
아두이노 시뮬레이션 프로그래밍

v1.0

김영준 저

공학박사, 목원대학교 겸임교수
前 Microsoft 수석연구원

헬로앱스

<http://www.helloapps.co.kr>

10 자율주행 로봇 구현하기

학습 목표

- 시뮬레이션 로봇의 동작을 제어할 수 있다.
- 시뮬레이션 로봇의 센서를 활용하여 장애물을 회피할 수 있다.

실습 개요

- 로봇의 거리 센서 값을 확인해 본다.
- 거리 센서 값을 이용하여 서보 모터를 제어한다.
- 3D 장애물을 추가하는 명령어를 배워 본다.
- 장애물을 회피하는 미션을 수행해 본다.

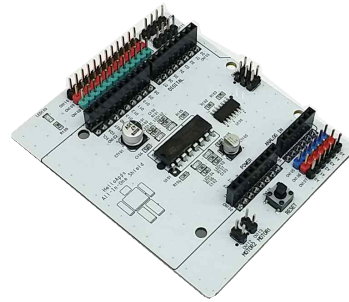
10.1 준비하기

준비물

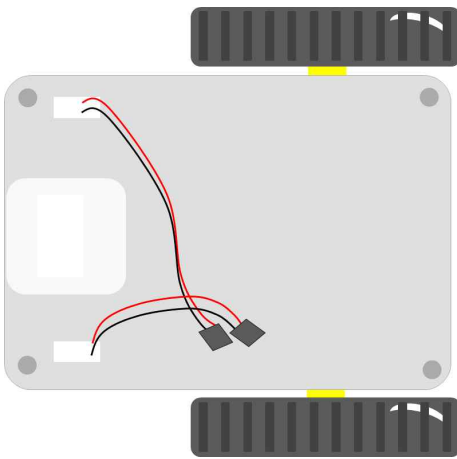
- 아두이노 보드, 올인원 쉴드, 아날로그 PSD센서, LED 모듈, 스피커



아두이노 우노보드



올인원 쉴드



모바일 로봇 플랫폼

시뮬레이션 상에서의 연결 정보

- 시뮬레이션 상에서는 디지털 LED 소자가 각각 디지털 11번, 12번, 13번에 연결되어 있으며, 차단기에 부착된 전방 PSD 거리센서는 아날로그 4번에 연결되어 있다.

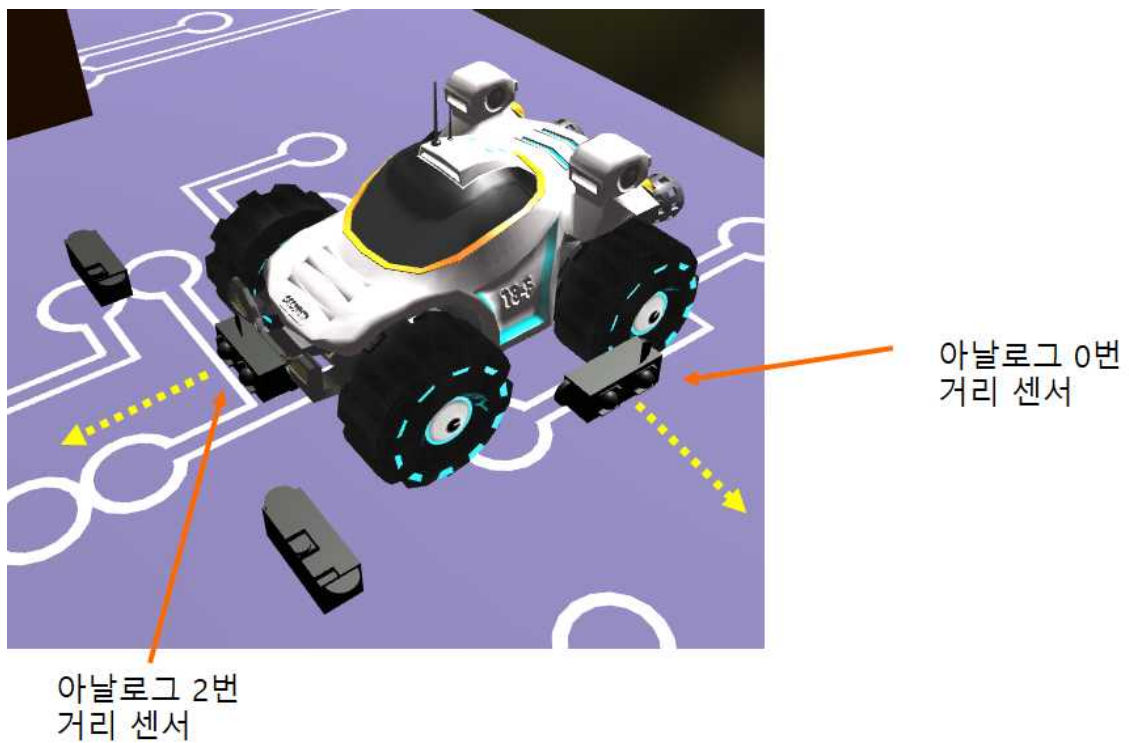


- 디지털/아날로그 핀에 연결된 부품
 - 디지털 4번 ~ 8번: 디지털 버튼
 - 아날로그 0번: 로봇 우측 PSD 거리 센서
 - 아날로그 2번: 로봇 전방에 있는 PSD 거리 센서
 - 아날로그 4번: 차단기에 부착된 PSD 거리 센서

10.2 로봇에 장착된 거리 센서

로봇 장착 센서

- 로봇에는 총 4개의 PSD 거리 센서가 장착되어 있다. 이 중에서 2개의 센서는 각각 왼쪽과 앞쪽을 향하는 거리 센서이다.



아날로그 0번 거리 센서: 왼쪽 거리 측정

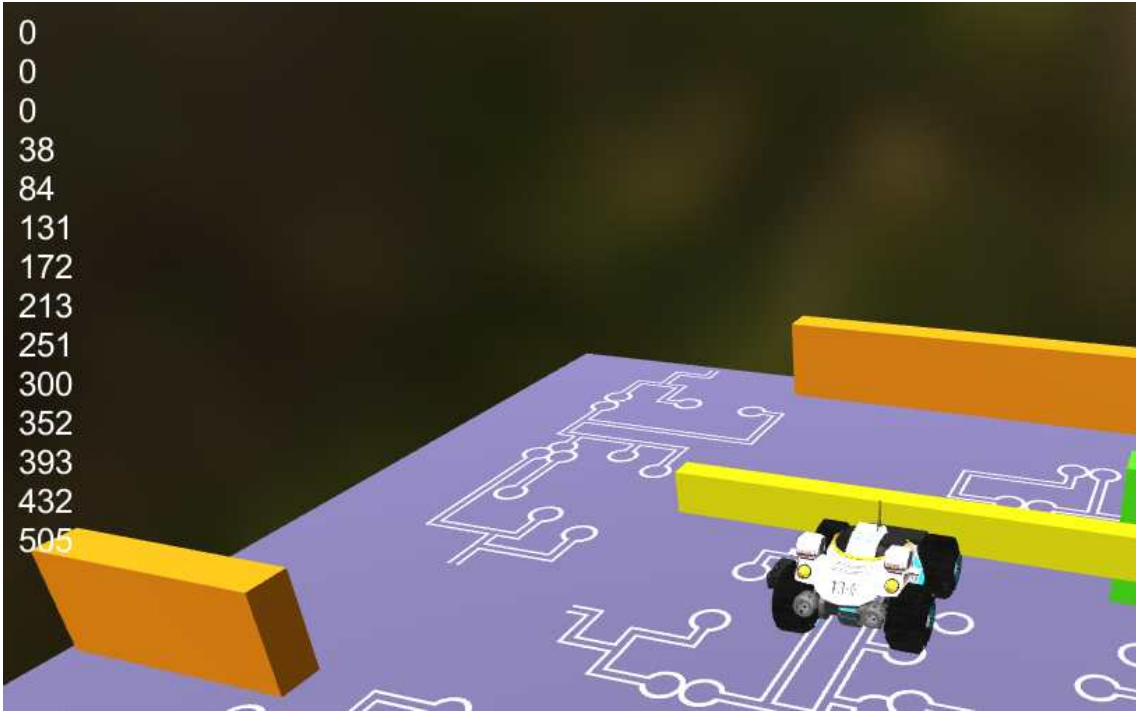
아날로그 2번 거리 센서: 앞쪽에서의 거리 측정

센서값 확인하기

- 아날로그 2번에 연결되어 있는 거리 센서의 값을 화면에 출력해 본다.



SPL 스크립트	스케치 코드
<pre>void setup() { DriveWrite(200, 200) } void loop() { a = AnalogRead(2) PrintLine(a) Delay(100) }</pre>	<pre>void setup() { Serial.begin(115200); DeriveWrite(200, 200); } void loop() { int a = analogRead(2); Serial.println(a); delay(100); }</pre>



10.3 서보모터 제어하기

서보모터 명령어

- 시뮬레이션 화면 상에서 로봇을 가로막고 있는 차단기에는 서보모터가 장착되어 있다. 서보모터를 제어하는 명령어는 다음과 같다.

ServoWrite(핀번호, 각도)

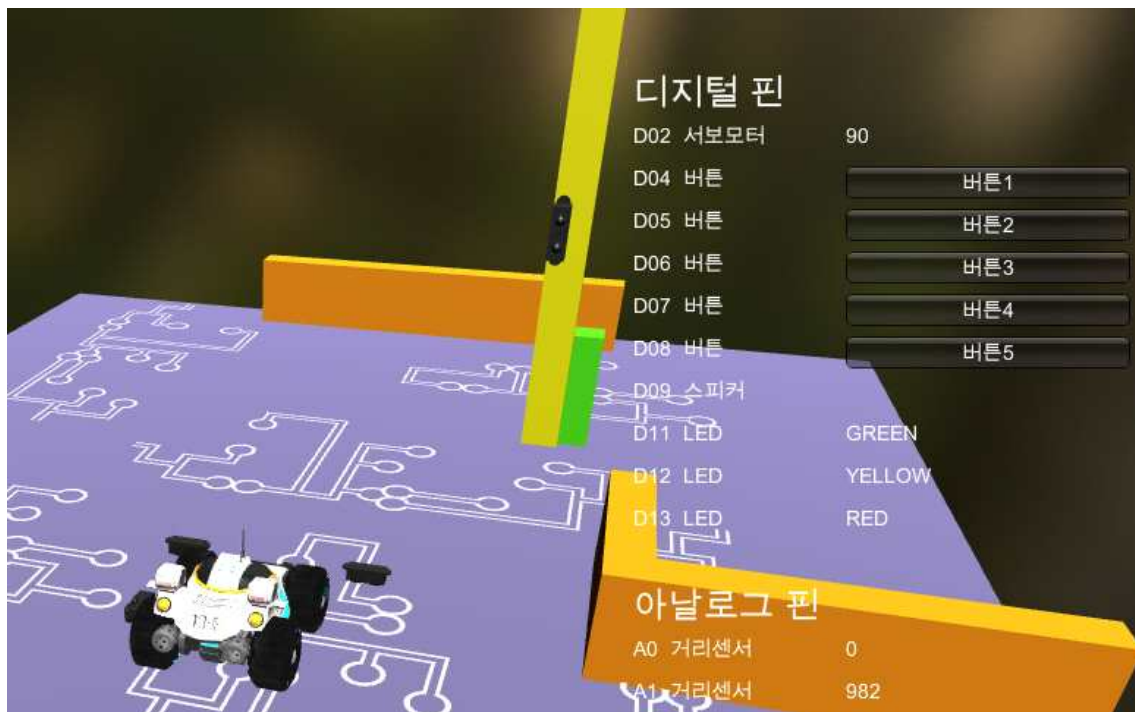
ServoWrite(2, 90)

서보모터 제어하기

- 1초 간격으로 차단기에 연결되어 있는 서보모터를 제어해 보자.



SPL 스크립트	스케치 코드
<pre>void setup() { } void loop() { ServoWrite(2, 90) Delay(1000) ServoWrite(2, 0) Delay(1000) }</pre>	<pre>void setup() { } void loop() { ServoWrite(2, 90); delay(1000); ServoWrite(2, 0); delay(1000); }</pre>



자동으로 차단기 작동시키기

- 로봇이 이동하면서 차단기에 접근하면 자동으로 차단기가 올라가도록 기능을 구현해 본다.



SPL 스크립트	스케치 코드
<pre> void setup() { DriveWrite(150, 150) } void loop() { a = AnalogRead(2) if (a > 300) ServoWrite(2, 90) Delay(100) } </pre>	<pre> void setup() { DriveWrite(150, 150) } void loop() { int a = analogRead(2); if (a > 300) ServoWrite(2, 90); delay(100); } </pre>



10.4 장애물을 회피하는 로봇

장애물을 만나면 멈추는 기능

- 로봇이 전진하다가 장애물 벽을 만나면 멈추는 기능을 구현해 본다.

```

Function Setup
  ServoWrite 2 (핀번호) 90 (값)
  DriveWrite 150 (Left) 150 (Right)

Function Loop
  a = AnalogRead 2 (핀번호)
  PrintLine a
  if a > 600
    DriveWrite 0 (Left) 0 (Right)
  Delay 100 (밀리초)
    
```

SPL 스크립트	스케치 코드
<pre> void setup() { ServoWrite(2, 90) DriveWrite(150, 150) } void loop() { a = AnalogRead(2) PrintLine(a) if (a > 600) DriveWrite(0, 0) Delay(100) } </pre>	<pre> void setup() { Serial.begin(115200); ServoWrite(2, 90); DriveWrite(150, 150); } void loop() { int a = analogRead(2); Serial.println(a); if (a > 600) DriveWrite(0, 0); delay(100); } </pre>

장애물을 만나면 후진/회전하는 기능 추가

- 이번에는 장애물을 만나면 로봇이 1초간 후진을 하고 1초간 회전을 한 후에 로봇이 다시 전진하는 기능을 구현해 보자.

The image shows a Scratch-style block diagram for an Arduino program. It consists of two main function blocks: 'Setup' and 'Loop'.

- Setup Function:** Contains a 'ServoWrite' block with pin number '2' and angle '90'.
- Loop Function:**
 - Starts with an 'AnalogRead' block on pin '2' assigned to variable 'a'.
 - Followed by an 'if' statement: 'if a > 600'.
 - True Path:**
 - 'DriveWrite' block: Left motor -150, Right motor -150.
 - 'Delay' block: 1000 milliseconds.
 - 'DriveWrite' block: Left motor 150, Right motor -150.
 - 'Delay' block: 1000 milliseconds.
 - False Path:**
 - 'DriveWrite' block: Left motor 150, Right motor 150.
 - 'Delay' block: 100 milliseconds.

SPL 스크립트	스케치 코드
<pre> void setup() { ServoWrite(2, 90) } void loop() { a = AnalogRead(2) if (a > 600) { DriveWrite(-150, -150) Delay(1000) DriveWrite(150, -150) Delay(1000) } DriveWrite(150, 150) Delay(100) } </pre>	<pre> void setup() { ServoWrite(2, 90) } void loop() { int a = analogRead(2); if (a > 600) { DriveWrite(-150, -150); delay(1000); DriveWrite(150, -150); delay(1000); } DriveWrite(150, 150); delay(100); } </pre>

10.5 장애물 벽 만들기

장애물 숨기기

- HideObstacle() 명령어를 setup() 함수에 추가하여 실행시 장애물이 숨겨지도록 기능을 수정한다.

The screenshot shows the HelloApps IDE interface with the following code blocks:

- Function Setup:**
 - Expression: HideObstacle()
- Function Loop:**
 - a = AnalogRead 2 (핀번호)
 - if a > 600:
 - DriveWrite -150 (Left) -150 (Right)
 - Delay 1000 (밀리초)
 - DriveWrite 150 (Left) -150 (Right)
 - Delay 1000 (밀리초)
 - DriveWrite 150 (Left) 150 (Right)
 - Delay 100 (밀리초)

SPL 스크립트	스케치 코드
<pre> void setup() { HideObstacle() } void loop() { a = AnalogRead(2) if (a > 600) { DriveWrite(-150, -150) Delay(1000) DriveWrite(150, -150) Delay(1000) } DriveWrite(150, 150) Delay(100) } </pre>	<pre> void setup() { HideObstacle(); } void loop() { int a = analogRead(2); if (a > 600) { DriveWrite(-150, -150); delay(1000); DriveWrite(150, -150); delay(1000); } DriveWrite(150, 150); delay(100); } </pre>



장애물 벽 만들기

- 다음과 같이 AddCube 명령어들을 setup 함수 안에 추가하여 외부의 벽을 만들어 준다.

AddCude(위치_X, 위치_Y, 위치_Z, 크기_X, 크기_Y, 크기_Z);

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

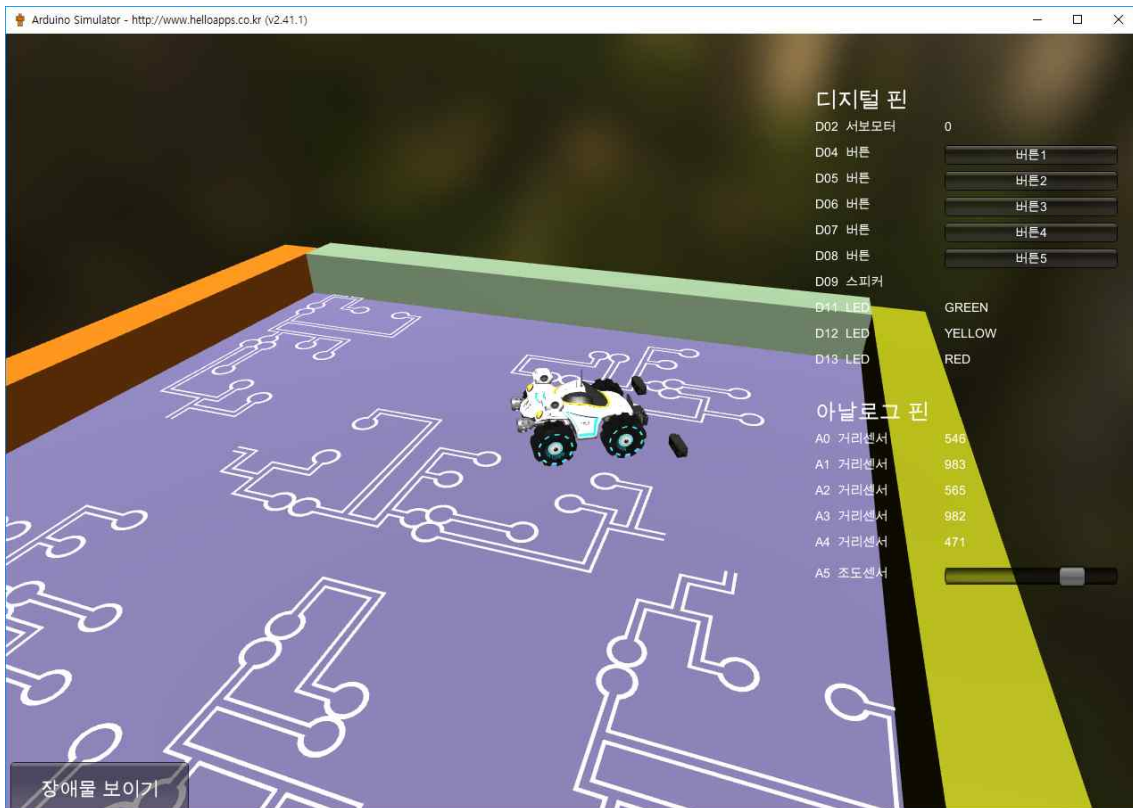
Function: Setup

- Expression: HideObstacle()
- Expression: AddCube(7, 0.5, 0, 1, 1, 14)
- Expression: AddCube(-7, 0.5, 0, 1, 1, 14)
- Expression: AddCube(0, 0.5, 7, 13, 1, 1)
- Expression: AddCube(0, 0.5, -7, 13, 1, 1)

Function: Loop

- Variable 'a' = AnalogRead 2 (핀번호)
- if a > 800
 - DriveWrite -150 (Left) -150 (Right)
 - Delay 1000 (밀리초)
 - DriveWrite 150 (Left) -150 (Right)
 - Delay 1000 (밀리초)
- DriveWrite 150 (Left) 150 (Right)
- Delay 100 (밀리초)

SPL 스크립트	스케치 코드
<pre> void setup() { HideObstacle() AddCube(7, 0.5, 0, 1, 1, 14) AddCube(-7, 0.5, 0, 1, 1, 14) AddCube(0, 0.5, 7, 13, 1, 1) AddCube(0, 0.5, -7, 13, 1, 1) } void loop() { a = AnalogRead(2) if (a > 800) { DriveWrite(-150, -150) Delay(1000) DriveWrite(150, -150) Delay(1000) } DriveWrite(150, 150) Delay(100) } </pre>	<pre> void setup() { HideObstacle(); AddCube(7, 0.5, 0, 1, 1, 14); AddCube(-7, 0.5, 0, 1, 1, 14); AddCube(0, 0.5, 7, 13, 1, 1); AddCube(0, 0.5, -7, 13, 1, 1); } void loop() { int a = analogRead(2); if (a > 600) { DriveWrite(-150, -150); delay(1000); DriveWrite(150, -150); delay(1000); } DriveWrite(150, 150); delay(100); } </pre>



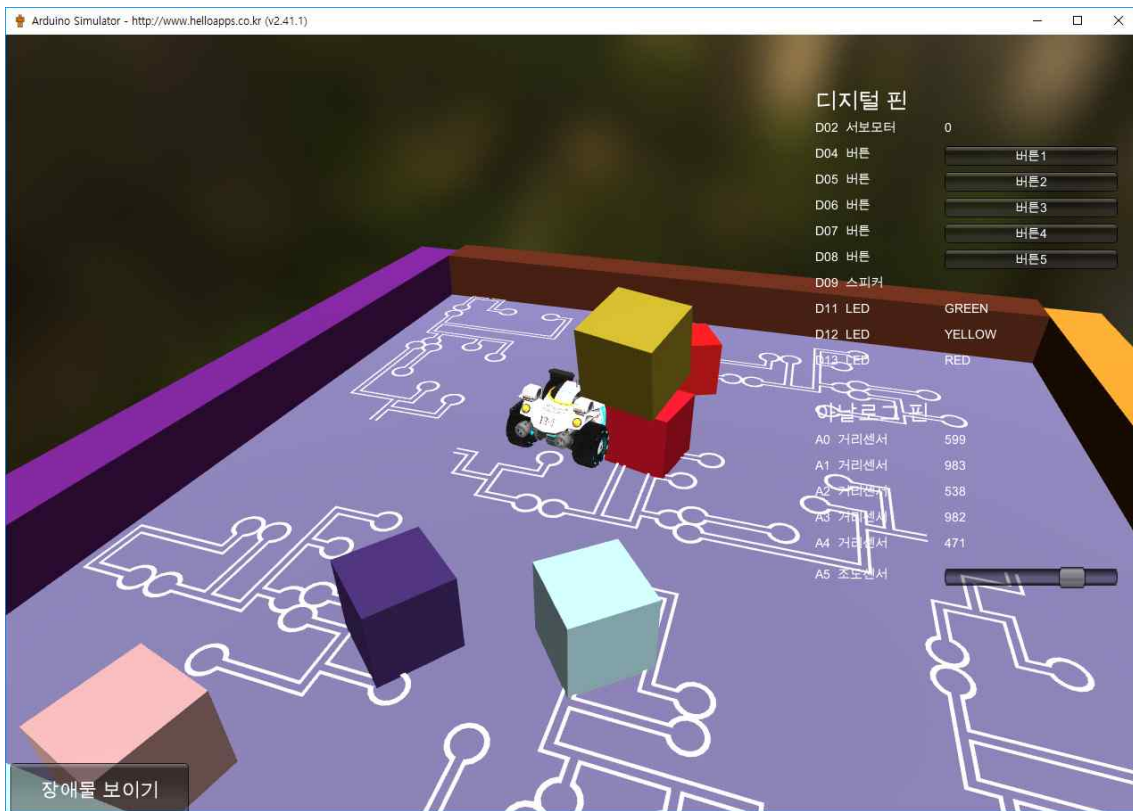
10.6 장애물 공 추가하기

장애물 공 추가하기

- 로봇이 장애물에 접근할 때 마다 동 모양을 하나씩 생성하여 떨어트리는 기능을 추가해 보자.

The image shows a Scratch-like code editor with two function blocks: 'Setup' and 'Loop'. The 'Setup' block contains five 'Expression' blocks: 'HideObstacle()', 'AddCube(7, 0.5, 0, 1, 1, 14)', 'AddCube(-7, 0.5, 0, 1, 1, 14)', 'AddCube(0, 0.5, 7, 13, 1, 1)', and 'AddCube(0, 0.5, -7, 13, 1, 1)'. The 'Loop' block starts with 'a = AnalogRead 2', followed by an 'if a > 800' condition. Inside the if block, there is an 'Expression' block 'AddCube(0, 5, 0)' highlighted with a red arrow, followed by 'DriveWrite -150 (Left) -150 (Right)', 'Delay 1000', 'DriveWrite 150 (Left) -150 (Right)', and 'Delay 1000'. Outside the if block, there is 'DriveWrite 150 (Left) 150 (Right)' and 'Delay 100'.

SPL 스크립트	스케치 코드
<pre> void setup() { HideObstacle() AddCube(7, 0.5, 0, 1, 1, 14) AddCube(-7, 0.5, 0, 1, 1, 14) AddCube(0, 0.5, 7, 13, 1, 1) AddCube(0, 0.5, -7, 13, 1, 1) } void loop() { a = AnalogRead(2) if (a > 800) { AddCube(0, 5, 0) DriveWrite(-150, -150) Delay(1000) DriveWrite(150, -150) Delay(1000) } DriveWrite(150, 150) Delay(100) } </pre>	<pre> void setup() { HideObstacle(); AddCube(7, 0.5, 0, 1, 1, 14); AddCube(-7, 0.5, 0, 1, 1, 14); AddCube(0, 0.5, 7, 13, 1, 1); AddCube(0, 0.5, -7, 13, 1, 1); } void loop() { int a = analogRead(2); if (a > 600) { AddCube(0, 5, 0); DriveWrite(-150, -150); delay(1000); DriveWrite(150, -150); delay(1000); } DriveWrite(150, 150); delay(100); } </pre>



10.7 나만의 장애물 공간 생성하기

창작 프로젝트 실습

- 다양한 3D 도형 명령어를 이용하여 나만의 장애물 공간을 생성해 본다.