아두이노 시뮬레이션 프로그래밍

v1.0

김영준 저

공학박사, 목원대학교 겸임교수

前 Microsoft 수석연구원

헬로앱스

http://www.helloapps.co.kr

12 로봇 조종기 구현하기

학습 목표

- 시뮬레이션 로봇의 주행을 버튼으로 제어할 수 있다.
- 로봇의 기능을 버튼으로 제어할 수 있다.

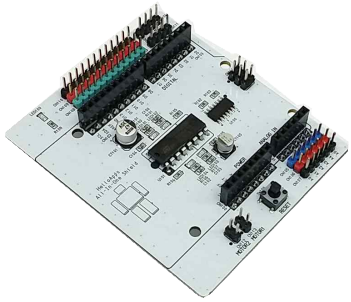
실습 개요

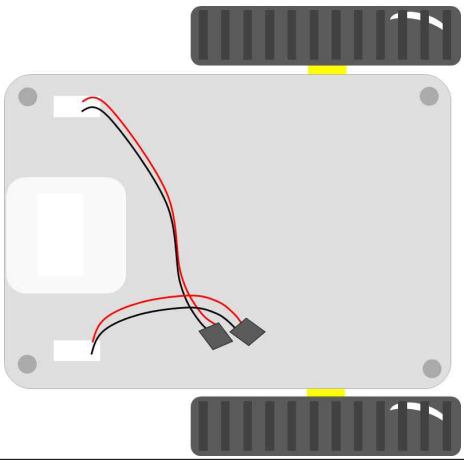
- 버튼을 활용한 로봇 주행 제어 기능을 구현해 본다.
- 로봇 조종기의 기능을 개선해 본다.
- 스마트 버튼 기능으로 장애물을 청소하는 기능을 만들어 본다.

12.1 준비하기

준비물

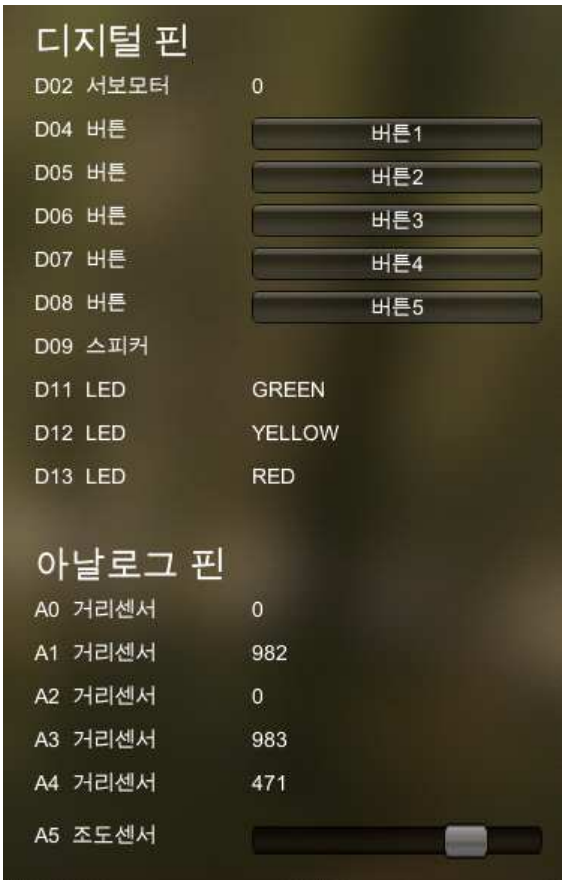
■ 아두이노 보드, 올인원 쉴드, 아날로그 PSD센서, LED 모듈, 스피커

	
아두이노 우노보드	올인원 쉴드

	
모바일 로봇 플랫폼	

시뮬레이션 상에서의 연결 정보

- 시뮬레이션 상에서는 디지털 버튼이 디지털 04번 ~ 08번에 연결되어 있으며, PSD 거리센서는 아날로그 0번 ~ 3번에 연결되어 있다.

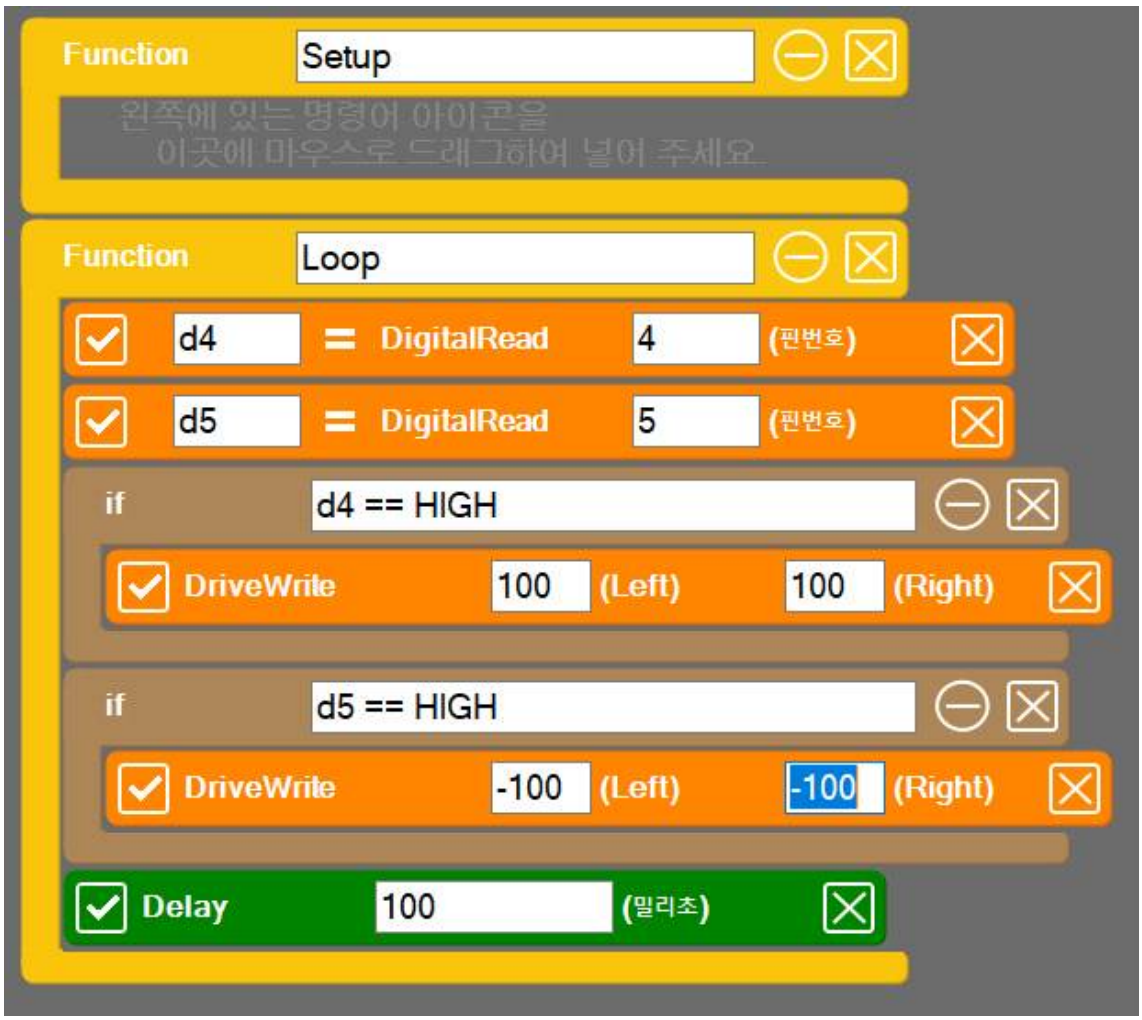


- 디지털/아날로그 핀에 연결된 부품
  - 디지털 4번 ~ 8번: 디지털 버튼
  - 아날로그 0번: 로봇 좌측 PSD 거리 센서
  - 아날로그 2번: 로봇 전방에 있는 PSD 거리 센서
  - 아날로그 4번: 차단기에 부착된 PSD 거리 센서

12.2 버튼으로 로봇 주행 제어하기

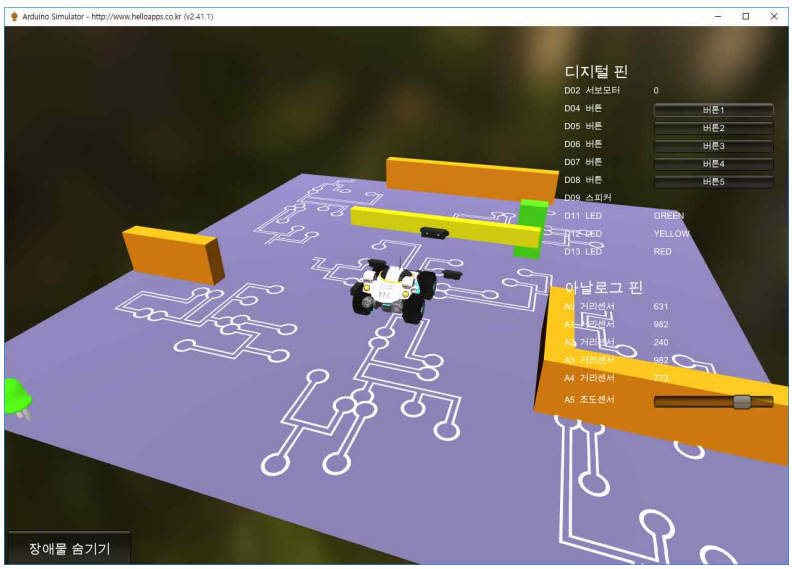
버튼으로 전후진 조종하기

- 버튼 1 (디지털 04번)과 버튼 2 (디지털 05번)를 이용하여 로봇을 전진 시키고 멈추는 기능을 구현해 본다.



SPL 스크립트

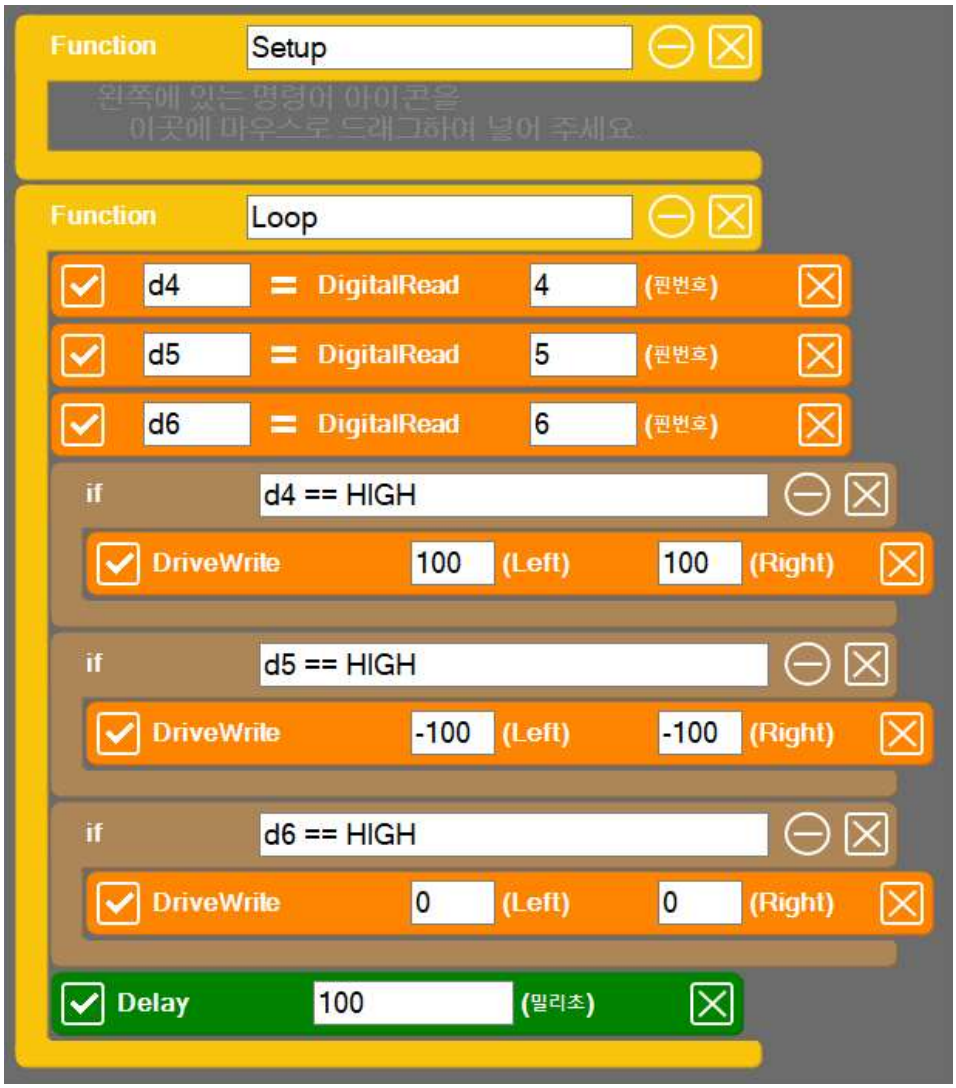
```
void setup()
{
}

void loop()
{
        d4 = DigitalRead(4)
        d5 = DigitalRead(5)

        if (d4 == HIGH)
        {
                DriveWrite(100, 100)
        }

        if (d5 == HIGH)
        {
                DriveWrite(-100, -100)
        }

        Delay(100)
}
```



■ 오른쪽 상단에 있는 “버튼1” 버튼 또는 “버튼2”버튼을 클릭하여 로봇의 동작을 관찰한다.

정지 기능 추가하기

■ 버튼 3(디지털 6번)에 정지 기능을 추가해 본다.



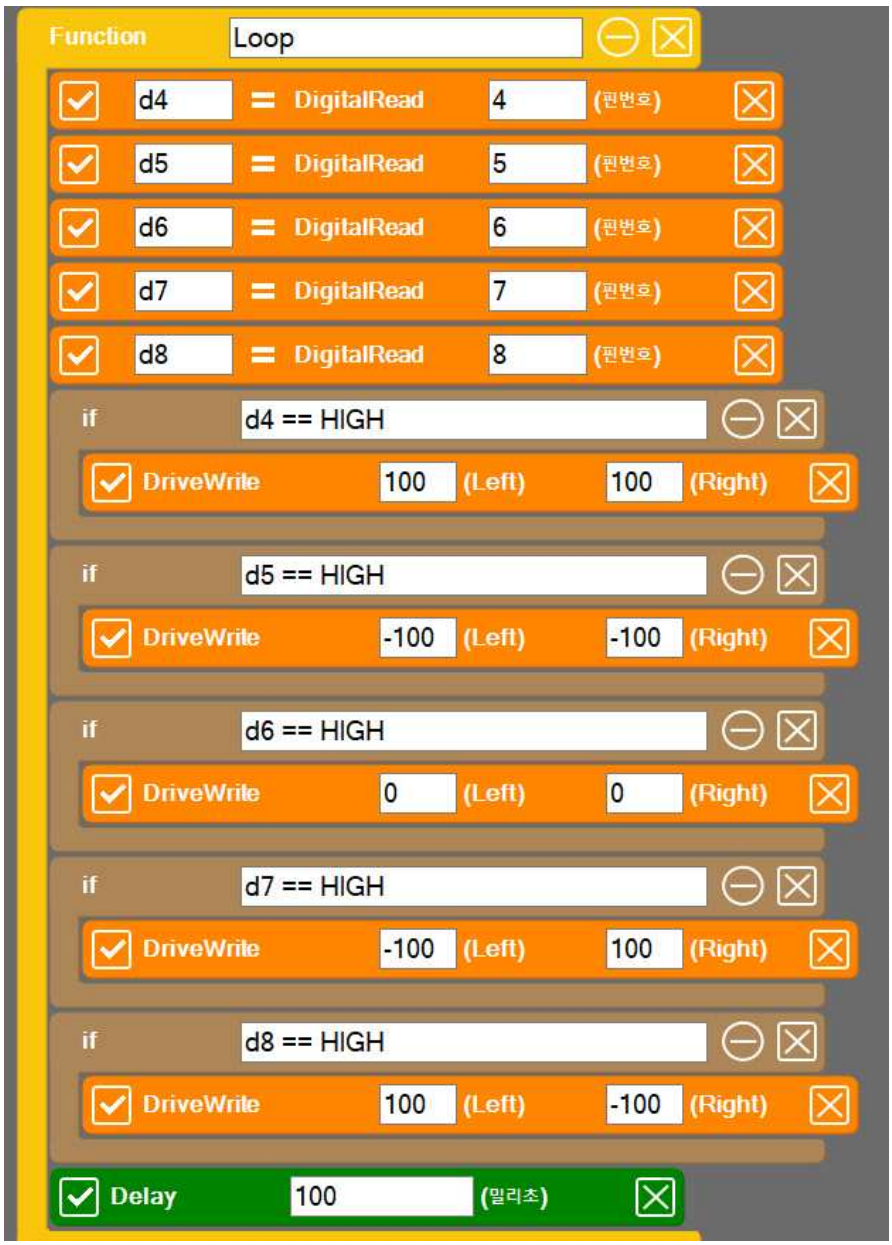
SPL 스크립트

```
void setup()
{
}

void loop()
{
        d4 = DigitalRead(4)
        d5 = DigitalRead(5)

        if (d4 == HIGH)
        {
                DriveWrite(100, 100)
        }

        if (d5 == HIGH)
        {
                DriveWrite(-100, -100)
        }

        if (d6 == HIGH)
        {
                DriveWrite(0, 0)
        }

        Delay(100)
}
```

■ 실행하여 기능을 확인해 본다.

회전 기능 추가하기

- 버튼 4(디지털 7번)와 버튼 5(디지털 8번)에 각각 회전 기능을 추가해 본다.



SPL 스크립트

```
void setup()
{
}

void loop()
{
        d4 = DigitalRead(4)
        d5 = DigitalRead(5)
        d6 = DigitalRead(6)
        d7 = DigitalRead(7)
        d8 = DigitalRead(8)

        if (d4 == HIGH)
        {
                DriveWrite(100, 100)
        }

        if (d5 == HIGH)
        {
                DriveWrite(-100, -100)
        }

        if (d6 == HIGH)
        {
                DriveWrite(0, 0)
        }

        if (d7 == HIGH)
        {
                DriveWrite(-100, 100)
        }

        if (d8 == HIGH)
        {
                DriveWrite(100, -100)
        }

        Delay(100)
}
```

로봇 조종기 기능 개선하기

- 기존의 기능은 버튼을 한번 누르면 해당 버튼의 기능이 계속 진행되기 때문에 로봇을 조종하기가 상당히 어렵다.

- loop 함수 맨 아래에 마지막 명령어로 다음과 같이 로봇을 정지시키는 명령어를 한 줄 추가하면 로봇이 버튼을 누르고 있는 동안에만 작동하기 때문에 훨씬 더 로봇을 조종하기가 수월해 진다.

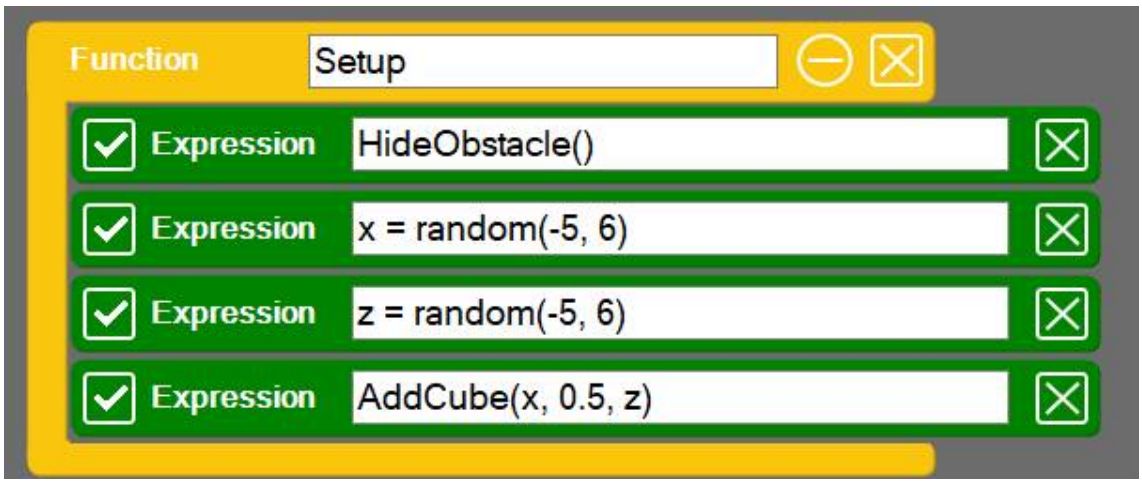
SPL 스크립트

```
void setup()
{


}

void loop()
{
        d4 = DigitalRead(4)
        d5 = DigitalRead(5)
        d6 = DigitalRead(6)
        d7 = DigitalRead(7)
        d8 = DigitalRead(8)

        if (d4 == HIGH)
        {
                DriveWrite(100, 100)
        }

        if (d5 == HIGH)
        {
                DriveWrite(-100, -100)
        }

        if (d6 == HIGH)
        {
                DriveWrite(0, 0)
        }

        if (d7 == HIGH)
        {
                DriveWrite(-100, 100)
        }

        if (d8 == HIGH)
        {
                DriveWrite(100, -100)
        }

        Delay(100)

        DriveWrite(0, 0)
}
```

12.3 장애물을 청소하는 로봇 구현하기

장애물 만들기

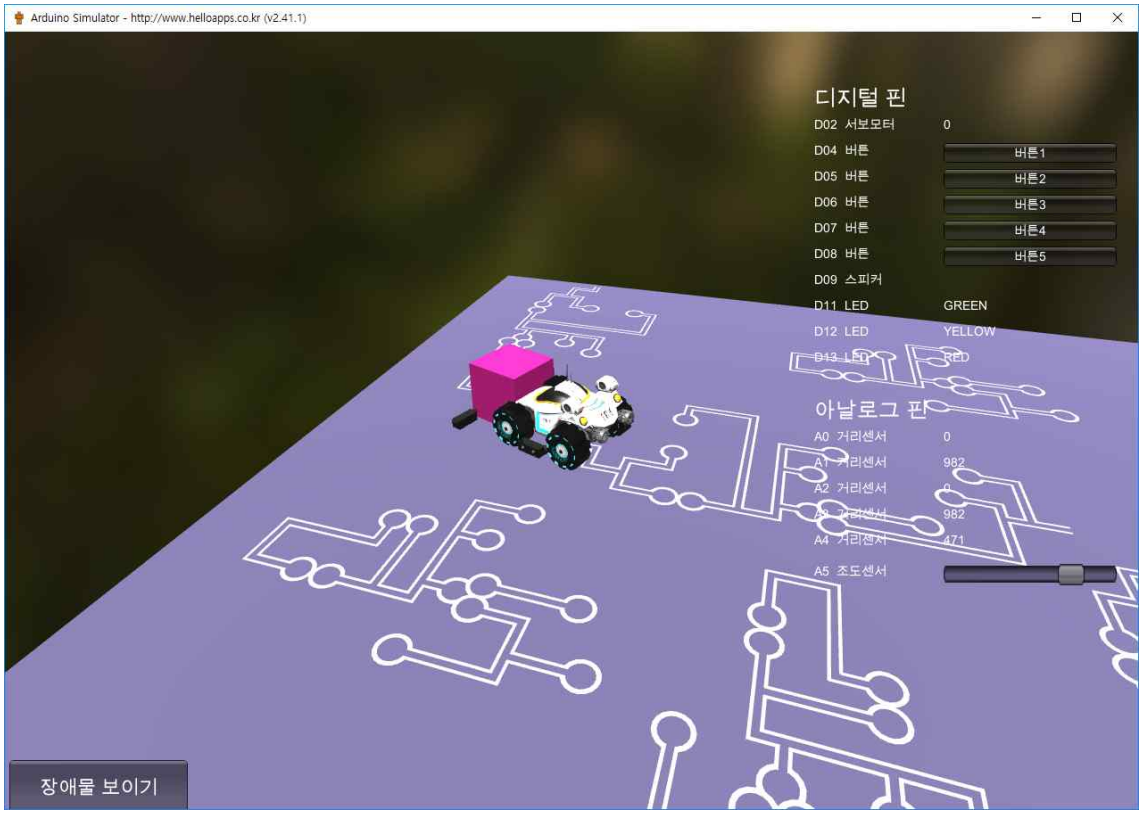
- 다양한 도형 명령으로 장애물을 임의의 위치에 만들어 본다.
- 다음과 같이 setup 함수 부분을 수정해 준다.
- random 함수는 임의의 숫자를 발생시키는 함수로서 아래 예에서는 -5와 5 사이의 임의의 정수 값을 생성시킨다. 6 값은 발생하지 안는다.



SPL 스크립트

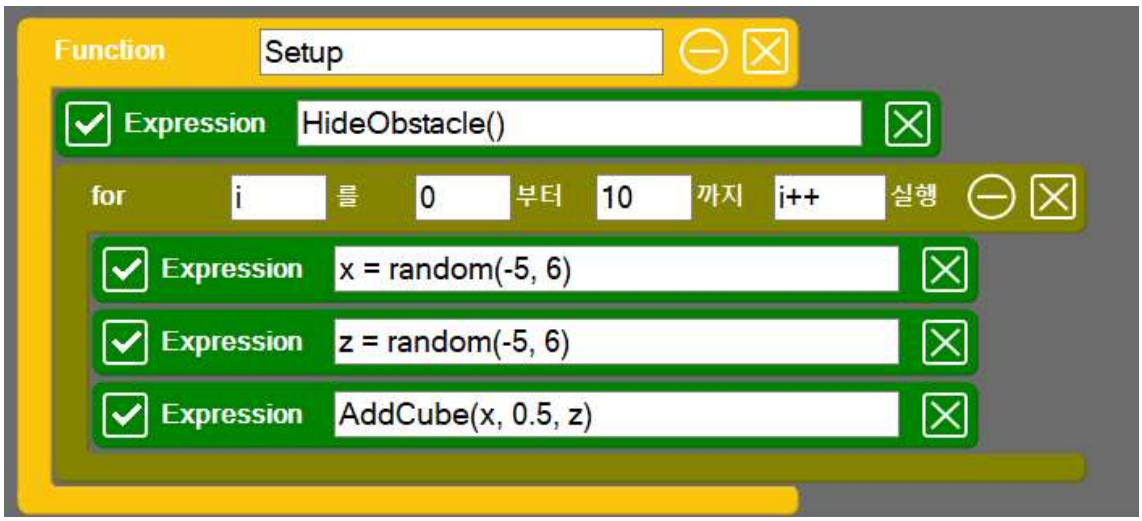
```
void setup()
{
	HideObstacle()

	x = random(-5, 6)
	z = random(-5, 6)

	AddCube(x, 0.5, z)
}

void loop()
{
	d4 = DigitalRead(4)
	d5 = DigitalRead(5)
	d6 = DigitalRead(6)
	d7 = DigitalRead(7)
	d8 = DigitalRead(8)

	if (d4 == HIGH)
	{
		DriveWrite(100, 100)
	}

	if (d5 == HIGH)
	{
		DriveWrite(-100, -100)
	}

	if (d6 == HIGH)
	{
		DriveWrite(0, 0)
	}

	if (d7 == HIGH)
	{
		DriveWrite(-100, 100)
	}

	if (d8 == HIGH)
	{
		DriveWrite(100, -100)
	}

	Delay(100)

	DriveWrite(0, 0)
}
```

■ 버튼으로 조종하여 장애물을 떨어트려 본다.



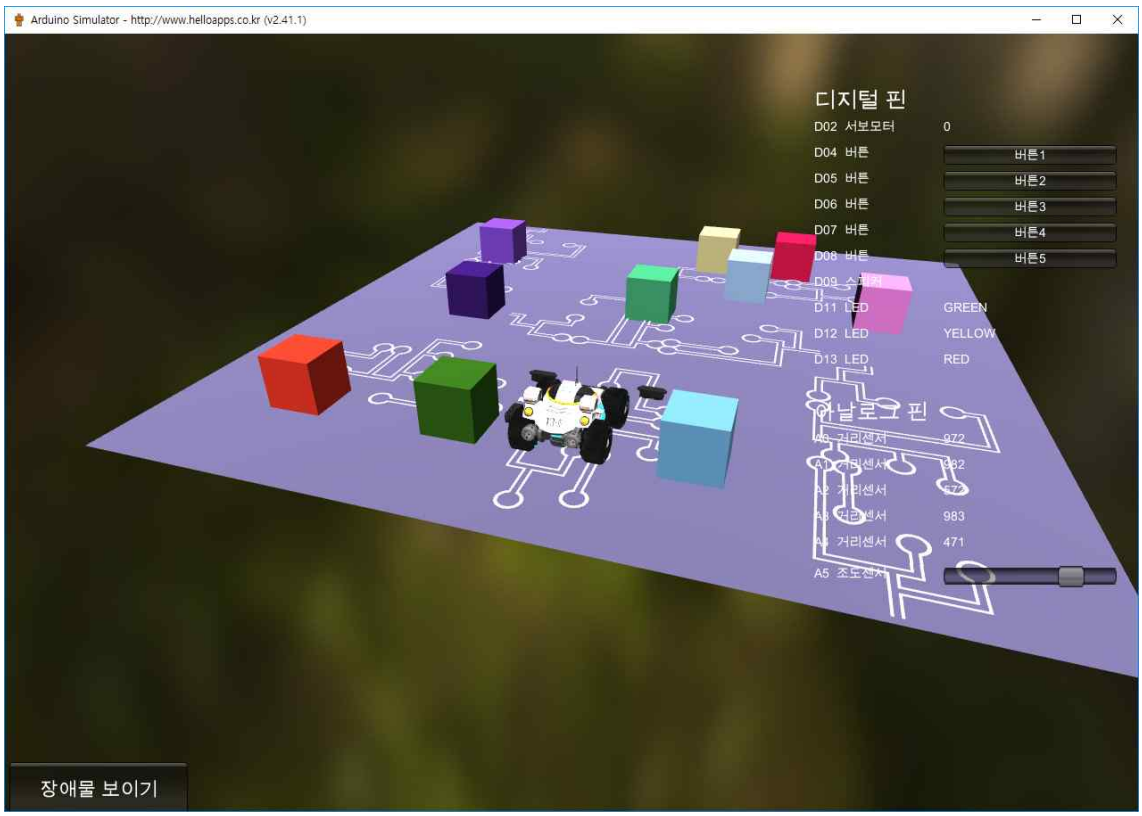
장애물 여러개 만들기

- 다음과 같이 setup 함수에 반복문을 추가하여 여러 개의 박스 모양을 임의의 위치에 추가해 준다.



SPL 스크립트

```
void setup()
{
	HideObstacle()

	for (i = 0; i < 10; i++)
	{
		x = random(-5, 6)
		z = random(-5, 6)
		AddCube(x, 0.5, z)
	}
}

void loop()
{
	d4 = DigitalRead(4)
	d5 = DigitalRead(5)
	d6 = DigitalRead(6)
	d7 = DigitalRead(7)
	d8 = DigitalRead(8)

	if (d4 == HIGH)
	{
		DriveWrite(100, 100)
	}

	if (d5 == HIGH)
	{
		DriveWrite(-100, -100)
	}

	if (d6 == HIGH)
	{
		DriveWrite(0, 0)
	}

	if (d7 == HIGH)
	{
		DriveWrite(-100, 100)
	}

	if (d8 == HIGH)
	{
		DriveWrite(100, -100)
	}

	Delay(100)

	DriveWrite(0, 0)
}
```



■ 박스들을 모두 떨어트려 본다.

박스의 개수 늘리기

■ 생성되는 박스의 개수를 늘려본다.

다른 모양 추가하기

■ 박스 외에 다른 모양의 도형도 추가해 본다.