 블록의 값은 '참'이 됩니다. 따라서 '만약 ~ 라면' 블록에서 '하단의 아니면' 아래에 연결된 블록들이 실행됩니다.

**예제 3** : 7조 센서값을 이용하여 18번 핀에 연결된 센서가 감지되면 '도' 음이 연주되고, 17번 핀에 연결된 센서가 감지되면 '레' 음이 연주되게 해봅시다. 이렇게 각 센서가 감지될 때 마다 각각의 음이 연주될 수 있는 스크립트를 작성해 보세요.



무한 반복하기

```

만약 7조 센서값 읽기 = 128 라면
    60 음을 0.5 박자로 연주하기
만약 7조 센서값 읽기 = 64 라면
    62 음을 0.5 박자로 연주하기
만약 7조 센서값 읽기 = 32 라면
    64 음을 0.5 박자로 연주하기
만약 7조 센서값 읽기 = 24 라면
    65 음을 0.5 박자로 연주하기
만약 7조 센서값 읽기 = 4 라면
    67 음을 0.5 박자로 연주하기
만약 7조 센서값 읽기 = 2 라면
    69 음을 0.5 박자로 연주하기
만약 7조 센서값 읽기 = 1 라면
    71 음을 0.5 박자로 연주하기
    
```

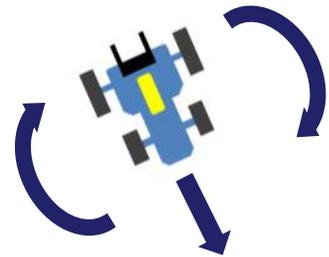
도움말 : 아무것도 감지되지 않은 상태에서 7조 센서의 값은 0 입니다. 스마트 인벤터 보드를 뒤집은 상태에서 가장 왼쪽만을 감지했을 때는 7조 센서의 값이  $2^7 = 128$ 이 됩니다. 왼쪽에서 두 번째 센서의 값을 감지했을 때는  $2^6 = 64$ 가 되는 것입니다.

여기서 가운데 센서는 하나만을 감지하더라도 마치 두 개의 센서가 감지된 것처럼  $2^4(16) + 2^3(8) = 24$ 의 값이 출력된다는 점을 유의하기 바랍니다.

이것은 8개의 핀을 7개의 센서에 연결해서 사용하도록 설계되어 있어 가운데 센서가 2개의 핀을 차지하기 때문입니다.

- 도 :  $2^7 = 128$
- 레 :  $2^6 = 64$
- 미 :  $2^5 = 32$
- 파 :  $2^4 + 2^3 = 24$
- 솔 :  $2^2 = 4$
- 라 :  $2^1 = 2$
- 시 :  $2^0 = 1$

**예제 4 :** 오른쪽 그림과 같은 레이싱 카 모양의 스프라이트가 있습니다([racingcar.png](http://racingcar.png)를 가져오기). 7조 센서의 왼쪽 센서(11번)를 감지시키면 왼쪽으로 회전하고 오른쪽 센서(18번)를 감지시키면 오른쪽으로 회전합니다. 또한 11번과 18번을 동시에 감지시키면 스프라이트가 전진할 수 있도록 스크립트를 작성해 봅시다.



무한 반복하기

```

만약 7조 센서값 읽기 = 128 + 1 라면
    5 만큼 움직이기
만약 7조 센서값 읽기 = 128 라면
    3 도 돌기
만약 7조 센서값 읽기 = 1 라면
    3 도 돌기
    
```

도움말 : 이 예제에서는 센서를 동시에 두 개 이상 감지시키는 경우, 어떻게 처리할 것인지 잘 보여줍니다. 맨 왼쪽 센서의 7조 센서값은 128이고 맨 오른쪽 센서의 7조 센서값은 1이므로 두 개를 동시에 감지시키면  $128+1$  즉, 129가 7조 센서값이 됩니다. 이러한 원리로 7조 센서 중 동시에 여러 개의 센서를 감지시키는 경우를 구별할 수 있습니다.

## 5.2 연습문제

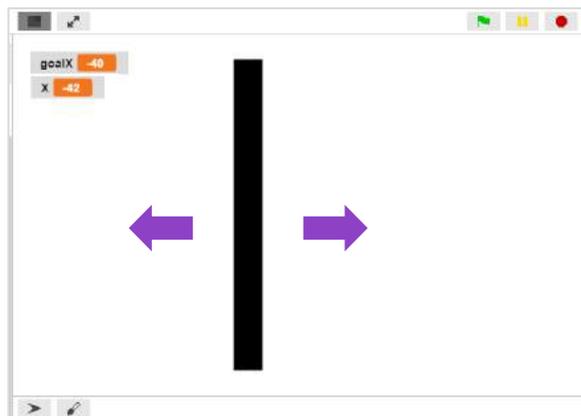
예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 교재 16페이지▶

- 오른쪽 그림과 같이 스테이지에 1번부터 3번까지 숫자가 적혀 있습니다. 각 숫자는 별개의 스프라이트로 이루어져 있습니다. 여기에서 특정 숫자의 디스위치를 올리면 스테이지 위에서 해당하는 숫자의 색상이 바뀌도록 해 봅시다. 그리고 스위치를 내리면 원래의 색상으로 돌아올 수 있도록 합니다.

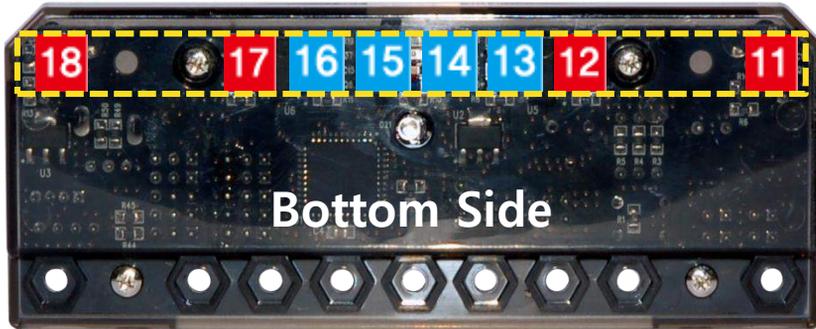


1 2 3

- 아래 그림처럼 스마트 인벤터 보드의 7조 센서를 폭이 약 2cm 정도인 검은 선에 감지시키며 좌우로 움직입니다. 이 때 스테이지에 나타난 검은 선 역시 7조 센서가 선을 감지하는 상태에 따라 좌우로 움직일 수 있도록 하여 봅시다. 예를 들어 12번 센서가 선을 감지하였다면 스테이지에 있는 선도 가운데를 기준으로 해서 12번 센서의 위치만큼 왼쪽으로 이동할 수 있도록 합니다.



3. 스마트 인벤터 보드의 7조 센서 중 11번과 12번 그리고 17번과 18번을 사용하여 타이밍 게임을 만들어 봅시다. 스테이지의 스프라이트가 특정 번호를 부르면 7조 센서 중 그 번호에 맞는 센서를 빨리 감지시키는 게임입니다. 만약 1초 내에 감지시키지 못하면 실패했다는 메시지가 뜨도록 합시다.



스테이지의 스프라이트

## 2. 디바이스 제어하기

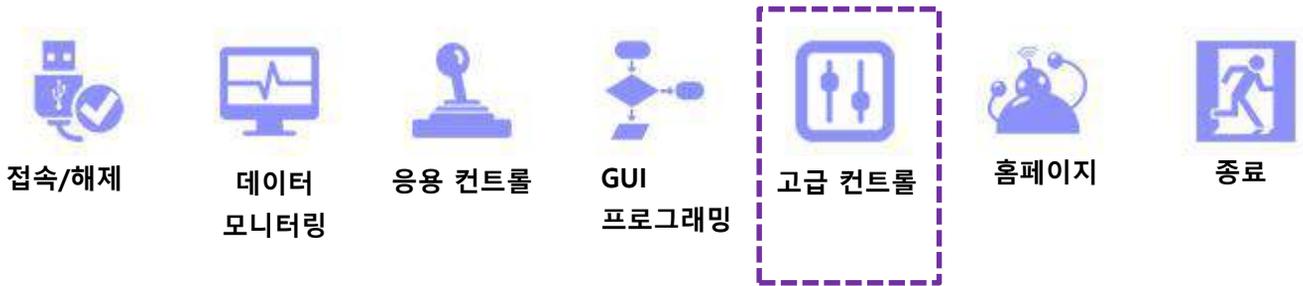
### 4장.

로킷 시뮬레이터

응용 시뮬레이터

# Rokit Simulator

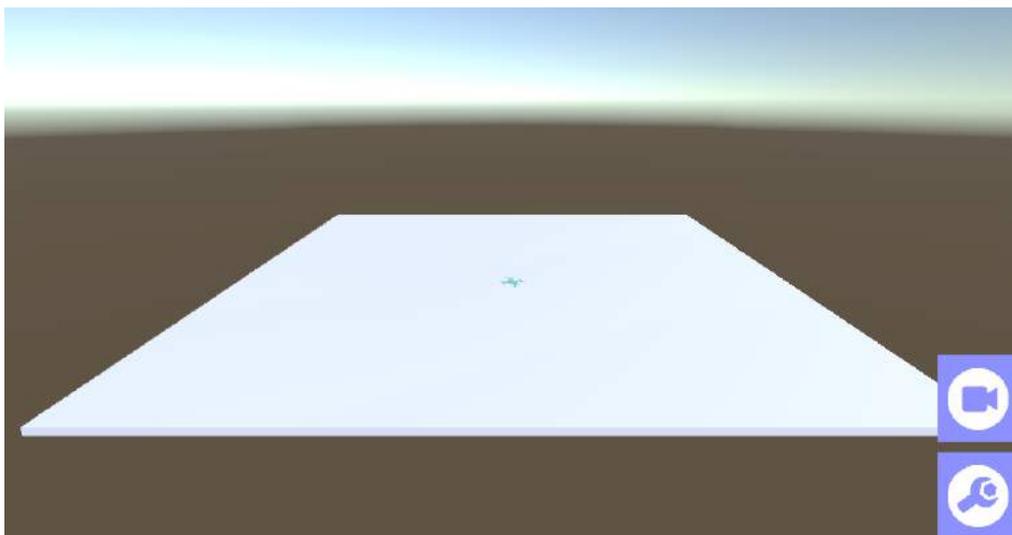
## 1.1 고급 컨트롤



### 고급 컨트롤

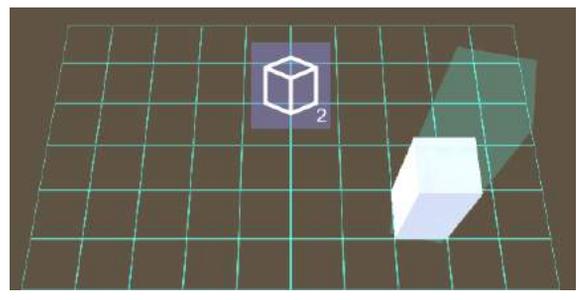
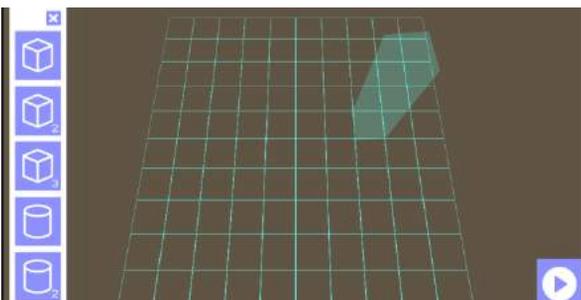
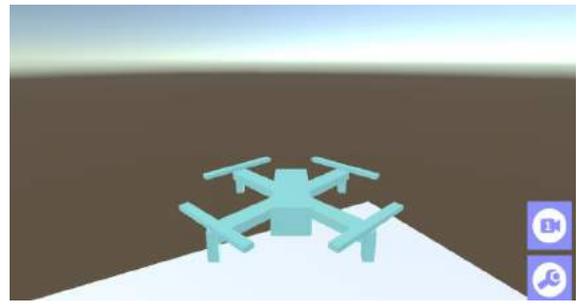
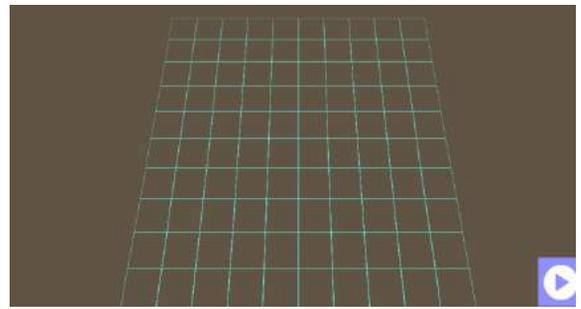
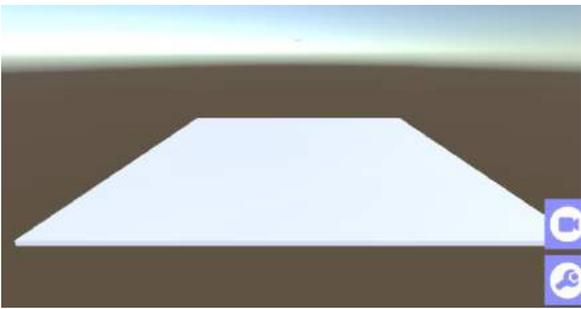
▶ 내가 만드는 시뮬레이션과 로봇 제어 환경

본인이 직접 환경을 만들고 인벤터 보드와의 구동방식을 설정해보는 로봇 시뮬레이션 환경



# Rokit Simulator

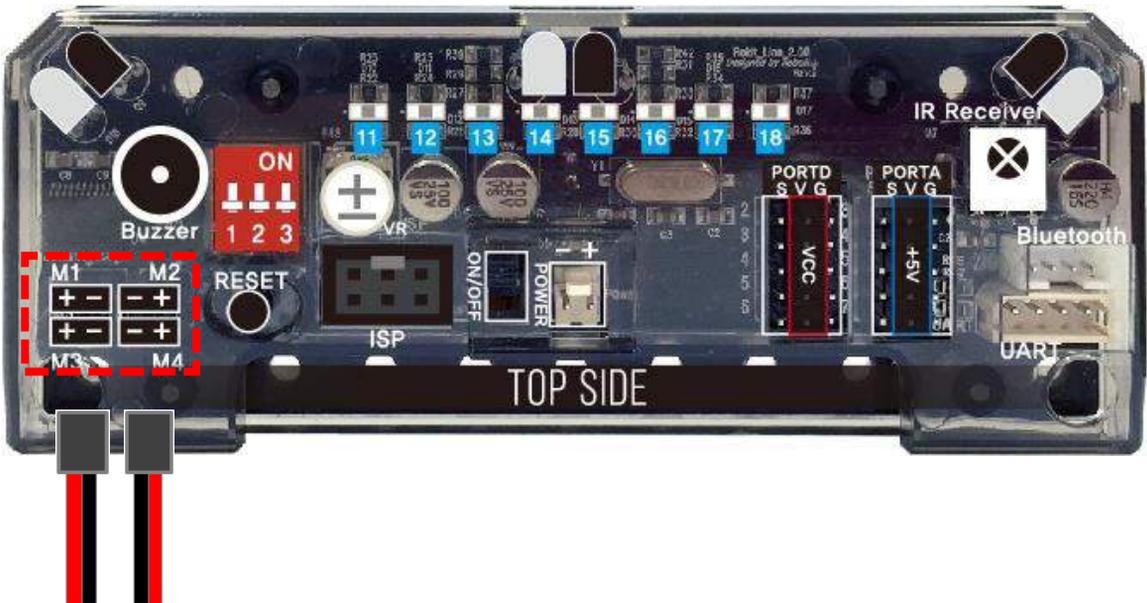
자신만의 시뮬레이터 환경을 만들어 봅시다.



## 2.1 DC모터

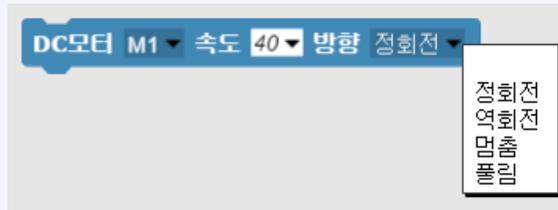
DC모터는 바퀴로 움직이는 로봇을 만들 때 필요한 모터입니다. 전원을 연결하면 360도로 계속 회전하며 DC모터에 바퀴 등을 달아서 로봇을 이동하게 할 수 있습니다.

## 2.2 DC모터의 연결

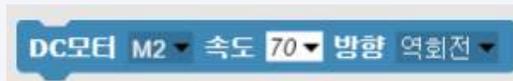


빨간색 사각형으로 표시된 부분이 DC모터가 연결되는 핀입니다. DC모터의 선을 이 부분에 연결합니다. 스마트 인벤터 보드는 2개의 DC모터를 제어할 수 있으며 따라서 핀 역시 M1과 M2 두 부분으로 이루어져 있습니다. DC모터 선을 연결할 때에는 +방향에 빨간색 선이 오도록 하면 됩니다. M3는 M1과 동일한 연결 핀으로서 독립적으로 제어되는 것이 아니라 M1의 모터 속도와 방향을 그대로 따릅니다. M4 역시도 M2와 동일한 연결 핀으로서 M2와 동일하게 움직입니다.

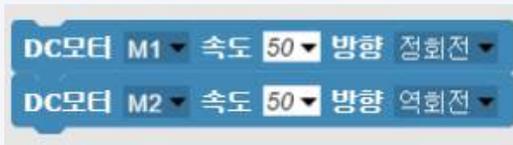
## 2.3 DC모터 블록 사용하기



DC모터 블록의 첫 번째 탭은 모터 번호, 두 번째는 모터의 속도, 마지막 탭은 모터의 동작을 결정합니다. 아래는 이미지는 여러가지 DC모터 블록의 사용법을 알려줍니다.



- M2핀에 연결된 DC모터를 속도 70, 방향은 역회전(시계 반대 방향)으로 움직입니다.



- M1핀과 M2핀에 연결된 두 개의 DC모터를 속도 50으로 동시에 움직입니다.  
단, 방향은 M1 모터가 정회전(시계 방향), M2 모터가 역회전(시계 반대 방향) 합니다.



- M1핀과 M2핀에 연결된 두 개의 DC모터를 속도 50, 각자 다른 방향으로 움직입니다.  
1초 동안 움직임을 유지하다가 동시에 두 모터가 멈추게 합니다.

### 3.1 예제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

**예제 1** : DC모터를 M1핀에 연결합니다. 스페이스 키를 누르면 1초 동안은 시계 방향(정회전)으로, 그리고 다음 1초 동안은 시계 반대 방향(역회전)으로 회전한 후 모터가 정지되도록 해봅시다.



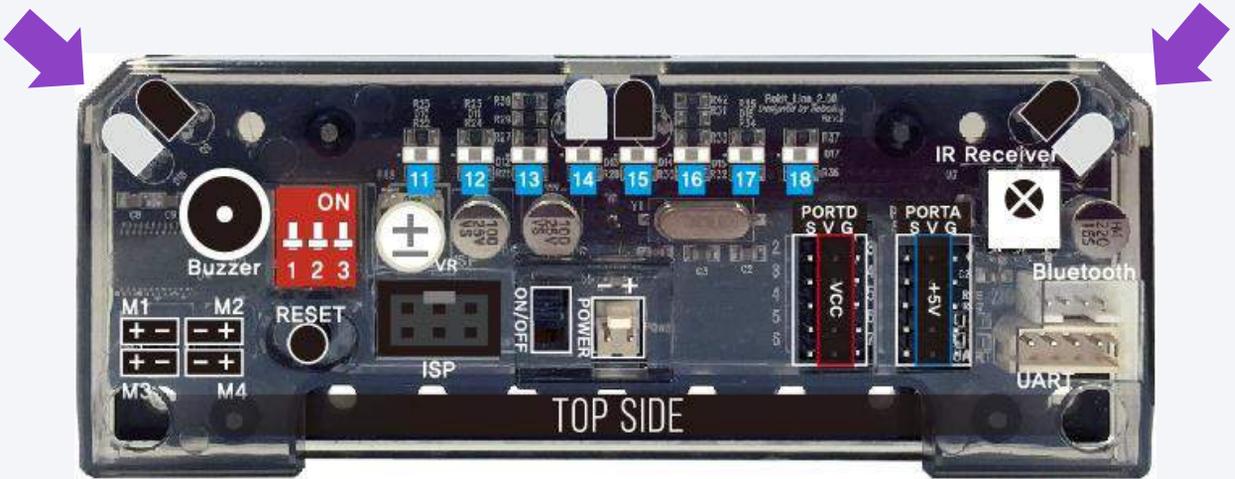
도움말: DC모터는 멈춤 블록으로 명령을 내려야만 멈추게 됩니다. 별도의 멈춤 명령이 없다면 스크립트 실행이 멈추었을 때, 심지어 보드와의 통신이 끊어진 경우에도 모터는 움직입니다. 이럴 경우에는 보드의 리셋 버튼을 누르면 모터가 멈춥니다.

**예제 2** : DC모터를 두 개 준비하여 하나는 M1핀에, 다른 하나는 M2에 연결합니다. 스크립트를 클릭하면 두 개의 모터 모두 1초 동안은 시계 방향으로 회전하며, 다음 1초 동안은 시계 반대 방향으로 회전한 후 멈출 수 있도록 스크립트를 작성해 봅시다.



도움말 : 멈춤 명령을 내린 경우 DC모터의 속도나 방향 설정은 중요하지 않습니다. 속도나 방향 설정과는 무관하게 멈춤이나 풀림은 동일한 효과를 가져옵니다.

**예제 3 :** DC모터 2개를 M1핀과 M2핀에 각각 연결합니다. 스마트 인벤터 보드의 앞쪽 세 개의 센서 중 왼쪽 센서를 감지하면 M1핀에 연결된 모터가 회전하고, 오른쪽 센서를 감지하면 M2핀에 연결된 모터가 회전하게 해 봅니다. 두 센서를 동시에 감지하면 두 개의 모터가 함께 회전하여야 합니다.



무한 반복하기

만약 **아날로그 값 읽기 19** < 100 라면  
**DC모터 M1** 속도 50 방향 정회전  
 아니면  
**DC모터 M1** 속도 50 방향 풀림

만약 **아날로그 값 읽기 21** < 100 라면  
**DC모터 M2** 속도 50 방향 정회전  
 아니면  
**DC모터 M2** 속도 50 방향 풀림

도움말 : '만약 ~ 이면' 블록을 사용하여 두 모터가 독립적으로 움직일 수 있도록 하였습니다. 따라서 두 센서를 한꺼번에 감지했을 경우 두 모터가 동시에 움직이려면 두 DC모터는 독립적으로 제어될 수 있어야 합니다.

## 3.2 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

1. DC모터를 M1핀에 연결한 후, 키보드의 위쪽 화살표 키를 누르면 모터의 속도가 서서히 증가하고 아래쪽 화살표 키를 누르면 속도가 천천히 내려갈 수 있도록 스크립트를 작성해 보세요.
  
2. 아래 그림과 같이 스프라이트에는 슬라이드 바가 있습니다. 슬라이더 바의 움직임에 따라 M1핀에 연결된 DC모터의 속도가 변화합니다.
  - 1) 슬라이더 바의 레버는 마우스에 따라 움직이는 데, 마우스가 양 끝으로 가게 되면 더 이상 따라오지 않고 제자리에서 멈춰있어야 하며 이 때 DC모터 역시 멈춰야 합니다.
  - 2) 레버가 중간에 위치할 때에도 DC모터는 멈춰야 합니다.
  - 3) 레버가 마우스의 움직임에 따라 왼쪽으로 움직일 때에는 모터가 시계 방향(정회전)으로 회전하며 속도는 레버의 움직인 양에 비례해서 빨라져야 합니다.
  - 4) 레버가 마우스의 움직임에 따라 오른쪽으로 움직일 때에는 모터는 시계 반대 방향(역회전)으로 회전하며 속도 역시 레버의 움직인 양에 비례해서 빨라져야 합니다.

이처럼 레버를 좌우로 조종하여 모터의 속도의 조절하는 패널을 만들어 봅시다.



3. DC모터를 M1핀에 연결합니다. 이 후 스크립트를 실행하면 모터의 속도와 방향을 묻는 질문과 입력창이 나옵니다. 입력 창에 입력 한 속도와 방향대로 움직일 수 있도록 스크립트를 작성해 봅시다.

속도는 0에서 100까지의 범위로 하며 입력받은 후에는 입력받은 것이 문자인지 숫자인지 구분될 수 있어야 합니다.

방향을 입력받을 때에는 네 가지(cw, ccw, stop, loose)로 입력받도록 하고 다른 문자나 숫자를 입력받았을 때에는 무시하도록 합니다.

### 3. 로봇 제어하기

## 0장.

# 로봇이란?

# 로봇의 기초

## 1. 로봇이란?

체코어 "Robota(일한다)"는 단어에서 나왔으며, 사람을 대신해 기능을 수행하는 기계를 말하며, 최근에는 사람에게 즐거움을 주는 로봇들도 지칭합니다.

## 2. 로봇 교육

어린이부터 청소년, 성인에게까지도 친근하고 호기심이 있는 과학 분야인 로봇에 대해 쉽게 다가가고 전자, 기계, 컴퓨터 등을 통해 로봇을 제어할 수 있을 뿐만 아니라, 소프트웨어 코딩 교육을 통해 구현되는 로봇 교육을 포함 합니다.

## 3. 로봇 교육 (Software Coding Education) 의 효과

로봇기술은 미래의 첨단 산업기술의 기초이자, 첨단 과학이므로, 앞으로 어린이들에게 필요한 교육이며, 로봇제작을 통해 (문제 해결을 위한) 사고력, (무에서 유를 만드는)창의력, (자신만의 것으로 활용 가능한)응용력 을 얻을 수 있습니다.

# 로봇의 기초

## 4. 왜 로봇을 만들까?

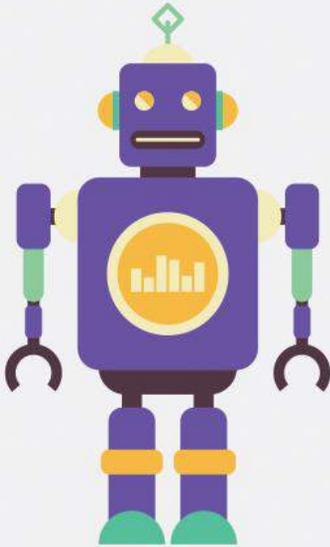
로봇은 인간의 모든 부분에 있어서 편리하게 하고 삶을 윤택하게 만들어 줍니다.



# 로봇의 기초

## 5. 로봇의 구조

인간처럼 몸과 두뇌, 감각 등을 지니고 있습니다.



두뇌 : 메인보드(컴퓨터) = 머리

몸체 : 프레임 = 사람의 뼈대

감각 : 센서 = 눈,코,입,귀 등

관절 : 모터 = 사람의 관절

힘 : 건전지 = 음식(밥)

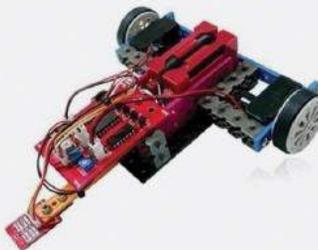
## 6. 로봇의 종류

다양한 분류 방법이 있지만 크게는 로봇의 움직임에 따라 분류합니다.

보행로봇



바퀴로봇



기구로봇



### 3. 로봇 제어하기

## 1장.

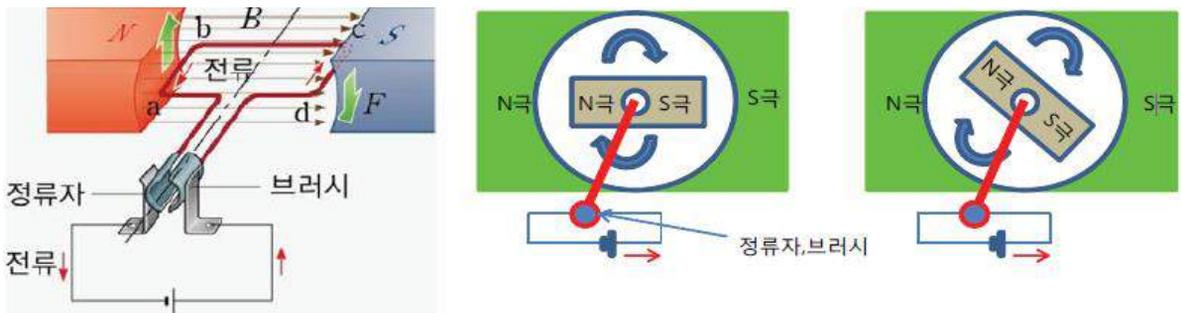
# 지능 로봇 조립

## 1.1 DC모터란?

모터는 전기를 이용하여 회전운동을 얻어내는 대표적인 동력장치입니다.



우리가 사용하는 모터의 내부에는 영구자석, 코일과 같은 부품이 들어있으며, 외부에서 들어오는 전기의 방향과 자기장의 방향에 따라 모터가 회전하는 방향이 결정됩니다.



모터는 자석의 N극과 S극이 서로 당기고 "N극과 N극", "S극과 S극" 같이 같은 극끼리는 밀어내는 원리를 이용하여 회전합니다.



특히, DC모터는 360도 회전하므로, 바퀴 구동형 로봇을 만들 때 적합한 모터입니다.

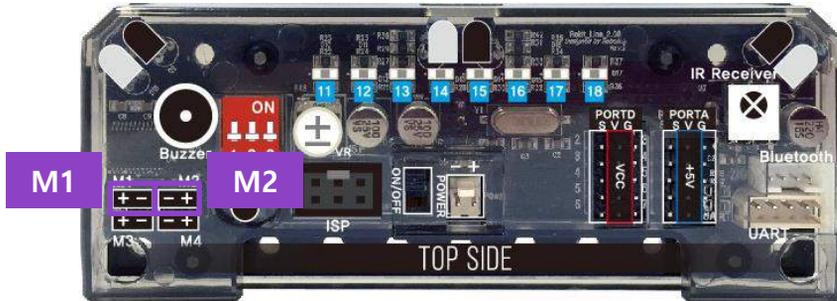
2개의 모터를 사용하여 2륜 구동 로봇을 만들거나 4개의 모터를 사용하여 4륜 구동 로봇을 만드는데 사용합니다.

## 1.2 DC모터 블록 사용하기

DC모터 블록은 다음과 같이 세 가지의 값을 입력하도록 되어 있습니다.



### 1. 모터 선택 : 어떤 모터를?



스마트 보드는 두 개의 모터를 제어 할 수 있습니다. 두 개의 모터를 각각 M1과 M2 라고 합니다. 움직이고 싶은 모터를 연결하고, 알맞은 값을 선택하도록 합니다.

### 2. 모터 속도 : 회전 속도는?

모터의 속도를 정합니다. 모터의 속도는 0부터 100까지 입니다. 숫자가 클수록 모터의 속도가 빨라집니다.

### 3. 모터 속도 : 어떤 동작을?

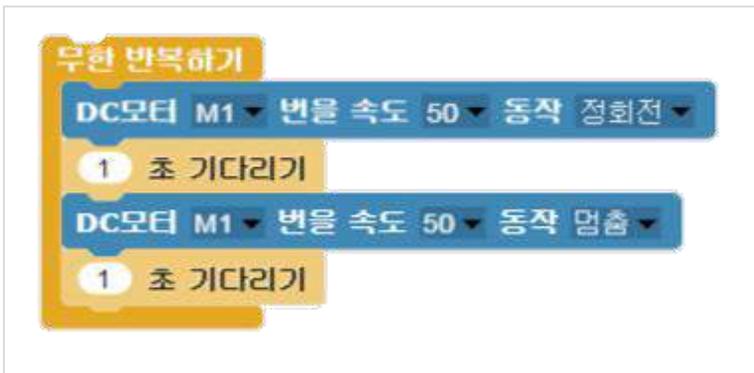
모터의 동작을 정합니다. DC모터는 다음과 같은 동작을 합니다.

- 정회전 : 모터를 시계 방향으로 회전합니다.
- 역회전 : 모터를 시계 반대 방향으로 회전합니다.
- 멈춤 : 모터를 정지시킵니다. (자전거의 브레이크를 밟듯이 정지)
- 풀림 : 모터를 정지시킵니다. (자전거의 페달을 놓듯이 정지)

## 1.2 DC모터 블록 사용하기



다음의 예제로 모터의 동작을 살펴봅시다.

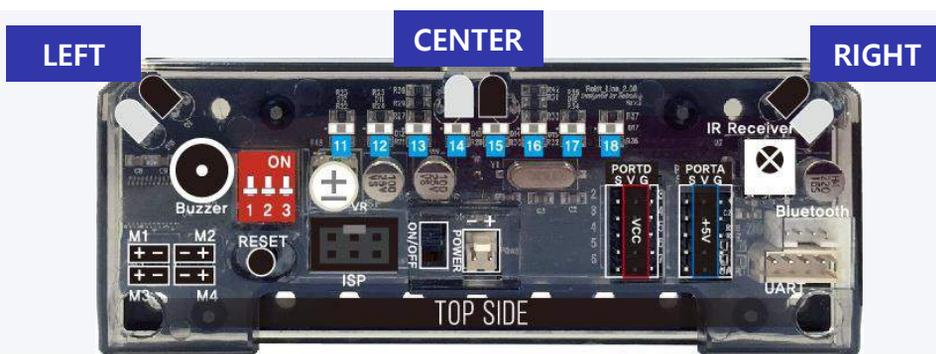


M1에 해당하는 모터를 50의 속도로 시계 방향으로 회전하고, 1초를 기다린 후에, 멈추고 다시 1초를 기다리는 것을 반복하는 예제입니다.

## 1.3 DC모터 속도와 방향을 센서로 제어하기



스크립트를 작성하기 전에 먼저 센서 입력 값을 저장할 3개의 변수 블록을 미리 만들어 놓도록 합니다.



변수는 각각 다음의 위치에 해당하는 센서값을 저장하게 됩니다.

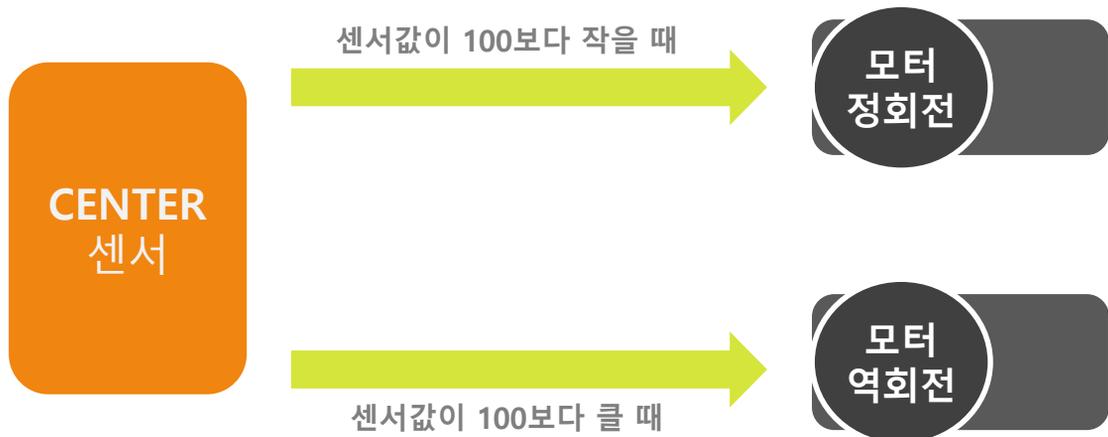
## 1.3 DC모터 속도와 방향을 센서로 제어하기

### (1) 센서로 모터 방향 제어하기



보드의 적외선 센서는 감지가 되면 값이 작아지게 됩니다.

센서에 물체가 감지되면 센서값이 100보다 작아지면, M1 모터가 정회전을 하게 됩니다. 반대로, 센서가 감지가 되지 않을 때는, M1 모터가 역회전을 하게 됩니다.



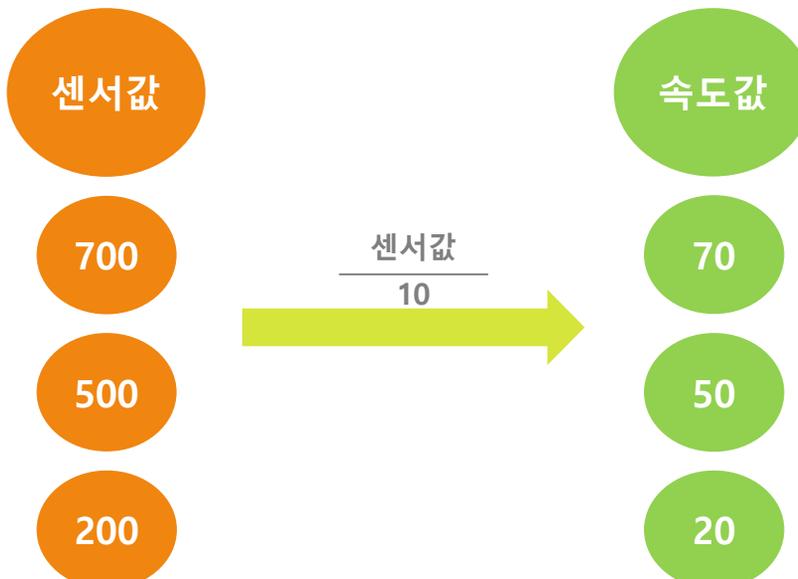
## 1.3 DC모터 속도와 방향을 센서로 제어하기

### (2) 센서로 모터 속도 제어하기



센서값으로 모터의 속도를 제어합니다. 센서값은 0부터 1023까지 입력이 되며, 속도 값은 0부터 100까지 입력이 가능하므로, 센서값과 속도 값의 범위를 맞춰 주기 위해서 센서값을 10으로 나누어서 그 값을 속도에 입력합니다.

LEFT 센서가 감지 되면 M1 모터의 속도가 줄어들게 되고, RIGHT 센서가 감지되면 M2 모터의 속도가 줄어들게 됩니다.



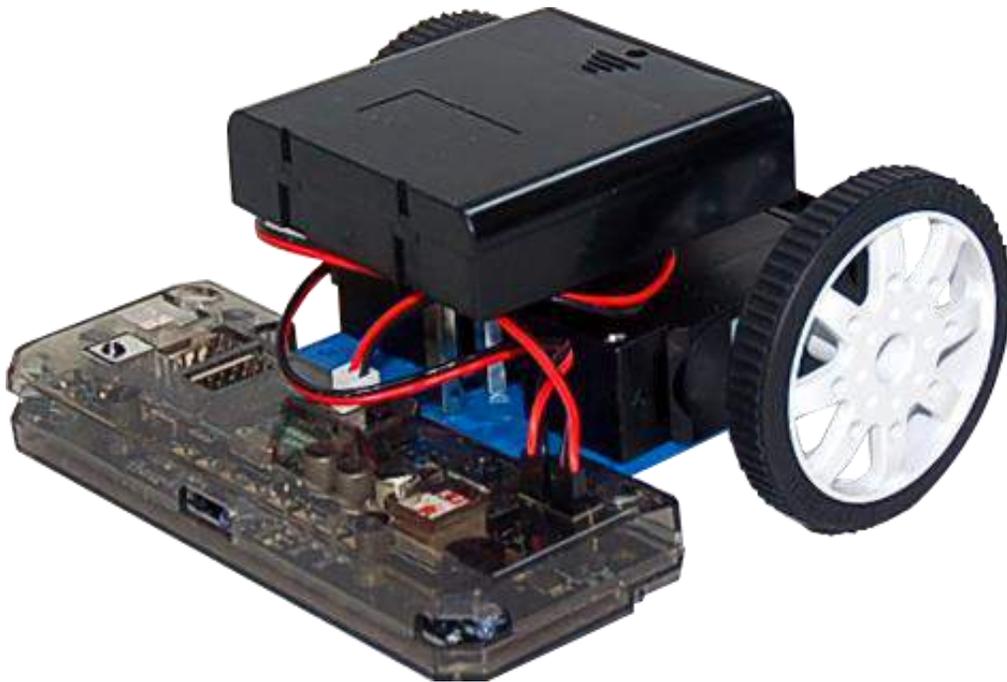
## 2.1 지능형 로봇이란?

어떤 로봇을 지능형 로봇이라고 부를 수 있을까요?

단순한 조립, 용접, 운반 등의 노동을 수행하는 (반)자동화된 로봇이 아닌, 주변 환경을 스스로 인식한 후 자신의 행동을 조절하고 결정하는 로봇을 '지능형 로봇'이라고 부릅니다.

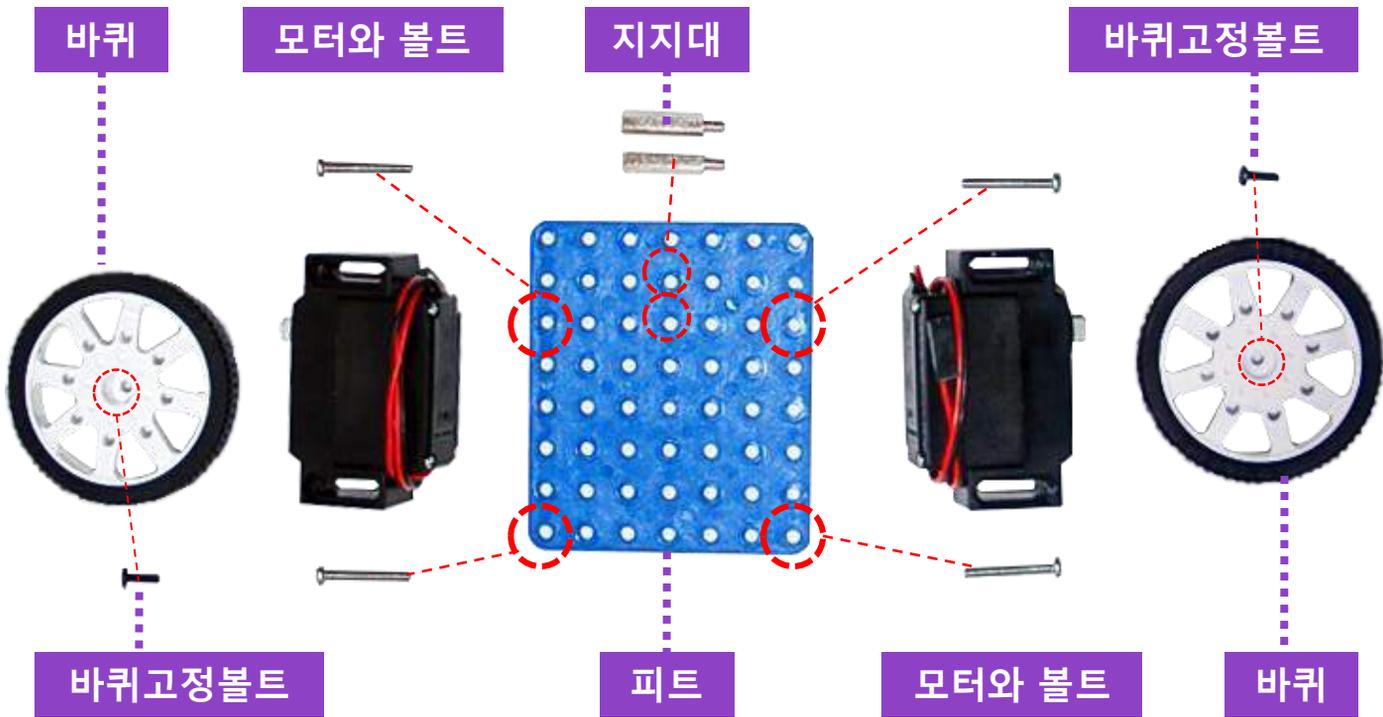
지능형을 결정하는 요소는 인간과 비교하여 로봇이 가질 수 있는 센서와 모듈을 이용하여 프로그램을 어떻게 적용하느냐에 결정됩니다.

여러분이 직접 지능형 로봇을 프로그래밍 해 보세요.

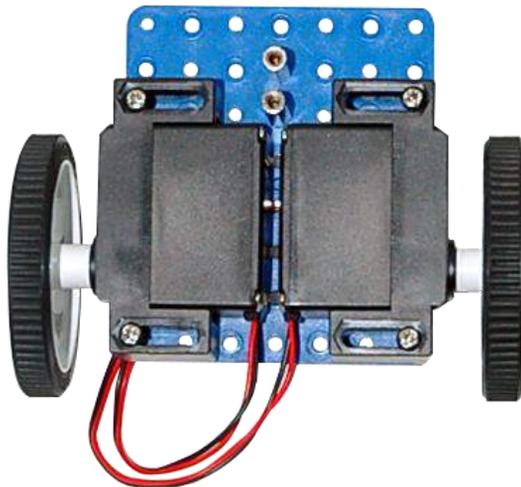


## 2.2 지능형 로봇 조립(1)

 : 볼트와 너트를 이용하여 조립합니다.



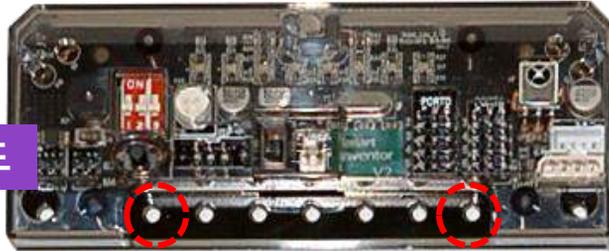
가로(7홀)와 세로(8홀)  
방향에 유의합니다.



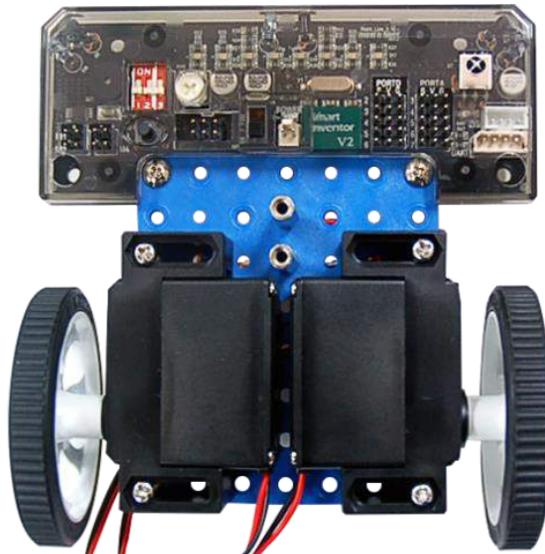
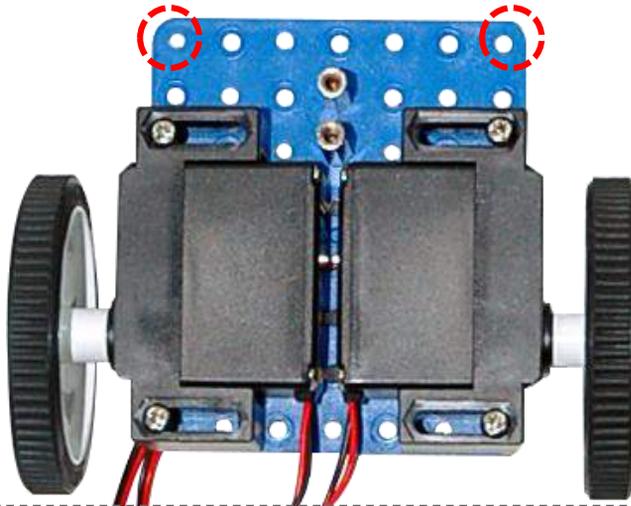
## 2.2 지능형 로봇 조립(2)

 : 긴볼트와 너트를 이용하여 인벤터 보드를 조립합니다.

스마트 인벤터 보드



긴볼트

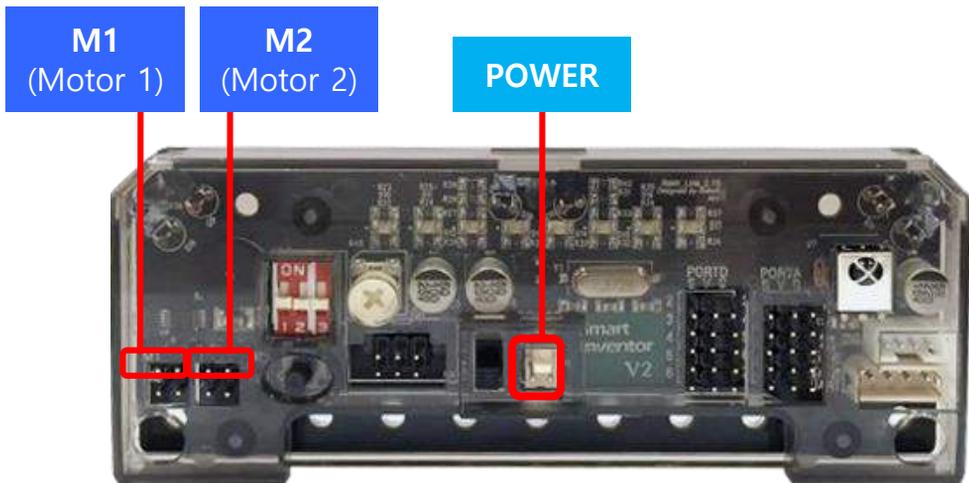
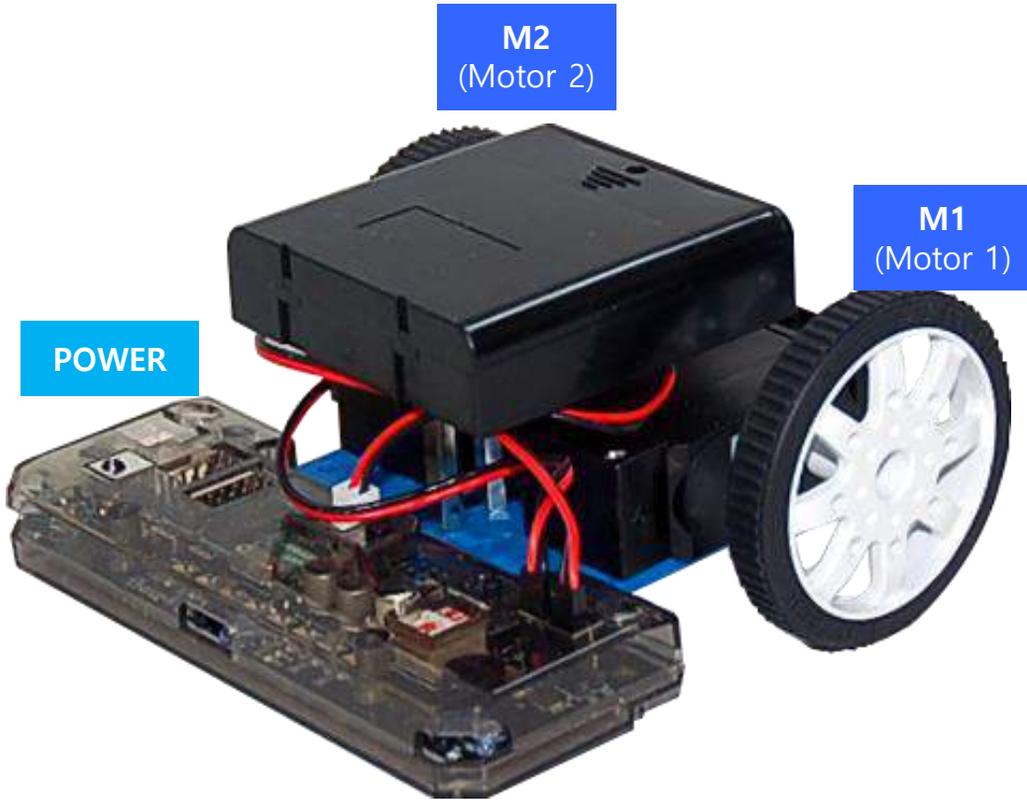


## 2.2 지능형 로봇 조립(3)

 : 짧은볼트와 너트를 이용하여 건전지케이스를 조립합니다.



## 2.2 지능형 로봇 조립(4-배선)



### 3.1 지능형 로봇 구동하기 (내장 프로그램 사용하기)

조립한 로봇의 전원을 켜고 손으로 다양하게 구동해봅시다.  
 완성된 로봇에 입력하는 프로그램 모드에 따라 각각 다른 로봇으로 구동이 가능합니다.



**주의**  
 DIP 스위치가 모두 OFF되어야 합니다.

프로그램 모드는 아래와 같습니다.

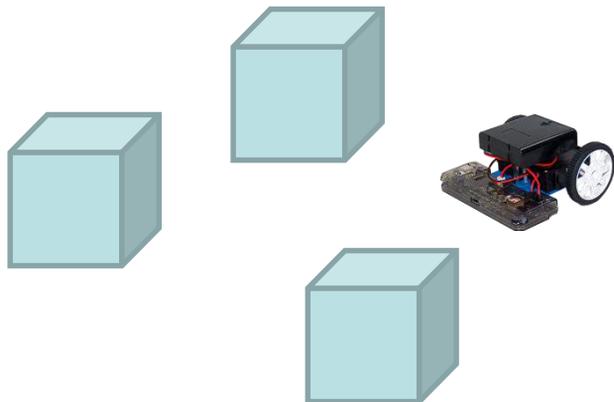
**어보이더**  
 (프로그램 1번 모드)

**바닥감지로봇**  
 (프로그램 2번 모드)

**라인트레이서**  
 (프로그램 12번 모드)

#### 1. 어보이더 구동 : 장애물을 두고 로봇이 정상적으로 회피하는지 확인합니다.

**어보이더**  
 (프로그램 1번 모드)

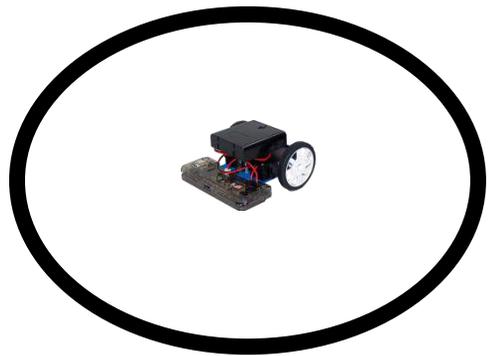
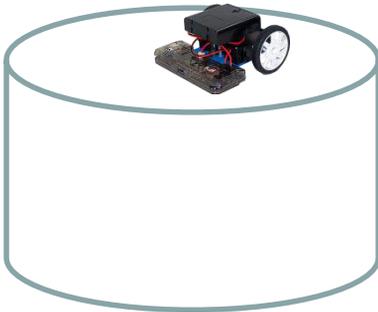


## 2. 바닥감지로봇 구동

: 흰색의 바닥이나 책상 위에 검은색의 테이프를 붙여 로봇이 움직이는 것을 확인합니다.



바닥감지로봇  
(프로그램 2번 모드)

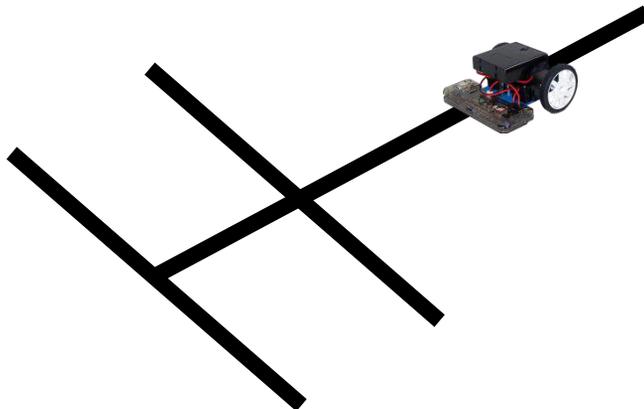


## 3. 라인트레이서 구동

: 흰색의 바닥이나 책상 위에 검은색의 테이프를 붙여 로봇이 라인을 따라 움직이는지 확인합니다.



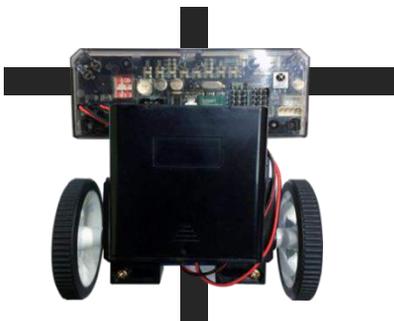
라인트레이서  
(프로그램 12번 모드)



### 3. 라인트레이서 감도 조절하기

라인트레이서의 작동이 시작되면 우선 **감도 체크 모드**로 들어갑니다.

이 모드에서는 라인트레이서를 검은 선 위에 올려두면 검은 선이 감지되는 부분의 LED만 꺼지게 됩니다. 만약 선에 올려놓았을 때 모든 LED가 꺼지거나 켜진다면 감도에 문제가 있는 것으로, **가변저항**을 통해 감도 조절을 하게 됩니다. 감도 조절 후에는 중앙 센서를 감지시켜 줍니다.



### 3. 라인트레이서 프로그램하기

감도 조절이 끝난 후 프로그래밍 모드로 들어가게 됩니다. 아래 설명을 보고 직접 라인을 따라가게 프로그래밍 해봅시다. 입력을 완료했다면, 중앙의 센서를 감지시켜 프로그램을 실행해 봅니다.



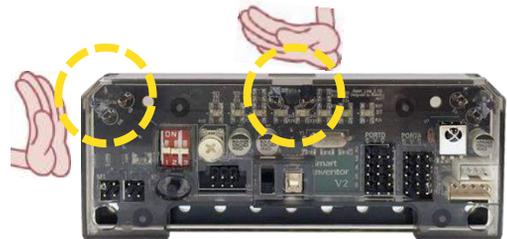
교차로에서 왼쪽으로



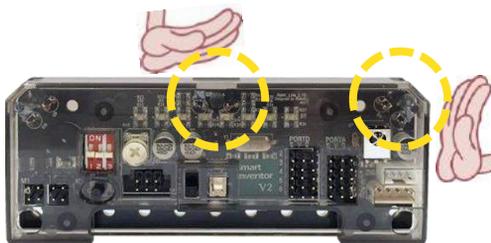
교차로에서 오른쪽으로



교차로를 지나 앞으로 직진



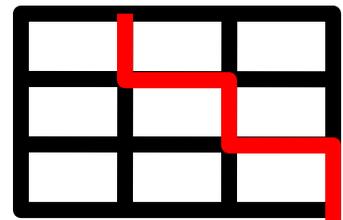
교차로에서 유턴



한 단계 실행 취소



구동 시작

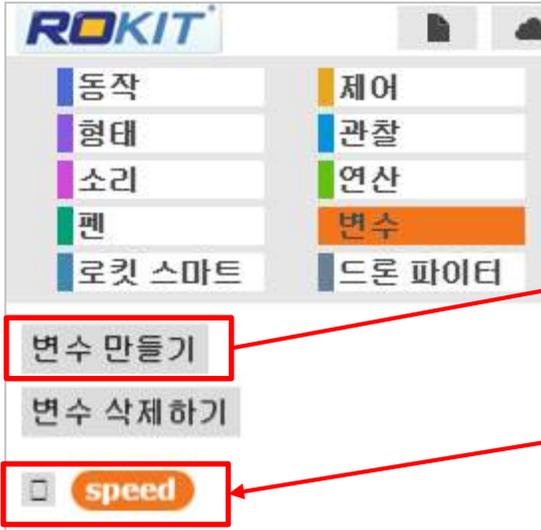


START ▲

제시된 순서대로 프로그램을 입력하면  
로봇이 어떻게 움직이는지 확인해 봅시다.

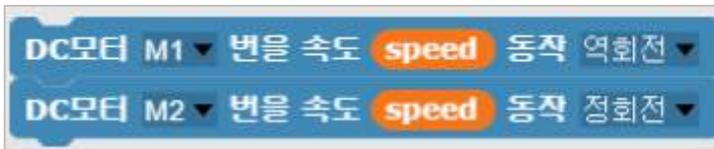
## 4.1 키보드를 활용한 구동 방법

### 1-1. 변수 만들기



변수 만들기를 사용하여  speed 수를 만들어 줍니다.

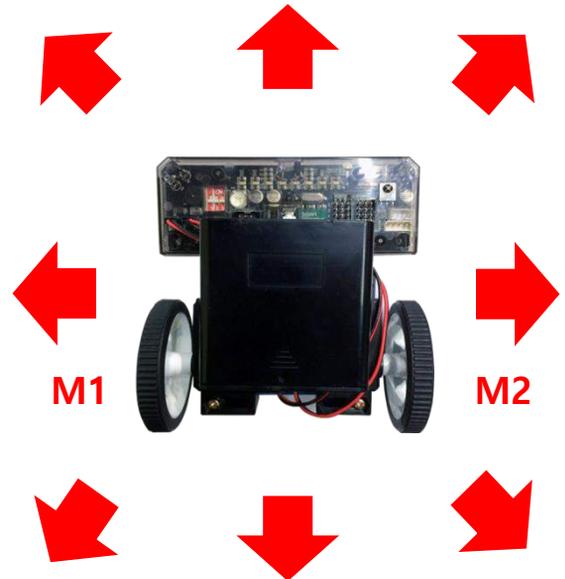
### 1-2. 모터의 회전 조건



모터의 기준에서 시계 반대 방향으로 돌아가는 것을 역회전



모터의 기준에서 시계 방향으로 돌아가는 것을 정회전



이러한 기준으로 모터의 전진, 후진, 좌회전, 우회전을 만들 수 있습니다.

### 1-3. 스크립트 만들기



Speed를 80으로 넣어줌 (숫자 크기에 속도변화)

조건문을 무한반복하기

조건1> 위쪽 화살표 M1:역회전,M2:정회전 전진

조건2>아래쪽 화살표 M1:정회전,M2:역회전 후진

조건3>왼쪽 화살표 M1:정회전,M2:정회전 좌회전

조건4>오른쪽 화살표 M1:역회전, M2:역회전 우회전

조건이 아닐 때 (키를 안 누를 때) 정지

※ 로봇과 연결하여 어떻게 구동이 되는지 확인하여 봅시다. ※

## 4.2 마우스를 활용한 구동 방법

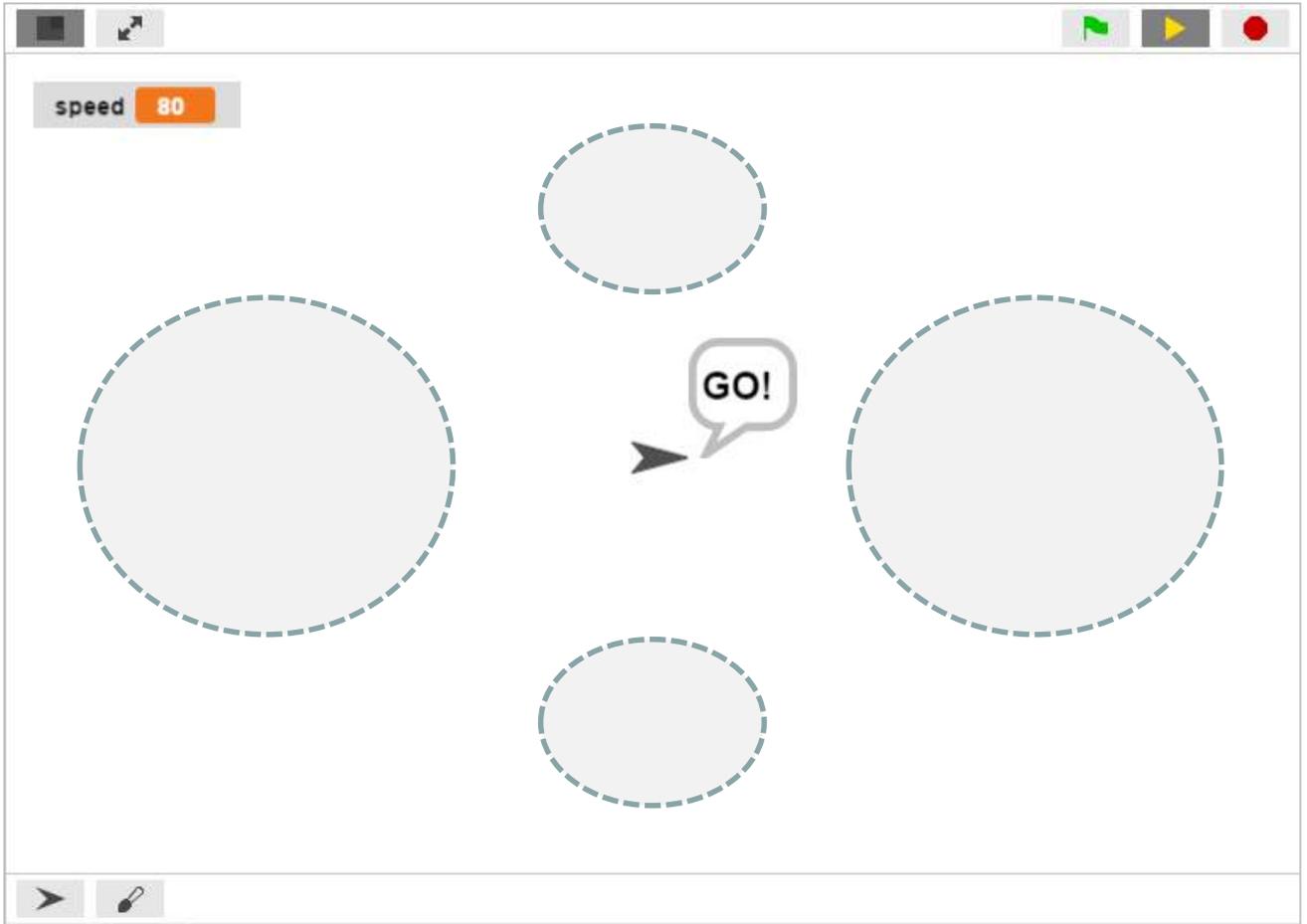
### 1-1. 스크립트 만들기

The image shows a Scratch script for controlling a robot's movement based on mouse clicks and coordinates. The script is annotated with red boxes and arrows explaining its components:

- Speed 설정:** A block to set the variable 'speed' to 80 is annotated with "Speed를 80으로 넣어줌 (숫자 크기에 속도변화)".
- 무한 반복하기:** A 'Repeat Forever' loop block is annotated with "조건문을 무한반복하기".
- 조건문 1:** An 'If Mouse Clicked?' block is annotated with "조건문1 전진".
- 조건문 2:** An 'If Mouse Y Coordinate > 50?' block is annotated with "조건1 전진".
- 조건문 3:** An 'If Mouse Y Coordinate < -50?' block is annotated with "조건2 후진".
- 조건문 4:** An 'If Mouse X Coordinate > 50?' block is annotated with "조건3 우회전".
- 조건문 5:** An 'If Mouse X Coordinate < -50?' block is annotated with "조건4 좌회전".
- 조건이 아닐 때:** An 'Else' block is annotated with "조건이 아닐 때 정지".

Visual feedback for each condition is shown in speech bubbles: "GO!", "BACK!", "RIGHT!", and "LEFT!".

## 1-2. 로봇 조종하기



Rokit brick의 무대에서 x축(가로), y축(세로)을 마우스로 클릭하면 전, 후, 좌, 우로 로봇이 구동합니다.

## 5.1 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

1. 순차적으로 출발지점부터 목표지점까지 자동으로 도착할 수 있도록 스크립트를 작성해 보세요.

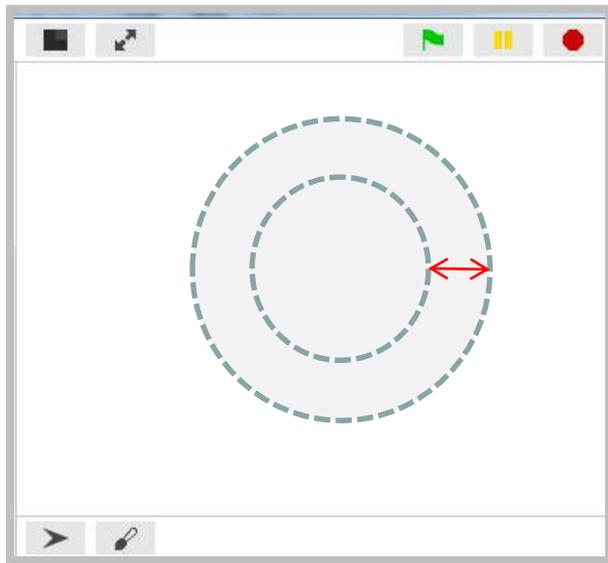
1) 직선 코스



2) 직진후 방향 이동 코스



2. Rokit brick의 무대에서 원의 지름을 조절하여 지름의 변화만큼 로봇이 원을 그리는 스크립트를 작성해보세요.



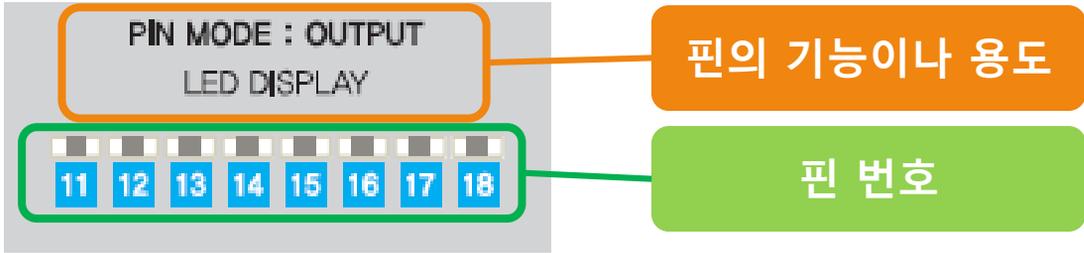
### 3. 로봇 제어하기

## 2장.

# 어보이더

## 1.1 핀맵 사용하기

핀맵에는 다음과 같은 정보가 표시됩니다.  
 (핀맵의 정보는 40쪽 2.디바이스제어하기 1장 참조)



핀 번호는 코딩 중에 해당하는 핀을 제어해야 하는 경우 사용됩니다.



핀의 용도를 보면 11번부터 18번까지는 LED DISPLAY 라고 적혀있습니다. 따라서 LED를 켜고 끌 수 있는 용도라는 것을 알 수 있습니다.



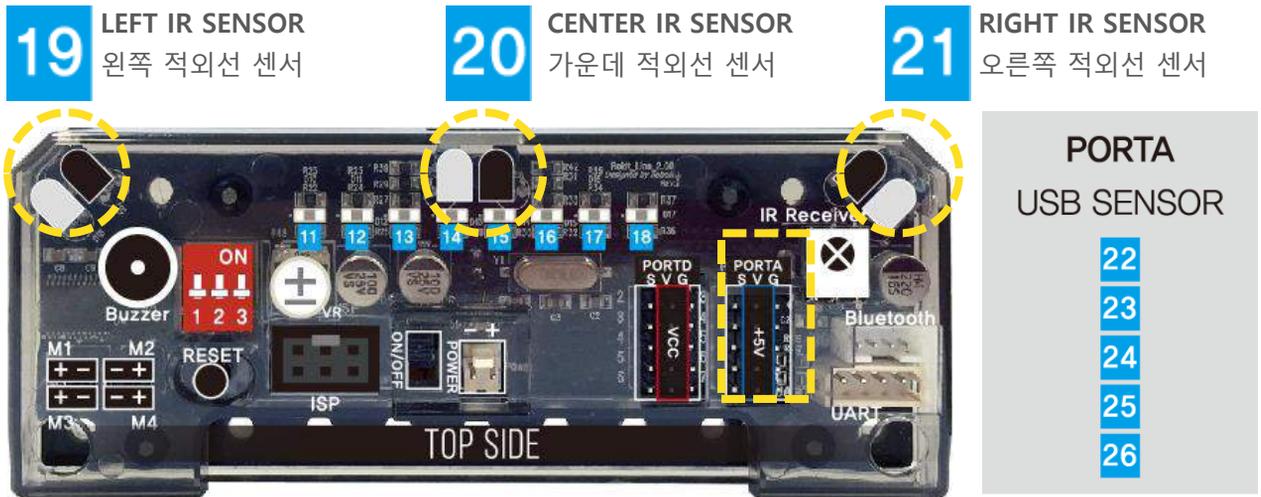
## 2.1 아날로그 블록 사용하기

아날로그 블록은 아날로그 핀 번호를 입력하도록 합니다.

아날로그 값 읽기 19

### 아날로그 핀 번호 입력

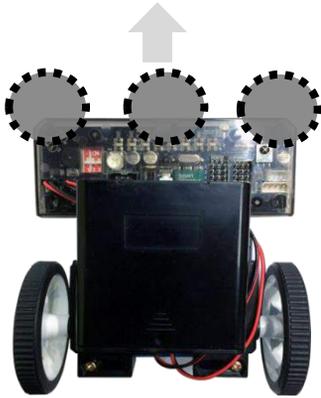
아날로그 값을 읽기 위해서는 아날로그 값을 읽어올 수 있는 핀을 선택하여야 합니다.  
모든 핀이 아날로그 값을 읽을 수 있는 것은 아닙니다.



19번부터 26번까지 총 8개의 핀 만이 아날로그 값을 읽어오는 기능을 가지고 있습니다.  
보드의 전방에 위치한 세 개의 적외선(IR) 센서와 제어기 오른쪽의 PORTA라고 표기된 부분  
입니다.

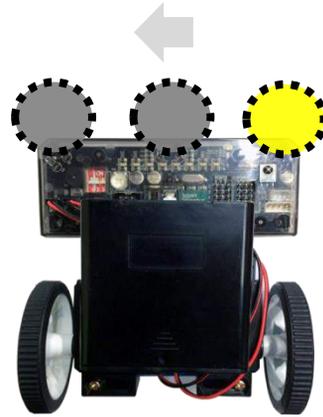
### 3.1 어보이더의 구동 원리

장애물을 피하는 방향으로 움직입니다.



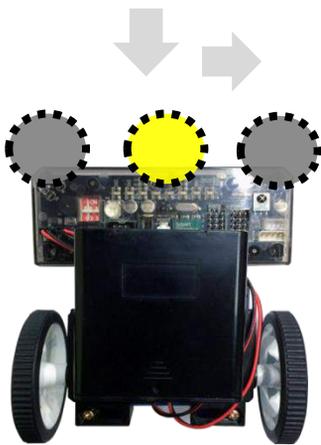
두개의 센서에 입력이 없을 때  
(장애물이 없을 때)

**전진**



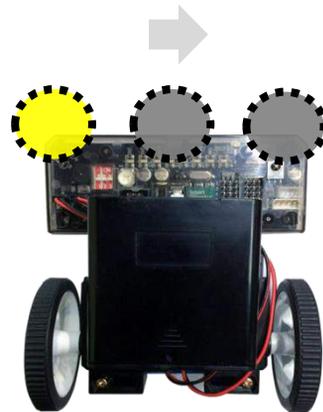
오른쪽 센서에  
입력이 들어 올 때

**좌회전**



가운데 센서에 입력이 들어 올 때

**후진후 우회전**

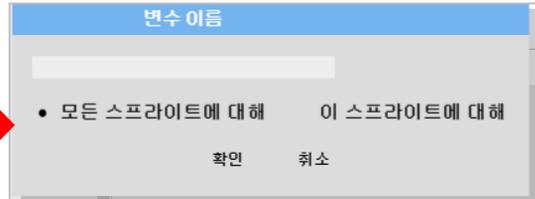


왼쪽 센서에 입력이 들어 올 때

**우회전**

## 4.1 프로그램 설명 (1)

### 1-1. 변수 설명



변수 만들기를 사용하여 5개의 변수 블록을 만들어 줍니다.

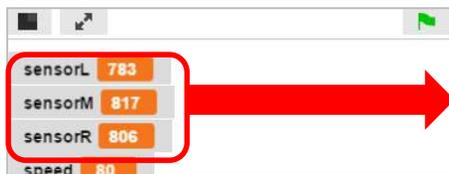
1. sensorL - 왼쪽 센서의 아날로그값
2. sensorM - 중앙 센서의 아날로그값
3. sensorR - 오른쪽 센서의 아날로그값
4. speed - 모든 모터의 속도
5. limit - 기준이 되는 값

### 1-2. 해당 변수의 아날로그값

만들어진 sensor 변수에는 앞서 배웠던 보드 전방의 IR센서 보정값(0 - 1023)이 들어갑니다.



이러한 값이 들어가려면 각 센서의 해당 핀 번호가 입력되어야만 메인보드로 입력되는 값이 실시간으로 감지되어 변수값으로 전달됩니다.



실시간 아날로그값

## 4.2 프로그램 설명 (2)

### 변수 선언

반복

변수 limit 에 200 저장하기  
 변수 speed 에 80 저장하기

변수 speed와 limit는 스크립트가 시작할 때 각각 특정한 값을 가져야 합니다. 이것을 초기값이라 합니다. 모터 속도를 제어하는 값인 speed는 초기값이 80입니다. 또한 센서의 감지 여부를 결정하는 값이 들어 있는 변수 limit의 초기값은 200입니다.

아날로그 값

변수 sensorL 에 아날로그 값 읽기 19 저장하기  
 변수 sensorM 에 아날로그 값 읽기 20 저장하기  
 변수 sensorR 에 아날로그 값 읽기 21 저장하기

앞부분 세 개의 센서값을 읽어서 각각의 변수에 저장합니다. 이 스크립트들이 무한 반복하기 블록 안에 들어 있다는 점을 유의해 주세요.

조건 1

만약 sensorL < limit 라면  
 DC모터 M1 속도 speed 방향 역회전  
 DC모터 M2 속도 speed 방향 역회전  
 0.5 초 기다리기

왼쪽 센서 값이 변수 limit 보다 작으면 물체가 감지된 것으로 판단하고 모터를 움직여서 오른쪽으로 회전합니다. 이 때 모터의 회전방향은 둘 다 역회전이어야 오른쪽으로 회전할 수 있습니다.

조건 2

만약 sensorM < limit 라면  
 DC모터 M1 속도 speed 방향 정회전  
 DC모터 M2 속도 speed 방향 역회전  
 0.5 초 기다리기  
 DC모터 M1 속도 speed 방향 역회전  
 DC모터 M2 속도 speed 방향 역회전  
 0.5 초 기다리기

가운데 센서 값이 변수 limit 보다 작으면 물체가 감지된 것으로 판단하고 모터를 움직여서 0.5초 정도 후진한 후 오른쪽으로 회전합니다. 이렇게 후진한 다음 회전하면 더욱 안정적으로 물체를 피할 수 있습니다.

조건 3

만약 sensorR < limit 라면  
 DC모터 M1 속도 speed 방향 정회전  
 DC모터 M2 속도 speed 방향 정회전  
 0.5 초 기다리기

오른쪽 센서 값이 변수 limit 보다 작으면 물체가 감지된 것으로 판단하고 모터를 움직여서 왼쪽으로 회전합니다. 이 때 모터의 회전방향은 둘 다 정회전이어야 왼쪽으로 회전합니다.

조건 아니면

DC모터 M1 속도 speed 방향 역회전  
 DC모터 M2 속도 speed 방향 정회전

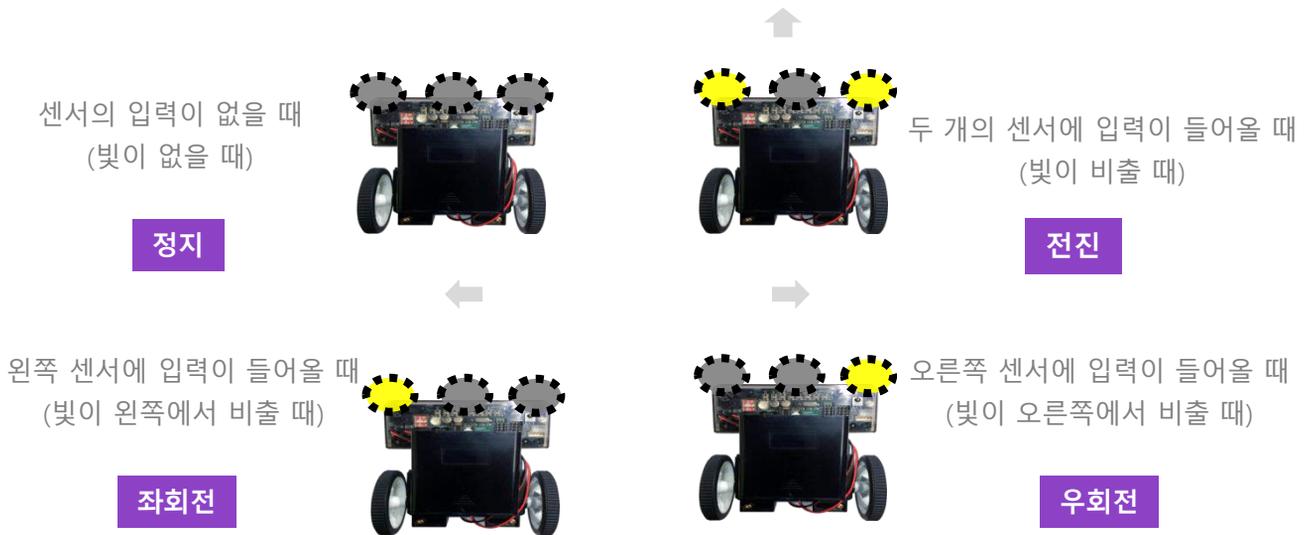
3개 중 어떤 센서도 감지되지 않은 경우에는 로봇은 전진합니다. 두 개의 모터 중 왼쪽(1번) 모터는 역회전, 오른쪽(2번) 모터는 정회전할 때 전진하게 됩니다.

## 5.1 연습문제

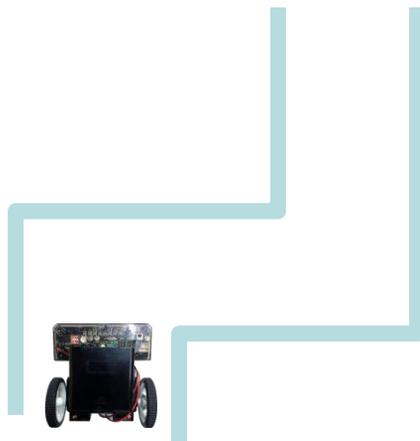
예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
**DOWNLOADS**를 확인해주세요. 자세한 내용은 **교재 16페이지**▶

1. 다음과 같이 빛을 따라오는 로봇 응용 스크립트를 작성해보세요.

센서로 입력된 빛을 따라가며 움직입니다.



2. 다음과 같은 미로를 탈출하기 위한 어보이더 응용 스크립트를 작성해보세요



### 3. 로봇 제어하기

## 3장.

# 바닥감지로봇

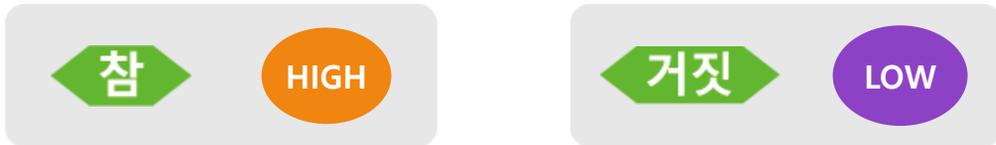
## 1.1 디지털 핀 출력에 대하여

디지털 핀은 입력과 출력의 기능을 사용할 수 있습니다. 출력이라는 말에서 알 수 있듯이 전기적인 신호 값을 내보내는 것을 말합니다.



디지털 핀의 출력은 HIGH 와 LOW 라는 상태로 표시 되는데, 하드웨어적인 전압의 상태로써 HIGH는 5V를 LOW 는 0V를 의미합니다.

논리적인 개념으로는 HIGH는 참, LOW는 거짓으로 표현됩니다.



쉽게 말해서, 방 안의 스위치 역할을 한다고 볼 수 있습니다.

스위치를 올린다 (HIGH) → 불이 켜짐 (상태 : 참)

스위치를 내린다 (LOW) → 불이 꺼짐 (상태 : 거짓)

## 1.2 디지털 핀 블록 사용하기

디지털 블록은 핀 번호를 입력하도록 합니다.



### 1. 핀 번호 입력

사용할 핀 번호를 입력합니다.

### 2. 참.거짓 설정

지정한 핀에 논리 신호인 참(HIGH)와 거짓(LOW) 신호를 보냅니다. '참' 블록과 '거짓' 블록은 연산 팔레트에서 얻을 수 있습니다.

**참** : 해당하는 핀을 켭니다.

**거짓** : 해당하는 핀을 끕니다.

설정 값에 따라 해당하는 핀의 상태가 달라집니다.

11 번 핀이 켜집니다.



11번 핀이 꺼집니다.

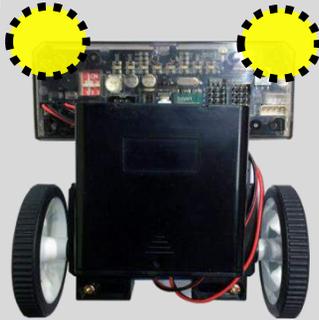


## 2.1 바닥감지로봇의 구동 원리

센서로 입력된 값이 없는 경우, 후진 후 방향을 틀어 다른 쪽으로 이동합니다.

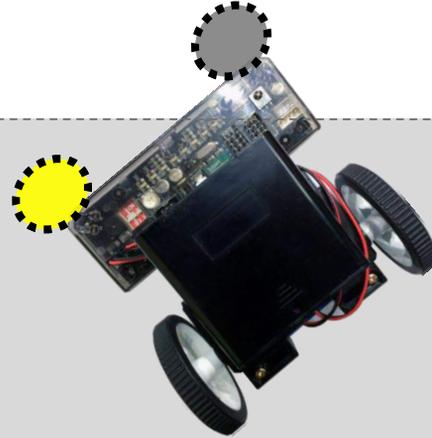
낭떠러지 (입력 없음)

지면 (입력이 들어옴)



두 개의 센서에  
입력이 들어올 때

**전진**

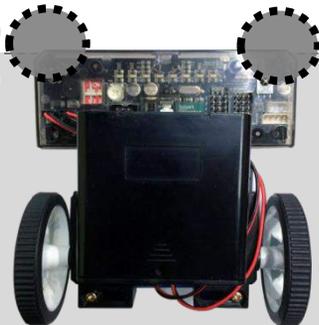


오른쪽 센서에만  
입력이 없을 때

**후진 후 좌회전**

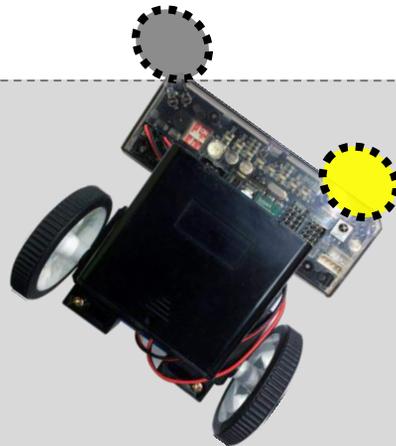
낭떠러지 (입력 없음)

지면 (입력이 들어옴)



두 개의 센서에 입력이 없거나  
왼쪽 센서에만 입력이 없을 때

**후진 후 우회전**

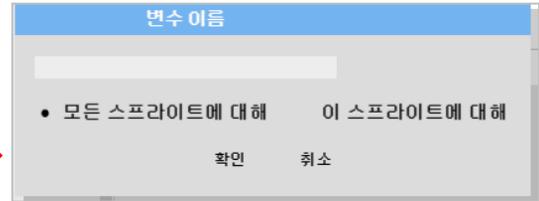


**주의**

낭떠러지가 없는 경우, 하얀색 바닥에 검은색 절연테이프를 원이나 네모모양으로 붙으면 낭떠러지와 같은 환경이 만들어 집니다.

## 3.1 프로그램 설명 (1)

### 1-1. 변수 설명



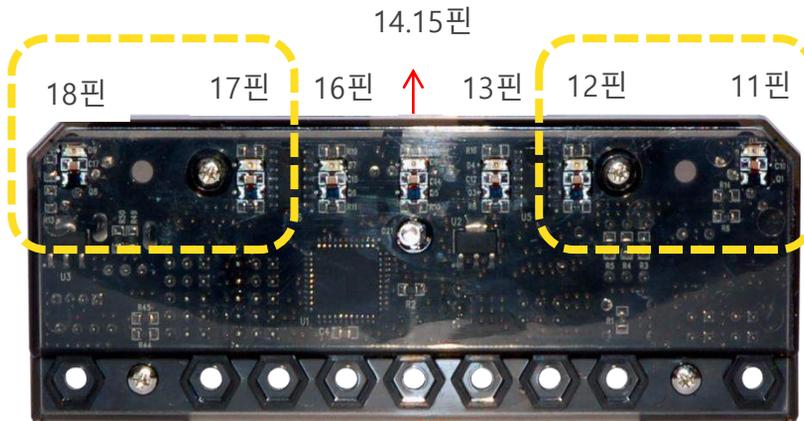
변수 만들기를 사용하여 한 개의 변수 블록을 만들어 줍니다.

1. speed – 모든 모터의 속도

### 1-2. 보드 뒷면 7개 센서의 디지털값

보드 뒷면의 7개의 센서가 디지털방식의 IR 센서 입니다.

디지털 센서란 0 또는 1과 같은 연속되지 않는 두개의 숫자로 정보를 생성하는 것을 뜻합니다.



센서 4개 : 11,12,17,18 사용

### 3.2 프로그램 설명 (2)

변수 선언

변수 speed 에 80 저장하기

변수 speed 에 80 을 저장

반복

무한 반복하기

조건문을 무한 반복

조건1

만약

디지탈 값 읽기 11 = 거짓 또는 디지탈 값 읽기 12 = 거짓

라면

DC모터 M1 ▾ 변을 속도 speed 동작 정회전 ▾  
 DC모터 M2 ▾ 변을 속도 speed 동작 역회전 ▾

바닥에 있는 디지탈 센서중 11,12번 센서가 바닥을 감지 안할때 후진  
 0.5초간 대기

0.5 초 기다리기

DC모터 M1 ▾ 변을 속도 speed 동작 역회전 ▾  
 DC모터 M2 ▾ 변을 속도 speed 동작 역회전 ▾

오른쪽으로 회전

0.5 초 기다리기

0.5초간 대기

조건2

만약

디지탈 값 읽기 17 = 거짓 또는 디지탈 값 읽기 18 = 거짓

라면

DC모터 M1 ▾ 변을 속도 speed 동작 정회전 ▾  
 DC모터 M2 ▾ 변을 속도 speed 동작 역회전 ▾

바닥에 있는 디지탈 센서중 17,18번 센서가 바닥을 감지 안할때 후진  
 0.5초간 대기

0.5 초 기다리기

DC모터 M1 ▾ 변을 속도 speed 동작 정회전 ▾  
 DC모터 M2 ▾ 변을 속도 speed 동작 정회전 ▾

왼쪽으로 회전

0.5 초 기다리기

0.5초간 대기

조건이 아닐때

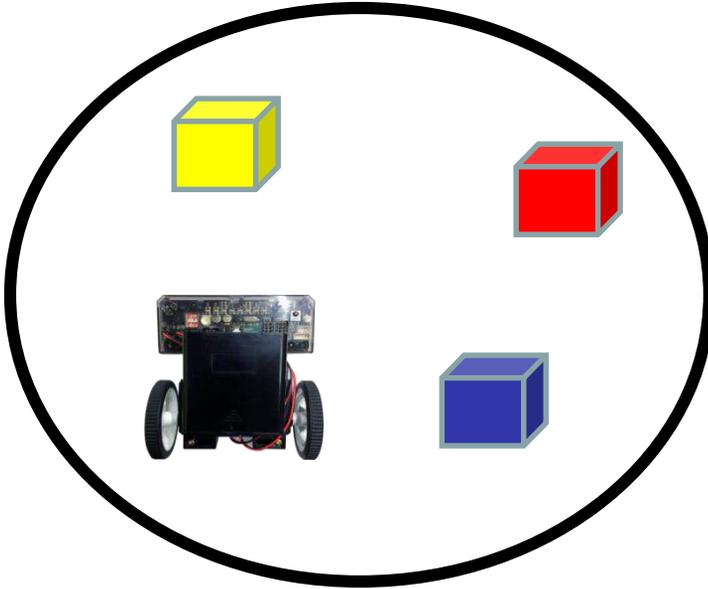
DC모터 M1 ▾ 변을 속도 speed 동작 역회전 ▾  
 DC모터 M2 ▾ 변을 속도 speed 동작 정회전 ▾

조건아 아닐때 전진

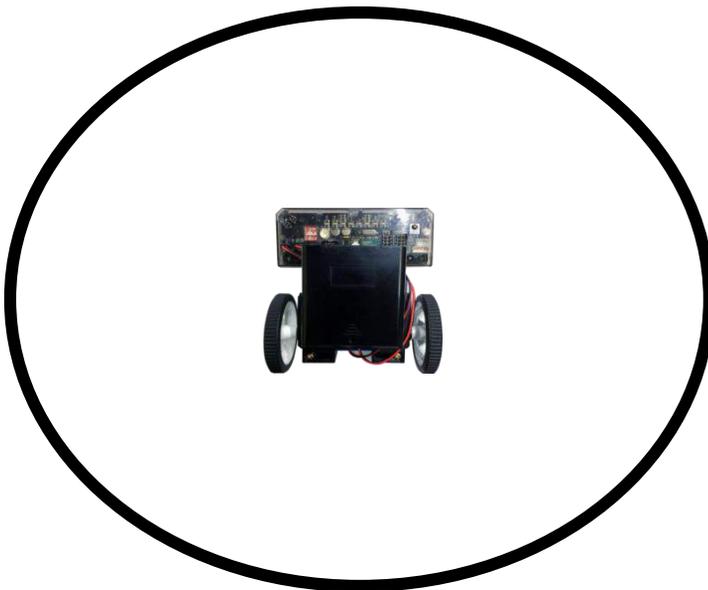
## 4.1 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
DOWNLOADS를 확인해주세요. 자세한 내용은 [교재 16페이지](#)▶

1. 바닥에 검은색 절연 테이프를 붙이고 물체를 모두 밖으로 내보내는 스모 형식의 로봇 응용 스크립트를 작성해보세요.



2. 낭떠러지를 열 번 감지하면 멈추는 스크립트를 작성해보세요.



### 3. 로봇 제어하기

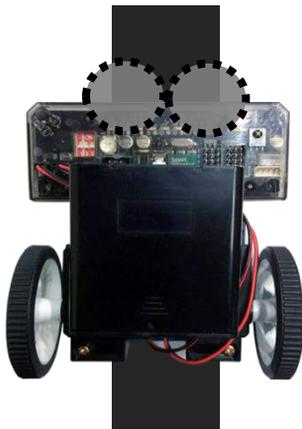
## 4장.

# 라인트레이서

## 1.1 라인트레이서의 구동 원리

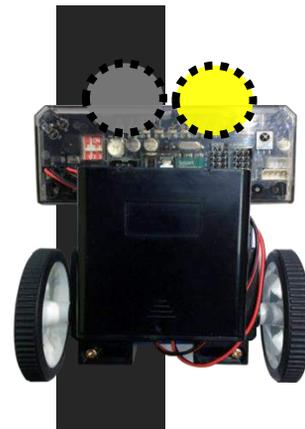
검정색 : 입력이 없음 (적외선을 흡수합니다)

흰색 : 입력이 들어옴 (적외선을 반사합니다)



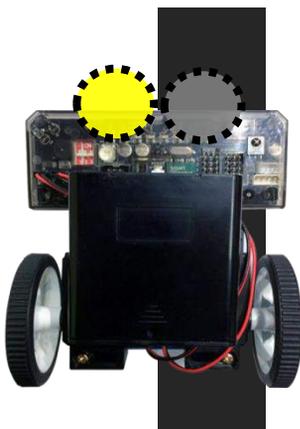
두 개의 센서에 입력이 없을 때  
(장애물이 없을 때)

**전진**



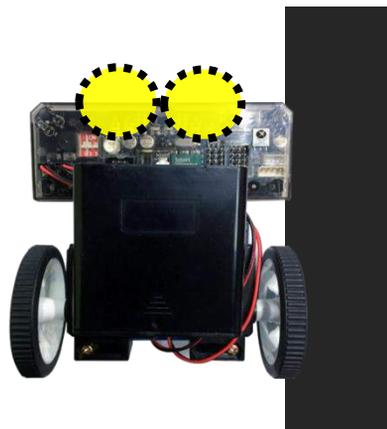
오른쪽 센서에만  
입력이 들어올 때

**좌회전**



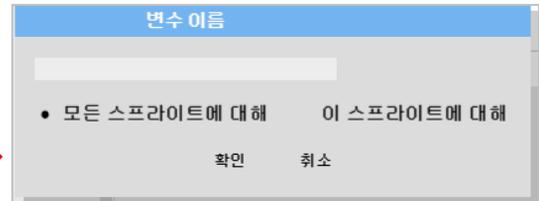
두 개의 센서에 입력이 들어오거나  
왼쪽 센서에만 입력이 들어올 때

**우회전**



## 2.1 프로그램 설명 (1)

### 1-1. 변수 설명



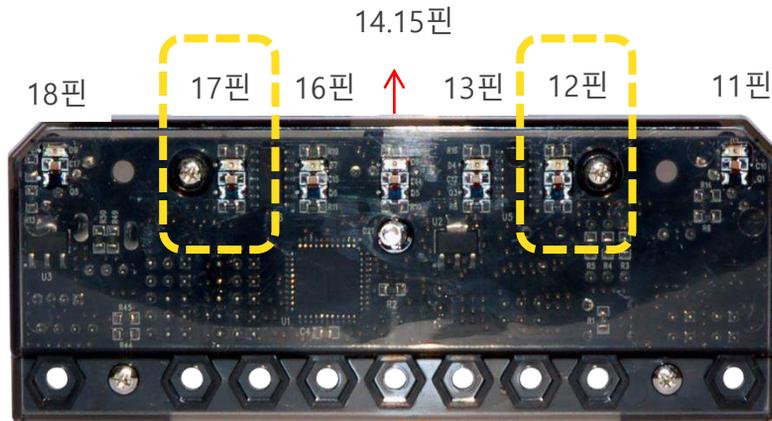
변수 만들기를 사용하여 3개의 변수 블록을 만들어 줍니다.

1. speed - 모든 모터의 속도
2. senL- 왼쪽 12핀 센서가 감지되면 참.  
감지되지 않으면 거짓.
3. senR- 오른쪽 17핀 센서가 감지되면 참.  
감지되지 않으면 거짓.

### 1-2. 보드 뒷면 7개 센서의 디지털값

보드 뒷면의 7개의 센서가 디지털방식의 IR 센서 입니다.

디지털 센서란 0 또는 1과 같은 연속되지 않는 두개의 숫자로 정보를 생성하는 것을 뜻합니다.



센서 2개 : 12, 17 사용

## 2.2 프로그램 설명 (2)

**변수 선언**  
 변수 speed 에 50 저장하기    변수 speed 에 50 을 저장

**반복**  
 무한 반복하기    조건문을 무한 반복

**조건1**  
 만약  
 디지털 값 읽기 12 = 거짓    그리고    디지털 값 읽기 17 = 참  
 라면  
 DC모터 M1    변수 속도 speed    동작 멈춤  
 DC모터 M2    변수 속도 speed    동작 정회전  
 바닥에 있는 디지털 센서중 12핀이 감지 안되고, 17핀이 감지 될때 왼쪽으로 회전

**조건2**  
 만약  
 디지털 값 읽기 12 = 참    그리고    디지털 값 읽기 17 = 거짓  
 라면  
 DC모터 M1    변수 속도 speed    동작 역회전  
 DC모터 M2    변수 속도 speed    동작 멈춤  
 바닥에 있는 디지털 센서중 12핀이 감지 되고, 17핀이 감지 안될때 오른쪽 회전

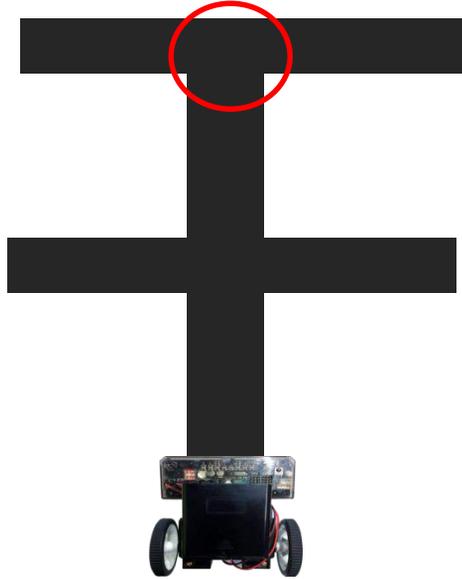
**조건3**  
 만약  
 디지털 값 읽기 12 = 참    그리고    디지털 값 읽기 17 = 참  
 라면  
 DC모터 M1    변수 속도 speed    동작 역회전  
 DC모터 M2    변수 속도 speed    동작 정회전  
 바닥에 있는 디지털 센서중 12핀,17핀 둘다 감지 될때 전진

**정지**  
 스페이스    키를 눌렀을 때  
 DC모터 M1    변수 속도 0    동작 멈춤  
 DC모터 M2    변수 속도 0    동작 멈춤  
 키보드의 스페이스 바를 눌렀을때 정지

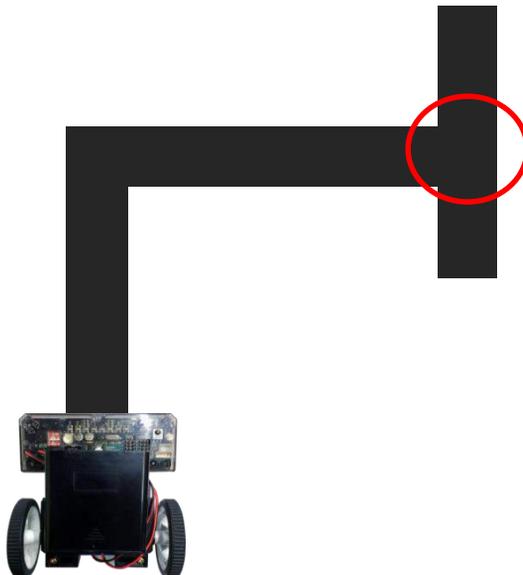
### 3.1 연습문제

예제와 연습문제 파일 다운로드는 [rokitarduino.org](http://rokitarduino.org) 웹사이트  
DOWNLOADS를 확인해주세요. 자세한 내용은 교재 16페이지▶

1. 바닥에 검은색 절연 테이프를 다음과 같이 붙이고, 두 번째 교차로 지점에서 멈추는 응용 스크립트를 작성해보세요.



2. 다음과 같은 직각형태의 라인을 통과하여 첫 번째 교차로에서 멈추는 응용 스크립트를 작성해 보세요.





**ROKIT**  
**Smart**

[www.rokitarduino.org](http://www.rokitarduino.org)

**BUILD, CODE, PLAY**