

코딩체험교실

아두이노 로봇 코딩

4차산업기술 체험
(SW코딩/자율주행기술)

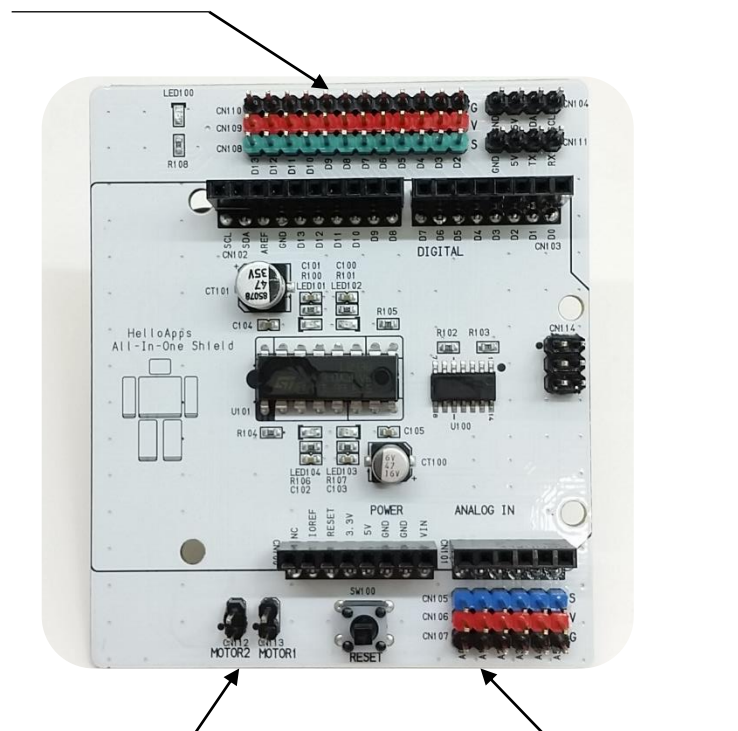
아두이노 소개

- 아두이노(Arduino)란?

- 2005년 이탈리아에서 만들어진 오픈 소스 하드웨어
- 초보자들이 전문 지식이 없어서 다양한 스마트 장치나 로봇을 만들 수 있도록 지원
- 국내 SW 교육에서 피지컬 컴퓨팅 단원에서 활용

- 아두이노 핀 설명

디지털 핀 (2번 ~ 13번)



로봇 모터 핀
MOTOR1, MOTOR2

아날로그 핀
(0번 ~ 5번)

실습 키트의 구성

- 아두이노 부품의 이름

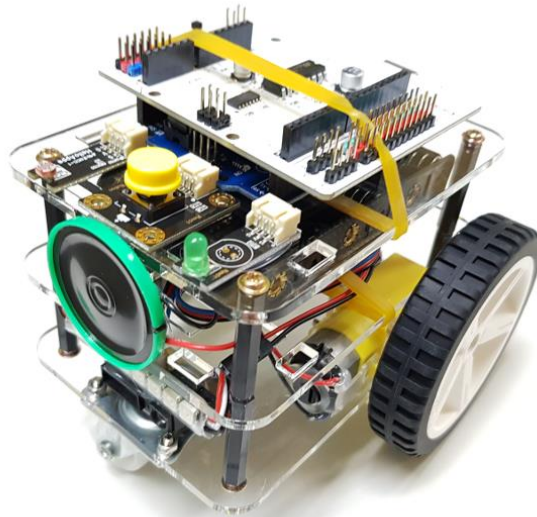
아날로그
조도센서



디지털
버튼센서



디지털
LED모듈



자율주행
로봇



아날로그
거리센서



디지털
스피커



디지털
컬러바

실습 키트의 구성

• 아두이노 부품의 용도

아날로그
조도센서



빛의 밝기를
측정합니다.

디지털
버튼센서



버튼의 눌림을
0과 1로 측정

디지털
LED모듈



0과 1로
LED 점멸

장애물과의
거리를
측정합니다.



아날로그
거리센서

소리를
발생시킵니다.



디지털
스피커

컬러
색상을
표시합니다.



디지털
컬러바

실습 키트의 구성

- 아두이노 부품의 연결 상태

아날로그
조도센서



아날로그
0번

디지털
버튼센서



디지털
2번

디지털
LED모듈



디지털
13번

아날로그
1번



아날로그
거리센서

디지털
3번



디지털
스피커

디지털
11번

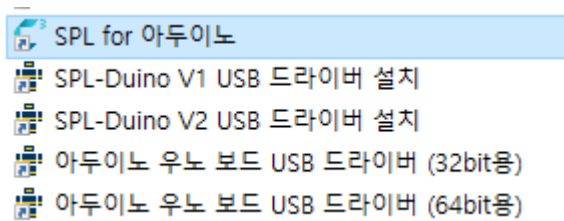


디지털
컬러바

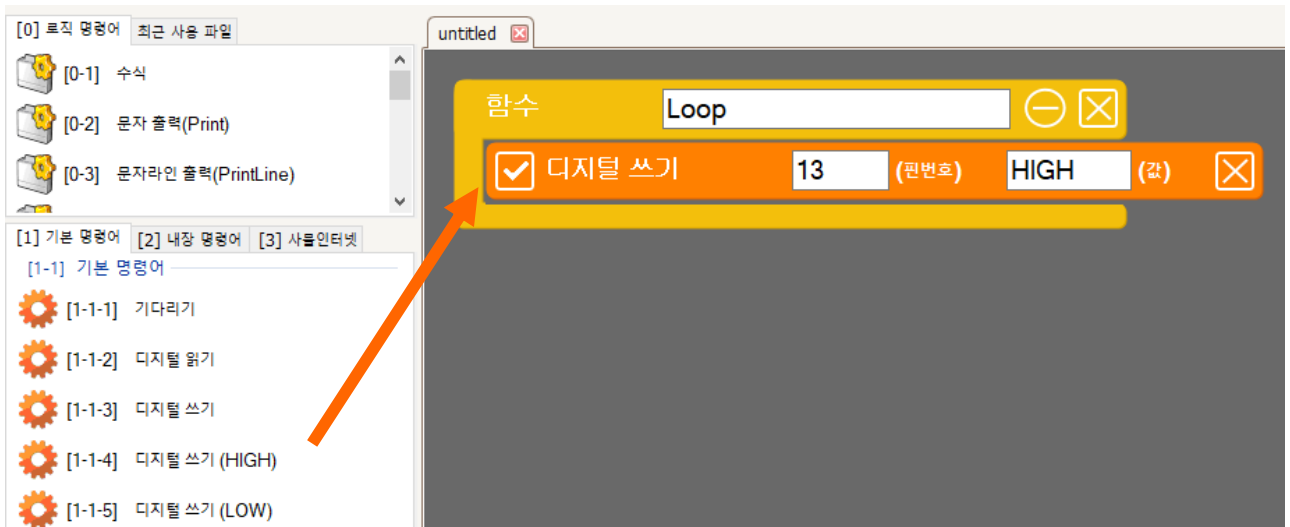
명령어 입력하기

• 아두이노 코딩 SW 실행하기

- 컴퓨터 바탕화면에 있는 SPL 이름의 폴더를 찾습니다.
- SPL 폴더 안에서 SPL for 아두이노 라는 프로그램을 실행시킵니다.



• 블록 편집기에서 명령어를 마우스로 이동시킵니다.



- 함수 Loop는 영역 안에 있는 명령어 들을 무한히 반복하여 실행시킵니다.

활동1) 깜박이는 LED

• 기본 명령어 설명

[1-1-4]

디지털 쓰기 13 (핀번호) HIGH (값)

- 13번 핀에 연결된 LED를 켭니다 (HIGH)
- HIGH: 전압을 높게하라는 의미입니다.

[1-1-5]

디지털 쓰기 13 (핀번호) LOW (값)

- 13번 핀에 연결된 LED를 끕니다 (LOW)
- LOW: 전압을 낮게하라는 의미입니다.

[1-1-1]

기다리기 1000 (밀리초)

- 1000밀리초(1초) 동안 실행을 멈추고 기다리라는 의미입니다.

• LED를 1초 마다 점멸하기

함수 Loop

디지털 쓰기 13 (핀번호) HIGH (값)

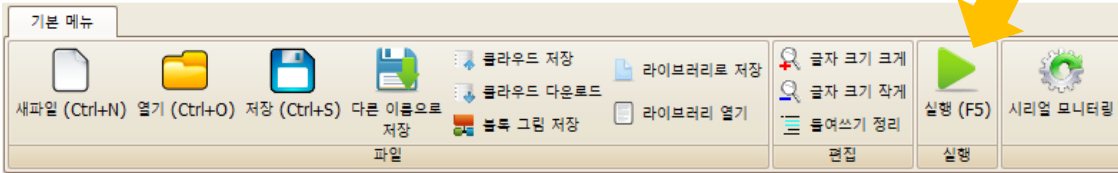
기다리기 1000 (밀리초)

디지털 쓰기 13 (핀번호) LOW (값)

기다리기 1000 (밀리초)

프로그램 실행하기

- 프로그램을 실행하기 위해서는 화면 상단 가운데에 있는 실행 아이콘을 클릭합니다.



- LED가 더 빠르게 점멸하도록 하려면 어느 값을 수정해야 할까요?

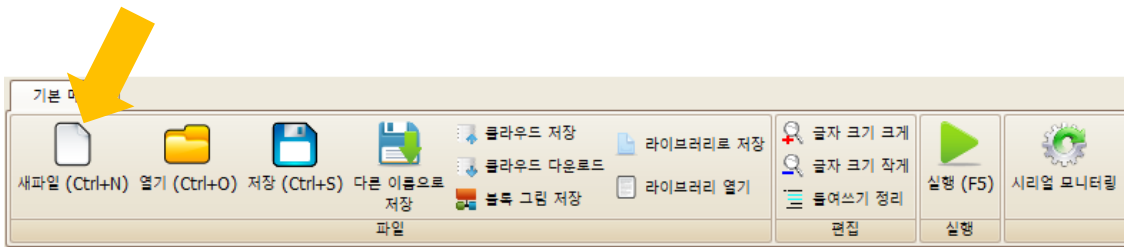


- 아주 빠른 속도로 깜빡이는 것이 보이도록 하려면 어느 정도 까지 값을 줄여야 할까요?
- LED 점멸은 일상 생활에서 어디에 응용될 수 있을까요?

활동2) 버튼으로 LED 켜기

• 새파일 만들기

- 왼쪽 맨 위에 있는 새파일 메뉴를 클릭하여 새로운 코딩 입력 창을 생성한 후, 프로그램을 작성합니다.



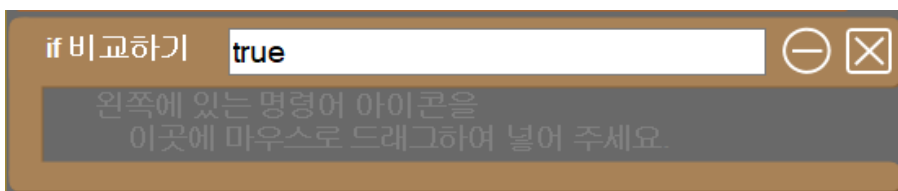
• 기본 명령어 설명

[1-1-2]

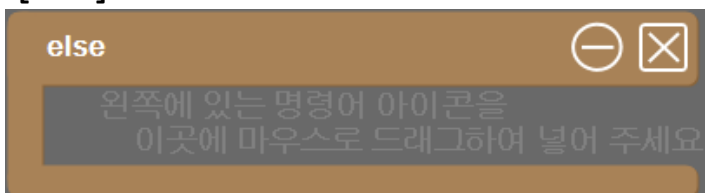


- 디지털 2번 핀에 연결된 센서에서 값을 읽어서 d라는 이름으로 저장합니다. d는 변수라고 부릅니다.

[0-6] - 조건을 비교하여 값이 참(True)이면 명령어를 실행합니다.

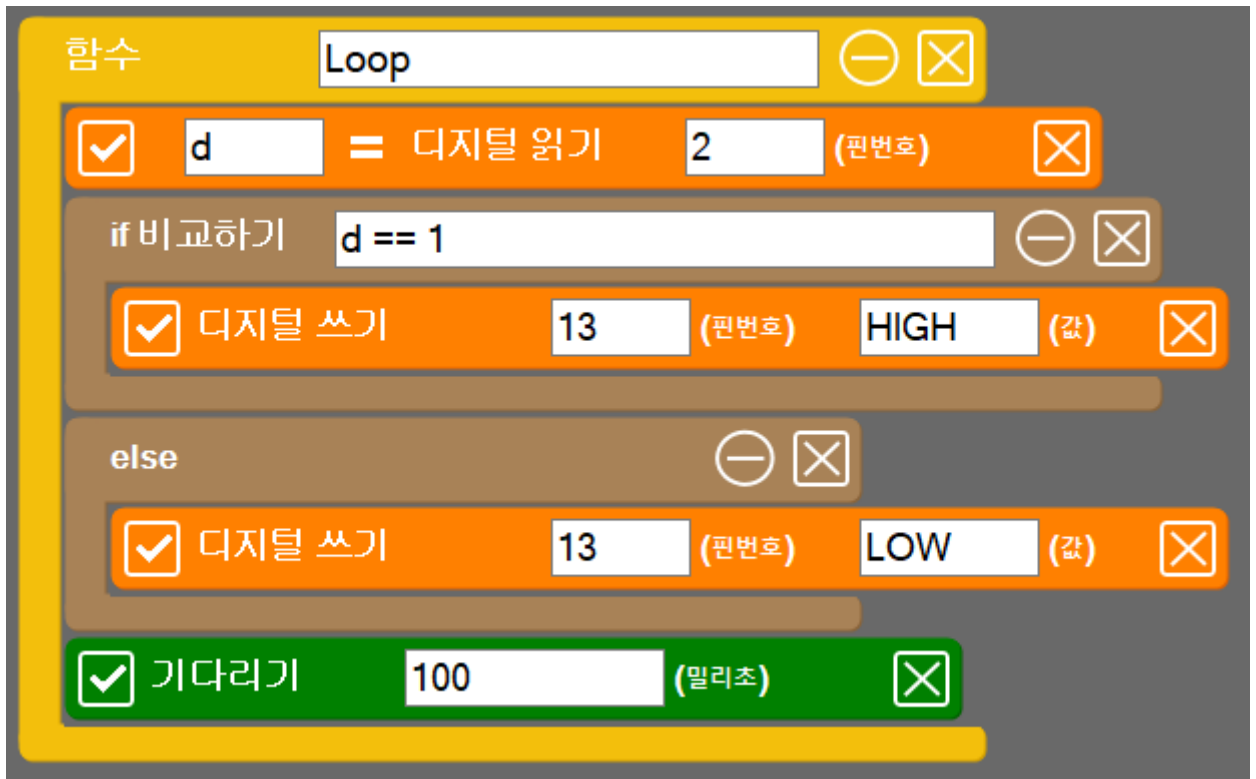


[0-8] - if 조건 비교문이 참이 아니면 else 그룹의 명령어를 실행합니다.



활동2) 버튼으로 LED 켜기

- 버튼 값을 비교하여 값이 1이면 LED를 켜고 그렇지 않으면 LED가 꺼지도록 기능을 구현합니다.



- LED는 디지털 2번 핀에 연결되어 있습니다.
- 값을 비교할 때에는 “==” 와 같이 “=” 기호를 2개 붙여서 사용합니다.
- 기다리기는 100밀리초 설정하여 0.1초 마다 위의 명령어들을 반복하도록 합니다.
- 버튼이 눌러지면 변수 d에는 1이, 그렇지 않으면 0이 저장됩니다. d가 1이면, 즉 d == 1이면 LED가 켜집니다.

활동3) 밝기 측정 및 LED 제어

- 아날로그 조도센서를 이용하면 밝기를 측정할 수 있습니다.

- 기본 명령어 설명

[1-1-6]

a = 아날로그 읽기 0 (핀번호)

- 아날로그 0번 핀에 연결되어 있는 조도센서에서 값을 읽어서 변수 a에 저장합니다.

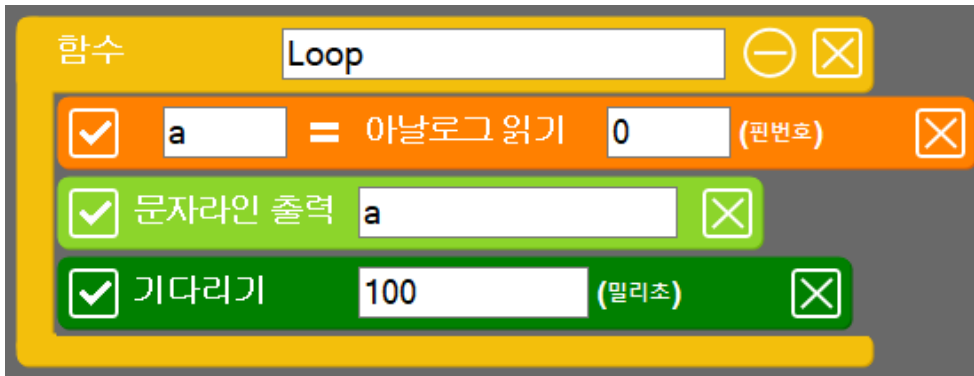
[0-3]

문자라인 출력 a

- 변수 a의 내용을 한 줄씩 콘솔창에 출력합니다.

활동3) 밝기 측정 및 LED 제어

- 밝기를 측정하는 기능을 구현해 봅니다.



- 어두워지면 변수 a의 값이 작아 집니다.

- 어두워 지면 저절로 켜지는 기능을 구현해 봅니다.



- 어두워지면 변수 a의 값이 작아 지며, 500보다 작으면 어두운 상태이기 때문에 LED를 켭니다. 그렇지 않으면 끕니다.

활동4) 멜로디 만들기

✓ 스피커는 디지털 3번 핀에 연결되어 있습니다.

• 톤[1-1-8] 명령어를 이용하여 멜로디 만들기

핀번호 진동수 소리나는 시간 (밀리초)

함수 Loop

- 톤(Tone) 3 (핀번호) 262 (진동수) 500 (지속시간)
- 기다리기 500 (밀리초)
- 톤(Tone) 3 (핀번호) 294 (진동수) 500 (지속시간)
- 기다리기 500 (밀리초)
- 톤(Tone) 3 (핀번호) 330 (진동수) 500 (지속시간)
- 기다리기 500 (밀리초)

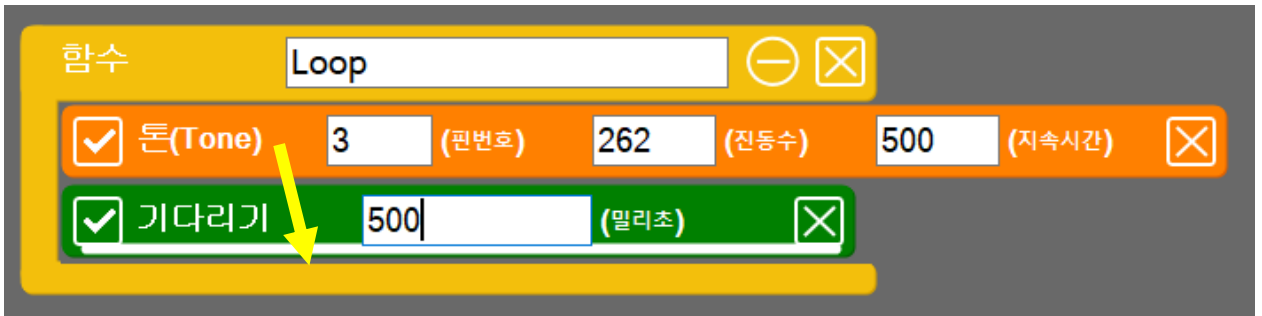
	도	레	미	파	솔	라	시					
	C	C#	D	D#	E	F	F#	G	G#	A	A3	B
0	16	17	18	20	21	22	23	25	26	28	29	31
1	33	35	37	39	41	44	46	49	52	55	58	62
2	65	69	73	78	82	87	93	98	104	110	117	124
3	131	139	147	156	165	175	185	196	208	220	233	247
4	262	278	294	311	330	349	370	392	415	440	466	494
5	523	554	587	622	659	699	740	784	831	880	932	988
6	1047	1109	1175	1245	1319	1397	1475	1568	1661	1760	1865	1976
7	2093	2218	2349	2489	2637	2794	2960	3136	3322	3520	3729	3951
8	4186	4435	4699	4978	5274	5588	5920	6272	6645	7040	7459	7902

• 위의 표에서 진동수 값을 찾아 봅니다.

활동4) 멜로디 만들기

- 블록 명령어 복사하기

- 시프트 (Shift) 키를 누른채 블록 명령어를 이동하면 해당 블록 명령어가 복사됩니다.



- 시프트 (Shift) 키를 누르고 있는 상태에서 계속 블록 명령어를 복사합니다.

활동5) 멜로디 만들기 실습

자 전 거

♬ 풀잎 동요마을

목일신 작사
김대현 작곡

1. 따 르 릅 따 르 릅 비 켜 나 세 요
2. 따 르 릅 따 르 릅 이 자 전 거 는

자 전 거 가 나 갑 니 다 따 르 르 릅
을 아 버 지 장 에 갔 다 돌 아 오 시 면

저 기 가 는 저 사 람 조 심 하 셔 요
교 부 랑 - 교 부 랑 고 개 를 넘 어

어 물 어 물 하 다 가 는 큰 일 납 니 다
비 탈 길 로 스 르 르 르 타 고 온 다 오

• 사이렌 소리 만들기

함수 Loop

for 반복 i 를 1 부터 100 까지 i=i+1 실행

수식 $f = i * i$

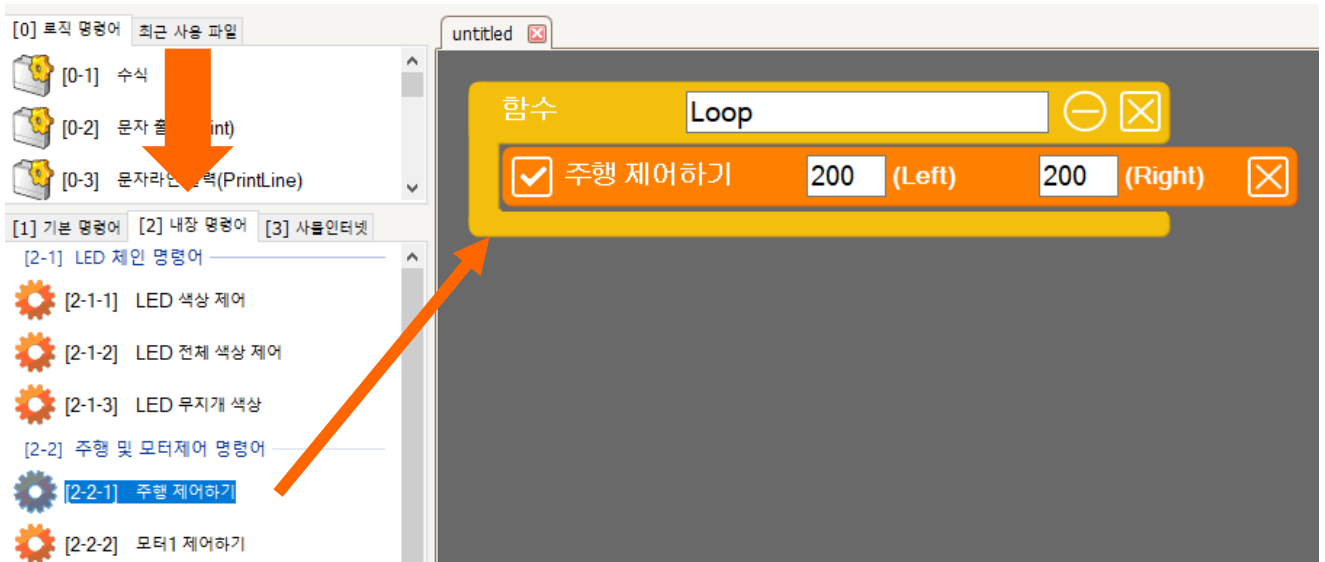
톤(Tone) 3 (편번호) f (진동수) 100 (지속시간)

기다리기 1 (밀리초)

- [0-9] For 반복문을 추가한 후, 값을 100으로 수정합니다.
- [0-1] 수식 명령어를 추가한 후, $f = i * i$ 수식을 입력합니다.
- 기다리기 값을 1 ~ 10 사이로 변화시켜 봅니다.

활동6) 로봇 주행 제어 실습

- 로봇 주행 제어 명령어 추가하기



- [2] 내장명령어 탭을 클릭합니다.
- [2-2-1] 에 있는 주행제어하기 명령어를 추가합니다.

[2-2-1]



왼쪽 모터값
-255 ~ 255

오른쪽 모터값
-255 ~ 255

값	설명
255, 255	최대 속도 전진
200, 200	보통 속도 전진
0, 0	정지
-200, -200	후진
200, -200	우회전
-200, 200	좌회전

활동6) 로봇 주행 제어 실습

- 1초간 전진 -> 1초간 정지 기능을 반복해 봅니다.

함수 Loop

- 주행 제어하기 200 (Left) 200 (Right)
- 기다리기 1000 (밀리초)
- 주행 제어하기 0 (Left) 0 (Right)
- 기다리기 1000 (밀리초)

- 아래의 동작을 반복해 봅니다.

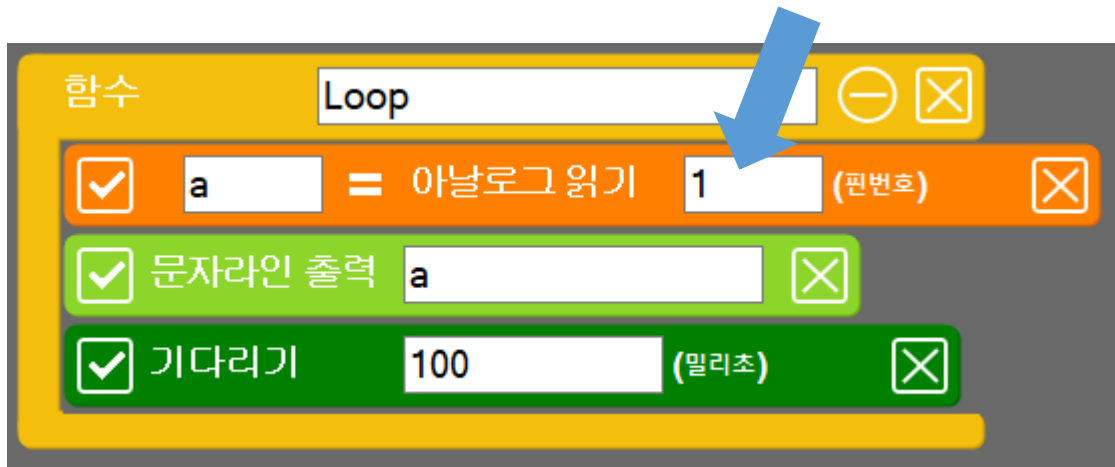
1초 전진 -> 1초 후진 -> 1초 회전

함수 Loop

- 주행 제어하기 200 (Left) 200 (Right)
- 기다리기 1000 (밀리초)
- 주행 제어하기 -200 (Left) -200 (Right)
- 기다리기 1000 (밀리초)
- 주행 제어하기 200 (Left) -200 (Right)
- 기다리기 1000 (밀리초)

활동7) 장애물 거리 측정하기

- 아날로그 1번에 연결된 거리센서의 센서값을 화면에 출력해 봅니다.



- 장애물과의 거리가 멀면 값이 작고, 거리가 가까워지면 값이 더 커집니다.

활동8) 로봇 자율 주행 실습

- 장애물을 만나면 피하는 로봇

The image shows a Scratch-style code editor with a yellow function block titled "함수 Loop". The code inside the loop is as follows:

- `a` = 아날로그 읽기 `1` (핀번호)
- if 비교하기** `a > 300`
 - 주행 제어하기 `-200` (Left) `-200` (Right)
 - 기다리기 `1000` (밀리초)
 - 주행 제어하기 `200` (Left) `-200` (Right)
 - 기다리기 `1000` (밀리초)
- 주행 제어하기 `200` (Left) `200` (Right)
- 기다리기 `100` (밀리초)

- 프로젝트 실습

✓ 로봇이 전진할 때 LED가 깜박이도록 해 봅니다.