

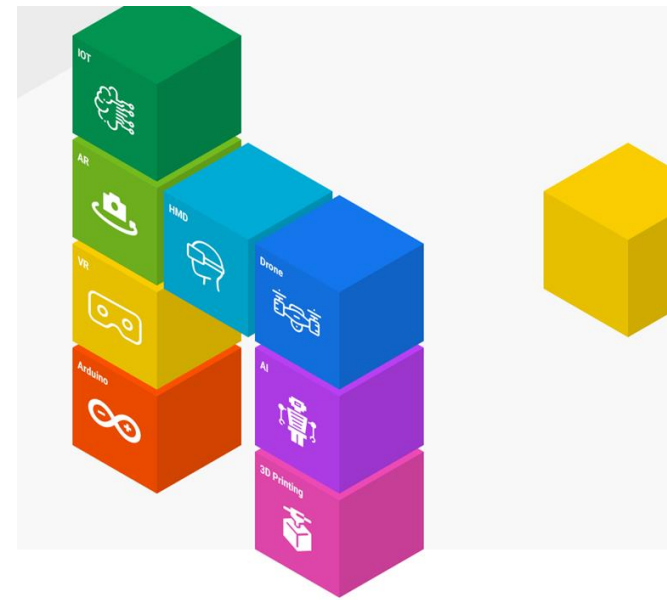
[아두이노 시뮬레이션 코딩] PSD 거리센서와 서보모터



www.helloapps.co.kr

김 영 준 / 070-4417-1559 / splduino@gmail.com

PSD 거리 센서



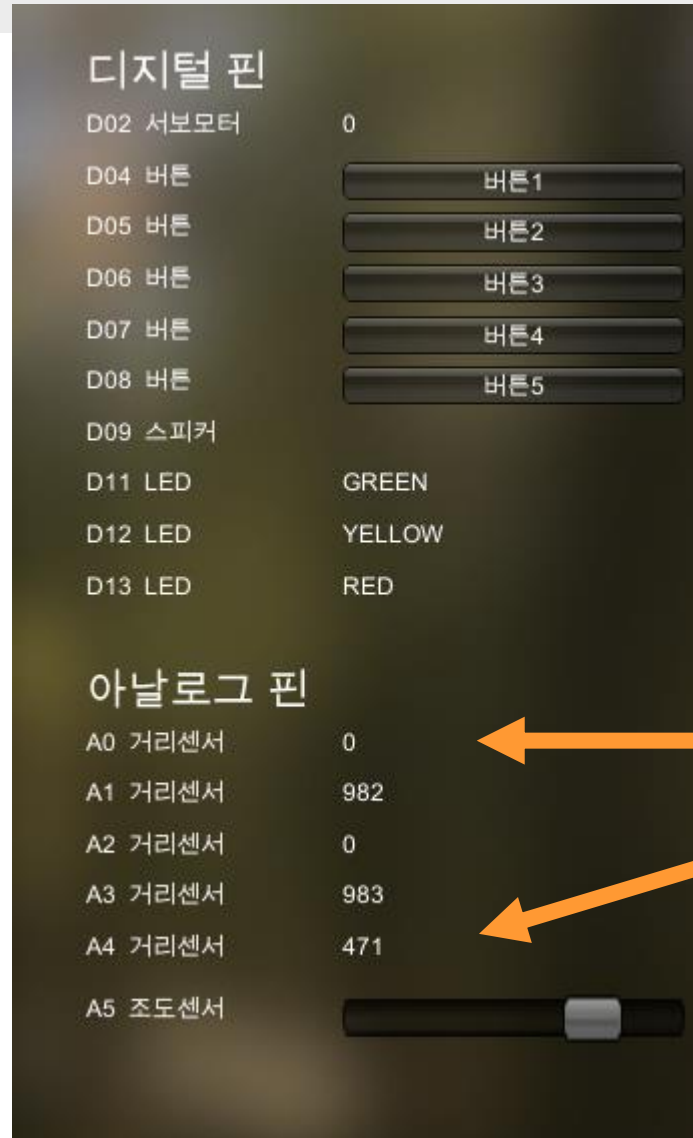
아날로그 PSD 거리 센서 종류



센서와 장애물 사이의 거리를
0 ~ 1023 사이의 값으로 표시

센서와 장애물 사이의 거리가
가까울 수록 값이 크고
거리가 멀 수록 값이 작아짐

시뮬레이션 연결 환경 이해하기

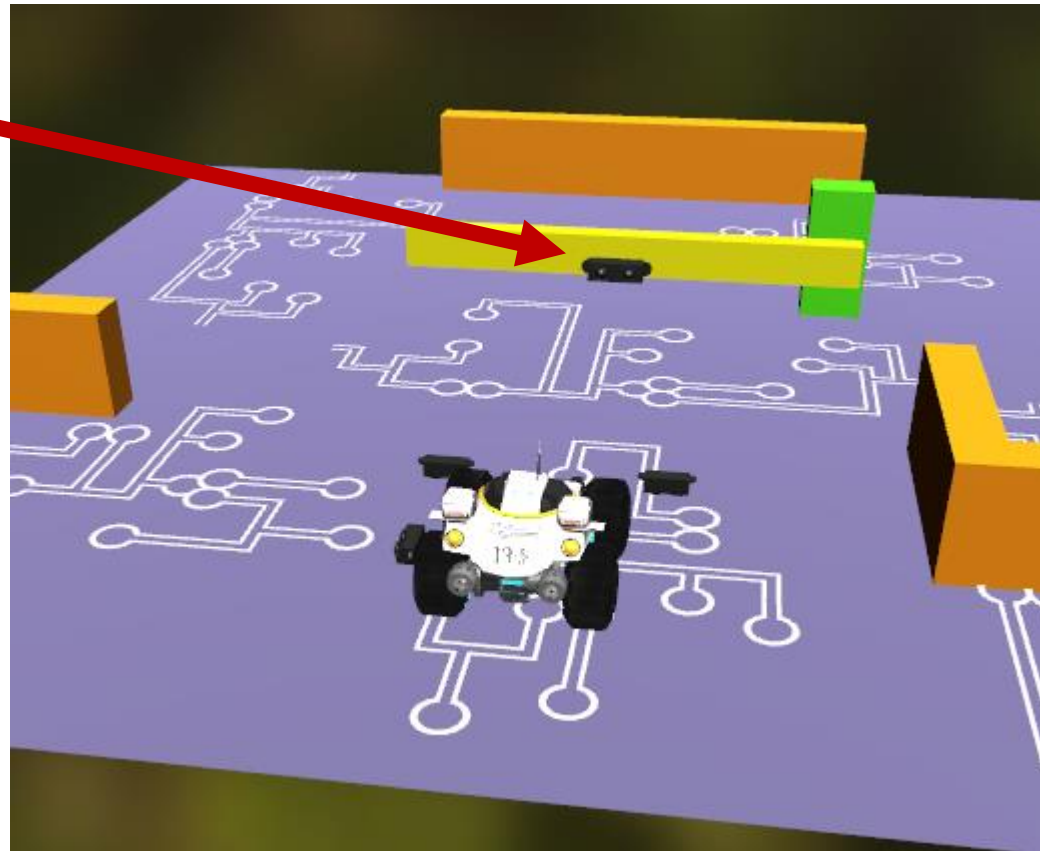


5개의 아날로그 PSD 거리센서가
연결되어 있음

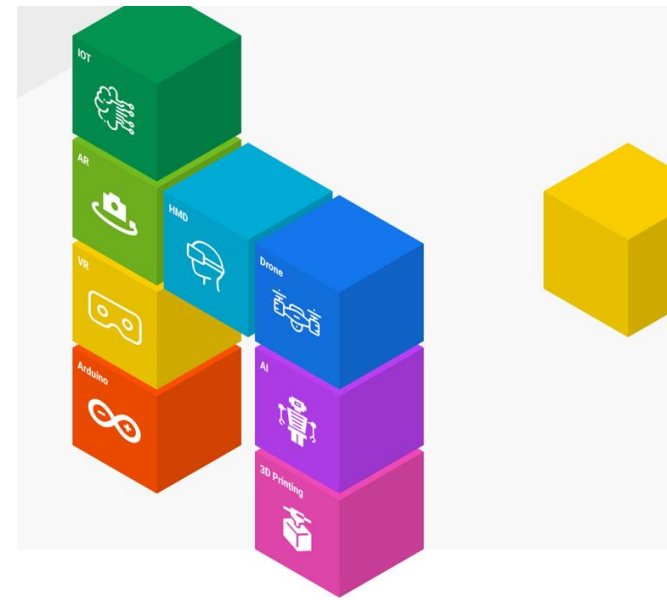
아날로그 센서 연결 상태



아날로그 4번 핀에 연결된
PSD 거리센서



PSD 거리 센서값 출력하기



PSD 거리 센서값 출력하기



아날로그 4번 핀에 연결된
PSD 거리센서의 값을
화면에 출력함

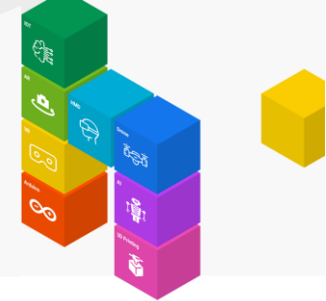
Function Setup

왼쪽에 있는 명령어 아이콘을
이곳에 마우스로 드래그하여 넣어 주세요.

Function Loop

- ☒ a4 = AnalogRead 4 (핀번호)
- ☒ PrintLine a4
- ☒ Delay 100 (밀리초)

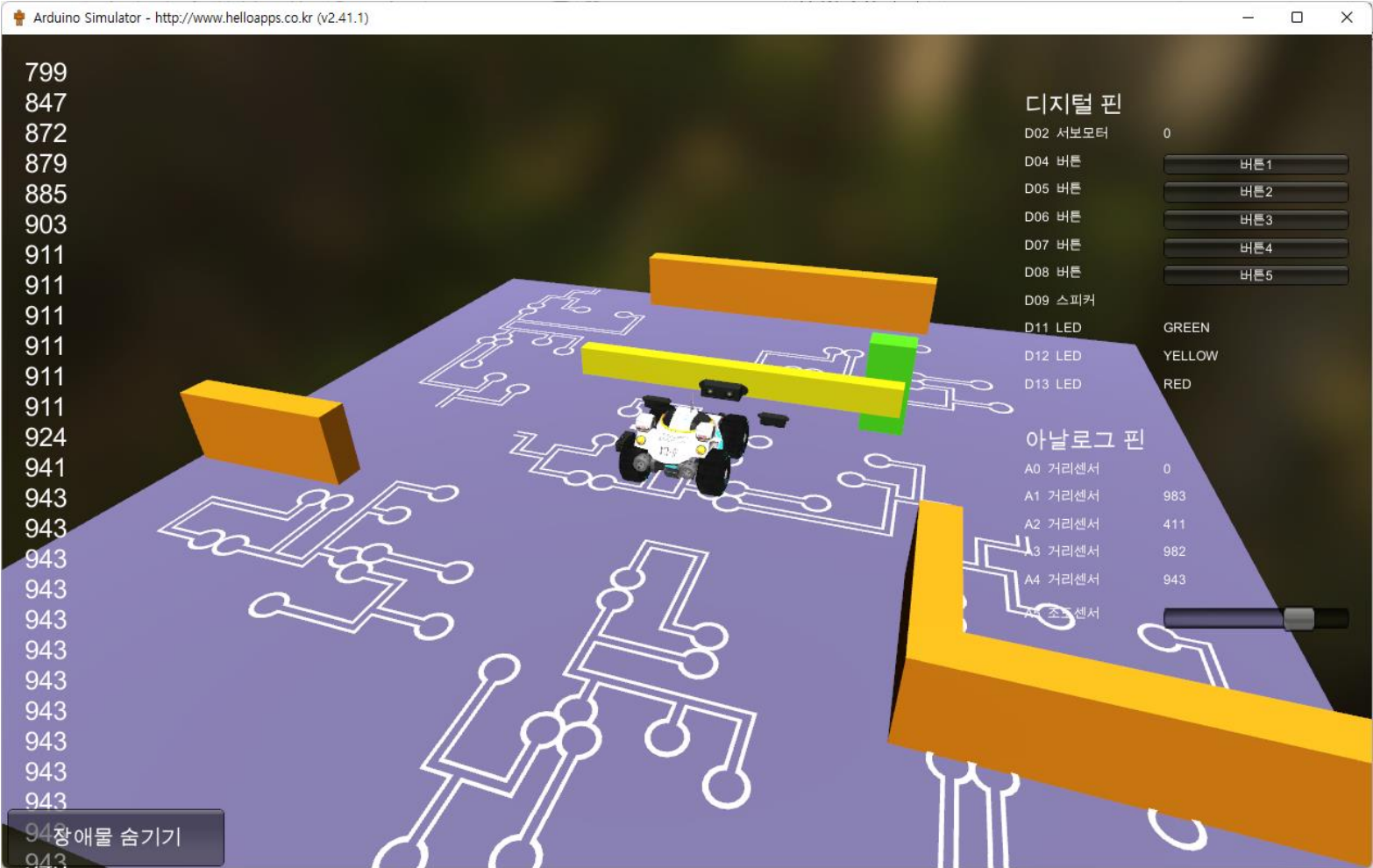
실행하기



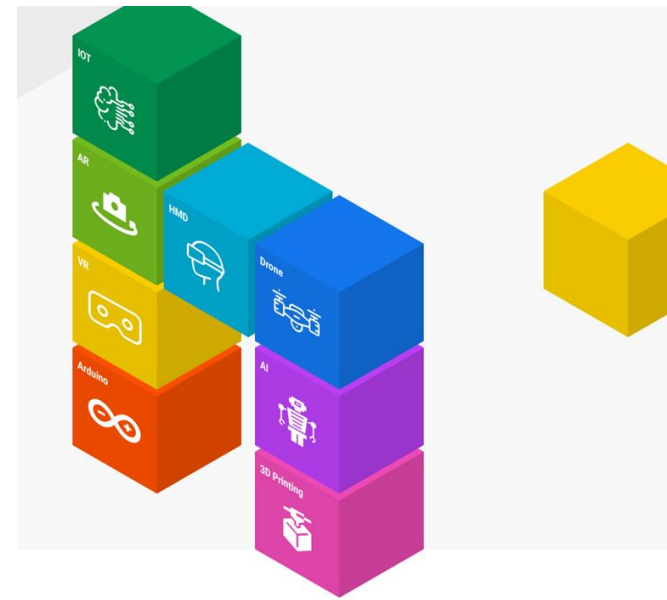
- 실행 버튼을 클릭합니다.



로봇이 PSD 센서에
접근할 수록
센서의 값이 커지는
것을 볼 수 있습니다



서보 모터



시뮬레이션 연결 환경 이해하기



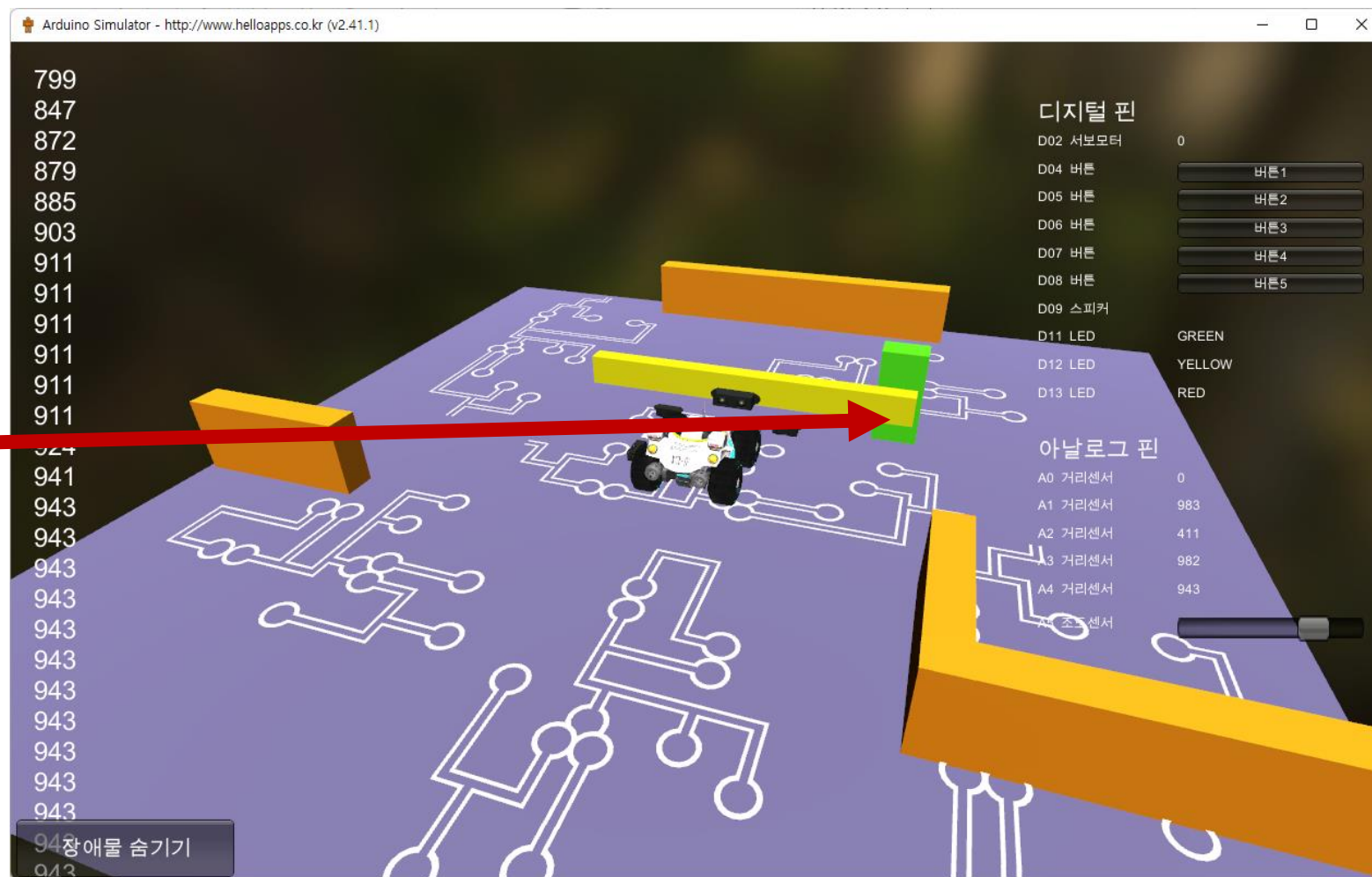
디지털 핀	
D02 서보모터	0
D04 버튼	버튼1
D05 버튼	버튼2
D06 버튼	버튼3
D07 버튼	버튼4
D08 버튼	버튼5
D09 스피커	
D11 LED	GREEN
D12 LED	YELLOW
D13 LED	RED
아날로그 핀	
A0 거리센서	0
A1 거리센서	982
A2 거리센서	0
A3 거리센서	983
A4 거리센서	471
A5 조도센서	

디지털 2번 핀에 서보 모터
연결되어 있음

서보 모터 장치



차단기에 서보모터
장착되어 있음



서보 모터 명령어



0도에서 180도 사이의 각도로 회전시킬 수 있음

☒ ServoWrite 2 (핀번호) 90 (값)

핀번호

각도
(0도 ~ 180도)

디지털 2번 핀에 연결된 서보 모터를 90도 회전시키는 예제입니다.

서보 모터 명령어



내장 명령어
탭을 클릭합니다.

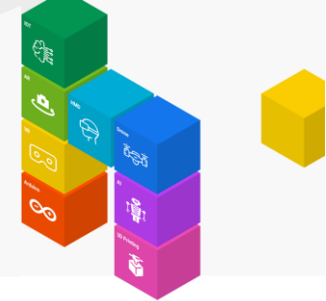
ServoWrite
명령어를 추가합니다.

The screenshot shows the HelloApps software interface. On the left, a command palette is open with the following categories and commands:

- [0] 트릭 명령어
 - [0-1] Expression
 - [0-2] Print
 - [0-3] PrintLine
- [1] 기본 명령어
- [2] 내장 명령어
 - [2-1] LED 제어 명령어
 - [2-1-1] SetColor
 - [2-1-2] SetAllColor
 - [2-1-2] SetRainbowCycle
 - [2-2] 주행 및 모터제어 명령어
 - [2-2-1] DriveWrite
 - [2-2-2] Motor1Write
 - [2-2-3] Motor2Write
 - [2-2-4] MotorDrive
 - [2-3] 서보모터 명령어
 - [2-3-1] ServoWrite
- [3] 3D 명령어

On the right, a block diagram is shown with two function blocks: "Function Setup" and "Function Loop". The "Function Setup" block contains a "ServoWrite" block with the following parameters: "2" (핀번호) and "90" (값). A red arrow points from the "ServoWrite" block in the command palette to the "ServoWrite" block in the "Function Setup" block. Another red arrow points from the text box "내장 명령어 탭을 클릭합니다." to the "[2] 내장 명령어" tab in the command palette.

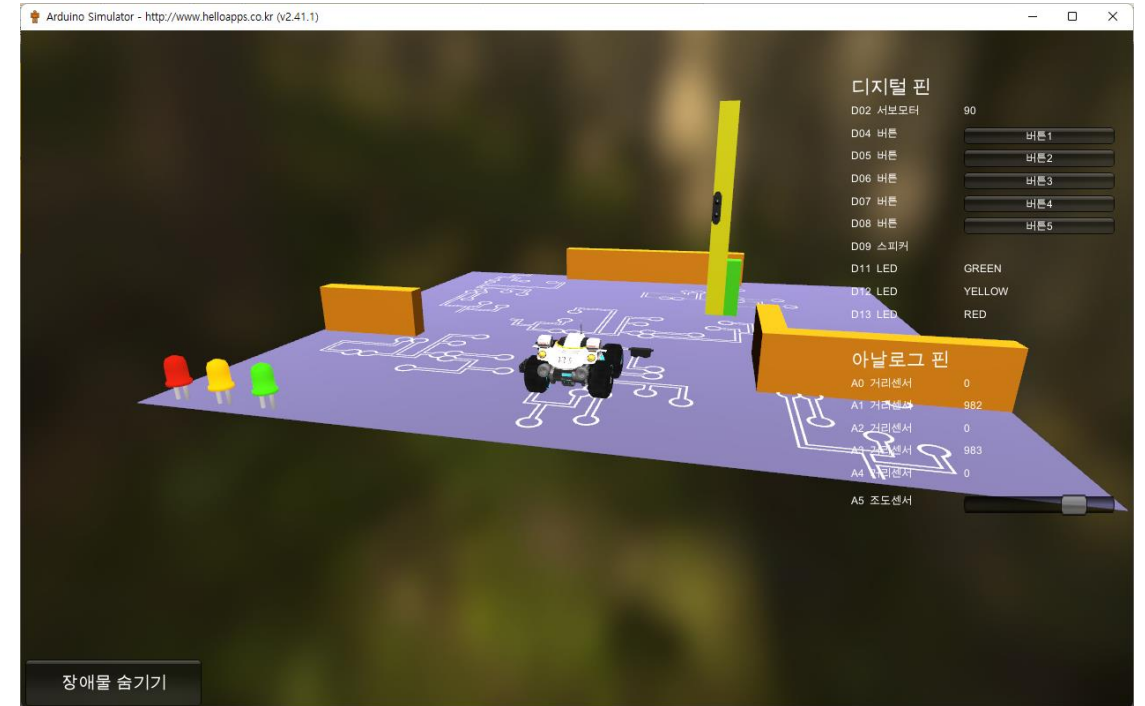
실행하기



- 실행 버튼을 클릭합니다.



서보 모터 명령어



서보 모터가 90도 회전하여 차단기가 올라가 있음

2초 후에 작동



2초 후에 차단기가
올라 갑니다.

2초 후에 차단기가
내려 옵니다.

The image shows a Scratch-style code editor with two function blocks: 'Setup' and 'Loop'.

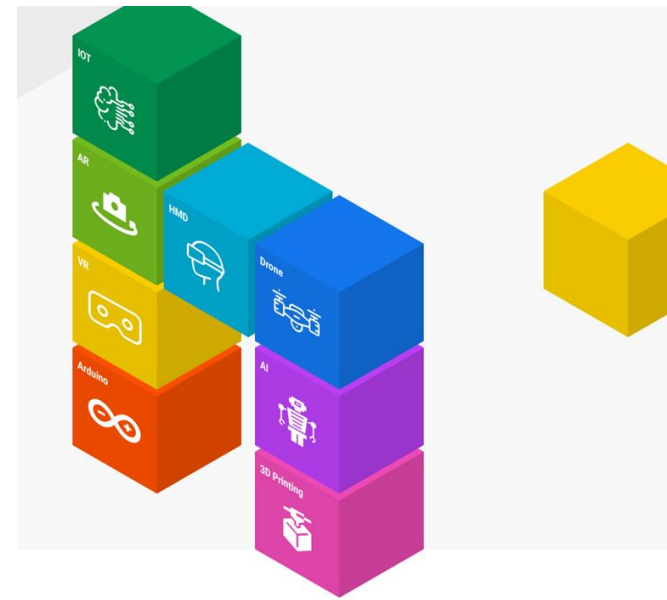
Function Setup

- ☒ Delay 2000 (밀리초)
- ☒ ServoWrite 2 (핀번호) 90 (값)
- ☒ Delay 2000 (밀리초)
- ☒ ServoWrite 2 (핀번호) 0 (값)

Function Loop

왼쪽에 있는 명령어 아이콘을 이곳에 마우스로 드래그하여 넣어 주세요.

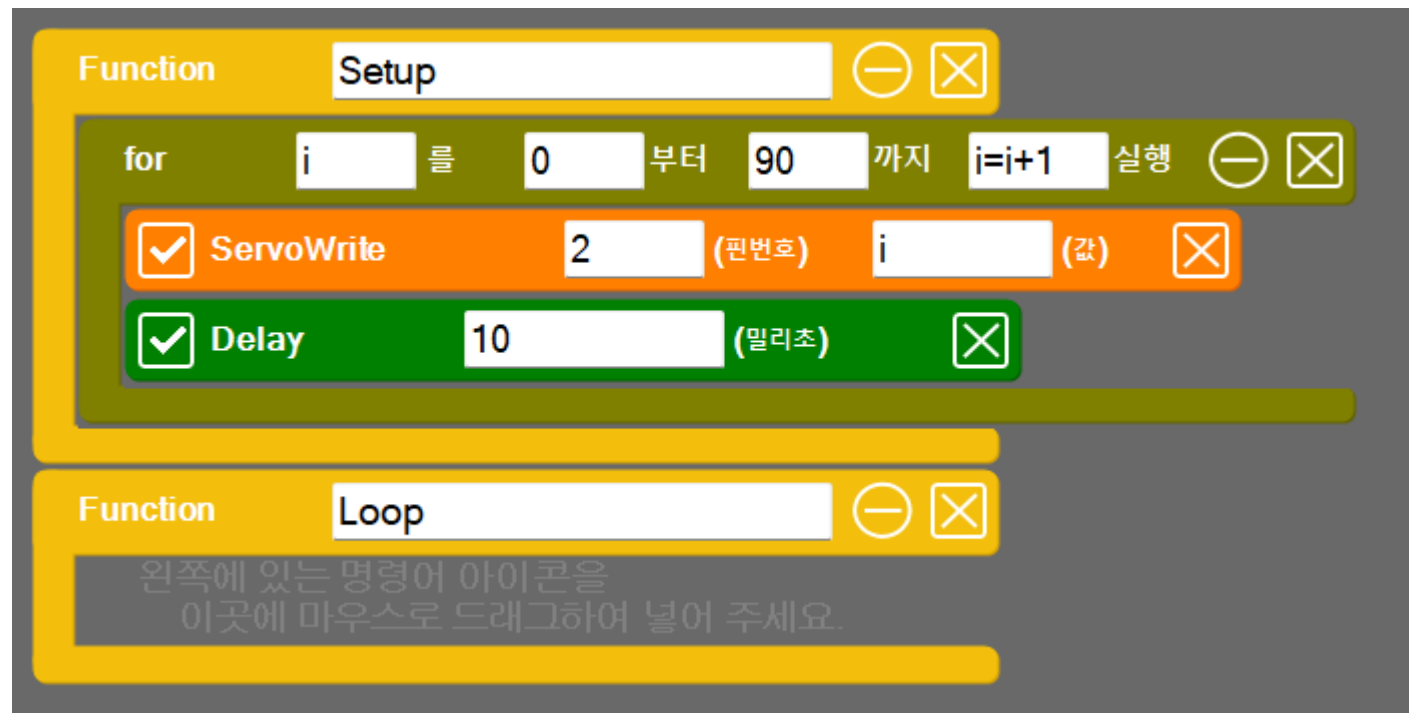
서보 모터를 부드럽게 회전시키기



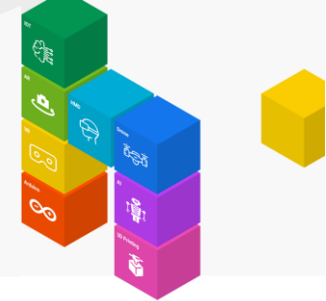
서보 모터를 반복문으로 회전시키기



for 반복문을 사용하여
0.1초 간격으로 1초 씩
회전시킴



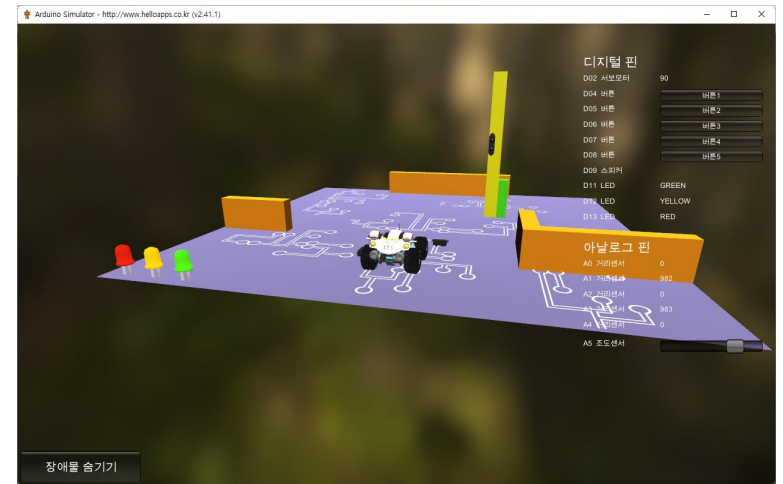
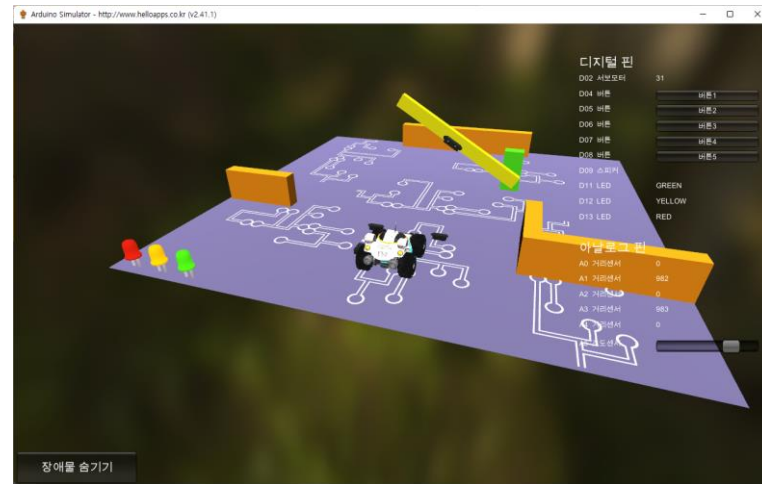
실행하기



- 실행 버튼을 클릭합니다.

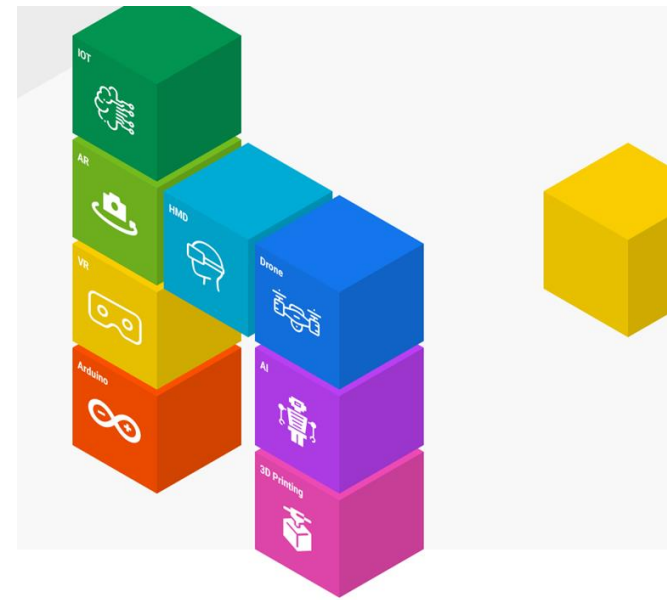


서보 모터를 반복문으로 회전시키기



차단기가 부드럽게 올라가는 것을 볼 수 있습니다.

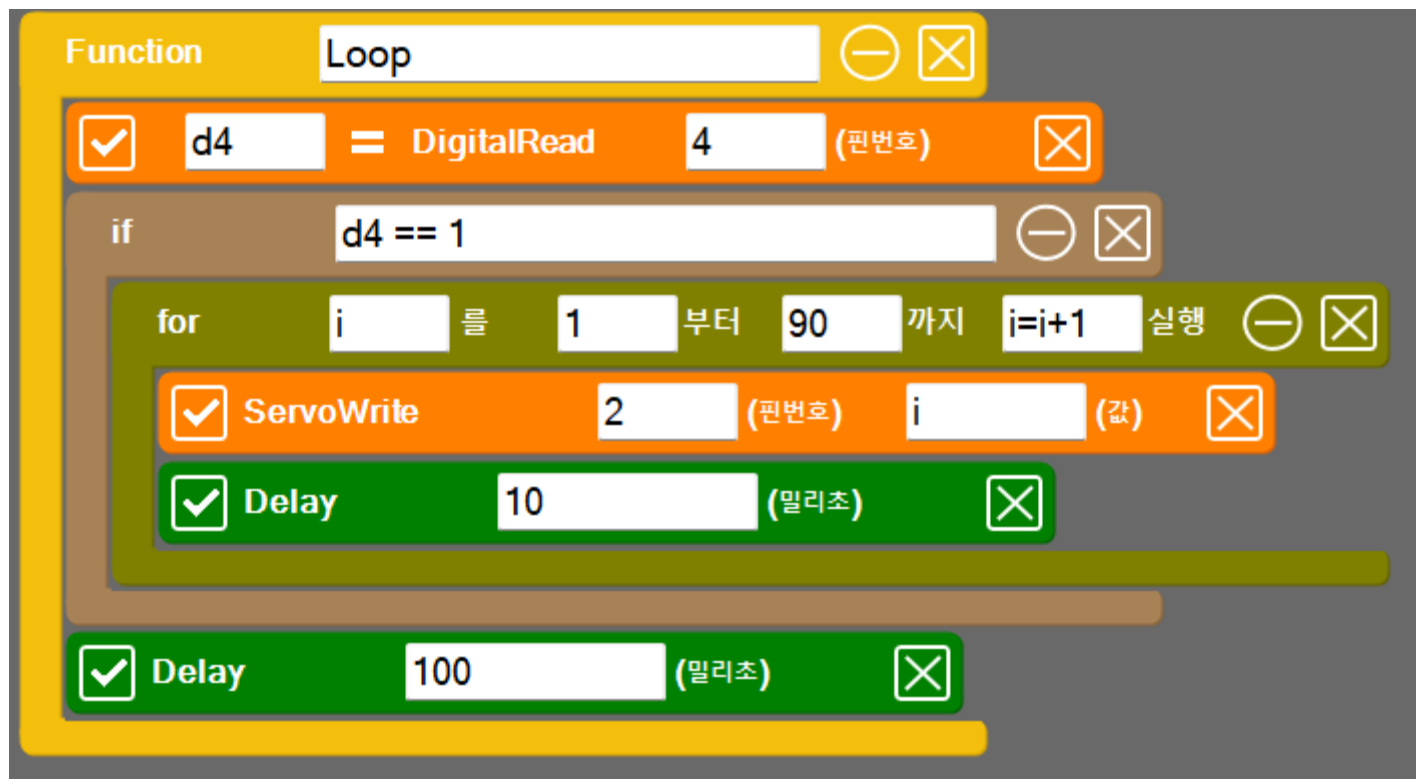
버튼으로 차단기 작동시키기



버튼으로 차단기 올리기



4번 핀의 버튼이
눌리면 차단기가
올라갑니다.



버튼으로 차단기 올리기

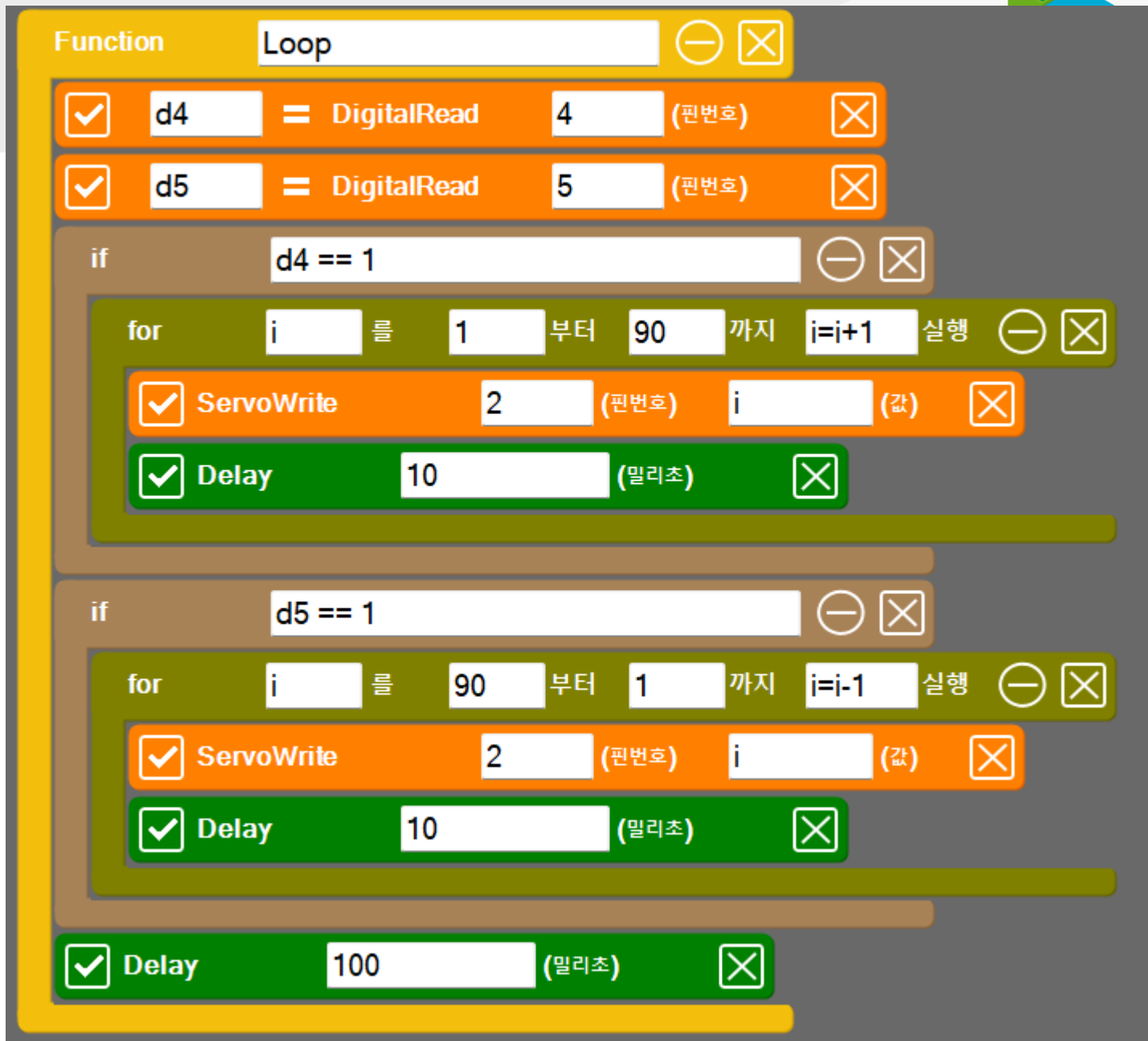


4번 핀의 버튼이
눌리면 차단기가
올라갑니다.



차단기 내리기

5번 핀의 버튼이
눌리면 차단기가
내려옵니다.



버튼으로 차단기 작동시키기

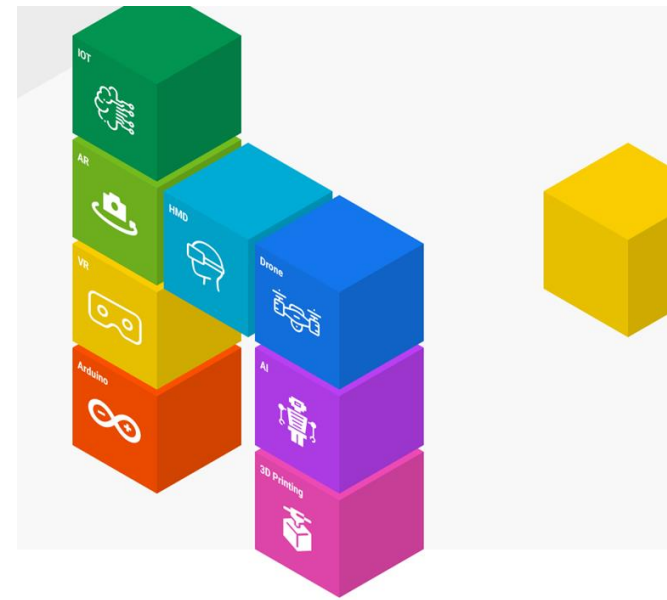


4번 핀의 버튼이
눌리면 차단기가
올라갑니다.



5번 핀의 버튼이
눌리면 차단기가
내려옵니다.

자동차가 접근하면 저절로 올라가는 차단기

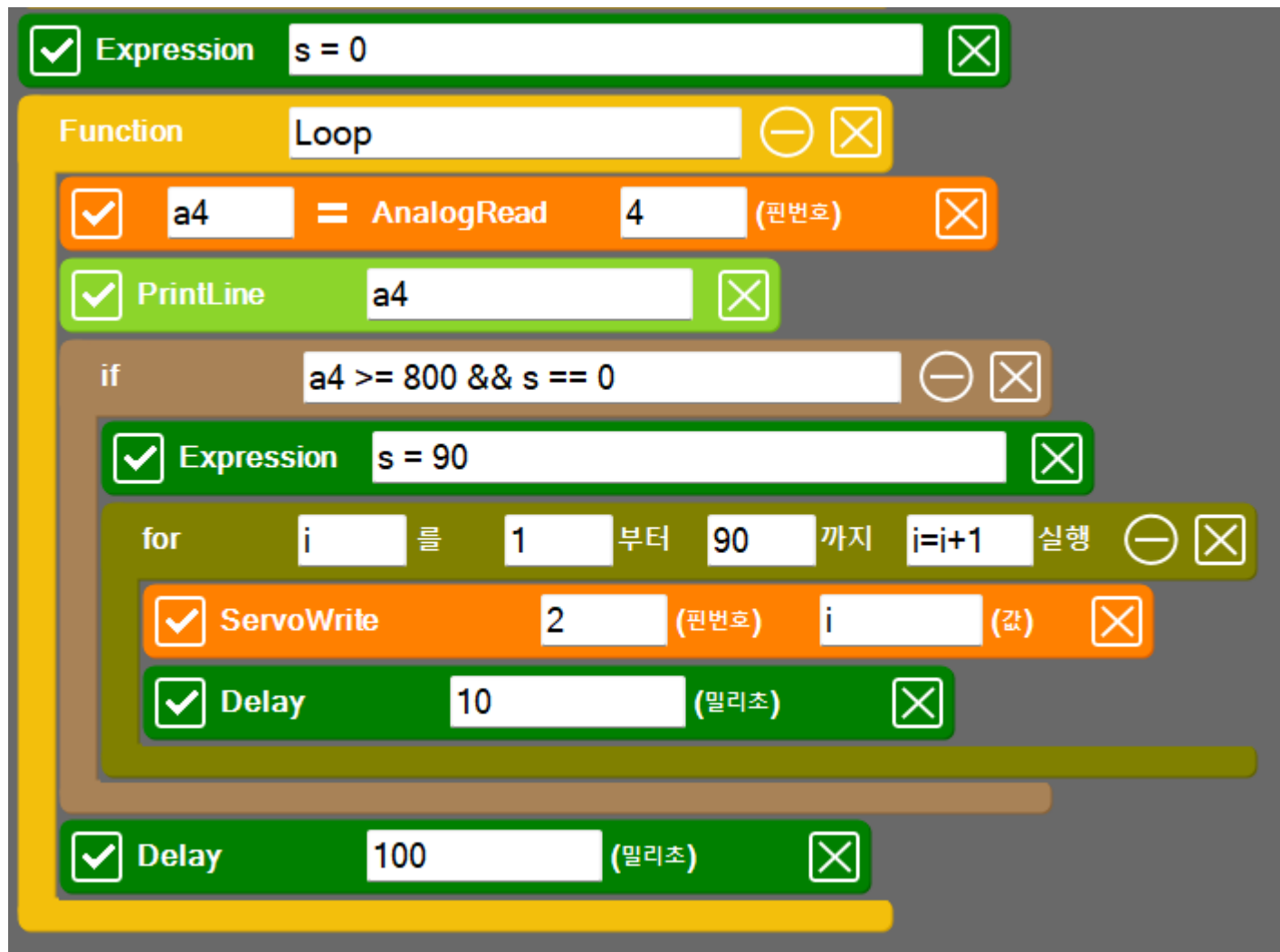


자동차가 접근하면 저절로 올라가는 차단기



a4 >= 800 이면
자동차가 가까이
접근한 것임

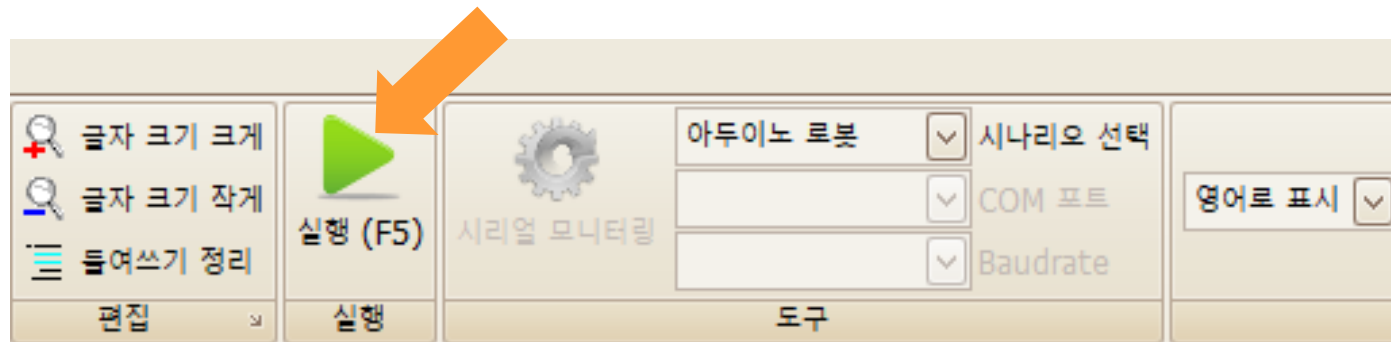
s == 0 이면
차단기가 아직 올라가
있지 않은 상태임



실행하기



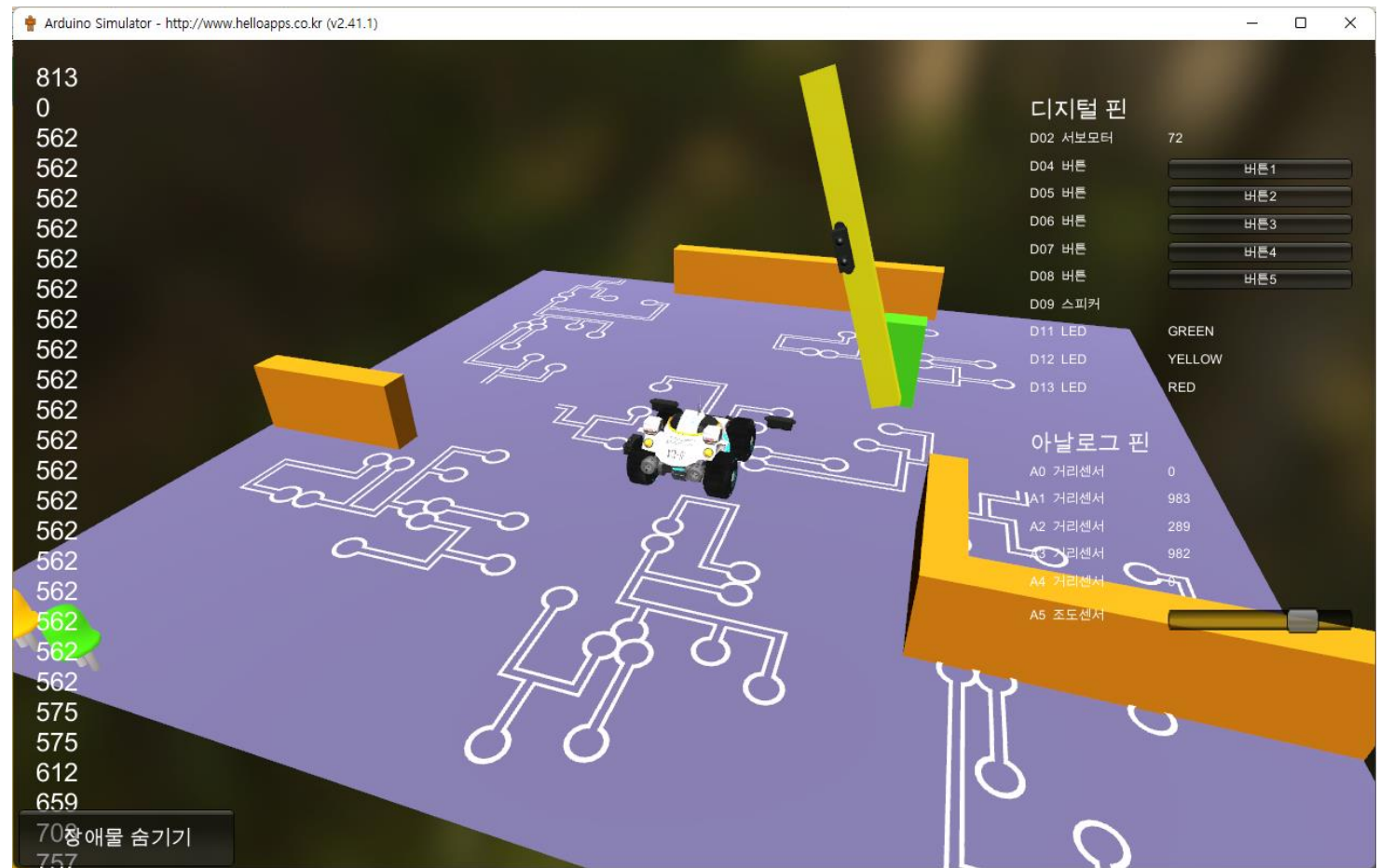
- 실행 버튼을 클릭합니다.



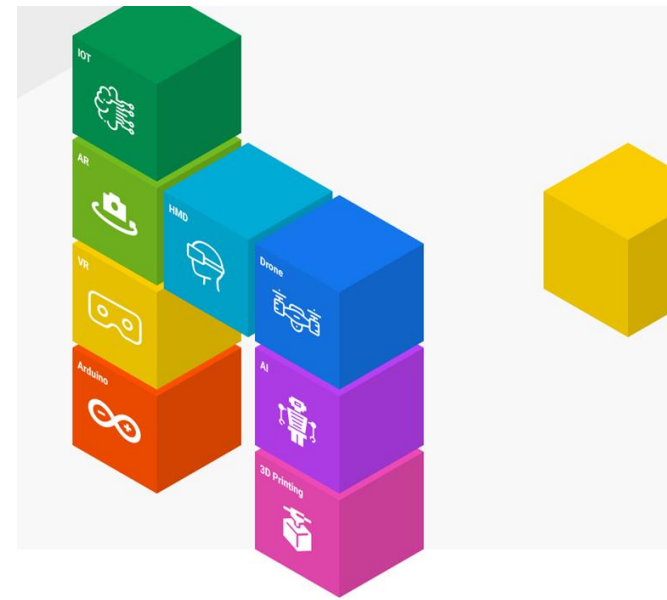
자동차가 접근하면 저절로 올라가는 차단기



로봇 자동차가 차단기에
접근하면 차단기가
자동으로 올라감



저절로 내려오는 기능 추가하기



내려오는 기능 추가

3초가 지나면
차단기가 다시
내려오는 기능을
추가합니다.

The image shows a Scratch script designed to control a servo motor. The script is organized as follows:

- Expression:** `s = 0`
- Function:** `Loop`
- Block 1:** `a4 = AnalogRead 4` (핀번호)
- Block 2:** `if a4 > 800 && s == 0`
- Block 3 (Inside if):** `for i 를 0 부터 90 까지 i=i+1 실행`
- Block 4 (Inside for):** `ServoWrite 2 (핀번호) i (값)`
- Block 5 (Inside for):** `Expression s = i`
- Block 6 (Inside for):** `Delay 10 (밀리초)`
- Block 7 (After for):** `Delay 3000 (밀리초)`
- Block 8 (After delay):** `for i 를 90 부터 0 까지 i=i-1 실행`
- Block 9 (Inside for):** `ServoWrite 2 (핀번호) i (값)`
- Block 10 (Inside for):** `Expression s = i`
- Block 11 (Inside for):** `Delay 10 (밀리초)`
- Block 12 (After for):** `Delay 100 (밀리초)`

실행하기



- 실행 버튼을 클릭합니다.

