

# 코딩키트

블록 코딩에서 사물인터넷 코딩까지  
쉬고 재미있게 배울 수 있어요.

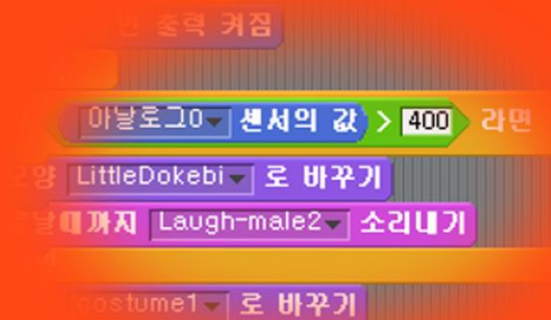
블록 코딩 / 아두이노 코딩 / 라즈베리파이 코딩 / 사물인터넷 코딩



## 코딩북2 - 블록코딩 스크래치



아빠와 함께하는 곰곰이의 코딩 놀이



SISO  
DREAM  
Silicon to Software

(주)시소드림 SISODREAM Inc.

- 본 문서의 저작권 -

본 문서에 대한 모든 저작권은 (주)시소드림에 있습니다. 본 문서는 자유롭게 배포할 수 있습니다. 단, 상업적인 목적으로 이용을 하거나 판매를 할 수는 없습니다. 또한 문서를 수정하여 배포할 수 없으며 인용할 경우 출처를 밝혀주시고, 인용한 부분이 1 페이지 이상 혹은 5 곳 이상일 경우에는 본 문서의 저작권자인 (주)시소드림에 허가를 득해야 합니다. 본 문서를 인쇄하여 누적수량 5 부 이상 배포할 경우에도 (주)시소드림의 허가를 득해야 합니다. 본 문서에 대한 이러한 사항이 지켜지지 않을 경우 민형사상의 책임을 물을 수 있습니다.

발행일 : 2017 년 7 월 27 일 (초판)

발행처 : (주)시소드림

연락처 : [ck@sisodream.com](mailto:ck@sisodream.com) / 031-757-7755

- 목 차 -

1. 아빠! 코딩이 뭐예요? .....	6
2. 블록 놀이? 블록 코딩!.....	7
3-1. S4A 프로그램 살펴보기 .....	8
3-2. S4A 맞보기 .....	12
4. 코딩키트! 코딩 놀이 친구 .....	21
4-1. 코딩키트 살펴보기.....	22
4-2. 코딩키트 동작시켜보기 .....	23
5. 깜박이는 LED 등 만들기.....	25
5-1. 코딩으로 LED를 켜보자 .....	26
5-2. LED가 깜박이는 동작을 만들어 보자.....	28
6. 초인종을 만들기 .....	32
6-1. 버튼이 눌리면 LED 가 켜지도록 해보자.....	33
6-2. 버튼을 누르면 “딩동” 소리가 나게 해보자.....	37
7. 크리스마스 전등 만들기 .....	44
7-1. 캐롤이 나오며 깜박이는 크리스마스 전등을 .....	45
만들어 보자.....	45
8. 도와주세요! 경보기.....	51
8-1. 부저 소리를 내보자 .....	52
8-2. 경보기 만들기 .....	55
9. 디지털과 아날로그.....	61
9-1. 가변 저항으로 화면을 조절하기 .....	63
9-2. 가변 저항으로 음량을 조절하기 .....	66
10. 부릉~ 부릉~ 자동차! .....	75
10-1. 바퀴를 움직여 볼까? .....	75

10-2.	자동차를 만들어 보자.....	78
11.	끼익~ 끼익~ 로봇팔! .....	87
11-1.	서보 모터를 움직여 보자.....	88
11-2.	원격조정 로봇 팔을 만들자.....	89
12.	인공지능 전등 만들기.....	95
12-1.	신기한 밝기 센서 .....	96
12-2.	인공지능 전등을 만들자.....	97
13.	잠자는 고양이 깨우기.....	101
13-1.	쉿 조용히.. 소리 센서.....	102
13-2.	잠자는 고양이를 깨우기 놀이를 해보자.....	104
14.	똑똑한 우리집 냉난방 시스템 .....	110
14-1.	온도를 알려드려요. 온도 센서. ....	111
14-2.	더울 때는 선풍기를, 추울 때는 난로를 켜자.....	112
15.	마법의 도깨비 집 .....	116
15-1.	폼작마! 적외선 센서 .....	117
15-2.	마법의 도깨비 상자를 만들자 .....	120
부록 A.	스피드 가이드.....	127
부록 B.	생각하기 해답.....	134
부록 C.	프로그램 설치 및 사용 가이드.....	140
1.	S4A(Scratch for Arduino) 소개 및 설치 .....	140
2.	아두이노 프로그램 설치하기 .....	147
3.	코딩킷에 S4AFirmware16 설치하기 .....	154
4.	코딩킷과 S4A 연결하기 .....	160
부록 D :	스위칭(Switching) ID .....	165
Release Note.....		167





# 1. 아빠! 코딩이 뭐예요?

곰곰이가 아빠에게 코딩에 관하여 묻습니다.



아빠! 코딩이 뭐예요?



응~ 코딩은 컴퓨터에게 일을 시키는 거야!

그런데, 컴퓨터는 사람 말을 알아듣지 못해. 그래서, 컴퓨터가 알아들을 수 있는 말을 사용해서 일을 시키지. 그것을 코딩이라고 해. 코딩 작업을 통해서 로봇을 움직이게 하고, 인터넷이나 게임을 할 수 있게 되는 거란다.

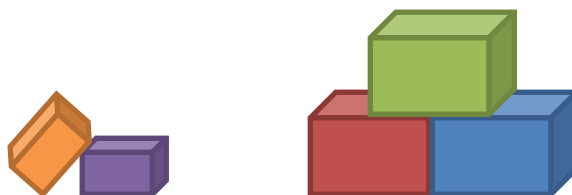


우와! 로봇을 움직이게 할 수 있다니..., 저도 코딩을 할 수 있다면 좋겠어요! 하지만, 너무 어려울 것 같아요.



코딩이 어려운 것만 있는 것은 아니란다. 곰곰이가 좋아하는 블록 놀이 같이 쉽고 재미있게 할 수 있는 코딩도 있단다.

곰곰이는 아빠와 함께 코딩을 배우기로 했습니다. 그리고, 코딩을 하기 위해 필요한 것이 무엇이 있는지 알아보기로 했습니다.



## 2. 블록 놀이? 블록 코딩!



요즘에는 어린이들이 쉽게 코딩을 배울 수 있는 프로그램이 많이 있단다. 그런 프로그램으로는 스크래치(Scratch), S4A(Scratch for Arduino), 스냅(Snap), 엔트리(Entry) 등이 있지. 이 프로그램들은 모두 블록 놀이를 하듯이 코딩을 할 수 있게 해준다.



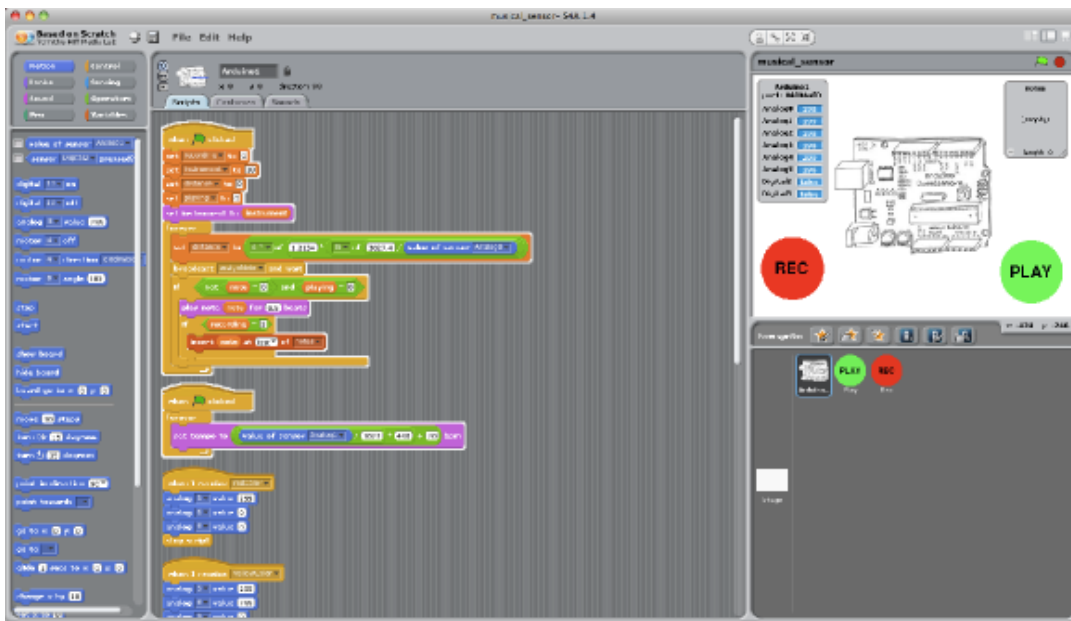
코딩이 블록 놀이랑 같다면 정말 재미있을 것 같아요!



블록 코딩 프로그램들 중에서, 블록 코딩도 배우고, 아두이노라는 작은 컴퓨터도 자유롭게 다룰 수 있도록 해주는 프로그램이 있더구나. 곰곰이가 로봇이나 자동차에 관심이 많으니 이 프로그램을 사용하여 코딩 공부를 해보는 것이 좋을 것 같다.

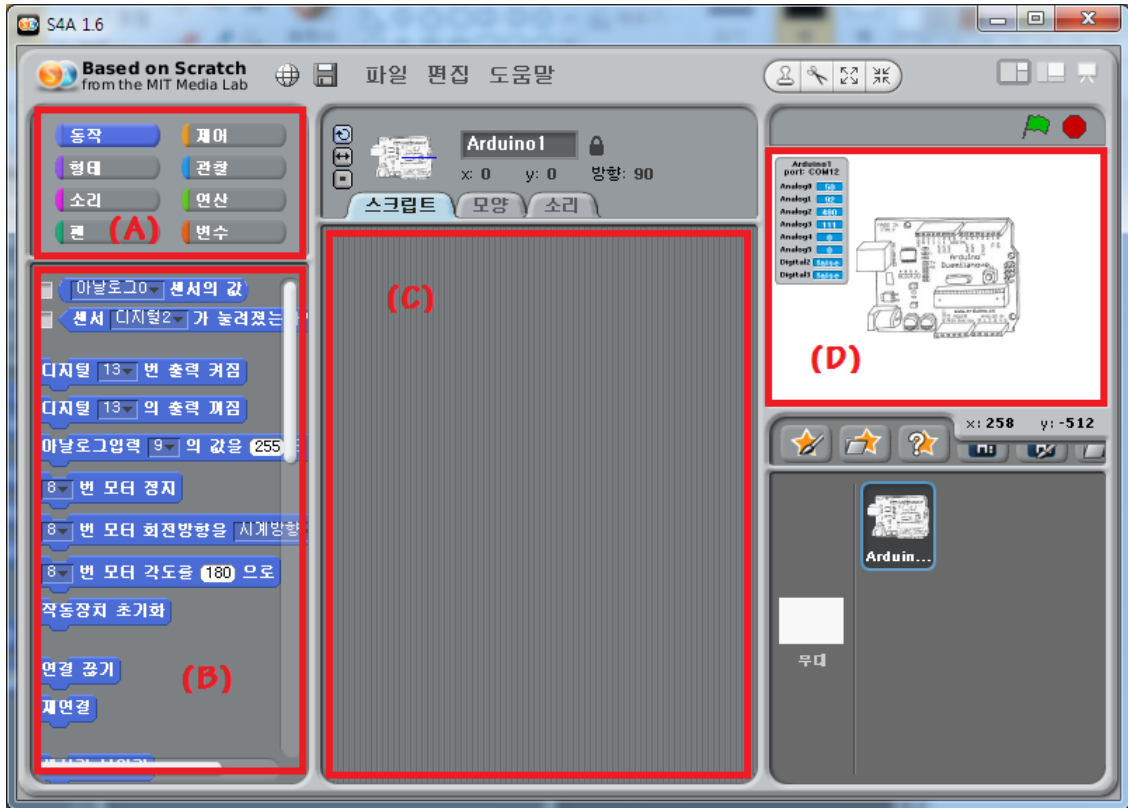
### 3-1. S4A 프로그램 살펴보기

블록 프로그램인 S4A(Scratch for Arduino) 프로그램에 대해 간단히 살펴보겠습니다. S4A 는 소형 컴퓨터인 아두이노를 다룰 수 있는 기능이 추가된 스크래치(Scratch) 프로그램입니다. 아두이노와 연결하여 다양한 장치들을 컨트롤 할 수 있습니다. 사용법은 스크래치와 동일합니다.



\* S4A 프로그램을 설치하는 과정은 “부록 C. 프로그램 설치 가이드”에 자세히 설명되어 있습니다. 아직 컴퓨터에 S4A 프로그램이 설치되어 있지 않다면 지금 설치해 주세요. 아빠가 도와 주세요!!

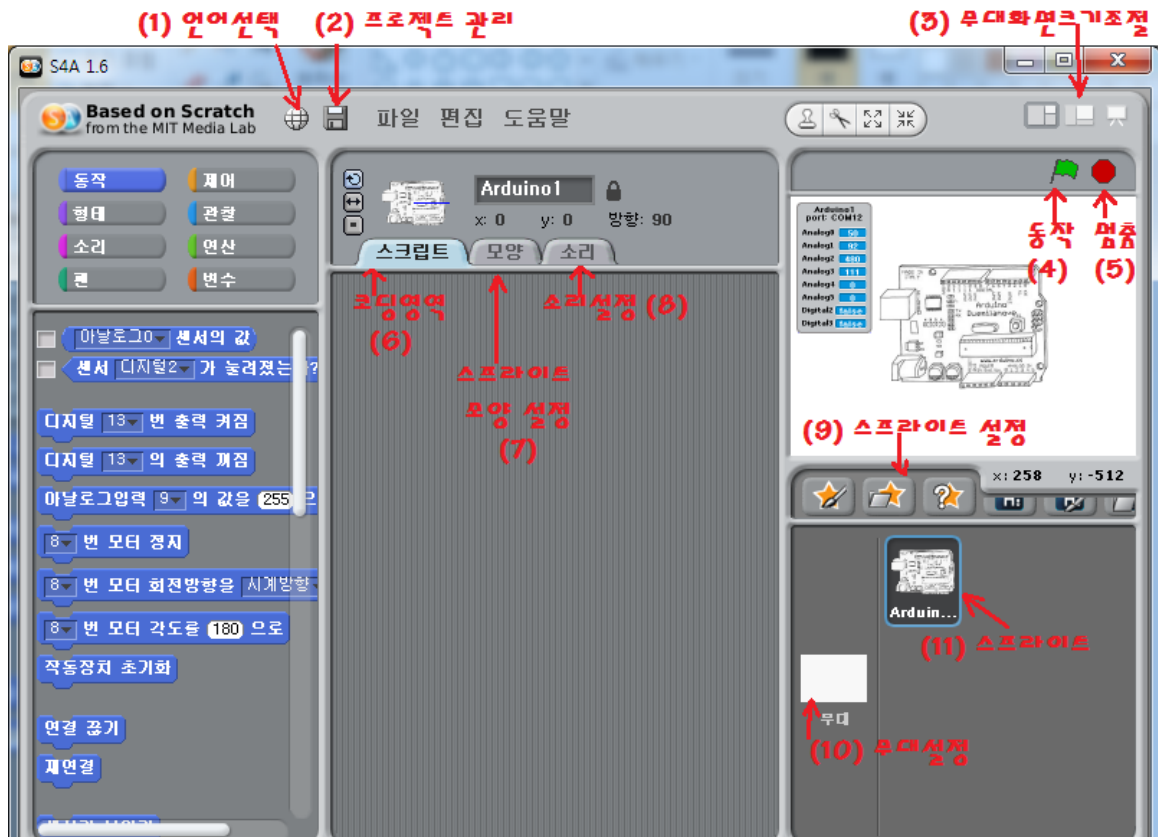
S4A 프로그램은 크게 블록 영역(A+B), 스크립트 영역(C), 무대 영역(D)으로 나누어 집니다.



블록 영역(A+B)은 스크립트를 작성하기 위한 블록들이 있습니다. 이 블록들은 특성에 따라 동작, 제어, 형태, 관찰, 소리, 연산, 펜, 변수로 구분되어 있습니다. 이 블록들은 드래그-앤-드롭 방법으로 스크립트 영역(C)으로 가져다 놓으면 동작을 하게 됩니다. S4A 에서 코딩이란 블록 영역에서 원하는 블록들을 스크립트 영역으로 가져와서 순서대로 배치하는 것입니다. 무대 영역(D)은 컴퓨터 화면에 블록 코딩에 따른 동작을 표현하기 위해서 사용되어 집니다.

지금부터 S4A 프로그램의 동작 및 기능을 간단히 설명할 것입니다. 지금은 이런 것이 있구나. 하는 정도로 이해하고 넘어 가셔도 됩니다. 실제 블록코딩을 하면서 프로그램에 익숙해질 것입니다.

다음 그림에서 보여지는 프로그램의 버튼과 아이콘들은 여러분들이 코딩을 하실 때 필요로 하는 기능들을 편리하고 빠르게 사용할 수 있도록 도와줍니다.



- (1) 언어 선택 : 각 국의 언어에 맞는 인터페이스로 바꿀 수 있습니다. 위 화면은 한국어를 선택한 화면입니다.
- (2) 프로젝트 관리 : 프로젝트의 저장, 열기 등의 작업을 수행합니다.
- (3) 무대 화면 크기 조절 : 무대의 크기를 작게 또는 크게 보이도록 화면을 조절해 줍니다.
- (4) 동작 : 스크립트를 동작 시킵니다.
- (5) 멈춤 : 스크립트의 동작을 멈추게 합니다.
- (6) 코딩 영역 탭 : 스크립트를 작성하는 창 입니다. 블록들을 이 창으로 드래그해서 순서대로 배치합니다. 이것이 블록 코딩입니다.
- (7) 모양 탭 : 이 탭에서 스프라이트나 무대 배경에 사용할 그림들을 관리합니다.
- (8) 소리 탭 : 이 탭에서 스크립트에서 사용할 소리 파일들을 관리합니다.
- (9) 스프라이트 설정 : 새로운 스프라이트를 설정하거나 변경하고자 할 때 사용합니다.



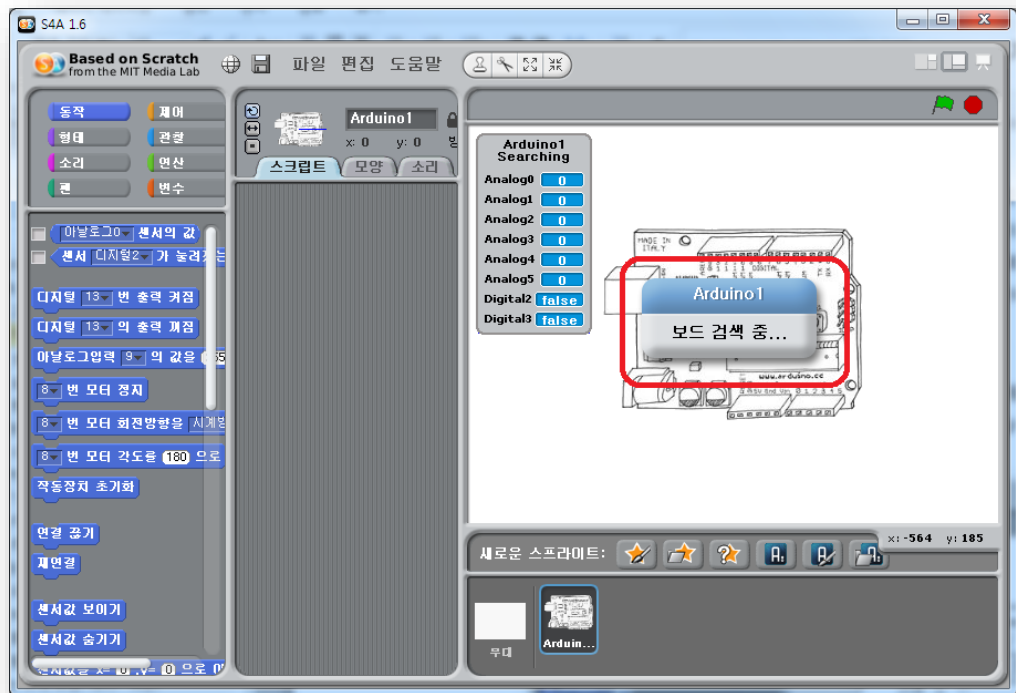
- 
- (10) 무대 화면 설정 : 이 버튼을 누르면 무대 화면을 만들거나 수정하는 작업을 할 수 있습니다.
- (11) 스프라이트 설정 : 이 버튼을 누르면 스크립트 영역이 해당 스프라이트의 작업 영역으로 설정됩니다.

## 3-2. S4A 맛보기

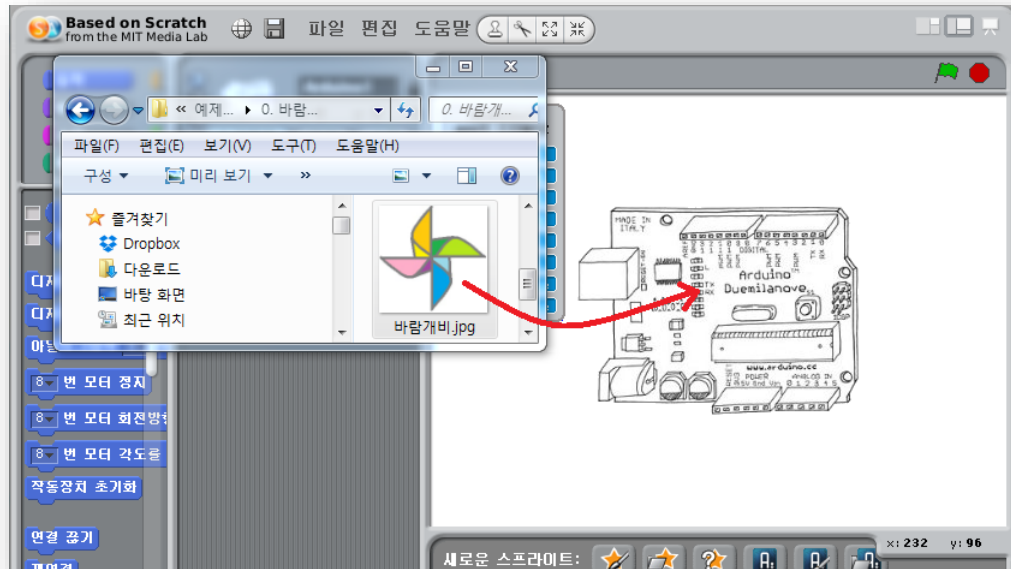
“백문이 불여일견”이라는 말이 있지요? 구구절절한 설명보다는 간단한 S4A 코딩을 직접 해보는 것이 이해하기가 쉬울 것입니다. 그러면, 빙글빙글 돌아가는 바람개비를 만들어 보겠습니다. 어려워하지 마세요. 초보자도 쉽게 만들 수 있습니다. 그럼, 천천히 따라와 보세요.

### 1 단계 : 바람개비 그림 올리기

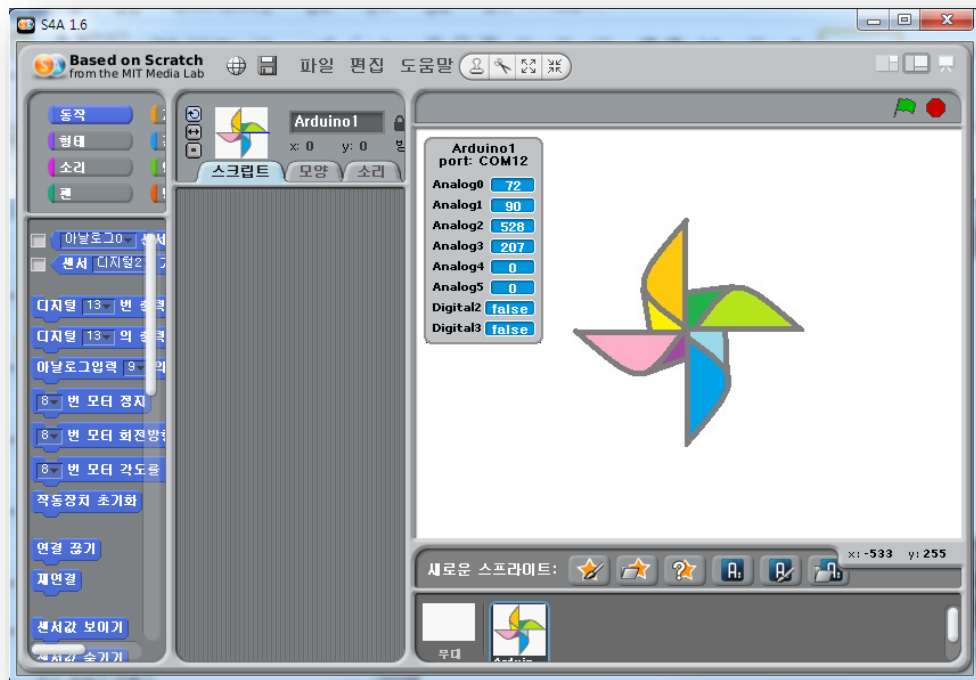
- ① S4A 프로그램을 시작합니다. S4A는 자동으로 아두이노를 찾으려고 할 것입니다. 아직 아두이노가 연결된 상태가 아니므로, 화면에 “보드 검색 중...”이라는 메시지 창이 뜰 것입니다. 지금은 아두이노를 사용하여 코딩을 할 것이 아니므로, 이 메시지는 무시합니다. 무대 화면의 보드 그림을 마우스로 드래그 해주면, 메시지 창이 그림 뒤로 가려지게 됩니다.



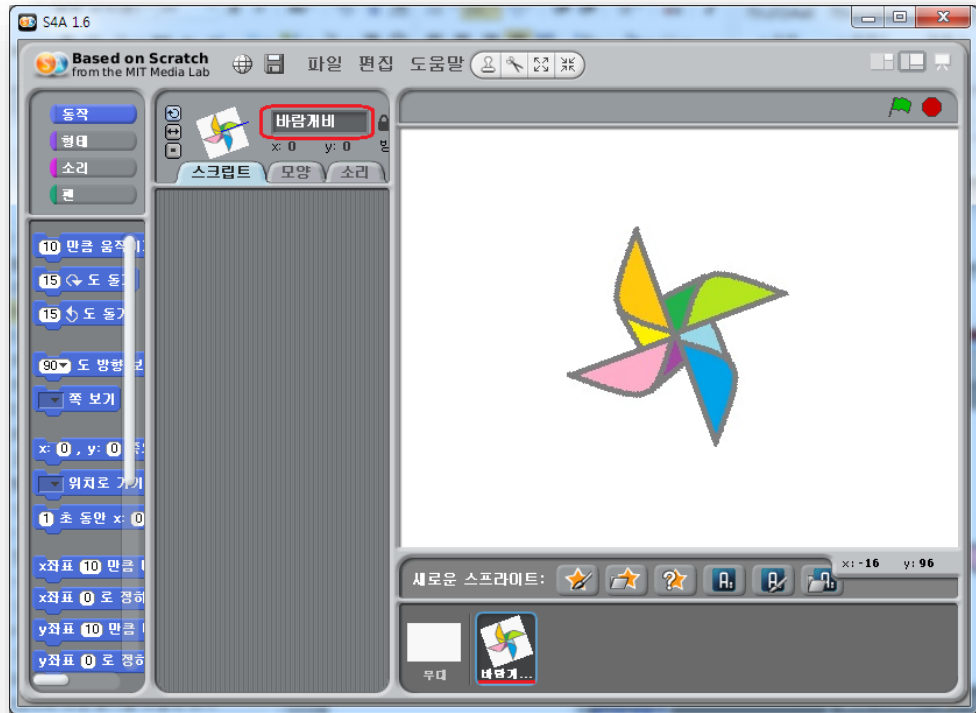
- ② 윈도우 탐색기를 열고, 본서와 함께 제공되는 파일들 중에서 “바람개비.jpg” 파일을 드래그-앤-드롭 방식으로 S4A 프로그램의 무대 화면에 가져다 놓습니다.



- ③ 다음 그림과 같이 무대 화면과 스프라이트 그림이 바람개비 그림으로 바뀌었나요?



- ④ 선택한 스프라이트의 이름을 변경해 보겠습니다. 프로그램의 중앙 상단에 다음 그림과 같이 스프라이트의 이름을 입력할 수 있는 창이 있습니다. 우리가 작업할 스프라이트의 이름을 “바람개비” 이라고 수정해 볼까요? 이름을 수정한 후에 엔터를 눌러 입력을 끝내면, 스프라이트의 이름이 “바람개비”로 변경됩니다.




## 2 단계 : 스크립트 작성하기

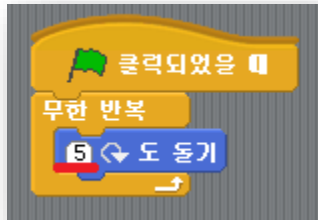
- ① 이제 스크립트를 작성해 보겠습니다. 블록 영역에서 **제어** 버튼을 눌러주세요. 그러면, “제어” 블록들이 프로그램의 좌측 하단 창에 나타납니다. 이 “제어” 블록들 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스크립트로 드래그해서 가져옵니다. 그리고 **클릭되었을 때** 블록 아래에 **무한 반복** 블록을 끼워 넣습니다.





- ② 이번에는, 블록 영역 상단에 있는 **동작** 버튼을 눌러주세요. 그리고, 동작 블록들 중에서 **15도 돌기** 블록을 찾아서 스크립트로 가져와 **무한 반복** 블록 안에 넣어 줍니다.



- ③ 바람개비가 도는 속도를 조절하기 위해  블록에 입력된 값을 5로 수정합니다. 바람개비를 좀 더 빨리 돌리게 하고 싶으면, 이 숫자를 줄여주면 됩니다. 반대로 바람개비를 천천히 돌리고 싶으면 숫자를 늘려주면 되겠지요?



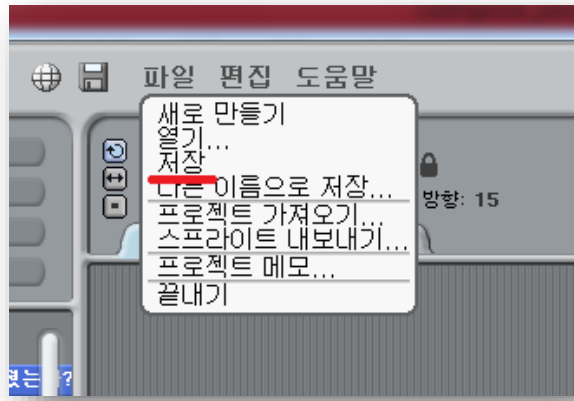
- ④ 이제 코딩이 끝났습니다. 어떠셨나요? 그리 어렵지 않지요? 그럼 잘 만들었는지 동작을 시켜보겠습니다. S4A 프로그램의 우측 상단에 있는  버튼을 클릭해 보세요. 바람개비가 잘 돌아가나요? 끝내기를 하시려면  버튼을 눌러줍니다.



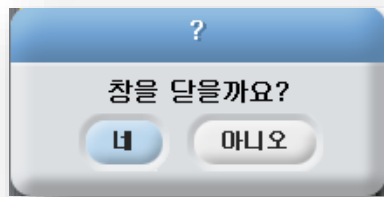
### 3 단계 : 스크립트 저장하기

- ① 작성한 스크립트를 저장하려면, S4A 프로그램의 중앙 상단에 있는 "파일" 메뉴를 눌러주세요. "저장"이나 "다른 이름으로 저장..." 메뉴를 선택하면, 작성된 스크립트를 파일로 저장할 수 있습니다.

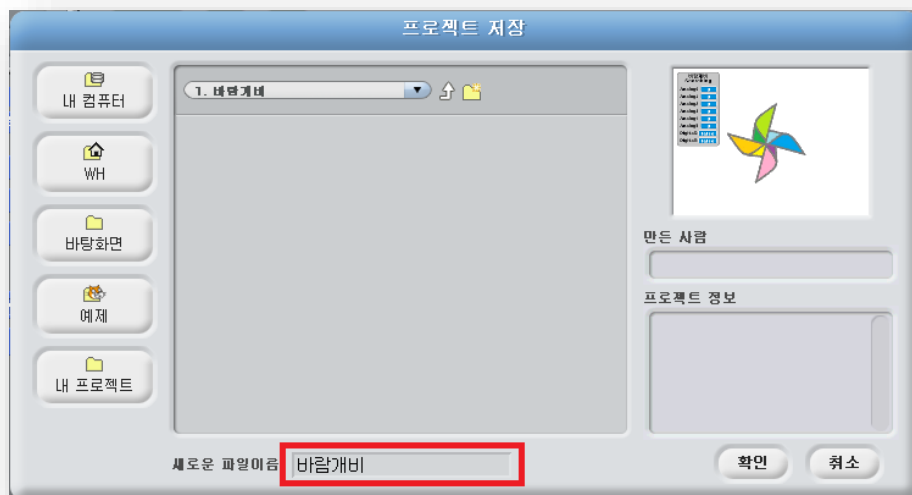




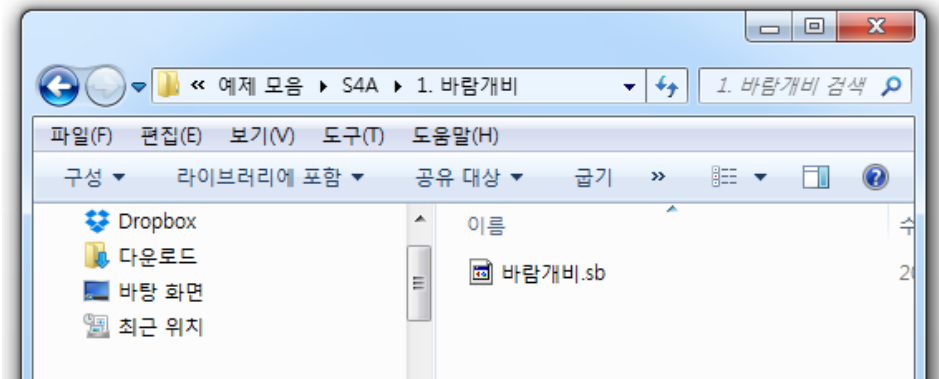
- ② “저장” 또는 “다른 이름으로 저장...” 메뉴를 선택하면 다음과 같은 메시지 창이 나옵니다. “네” 버튼을 눌러주세요.



- ③ 그러면, 다음과 같이 프로젝트를 저장하는 창이 뜹니다. 원하는 폴더를 선택합니다. 그리고, 하단 입력 창에 파일 이름을 적고, “확인” 버튼을 누릅니다.

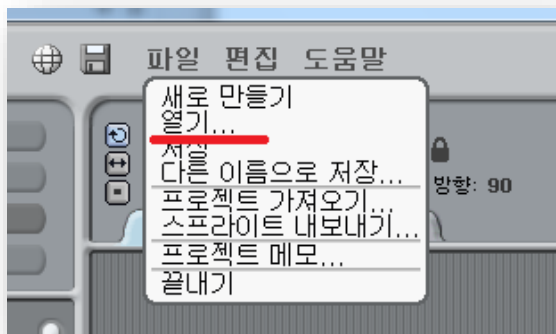


- ④ 다음 그림과 같이 "바람개비.sb"라는 이름의 프로젝트가 저장되었습니다.

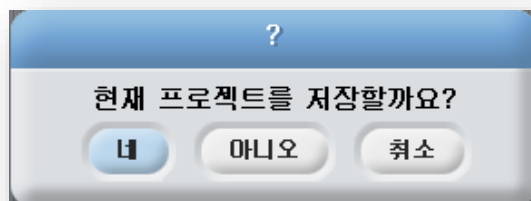


### 3 단계 : 스크립트 불러오기

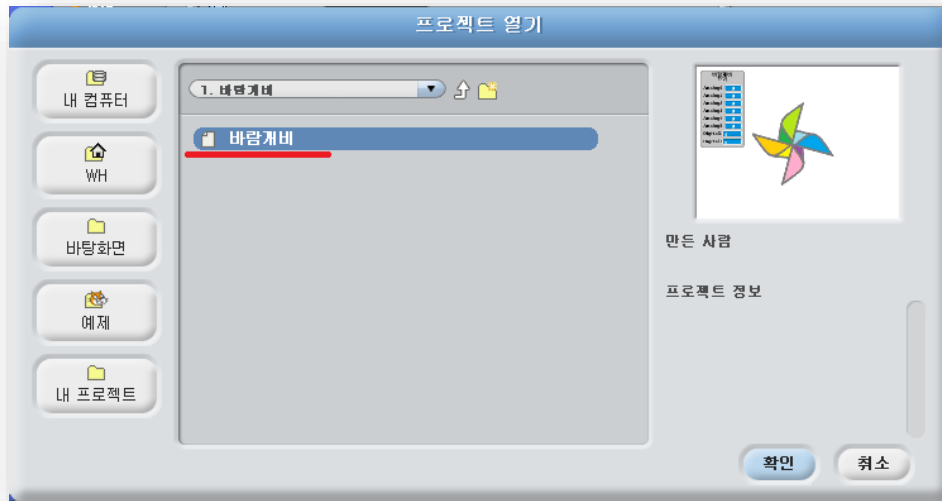
- ① 위와 같이 저장된 프로젝트를 가져오기 할 경우에, "파일" 메뉴에서 "열기..."를 선택합니다.



- ② 그러면, 다음 그림과 같은 메시지 창이 나타납니다. 저장해야 할 작업중인 프로젝트가 있다면, 저장하라는 메시지입니다. "아니오" 버튼을 누릅니다.





- ③ “프로젝트 열기” 창이 뜹니다. “바람개비.sb” 파일이 저장된 위치로 폴더를 이동하면, 다음 그림과 같이 “바람개비” 프로젝트를 선택하실 수 있습니다. “확인” 버튼을 눌러 “바람개비” 프로젝트를 프로그램으로 불러옵니다.




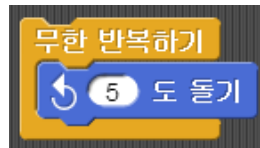
- ④ 프로젝트 열기가 성공적으로 되었다면, 다음 그림과 같이 가져온 프로젝트에 추가 작업을 수행하거나 스크립트를 실행 할 수 있는 상태가 됩니다.



## 설명 더하기

- ✓  블록은 보통 코딩의 시작 부분에 쓰입니다. 이것은 초록색 깃발을 클릭했을 때 코드를 시작하라는 뜻입니다. 그래서 여러분이 코딩을 다 하고 코드를 실행시킬 때  을 클릭합니다.

- ✓  는 블록이름 그대로 감싸고 있는 코드를 무한 반복하는 것입니다. 여기서는 다음과 같이 5도 돌리기를 무한 반복하는 것입니다.



- ✓ 이 다음과 같이 무한 반복하기를 빼고 코드를 실행해 보십시오. 그러면 바람개비는 5도만 돌고 멈출 것입니다.



- ✓ 스프라이트에서 쓰던 블록들을 지우는 방법으로는 블록들에서 마우스 오른쪽 버튼을 누르면 "삭제" 메뉴가 있습니다. 이 메뉴를 선택하면 블록이 지워집니다. 또는 이 블록들을 끌어다가 왼쪽에 있는 블록 모음들에 놓으면 없어집니다.

## 4. 코딩키트! 코딩 놀이 친구



어때? 블록 코딩을 한번 해보니까 쉽고 재미있지?



네~ 정말 쉽고 재미있어요! 빨리 다른 것도 만들어 보고 싶어요!



허허! 녀석도 참! 그럼, 지금 먼저 무엇을 만들고 싶니?



음.. 게임도 만들어 보고 싶고, 로봇도 움직이게 하고 싶어요. 그리고 우리 집 문도 자동 문으로 만들고...



허허허! 우리 곰곰이는 하고 싶은 것이 참 많구나! 아빠가 여러 가지 장치들을 다양하게 다루며 공부해 볼 수 있도록 코딩키트를 준비했는데, 한번 볼래?

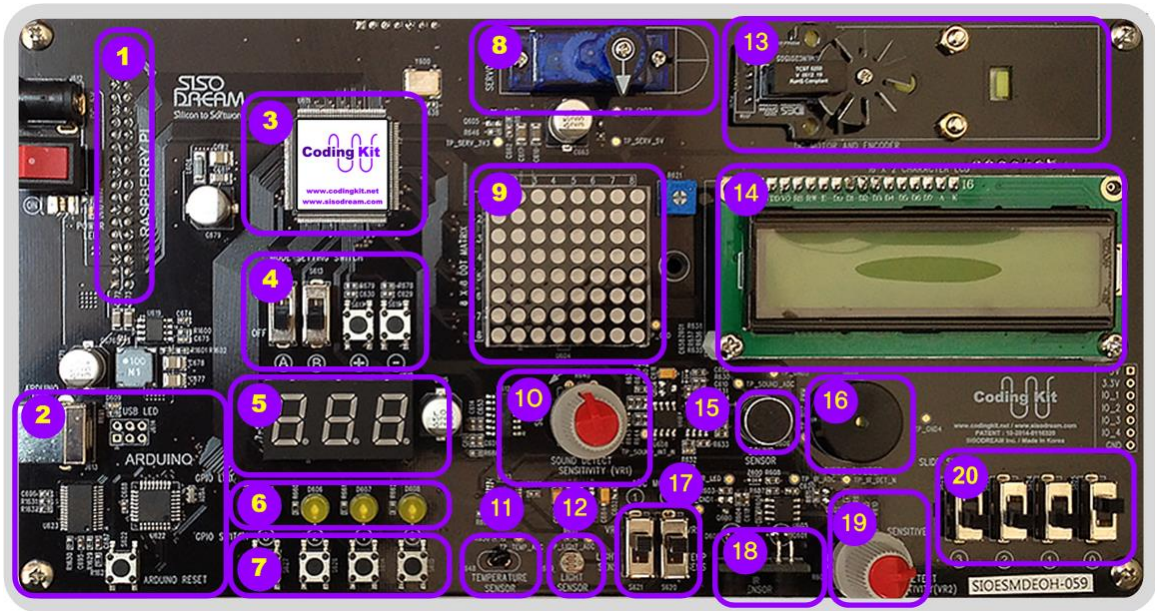
곰곰이는 아빠가 선물 해 주신 코딩키트 안에 작은 컴퓨터가 있다는 이야기를 듣고 깜짝 놀랐습니다. 그리고 작은 컴퓨터와 다양한 장치들을 손쉽게 동작 시켜 볼 수 있다고 하니 빨리 코딩을 하고 싶은 생각이 들었습니다.

## 4-1. 코딩킷 살펴보기

코딩킷에는 로봇 등을 만들 때 사용되어 질 수 있는 많은 전자 부품들이 장착되어 있어요. 그리고 이 부품들을 동작시킬 수 있는 소형 컴퓨터도 있고요. 이 컴퓨터는 세상에 나와 있는 많은 컴퓨터들 중 한 종류인데, 코딩킷에 장착된 컴퓨터는 아두이노라고 합니다. 여러분이 코딩을 하시면 이 아두이노에서 그 코드를 해석해서 여러 부품들을 동작시킵니다.

장착되어 있는 부품들로는 무언가를 보여줄 수 있는 LED, LCD, 도트매트릭스 등이 있습니다. 아두이노로 신호를 전달해 줄 수 있는 버튼, 스위치 등도 있습니다. 온도, 밝기 등을 체크할 수 있는 센서도 있습니다. 바퀴나 로봇 팔 등을 움직일 수 있는 DC 모터, 서보 모터 등도 있습니다. 앞으로 이런 부품들을 하나 하나 움직여 볼 것입니다. 여러분이 작성한 코드로 이 부품들이 움직이는 것을 보면서 매우 재미있게 공부하며 코딩의 기초를 탄탄히 쌓을 수 있을 것입니다.

다음은 코딩킷 보드에 대한 자세한 내용입니다.



1. 라즈베리 파이 커넥터 (Raspberry Pi Connector)
2. 아두이노 (Arduino)
3. 스위칭 시스템 스위칭 칩 (Switching System, Switching Chip)
4. 스위칭 시스템 스위치와 버튼 (Switching System, Switch & Button)
5. 스위칭 시스템 세븐세그먼트 (Switching System, 7-Segment)
6. LED 4 개



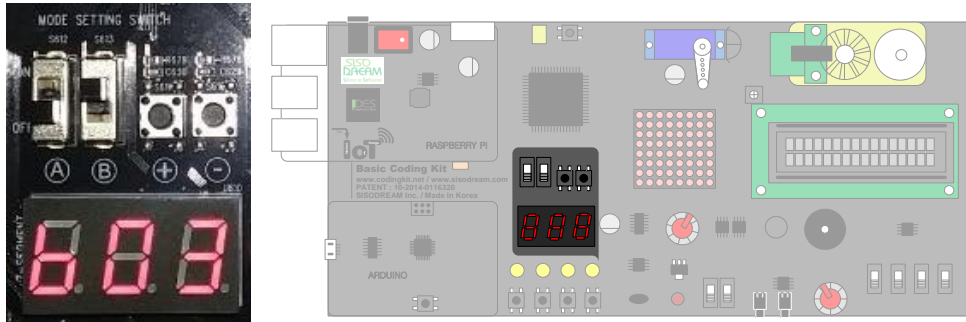
7. 버튼 4 개 (Button)
8. 서보 모터 (Servo Motor)
9. 도트매트릭스 (Dotmatrix)
10. 가변 저항 1 (Variable Resistor 1)
11. 온도 센서 (Temperature Sensor)
12. 밝기 센서 (Light Sensor)
13. DC 모터 (DC Motor)와 인코더(Encoder)
14. 캐릭터 LCD (Character LCD)
15. 소리 센서 (Sound Sensor)
16. 부저 (Buzzer)
17. ADC 모드 스위치
18. 적외선 센서 (IR Sensor)
19. 가변 저항 2 (Variable Resistor 2)
20. 스위치 4 개

## 4-2. 코딩킷 동작시켜보기

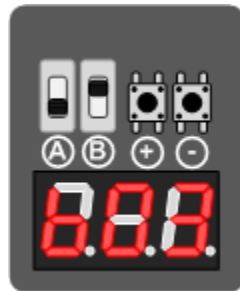
코딩킷에는 USB 케이블하고 아답터가 있습니다. USB 케이블은 PC 에, 아답터는 전원 콘센트와 연결해 주세요. 아답터의 콘센트 부분이 아닌 전원 케이블은 코딩킷에 연결해 주세요. 그리고 코딩킷의 전원 스위치를 켜주세요.



전원이 연결되면, 코딩킷의 7-SEGMENT 가 깜박일 것입니다. 이때 ⊕ 또는 ⊖ 스위치 버튼을 2~3 초간 눌러주시면, 7-SEGMENT 의 깜박임이 멈춥니다.



그 옆으로 ㉠, ㉡ 스위치가 보이시죠? ㉠ 스위치는 아래로 내리고, ㉡ 스위치는 위로 올려주세요.



⊕ 버튼을 누르면, b00, b01, b02, ... 으로 숫자가 커지고, ⊖ 버튼을 누르면 다시 숫자가 줄어듭니다. b03 에 맞추고 7-SEGMENT 가 깜박이면 다시 ⊕ 또는 ⊖ 스위치 버튼을 2~3 초간 눌러 주세요.

코딩킷이 잘 동작하나요? 코딩 공부를 시작하기 위해 몇 가지 할 일이 조금 더 남아 있습니다. 선생님이나 부모님들께서 조금 도와주셔야 할 것 같습니다. “부록 C. 프로그램 설치 및 사용 가이드”에 코딩 시작 전에 해야 할 작업들을 설명해 두었습니다.

## 5. 깜박이는 LED 등 만들기



어? 코딩 키트에 있는 버튼을 눌러도 LED가 켜지지 않아요. 고장이 났나 봐요. 어찌죠?



허허허! 아직 우리가 코딩을 하지 않아서 아무런 동작을 하지 않는 거란다. LED가 켜지도록 코딩을 해볼까?



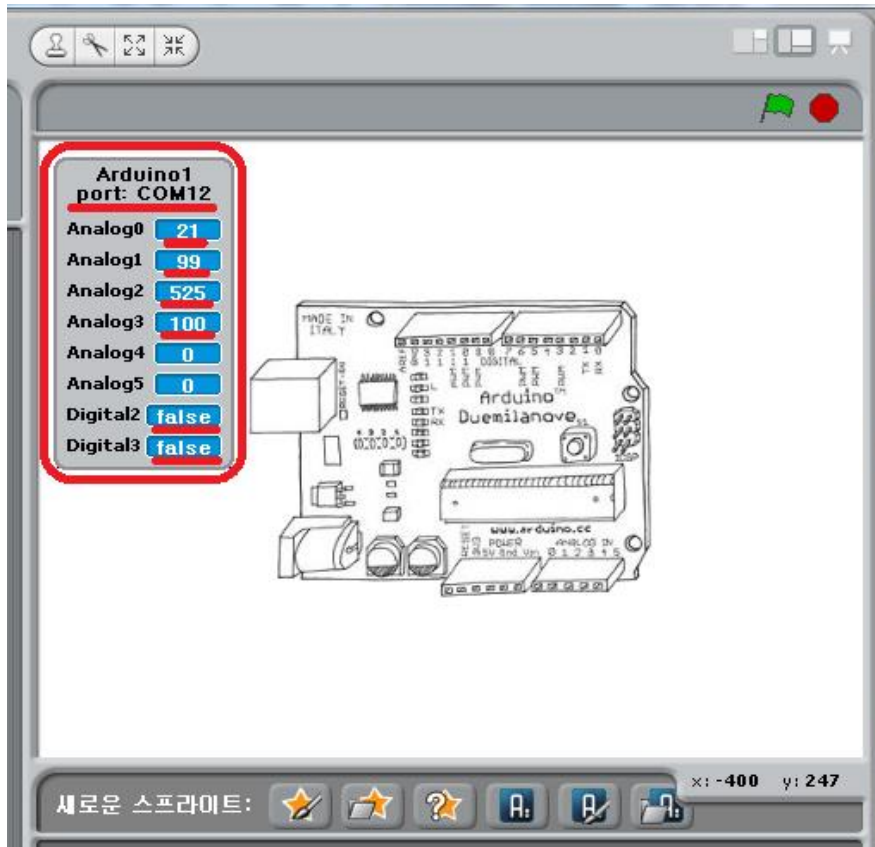
네~ 어떻게 LED가 켜지도록 코딩을 하는지 궁금해요.



LED를 깜빡이게 하는 동작의 원리는 간단하단다. 집에 있는 전원 스위치를 올렸다 내렸다 하면 어떻게 되지? 전원 스위치를 올리면 전등이 켜지고, 전원 스위치를 내리면 전등이 꺼지지? 이처럼 코딩키트의 LED로 전원을 공급하면 LED가 켜지고, 전원을 중단하면 LED가 꺼진단다. 그리고 이 동작을 반복하면 LED가 깜빡이게 되는 거지.

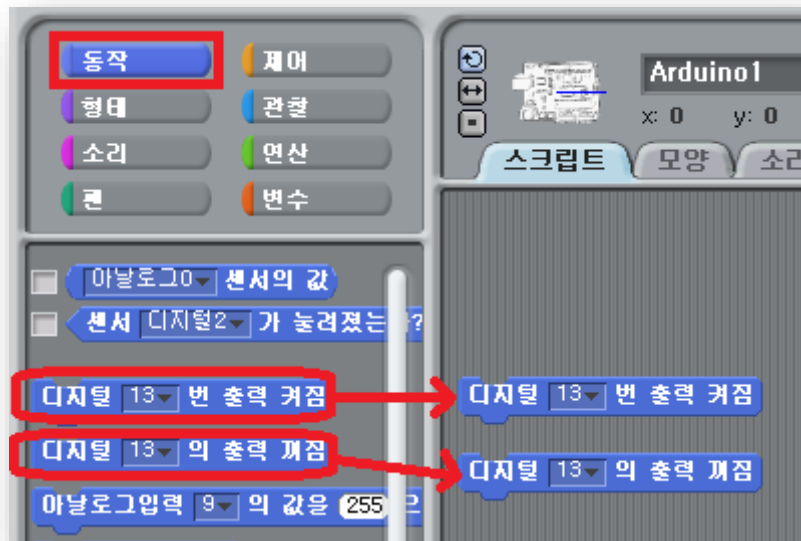
## 5-1. 코딩으로 LED를 켜보자

S4A 프로그램을 실행시킵니다. 코딩킷과 컴퓨터가 잘 연결이 되었다면 “보드 검색 중...” 이라는 메시지 창은 사라지고 다음 그림과 같이 무대 화면에 코딩킷과 연결된 포트정보와 각 핀들의 상태 값들이 나타납니다.

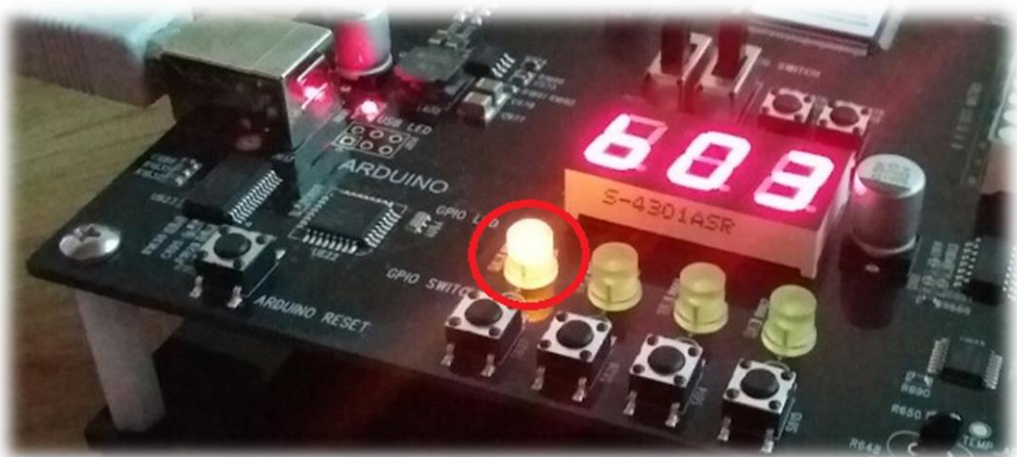


코딩킷과 S4A 프로그램의 통신이 잘 되고 있는 것을 확인하였다면, 코딩으로 LED가 켜지는 동작을 만들어 보겠습니다.

- ① 화면 좌측 상단에 있는 블록 모음들에서 **동작** 버튼을 눌러주세요. 그러면, 하단에 “동작” 블록들이 나타납니다.
- ② **디지털 13번 출력 켜짐** 블록과 **디지털 13번의 출력 꺼짐** 블록을 “스크립트”로 가져옵니다. 마우스를 사용하여 드래그-앤-드롭 방식으로 가져올 수 있습니다.



- ③ 스크립트로 가져온 **디지털 13번 출력 켜짐** 블록을 마우스로 클릭해 보세요. 그러면, 다음 사진과 같이 코딩킷 LED에 불이 켜지는 것을 확인 할 수 있습니다. 13번 핀이 LED ③과 연결되어 있어서, 13번에 출력 켜짐이 전달되면 3번 LED가 켜지게 됩니다.



- ④ LED 등이 계속 켜져 있는 상태로 있습니다. 이번에는 LED 등을 끄는 동작을 해보겠습니다. 스크립트에 있는 **디지털 13번 출력 꺼짐** 블록을 마우스로 클릭합니다. 그러면, LED 등이 꺼지는 것을 확인 하실 수 있습니다. 13 번에 출력 꺼짐이 전달되어 3 번 LED가 꺼지게 되는 것입니다.

## 5-2. LED가 깜박이는 동작을 만들어 보자.

이번에는 LED가 깜박이는 동작을 반복 하도록 만들어 볼까요? 위 5-1절에서 스크립트에 있는 **디지털 13번 출력 꺼짐** 블록과 **디지털 13번의 출력 켜짐** 블록을 번갈아 가며 마우스로 클릭하게 되면 LED가 깜박이는 것처럼 보일 것입니다. 이 동작을 코딩으로 만들어 보겠습니다.

- ① 화면 좌측 상단에 있는 블록 모음들에서 **제어** 버튼을 눌러 주세요.

- ② **제어** 창에서 **클릭되었을 때** 블록을 스크립트로 가져옵니다.

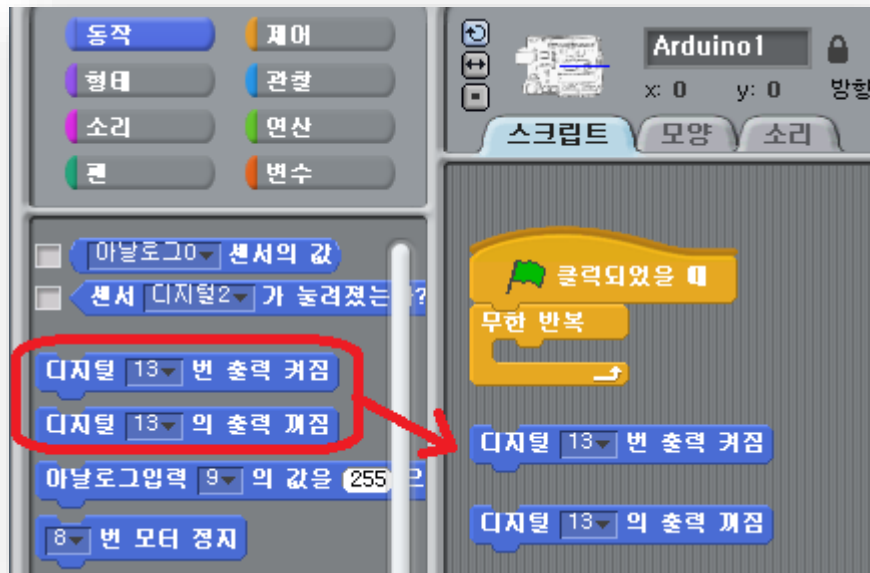


- ③ 그리고, LED 등이 꺼졌다 켜졌다 하는 동작을 반복하도록 하기 위해 **무한 반복** 블록도 스크립트로 가져옵니다.





- ④ 화면 좌측 상단에 있는 블록 모음들에서 **제어** 버튼을 눌러주세요.
- ⑤ **디지털 13**의 출력 꺼짐 블록과 **디지털 13**의 출력 켜짐 블록을 스크립트로 가져옵니다.

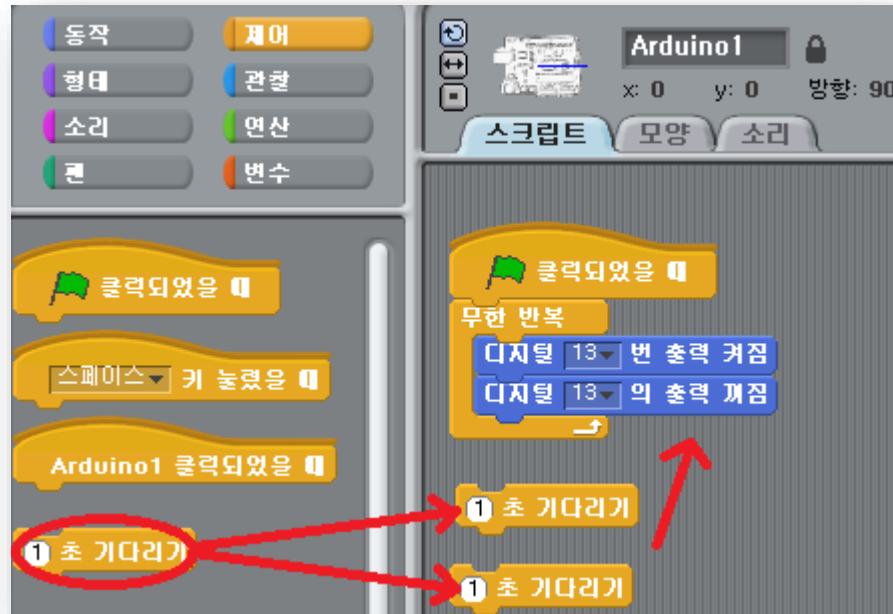



- ⑥ 반복 동작을 할 **디지털 13**의 출력 켜짐 블록과 **디지털 13**의 출력 꺼짐 블록을 **무한 반복** 블록 안으로 넣어 줍니다.

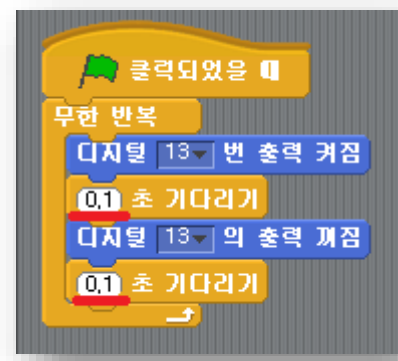
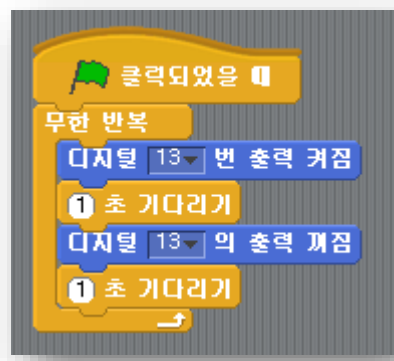


- ⑦ 그런데, LED에 불을 켜자마자 끄도록 하면, 너무 빨라서 우리 눈으로 확인할 수가 없습니다. 그래서 LED 불이 켜지고 꺼지는 시간 간격을 1초씩 주도록 하겠습니다. 좌측 상단의 블록 모음들에서 **제어** 블록 모음을 선택하세요.

- ⑧ **1 초 기다리기** 블록을 스크립트로 두 개 가져와, **디지털 13번 출력 꺼짐** 블록과 **디지털 13번의 출력 꺼짐** 블록 다음에 각각 넣어 줍니다.



- ⑨ 다음 그림과 같이 스크립트가 완성이 되었나요? 프로그램의 우측 상단에 있는  버튼을 눌러서 실행시켜 보세요. LED 가 깜박이는 동작을 반복합니다. 만약 좀 더 빨리 깜박이게 하고 싶으시다면, **1 초 기다리기** 블록의 초 값을 0.5초, 0.1초 등으로 수정 입력하여 주시면 됩니다.



## 🔗 생각하기 1

지금까지 LED 한 개를 깜박이도록 해 보았습니다. LED 두 개를 깜박이게 하고 싶습니다. 어떻게 하면 좋을까요? **디지털 13번 출력 커점** 블록과 **디지털 13번의 출력 꺼짐** 블록에서 핀 번호 옆에 있는 아래 화살표를 누르면 핀 번호를 바꿀 수 있습니다. (핀 번호 13, 12, 11, 10은 LED ③, ②, ①, ④과 각각 연결되어 있습니다.)



## 6. 초인종을 만들기

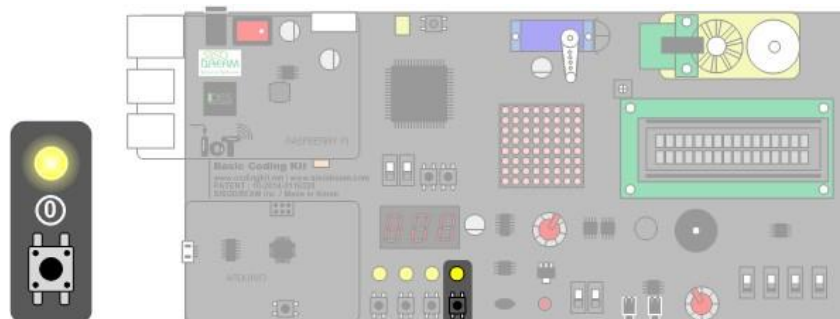
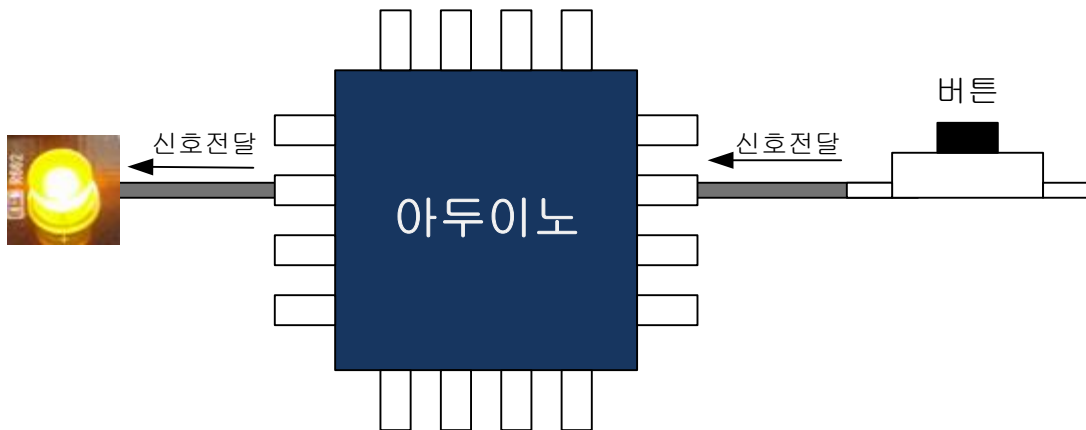


아빠! 코딩으로 LED를 깜박이게 해보니 정말 신기해요!



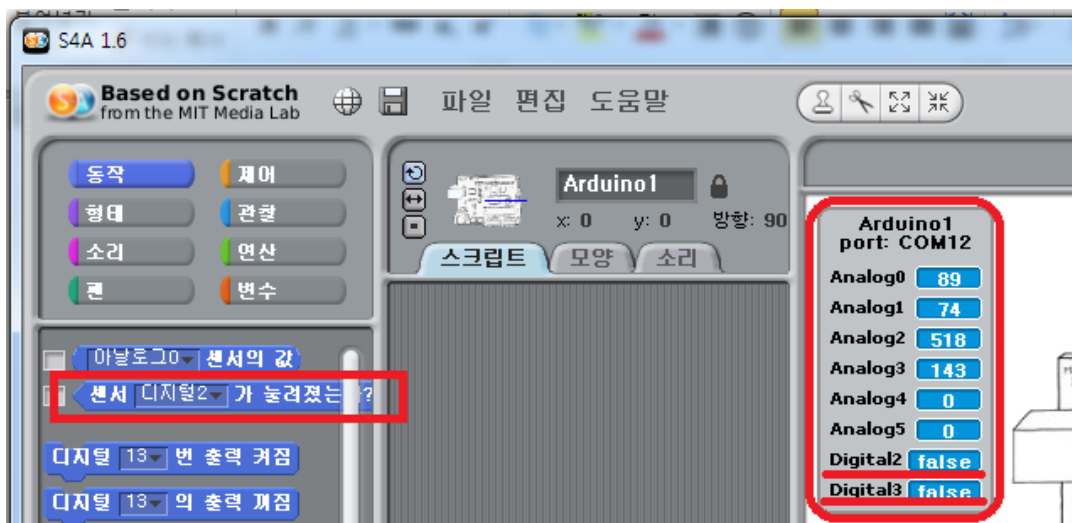
어때? 신기하고 재미있지?

그러면, 이번엔 버튼을 눌러서 LED가 켜지는 코딩을 배워볼까? 이 그림을 한번 보렴. 버튼을 누르면 그 신호가 아두이노에 전달되고, 아두이노에서는 곰곰이가 코딩한 코드를 실행시키지. 그 코드에서 버튼이 눌렀다는 신호가 들어오면 LED를 켜도록 만드는 거야.



## 6-1. 버튼이 눌리면 LED 가 켜지도록 해보자.

이번 장에서는 버튼을 다루어 보겠습니다. 프로그램의 우측 상단에 위치한 무대를 보시면, 장치들의 상태 정보를 보여주는 상태 창이 하나 있습니다. 보여지는 상태 정보들 중에 Digital2와 Digital3가 코딩키트의 버튼과 연결되어 있습니다. 코딩키트의 ⑩번 버튼과 ①번 버튼을 눌러보세요. 상태 창의 "Digital2"와 "Digital3"의 값이 "false"에서 "true"로 바뀝니다. 다시 버튼에서 손을 떼면, 원래대로 "false" 값이 됩니다.



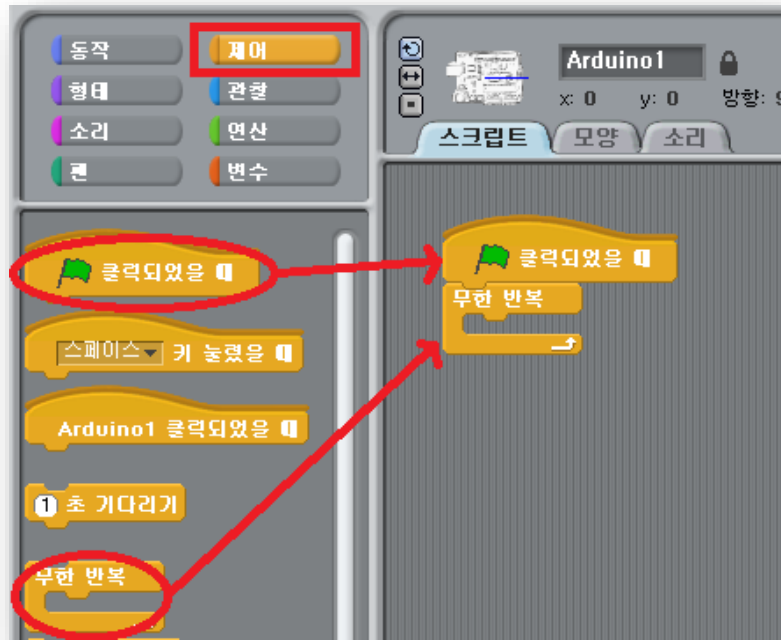
버튼의 동작에 따라 Digital2와 Digital3의 값이 어떻게 변하는지 알았습니다. 그런데, 코딩을 하려면 이 상태 값들을 가져와 쓸 수 있어야 합니다. 프로그램의 좌측에 **동작** 블록 모음을 클릭해 보세요. **센서 디지털2가 눌려졌는가?** 블록이 있습니다. 이 블록을 사용하면 Digital2와 Digital3의 상태 정보 값 즉, 버튼이 눌렸는지 아닌지를 값으로 얻어 올 수 있습니다.

그러면, 버튼이 눌렸을 경우에 LED가 켜지는 동작을 만들 수 있겠지요? 한번 만들어 보겠습니다.

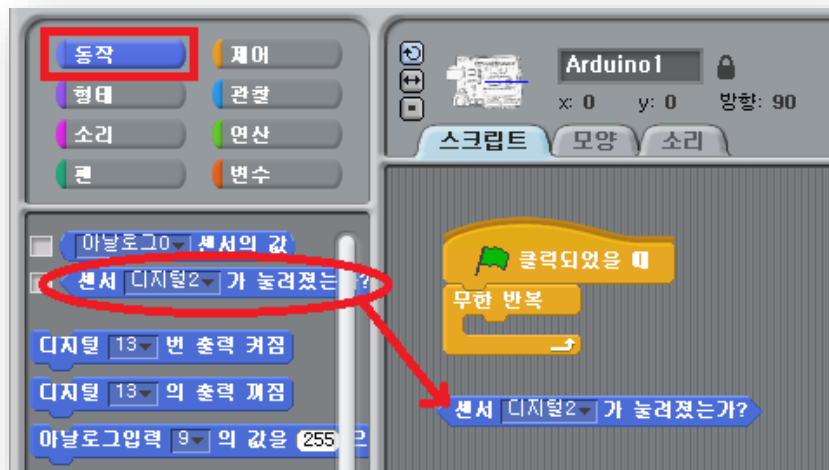
- ① 화면 좌측 상단에 있는 블록 모음들에서 **제어** 버튼을 눌러 주세요.

- ② **제어** 블록 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스

크립트로 가져옵니다.



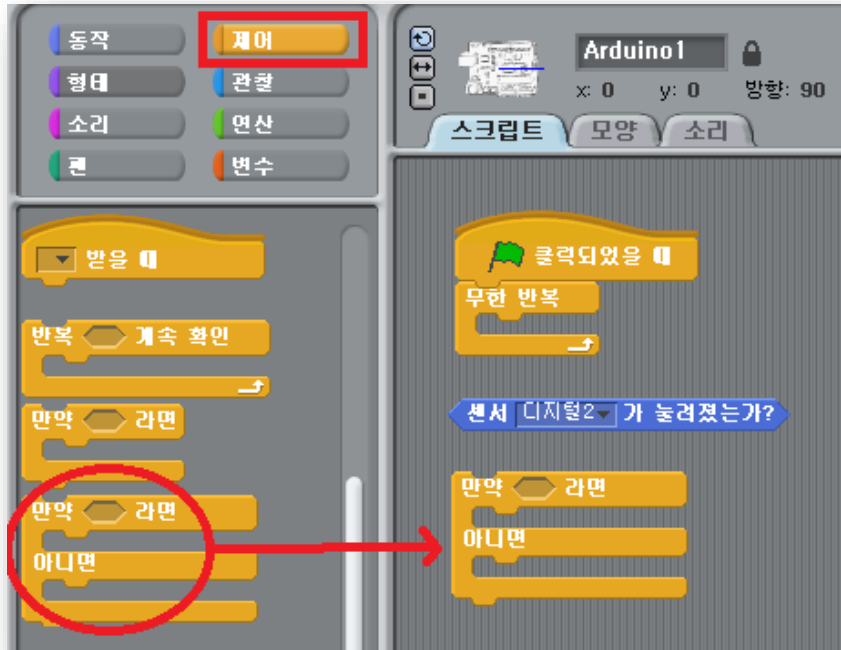
- ③ **동작** 블록 모음을 클릭하세요. **센서 디지털2가 눌러졌는가?** 블록을 스크립트로 가져옵니다.




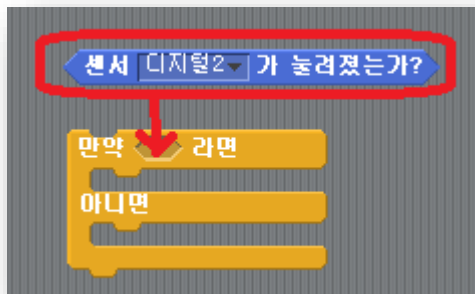
- ④ 다시 **제어** 블록 모음 버튼을 눌러 주세요. **제어** 블록 모음에서



블록을 스크립트로 가져옵니다.

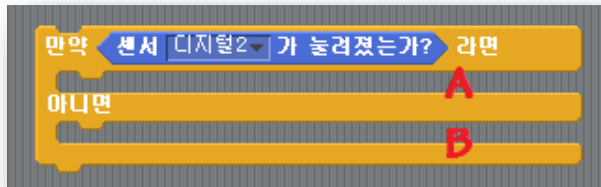


- ⑤ 그리고, **센서 디지털2가 눌러졌는가?** 블록을  의 첫 번째 입력 칸에 넣습니다.

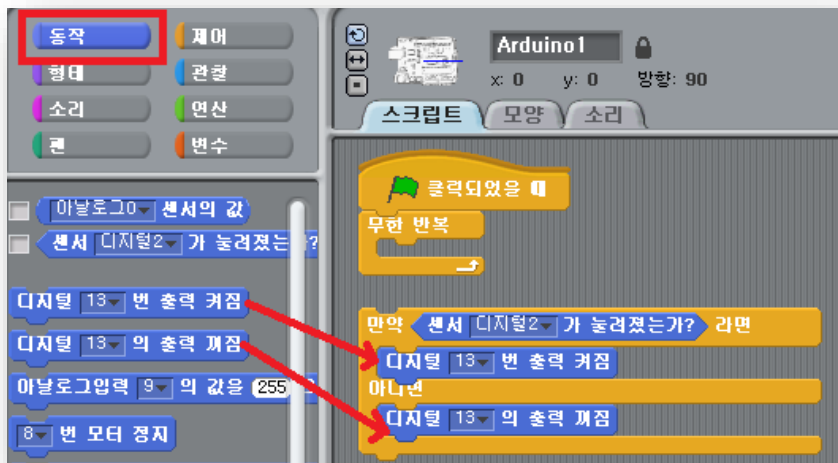


- ⑥ 다음 그림과 같이 블록이 만들어 졌나요? 디지털2 즉, 코딩킷의 버튼 ⑩이 눌렸으

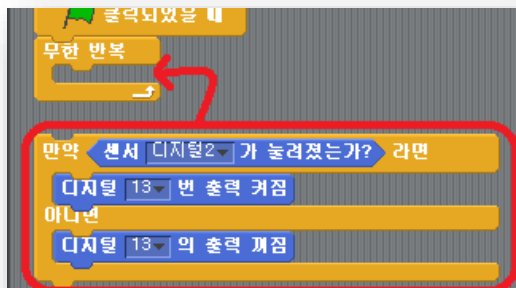
면 A 부분, 버튼 ⑩이 눌리지 않았으면 B 부분의 블록들이 실행되는 블록이 만들어졌습니다.




- ⑦ **동작** 블록 모음에서 **디지털 13번 출력 꺼짐** 과 **디지털 13번의 출력 꺼짐** 블록을 스크립트로 가져와 A와 B 부분에 다음 그림과 같이 넣어 줍니다.

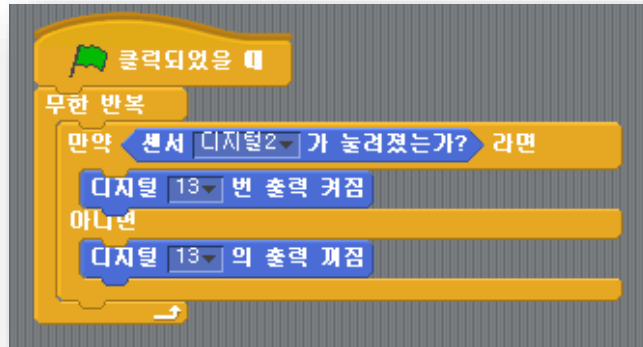


- ⑧ 위 ⑦에서 만들어진 블록을 **무한 반복** 블록 안으로 넣어 줍니다.





- ⑨ 다음 그림과 같이 블록이 완성되었나요? 그러면,  버튼을 눌러 실행시켜 보세요. 제일 오른쪽의 0번 버튼이 눌리면 LED가 켜지고 눌리지 않으면 LED가 꺼지는 결과를 확인할 수 있습니다.



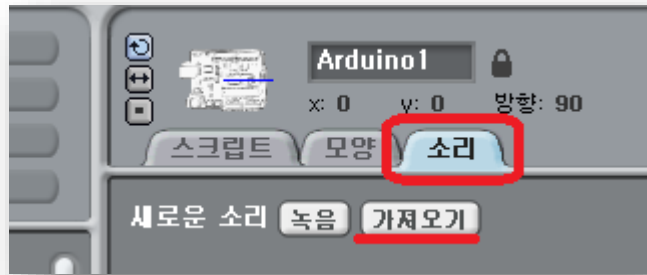
## 🚧 생각하기 2

⑩번 버튼이 눌리면 ⑩번 LED(디지털 10)가 켜지고 ⑪번 버튼이 눌리면 ⑪번 LED(디지털 11)가 켜지는 동작을 만들어 보세요.

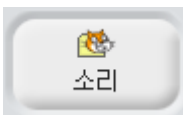
## 6-2. 버튼을 누르면 “딩동” 소리가 나게 해보자.

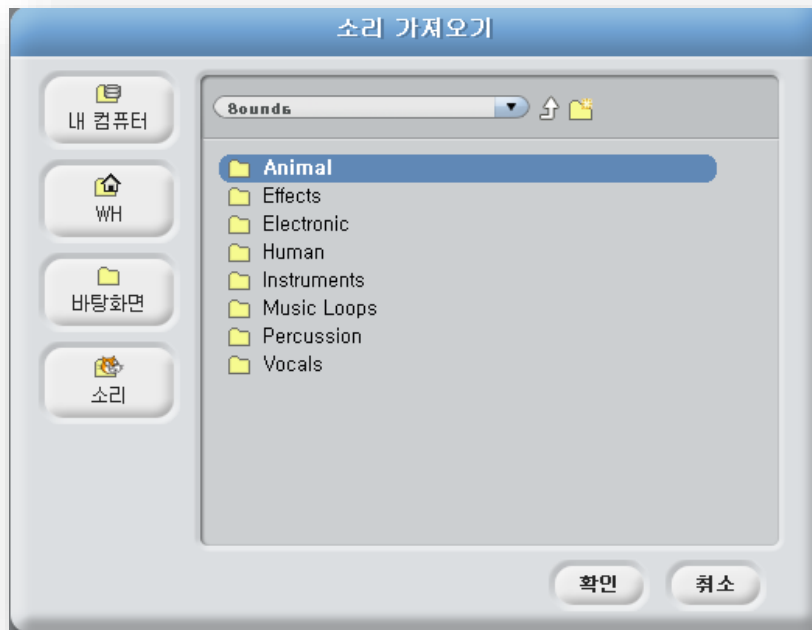
### 🎯 1 단계 : 소리 삽입하기

- ① 스크립트 화면의 상단을 보면, “스크립트”, “모양”, “소리” 탭이 있습니다. 여기에서 “소리” 탭을 누릅니다.

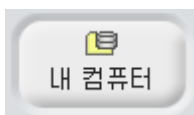


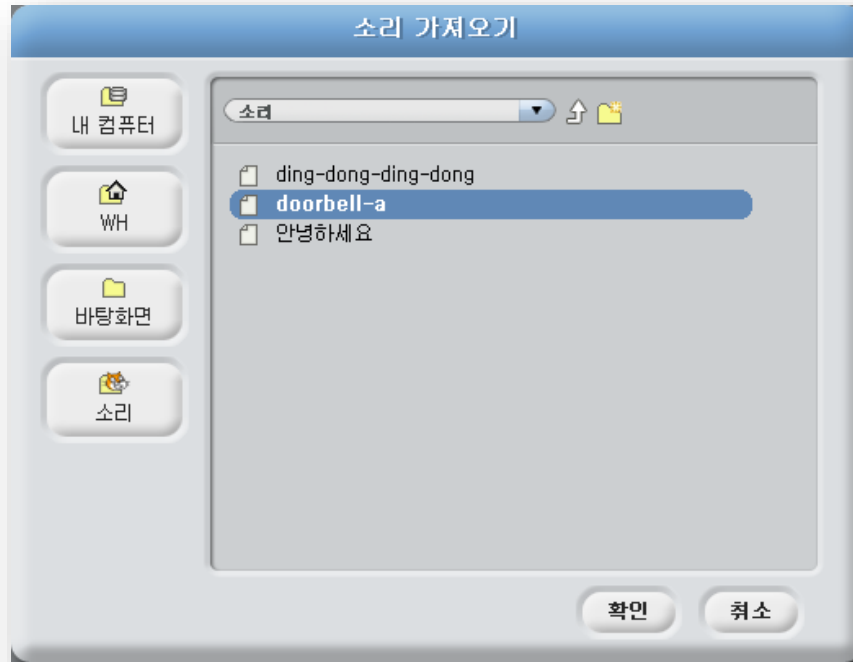
- ② “소리” 탭에서 “가져오기” 버튼을 누릅니다. 그러면, 다음 그림과 같이 “소리 가져오

기” 창이 나타납니다.  버튼을 누르면, S4A에서 제공되는 소리 파일 목록이 나타납니다.



- ③ S4A에서 제공되는 소리 파일 중에는 초인종 소리가 없으므로, 본 서와 함께 제공되

는 “doorbell-a.wav” 파일을 가져오겠습니다.  버튼을 눌러서, “doorbell-a.wav” 소리 파일이 있는 폴더로 갑니다. 그리고, “doorbell-a” 을 선택하세요. 그러면 “딩동~” 소리가 날 것입니다. “확인” 버튼을 누릅니다.

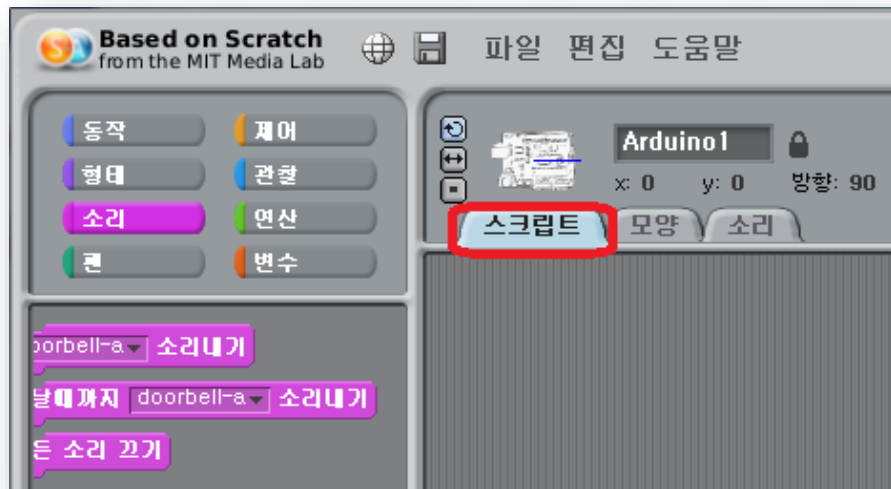


- ④ “소리” 탭에 “doorbell-a” 소리가 만들어졌습니다. **소리** 블록 모음 버튼을 누르면, **doorbell-a** 소리내기 블록과 **끝날 때까지 doorbell-a** 소리내기 블록이 나옵니다. 이 두 블록을 사용하여 “doorbell-a” 소리를 낼 수 있습니다.

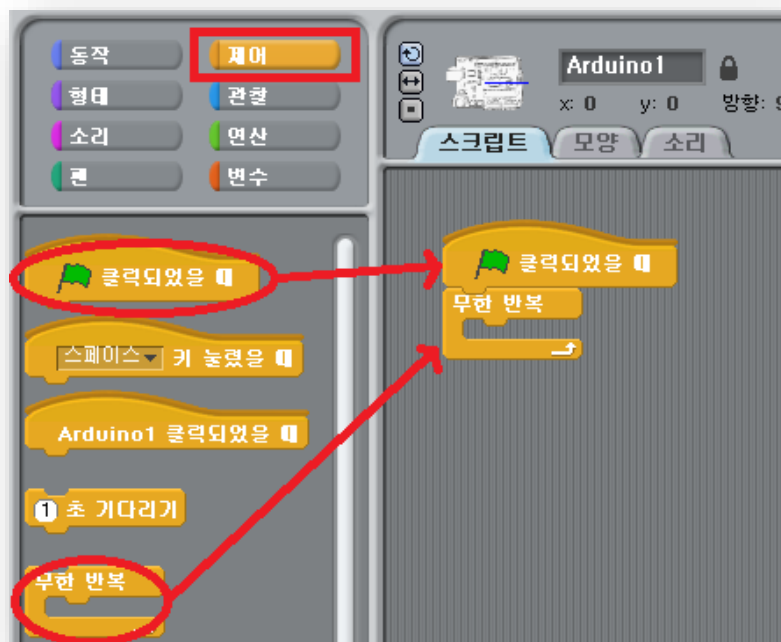


## 🌀 2 단계 : 초인종 만들기

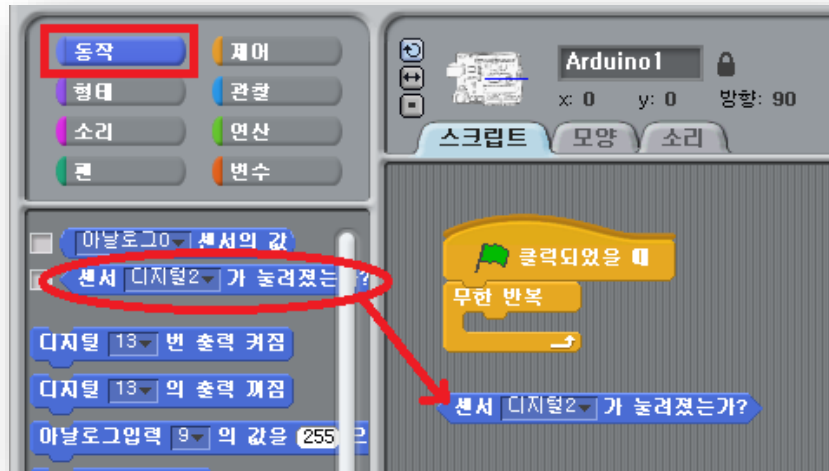
- ① 스크립트 탭으로 이동합니다.



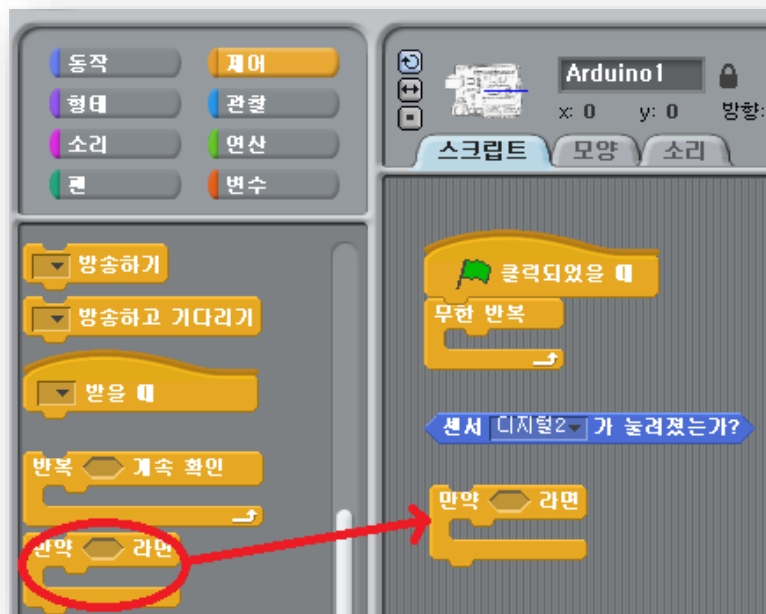
- ② **제어** 블록 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.




- ③ **동작** 블록 모음을 클릭하세요. **센서 디지털2가 눌러졌는가?** 블록을 스크립트로 가져옵니다.

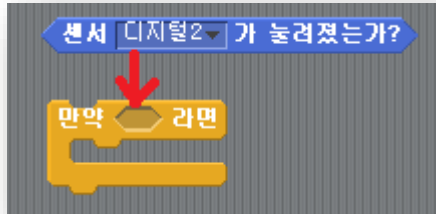





- ④ 다시 **제어** 블록 모음으로 가서 **만약 그러면** 블록을 스크립트로 가져옵니다.



- ⑤ 그리고, **센서 디지털2가 눌려졌는가?** 블록을 **만약 그러면** 블록의 첫 번째 입력 칸에

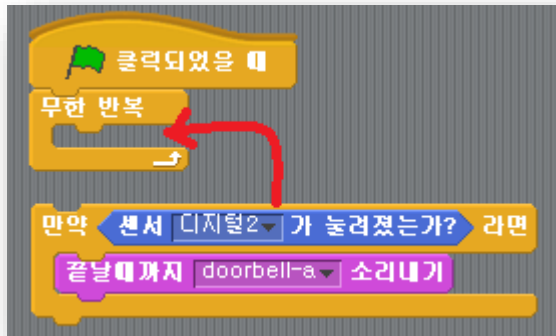
넣습니다. 버튼 ①이 눌리면  블록의 안쪽에 삽입되는 블록들이 동작을 하게 됩니다.




⑥  블록 모음에서  블록을 가져와서  블록 안에 넣습니다. 버튼 ①이 눌리면 "doorbell-a" 소리가 나도록 블록을 만들었습니다.



⑦ 위 ⑥에서 만들어진 블록을  블록 안으로 넣어 줍니다.



- ⑧ 다음 그림과 같이 블록이 완성되었나요? 그러면,  버튼을 눌러 실행시켜 보세요. 버튼이 눌리면, "딩동~" 소리가 날 것입니다.



### 🔗 생각하기 3

초인종을 누르면 벨소리와 함께 LED도 깜박이도록 만들어 보세요.

## 7. 크리스마스 전등 만들기



제가 코딩으로 초인종을 만들다니! 엄마가 아시면 깜짝 놀라시겠어요! ^^



코딩이 생각보다는 쉽고 재미있지?

벌써 LED도 깜박이게 해보고, 버튼도 다뤄보고, 소리 파일을 삽입하여 소리 내는 방법까지 배웠구나. 지금까지 배운걸 가지고 근사한 것을 만들어 볼 수 있을 것 같다. 예를 들면 캐롤이 흘러나오는 크리스마스 전등 같은 거 말이다.



생각만 해도 멋진데요! 한번 만들어 볼래요!



그럼 아빠랑 같이 만들어 볼까? 빠른 캐롤이 나오면 LED가 빠르게 깜박이고 느리게 나오면 느리게 깜박이게 하면 더 근사하겠구나.



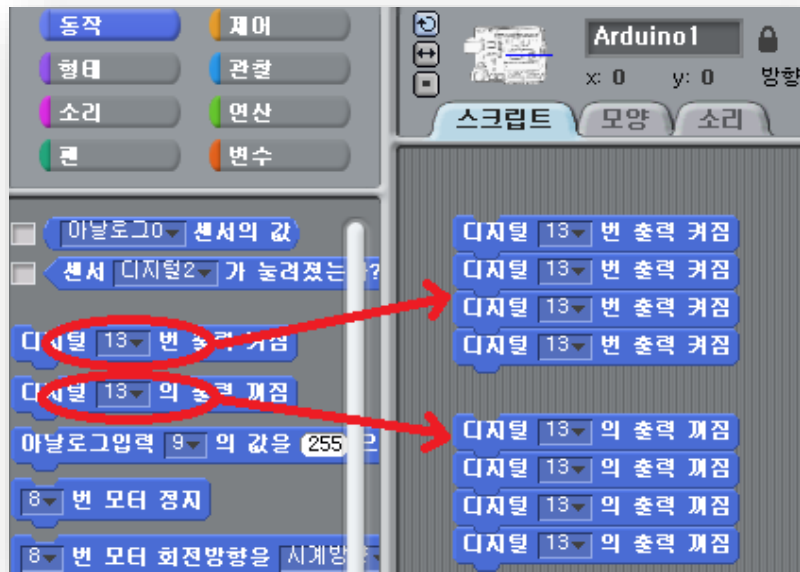


## 7-1. 캐롤이 나오며 깜박이는 크리스마스 전등을 만들어 보자.

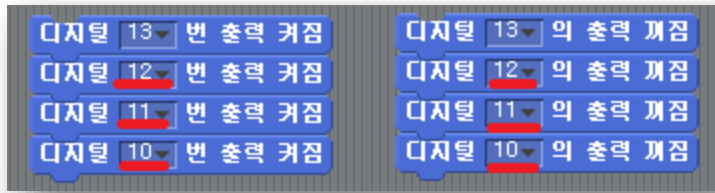
지금까지 배운 내용을 가지고 간단하게 크리스마스 전등을 만들어 볼 수 있습니다. LED를 서로 번갈아 깜박이게 하고, 크리스마스 캐롤이 나오는 소리 파일을 가져와 동시에 출력되도록 만들어 주면 됩니다.

### 1 단계 : 깜박이는 LED 등 만들기

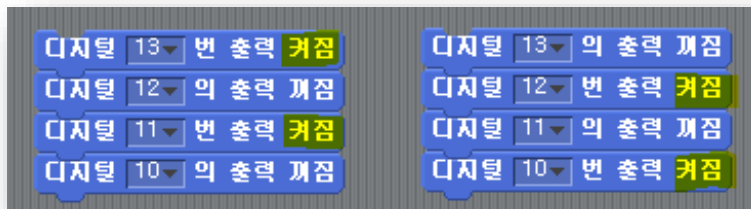
- ① 동작 블록 모음에서 **디지털 13번 출력 켜짐** 과 **디지털 13번의 출력 꺼짐** 블록을 스크립트로 각각 4개씩 가져옵니다.



- ② 숫자 옆의 아래 화살표를 누르면, 10~13번까지 번호를 선택할 수 있습니다. LED를 번갈아 가며 켜다 꺼다 하려고 합니다. 10~13번까지의 번호를 차례로 선택해 줍니다.



- ③ LED 4개가 교차하여 켜지고 꺼질 수 있도록 디지털 12번과 10번 출력 켜짐과 꺼짐을 서로 바꾸어 줍니다.

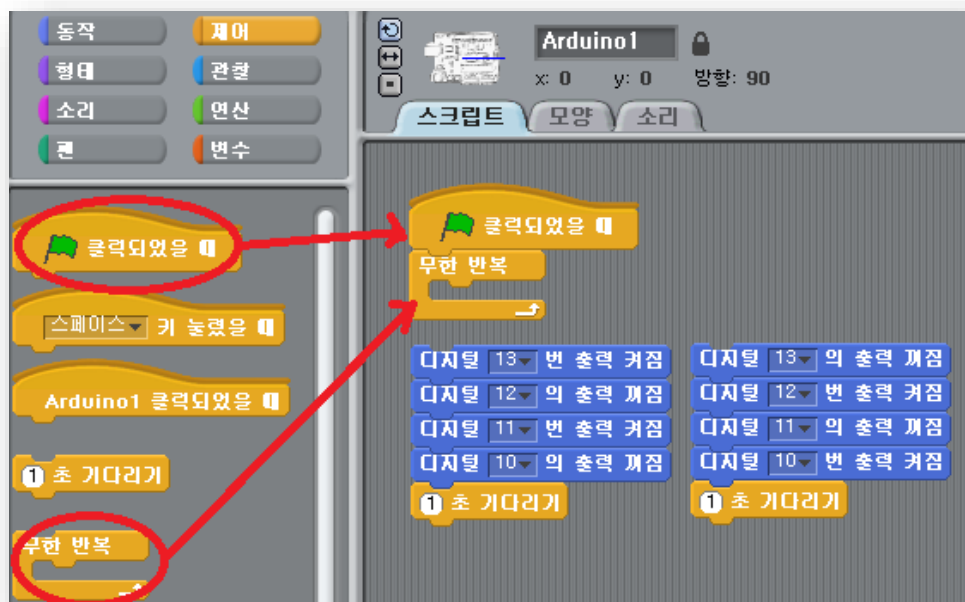


- ④ **제어** 블록 모음에서 **1 초 기다리기** 블록을 스크립트로 두 개 가져와 다음 그림과 같이 블록들 아래에 넣어줍니다.

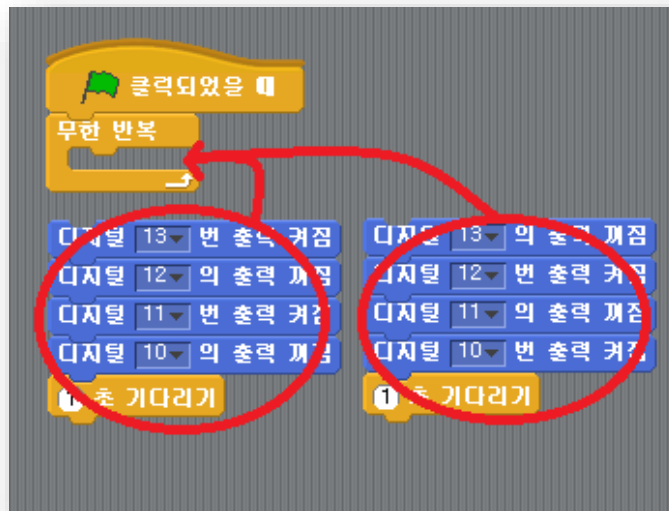



⑤ LED가 깜박이는 동작이 반복적으로 수행 되어야 하겠지요? **제어** 블록 모음

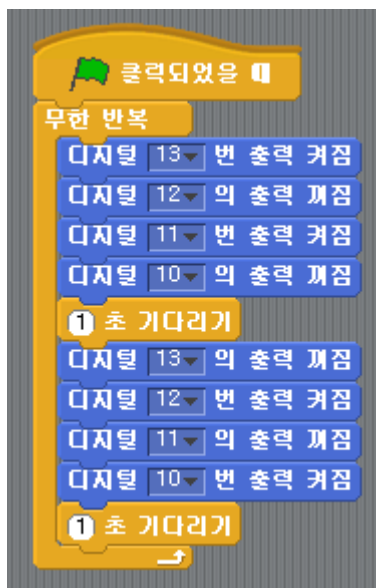
에서 **클릭되었을** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.



⑥ LED를 켜고 끄는 블록을 **무한 반복** 블록 안에 넣어 줍니다.

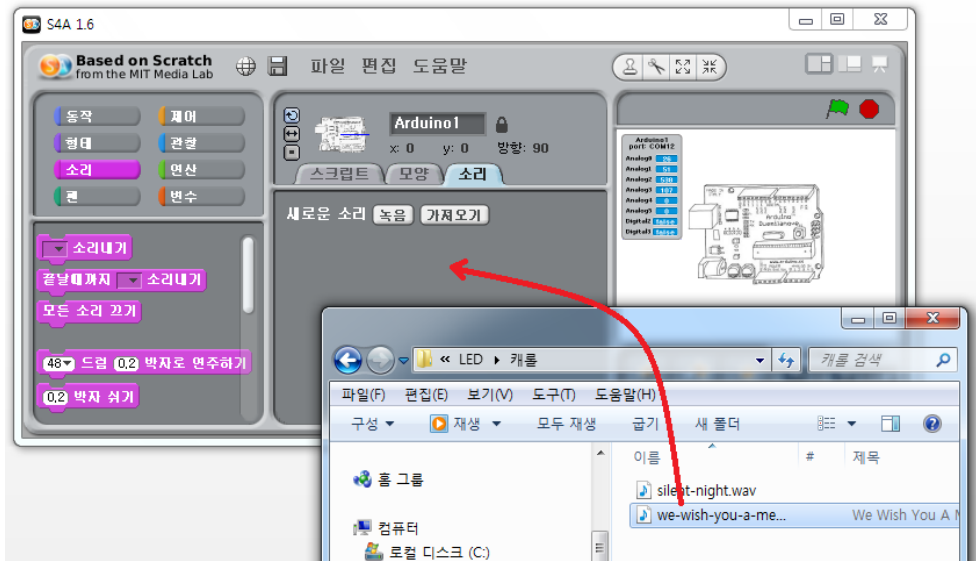


- ⑦ 다음 그림과 같이 블록이 완성되었나요?  버튼을 눌러 실행 시켜보세요. 4개의 LED들이 1초 간격으로 두 개씩 번갈아 깜박입니다. 좀더 빠르게 깜박이게 하기 위해서 초 간격을 0.5초 또는 0.3초 등으로 수정하여 입력해 주시면 됩니다.



## ㉔ 2 단계 : 크리스마스 캐롤 들려주기

- ① "소리" 탭을 누릅니다. "가져오기" 버튼을 눌러서 원하는 음악파일을 가져오거나, 다음 그림과 같이 탐색기에서 직접 파일을 드래그-앤-드롭 방식으로 가져옵니다.



② “we-wish-you-a-merry-christmas” 소리를 프로그램으로 가져왔습니다.



③ 이제, “스크립트” 탭을 눌러, 스크립트 작성 창을 띄웁니다. 그리고, **소리** 블록 모음에서 **끝날 때까지 we-wish-you-a-merry-christmas 소리내기** 블록을 스크립트로 가져옵니다.

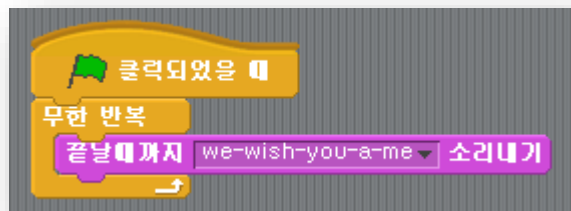


④ LED가 깜박일 때, 캐롤이 동시에 나오도록 해야 합니다. **제어** 블록 모음에서

**클릭되었을 때** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.



⑤ **끝날 때까지 we-wish-you-a-me 소리내기** 블록을 **무한 반복** 블록 안으로 넣습니다.



⑥ 다음 그림과 같이 블록이 완성이 되었나요? **클릭** 버튼을 클릭해 보세요. LED가 깜박이고, 캐롤이 동시에 흘러나오게 됩니다.



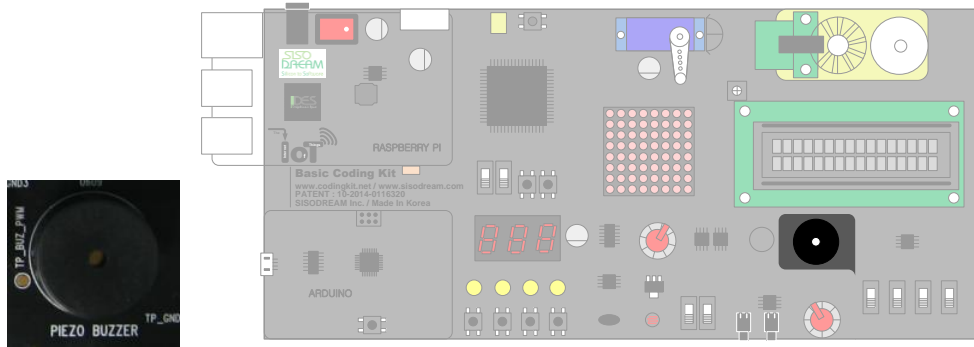
#### 🔗 생각하기 4

LED의 동작과 음악을 바꿔서 만들어 보세요. LED 는 오른쪽 2개, 왼쪽 2개가 번갈아 가면서 켜고 꺼지게 합니다. 그리고 음악은 "silent-night.wav" 파일을 이용하세요.

## 8. 도와주세요! 경보기



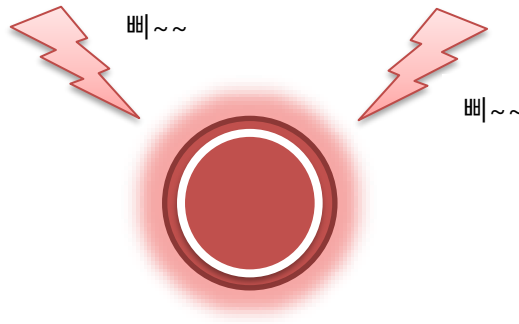
곰곰아! 코딩키트에 있는 이 까맣고 동그란 것이 뭔지 아니?



뭐예요? 잘 모르겠어요.



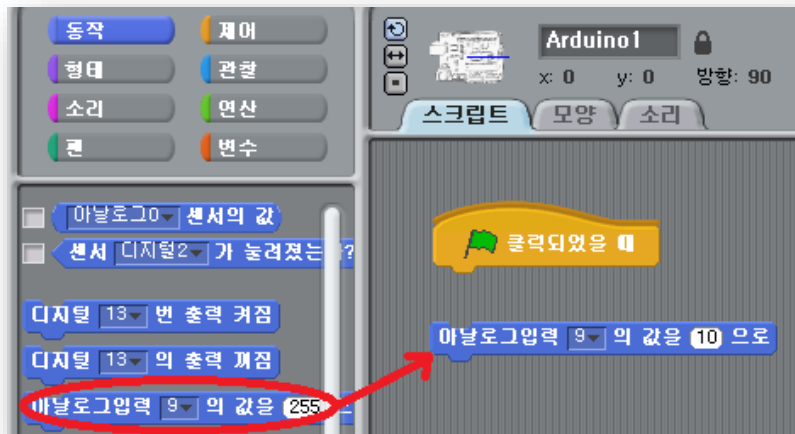
이것은 부저라는 거야! 부저는 신호를 주면 "삐~"하고 소리를 낸단다. 이 부저를 가지고 경보기를 만들어 보는 건 어떨까? 화재 경보기 같이 도움이 필요할 때, 소리를 내서 도움을 요청할 수 있는 장치를 만들어 보는 거야.



## 8-1. 부저 소리를 내보자

- ① **제어** 블록 모음에서 **클릭되었을** 블록을, **동작** 블록 모음에서 **아날로그입력 9**의 값을 **255**으로 블록을 스크립트로 가져옵니다.

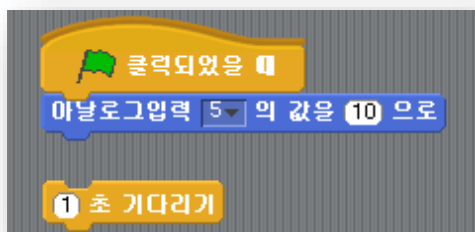




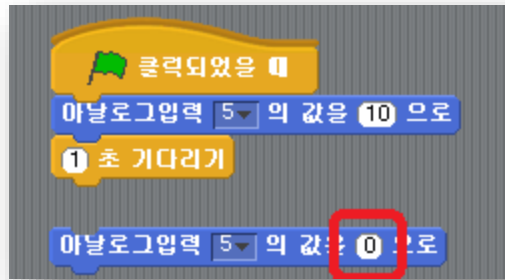
- ② **아날로그입력 9의 값을 255** 으로 블록의 첫 번째 입력 칸의 아래 화살표를 눌러 "5"를 선택하고 두 번째 입력 칸에 "255" 값 대신에 숫자 "10"을 입력합니다.




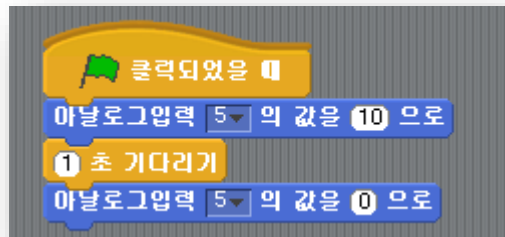
- ③ **제어** 블록 모음에서 **1 초 기다리기** 블록을 스크립트로 가져옵니다.



- ④ **동작** 블록 모음에서 **아날로그입력 9의 값을 255** 으로 블록을 스크립트로 한 개 더 가져옵니다. **아날로그입력 9의 값을 255** 으로 블록의 첫 번째 입력 칸의 아래 화살표를 눌러 "5"를 선택하고 두 번째 입력 칸에 "255" 값 대신에 숫자 "0"을 입력합니다. 1초 후 소리 크기가 0 이 되고 부저 소리가 나지 않게 됩니다.



- ⑤ 다음 그림과 같이 블록이 완성 되었나요?  버튼을 눌러보세요. “삐~” 소리가 1 초간 나고 멈춥니다.



## 생각하기 5

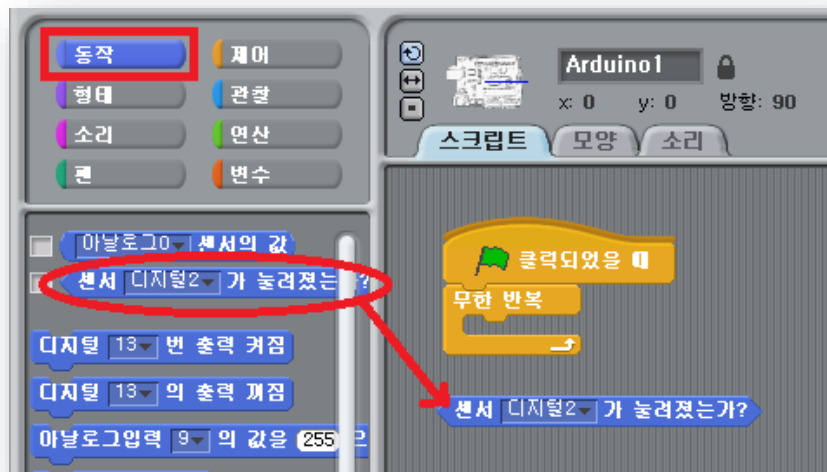
부저 소리의 크기를 변화시켜 보세요. (부저 소리는 0 ~ 10 까지만 사람의 귀로 확연한 차이를 느낄 수 있고 그 이상을 넘어가면 소리가 비슷하게 들립니다.)

## 8-2. 경보기 만들기

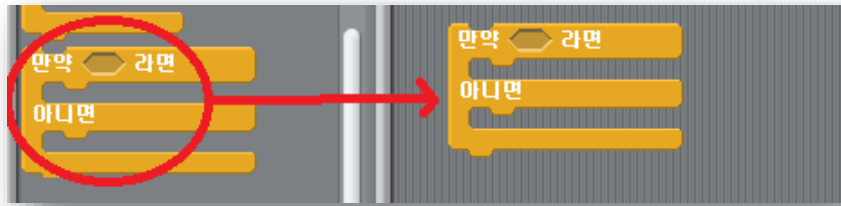
버튼이 눌리면, 부저 소리와 함께 비상등(LED)이 깜박이는 경보기를 만들어 보겠습니다. 지금까지 배웠던 LED, 버튼, 부저를 모두 사용해 볼 수 있는 예제입니다.


### 1 단계 : 경보기 동작 시키기

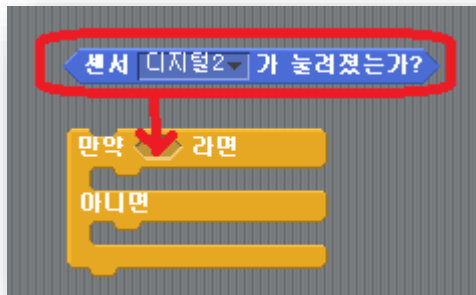
- ① **제어** 블록 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.
- ② **센서** 블록 모음을 클릭하세요. **센서 [디지털2] 가 눌려졌는가?** 블록을 스크립트로 가져옵니다.



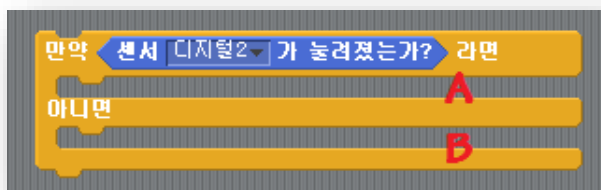
- ③ **제어** 블록 모음에서 **만약 ... 라면** / **아니면** 블록을 스크립트로 가져옵니다.



- ④ 그리고, **센서 디지털2가 눌러졌는가?** 블록을  의 첫 번째 입력 칸에 넣습니다.



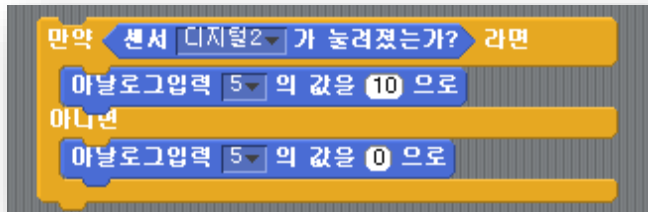
- ⑤ 디지털2 즉, 코딩킷의 버튼 ①이 눌렸으면 A 부분, 버튼 ①이 눌리지 않았으면 B 부분의 블록들이 실행되는 블록이 만들어 졌습니다.



- ⑥ **등적** 블록 모음에서 **아날로그입력 9의 값을 255으로** 블록 두 개를 스크립트로 가져옵니다. **아날로그입력 9의 값을 255으로** 블록의 첫 번째 입력 칸은 모두 "5"를 선택합니다. 그리고, 두 번째 입력 칸에 각각, 숫자 "10"과 숫자 "0"을 입력합니다.



- ⑦ 위 ⑥에서 만들어진 블록을 ⑤의 블록 A와 B에 다음 그림과 같이 각각 넣어 줍니다. ⑩번 버튼이 눌리면 부저가 울리고, 그렇지 않은 경우에는 부저가 울리지 않게 됩니다.



## ② 2 단계 : 경고등 깜박이기

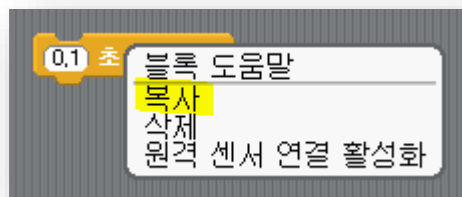
- ① 경고음이 울릴 때 경고등도 깜박이도록 하는 동작을 추가해보겠습니다. **동적** 블록 모음에서 **디지털 13번 출력 꺼짐** 블록과 **디지털 13의 출력 꺼짐** 블록을 스크립트로 가져옵니다.
- ② ⑩번 LED를 경고등으로 사용할 것 입니다. 첫 번째 입력 칸의 아래 화살표를 눌러서 "10"을 선택하여 주세요.



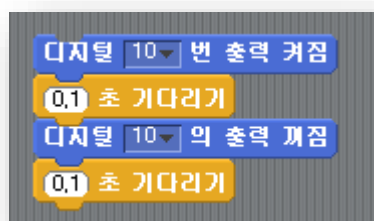
- ③ **제어** 블록 모음에서 **1 초 기다리기** 블록을 가져와 0.1초 기다리기 블록으로 수정합니다.



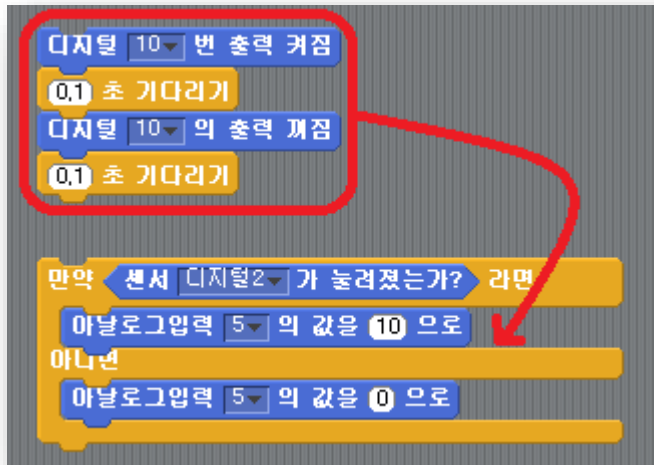
- ④ **0.1 초 기다리기** 블록을 한 개를 더 만들겠습니다. 위 **0.1 초 기다리기** 블록에 마우스 포인터를 가져가서 오른쪽 버튼을 클릭하시면 다음 그림과 같은 메뉴가 나옵니다. "복사"를 선택하여 주세요.



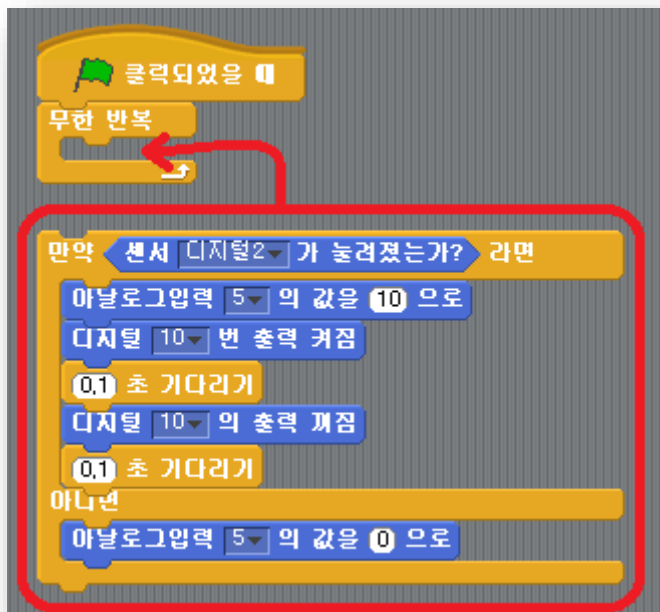
- ⑤ **디지털 10번 출력 꺼짐** 과 **디지털 10번의 출력 켜짐** 블록 사이에 **0.1 초 기다리기** 을 넣어 줍니다.




- ⑥ 경고음이 나오면 경고등이 깜박이도록 해야겠지요? 위 ⑤에서 만든 블록을 아래 그림과 같이 경고음이 울리는 블록 아래에 넣어 줍니다.



- ⑦ 마지막으로, ⑥의 완성된 블록을  블록 안에 넣어 줍니다.



- ⑧ 이제  버튼을 눌러서 잘 동작하는지 확인해 보세요. 코딩 키트의 ⑩번 버튼을 누르면 경보기가 동작합니다. 그리고 다시 버튼을 누르면 경보기가 멈출 것 입니다.





## 9. 디지털과 아날로그



곰곰이는 디지털과 아날로그라는 말 들어봤니?



아~ 네! TV 광고에서 디지털 TV라는 말을 들어봤어요.



그래! 맞아! TV, 핸드폰 이런 전자 제품들이 모두 디지털 신호를 이용하여 만들어진 거란다. 우리가 지금까지 LED를 켜고 끄기 위해 "참" 또는 "거짓" 신호를 보냈었던단다. 그리고 버튼이 눌렸는지에 대한 정보를 가져올 때도 "참" 또는 "거짓" 값으로 받아왔지. 이렇게 "참" 또는 "거짓" 이라는 서로 반대되는 2 가지 신호만을 사용하는 것을 **디지털 신호**라고 한단다. 이 "참" 또는 "거짓" 은 "0" 또는 "1", "ON" 또는 "OFF", "true" 또는 "false" 등으로도 표현한단다.

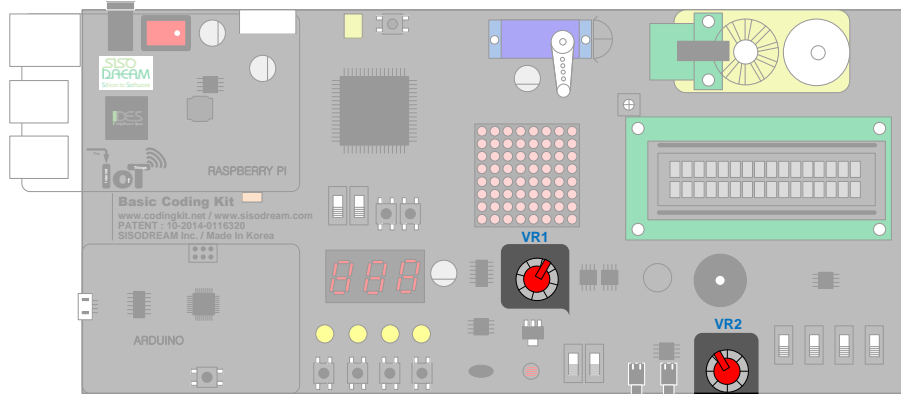


디지털과 다르게 연속적으로 변하는 값을 표현 할 때 주로 사용하는 것이 **아날로그 신호**라는 것이란다. 자동차의 바늘 속도계나, 무게를 달기 위해 사용하는 바늘 저울 등이 아날로그 신호로 표현하는 것이란다.



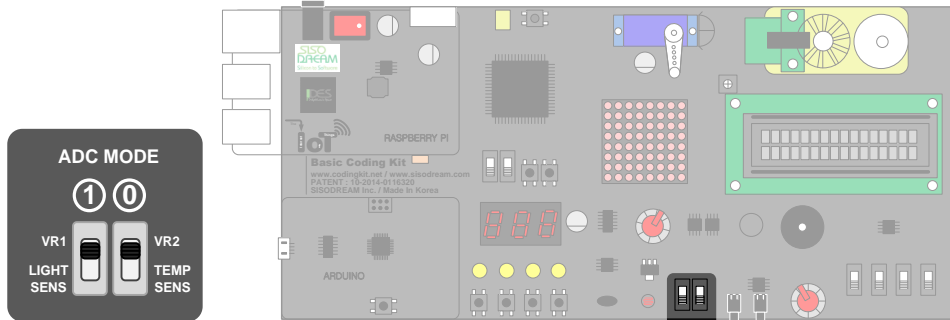


코딩키트에서 아날로그 신호를 다루어 볼 수 있는 가변 저항이라는 장치가 있단다. 다이얼을 돌리는 장치 두 개가 있는데 VR1, VR2 장치란다. 이 장치는 아날로그 신호를 0 에서부터 1023 까지의 값으로 전달해 준단다.

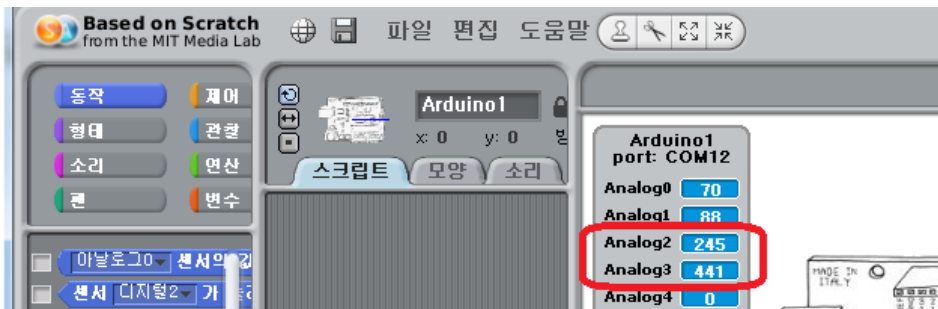


## 9-1. 가변 저항으로 화면을 조절하기

가변 저항을 사용하기 위해 ADC 모드 스위치를 모두 위로 올려주세요.

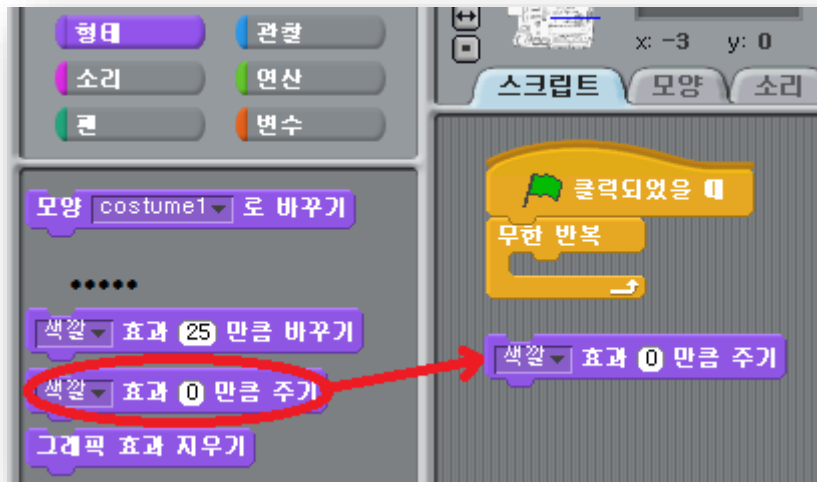


그리고, 가변 저항을 돌려보세요. 그러면, S4A 프로그램의 무대화면에 장치들의 상태 값 중에 Analog2와 Analog3의 값이 변하는 것을 확인 하실 수 있습니다.

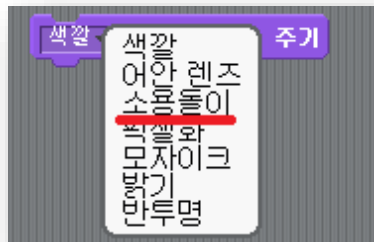


가변 저항을 사용하여 무대 화면을 조절하는 동작을 만들어 보겠습니다.

- ① **제어** 블록 모음에서 **클릭되었을** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.
- ② **형태** 블록 모음에서 **색깔** 효과 **0** 만큼 주기 블록을 스크립트로 가져옵니다.



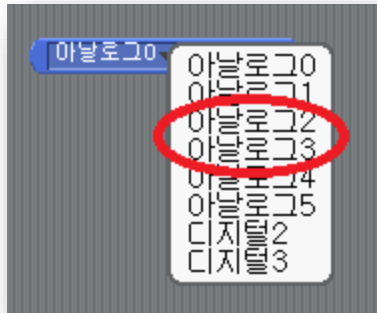
- ③ **색깔** > **효과 0 만큼 주기** 블록에서 첫 번째 입력 칸에 있는 아래 화살표를 클릭해 보세요. 다양하고 재미있는 효과를 주는 메뉴가 있습니다. 여기에서, "소용돌이"를 선택해 주세요.



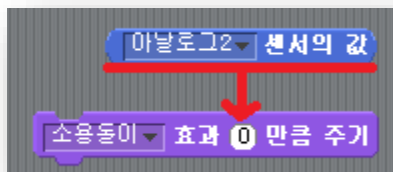
- ④ **동작** 블록 모음에서 **아날로그** > **센서의 값** 블록을 스크립트로 가져옵니다.



- ⑤ 입력 칸의 아래 화살표를 누르면, 다음 그림과 같이 아날로그0~아날로그5, 디지털 2~3의 값을 가져올 수 있습니다. 가변저항의 값을 쓸 것이므로, "아날로그 2"(VR2) 또는 "아날로그 3"(VR1)을 선택하세요.




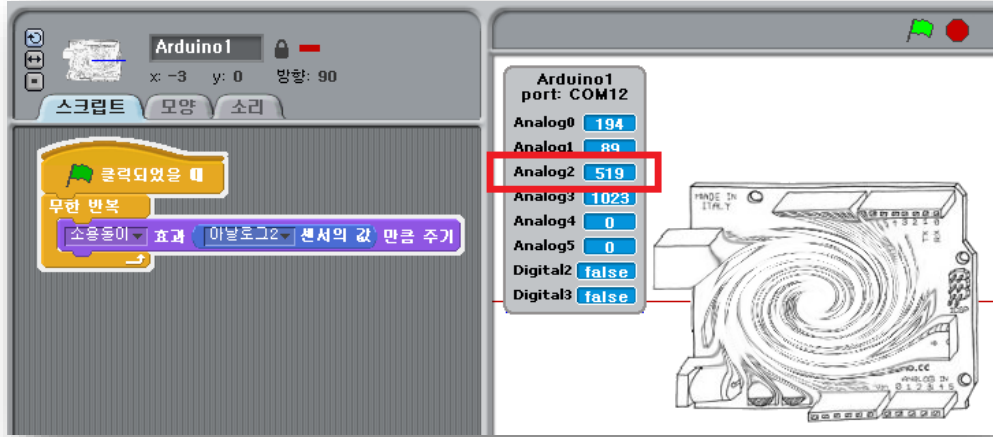
- ⑥ **색깔** ▾ **효과** 0 만큼 주기 블록의 두 번째 칸에는 위 ⑤의 블록을 넣어 줍니다. 가변 저항의 값이 들어가게 되는 것입니다.



- ⑦ 이제 ⑥에서 만들어진 "소용돌이 효과를 가변 저항에서 전달되는 값만큼 주는" 블록을 **무한 반복** 블록 안에 넣어줍니다.



- ⑧ 다음 그림과 같이 완성되었나요?  버튼을 누르고, 가변저항 VR2를 돌려보세요. 무대 화면의 그림이 가변 저항의 값에 따라 변하는 것을 보실 수 있습니다.

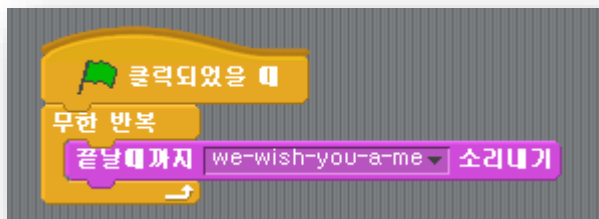


## 9-2. 가변 저항으로 음량을 조절하기

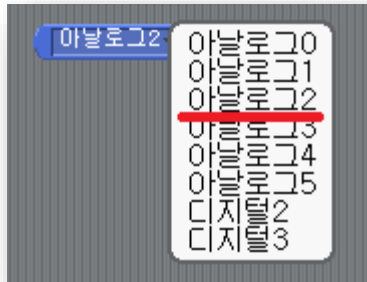
가변 저항의 다이얼을 돌려서 음량을 조절하는 동작을 만들어 보겠습니다. 여기에 LED로 음량의 크기를 보여주면 더 좋을 것 같습니다.

### 🌀 1 단계 : 가변 저항으로 음량 조절하기

- ① 음악파일을 가져와 들려주는 동작은 7장에서 설명하였습니다. 7-1절의 "2단계 : 크리스마스 캐롤 들려주기"를 참고하여 다음 그림과 같이 캐롤을 들려주는 블록을 작성합니다.



- ② **등록** 블록 모음에서 **아날로그2** 센서의 값 블록을 스크립트로 가져옵니다. 첫 번째 입력 칸에 있는 아래 화살표를 눌러서 "아날로그 2" 센서 값을 갖는 블록으로 설정합니다.





- ③ **소리** 블록 모음에서 **음량 100 % 로 맞추기** 블록을 스크립트로 가져옵니다.



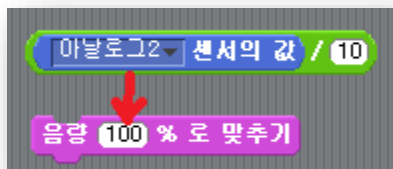
- ④ 가변 저항에서 전달되는 값은 0~1023까지의 값입니다. 음량을 최대 100%로 맞추어야 하므로, 전달되는 가변 저항 값을 10으로 나누어 줍니다. **연산** 블록 모음에서 **/** 블록을 가져옵니다.



- ⑤  블록의 첫 번째 입력 칸에는  블록을 넣어 주고, 두 번째 입력 칸에는 숫자 "10"을 입력합니다.




- ⑥ 위 ⑤에서 만들어진 블록을  블록의 첫 번째 칸에 넣어 줍니다.

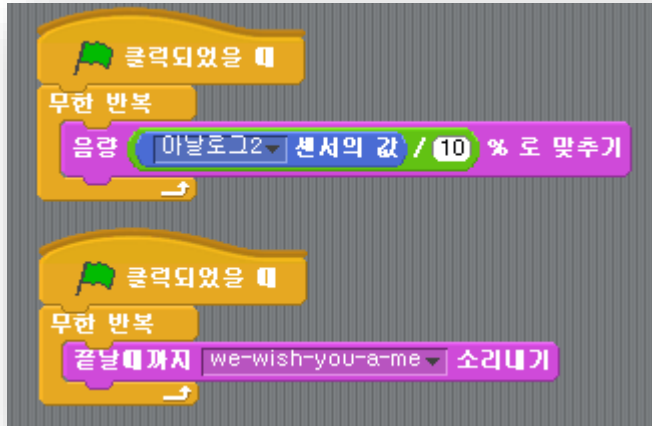


- ⑦  블록 모음에서  블록과  블록을 스크립트로 가져와서 ⑥에서 완성된 블록을  블록 안으로 넣어 줍니다.



- ⑧ 다음 그림과 같이 완성되었나요?  버튼을 눌러서 가변 저항 다이얼을 돌려 보세요. 다이얼을 왼쪽으로 돌리면 소리가 작아지고, 오른쪽으로 돌리면 소리가 커집니다.

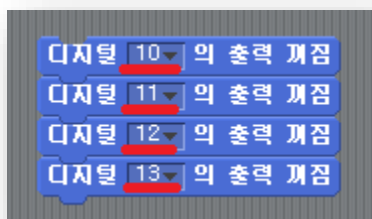




## 2 단계 : LED로 음량 보여주기

소리가 작아지거나 커지는 것을 눈으로도 확인 할 수 있도록 LED로 음량의 크기를 나타내는 기능을 추가해 보겠습니다. **디지털 13번 출력 켜짐**

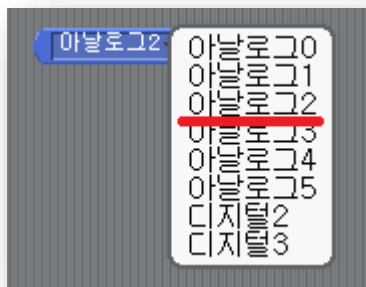
- LED를 모두 꺼짐 상태로 만들어 줍니다. **등적** 블록 모음에서 **디지털 13번의 출력 꺼짐** 블록을 스크립트로 4개를 가져옵니다. 그리고 첫 번째 입력 칸의 아래 화살표를 눌러서 10부터 13까지 차례로 선택해 줍니다.



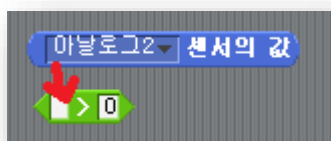
- 이제 가변 저항의 값에 따라서 LED를 켜지 말지를 결정해야 합니다. **등적** 블록 모음에서 **아날로그2 센서의 값** 블록을, **연산** 블록 모음에서 **IF >** 블록을, **제어** 블록 모음에서 **만약 < > 라면** 블록을 스크립트로 가져옵니다.



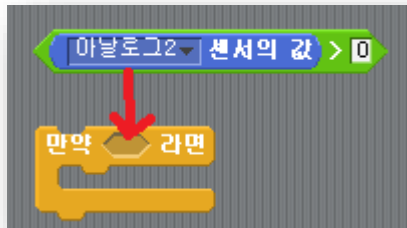
- ③ **아날로그0 센서의 값** 블록의 첫 번째 입력 칸에 있는 아래 화살표를 눌러서 “아날로그 2” 센서 값을 갖는 블록으로 설정합니다.



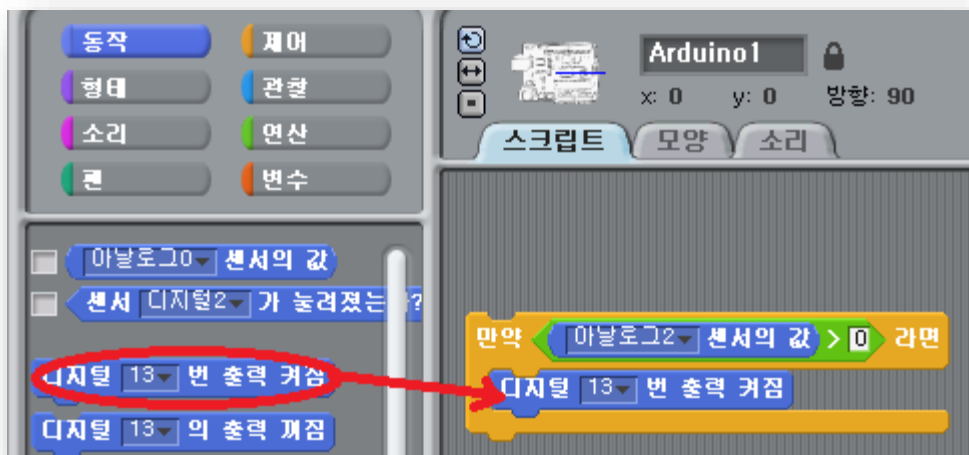
- ④ **>** 블록의 첫 번째 입력 칸에는 **아날로그2 센서의 값** 블록을 넣어 주고, 두 번째 입력 칸에는 숫자 “0”을 입력합니다.



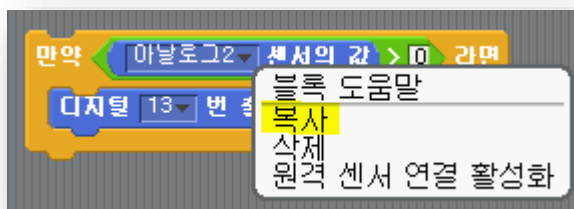
- ⑤ 위 ④에서 만들어진 블록을 **만약 라면** 블록의 첫 번째 입력 칸에 넣어 줍니다.



- ⑥ 첫 번째 LED는 가변 저항 값이 "0" 이상이면 켜지도록 합니다. **동작** 블록 모음에서 **디지털 13번 출력 켜짐** 블록을 가져와서 위 ⑤ 번에서 만들어진 블록의 **만약 > 0이면** 블록 안에 넣어 줍니다.



- ⑦ 6번에서 만들어진 블록에 마우스를 가져가서 오른쪽 클릭을 합니다. 그리고 "복사" 메뉴"를 선택합니다.



- ⑧ 코딩키트의 LED가 4개 이므로 3개를 더 복사합니다.




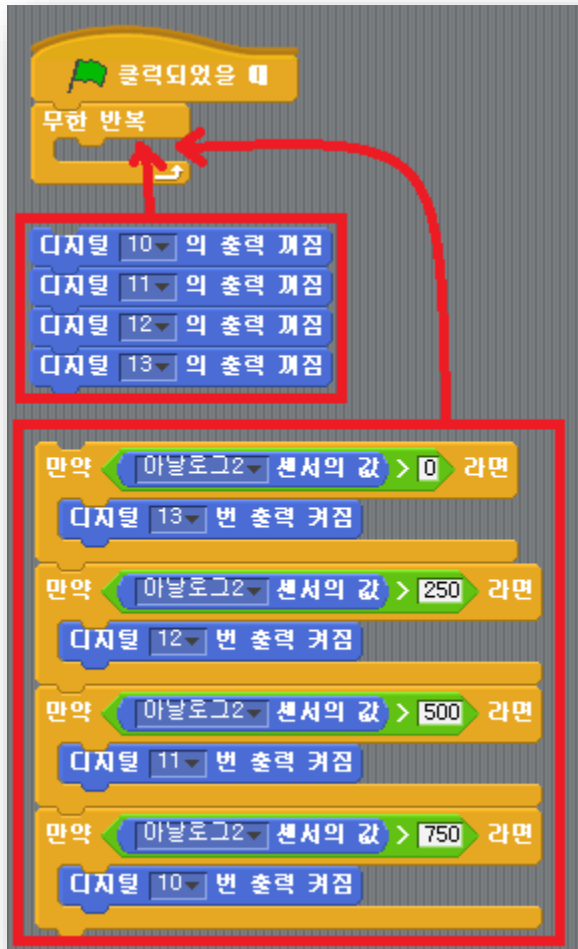
- ⑨ 가변 저항 값이, 250, 500, 750 값을 넘을 때 마다 LED를 추가로 하나씩 더 켜지도록 다음 그림과 같이 수정 합니다. **디지털 13번 출력 켜짐** 블록의 번호를 그림과 12, 11, 10 번으로 바꿉니다.




- ⑩ **제어** 블록 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스



크립트로 가져옵니다. 그리고, ①과 ⑨에서 완성된 블록을  블록 안으로 넣어 줍니다



- ① 캐롤이 나오는 블록, 소리 크기가 조절되는 블록, 소리 크기에 따라 LED 가 켜지는 블록 이렇게 3개의 블록이 완성되었습니다.  버튼을 누르고, 가변 저항의 다이얼을 오른쪽 또는 왼쪽으로 돌려보세요. 노래 소리가 커지거나 작아지고, LED 등도 노래 소리 만큼 켜졌다가 꺼졌다가 합니다.



## 10. 부릉~ 부릉~ 자동차!



아빠! 저랑 장난감 자동차로 자동차 경주 해요! 부릉~ 부릉~ 아! 빨리 커서 아빠처럼 진짜로 자동차를 운전해보고 싶어요.



허허허! 아빠도 우리 곰곰이가 운전하는 차를 타면 정말로 뿌듯할 것 같구나. ^^



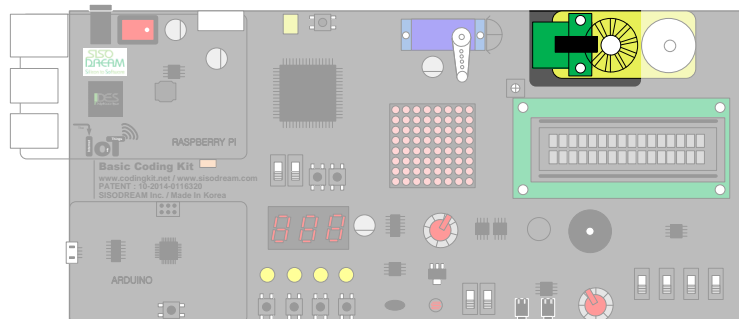
그런데 자동차는 어떻게 움직이는 거예요? 궁금해요.



음.. 그러면, 코딩키트로 자동차 바퀴가 굴러가는 동작을 한번 만들어 볼까? 코딩키트에 DC 모터라는 장치가 있는데, 장난감 자동차 바퀴를 만들 때는 이런 빙글 빙글 돌아가는 모터를 사용한단다.

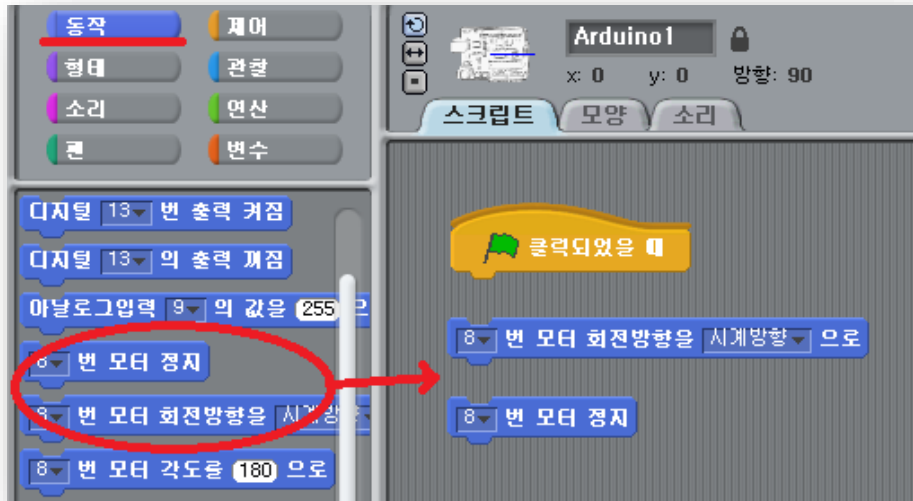
### 10-1. 바퀴를 움직여 볼까?

코딩키트에 DC 모터가 있습니다. 장난감 자동차 바퀴를 만들 때는 이런 빙글 빙글 돌아가는 모터를 사용합니다. 코딩키트의 DC 모터를 사용하여 자동차 바퀴가 굴러가는 동작을 만들어 보겠습니다.

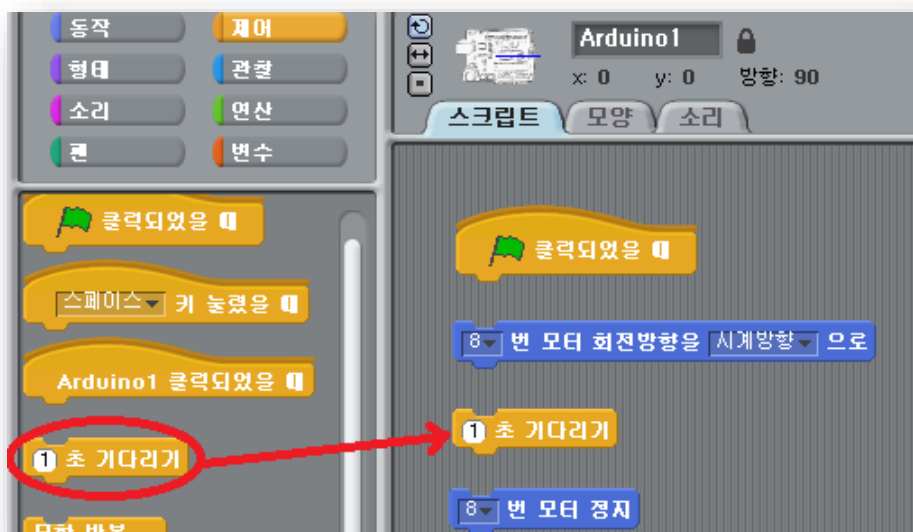


① **제어** 블록 모음에서 **클릭되었을** 블록을 스크립트로 가져옵니다.


② **동작** 블록 모음에서 **8번 모터 회전방향을 시계방향** 으로 블록과 **8번 모터 정지** 블록을 스크립트로 가져옵니다.

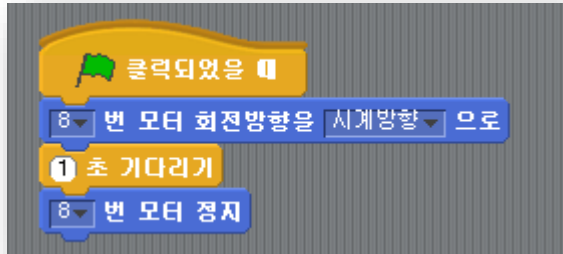


③ 모터를 1 초간 돌린 후 멈추게 하려고 합니다. **제어** 블록 모음에서 **1 초 기다리기** 블록을 스크립트로 가져옵니다.





- ④ 스크립트로 가져온 위 블록들을 모두 연결하여 다음 그림과 같이 완성합니다. 
- 버튼을 눌러 보세요. 모터가 시계방향으로 돌고 1초 후 멈춥니다.



## 생각하기 6

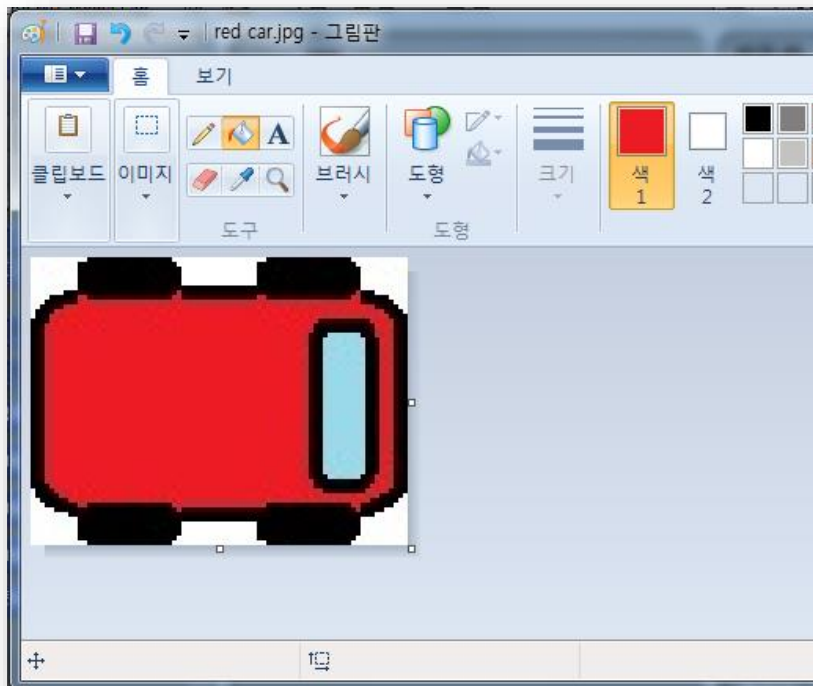
모터를 반시계방향으로 돌도록 만들어 보세요.

## 10-2. 자동차를 만들어 보자

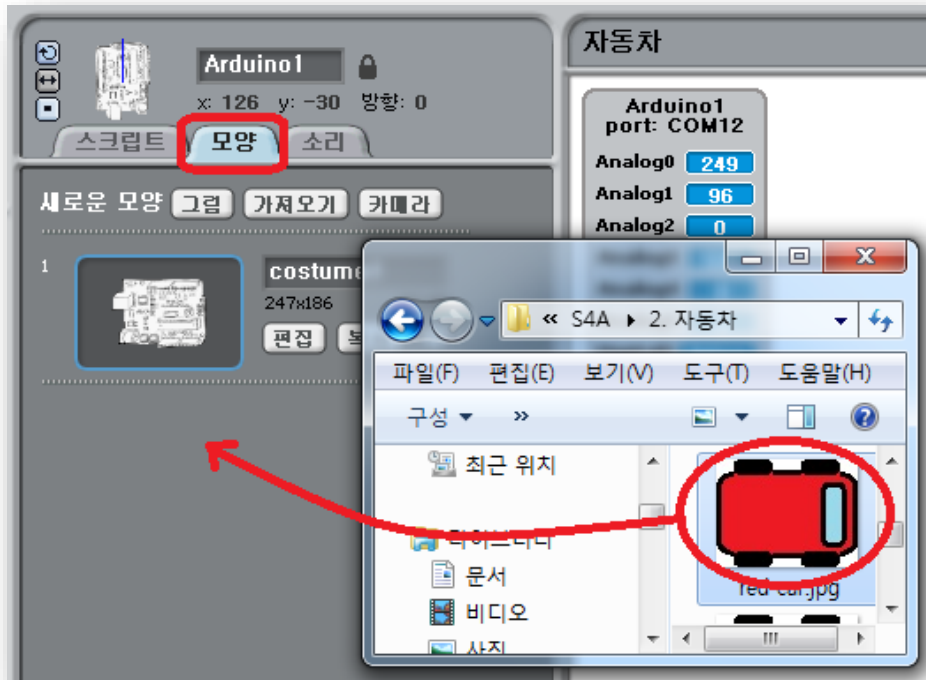
엑셀을 밟으면 자동차가 앞으로 가고, 엑셀에서 밟을 때면 멈추는 자동차를 만들어보겠습니다. 코딩키트의 버튼은 엑셀, 모터는 자동차 바퀴, 가변저항은 핸들이 됩니다. 자동차의 움직임은 무대 화면 상으로 보여지도록 하겠습니다.

### 1 단계 : 자동차 그림 삽입하기

- ① 그림판을 열어서 자동차 그림을 그립니다. "car.jpg"로 저장합니다. 그림 그리기가 번거로우면 "...WS4AW그림W자동차" 폴더에서 가져오기 해도 됩니다.



- ② S4A 프로그램에서 "모양"탭을 누릅니다. 그리고, 탐색기를 열어서 위에서 저장한 "car.jpg" 그림을 찾아 선택하고, "모양"탭 안으로 드래그-앤-드롭 방식으로 그림을 가져다 놓습니다. "가져오기" 버튼을 눌러 가져와도 됩니다.

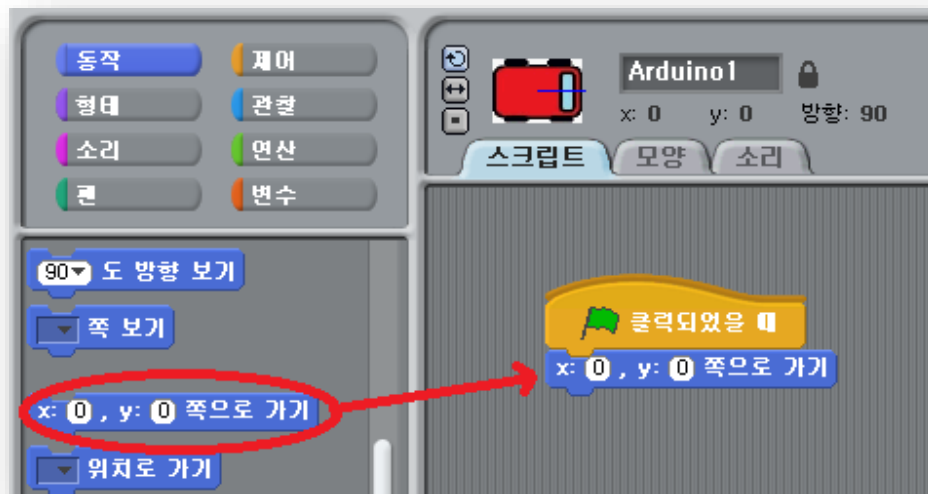


- ③ 다음 그림과 같이 "car.jpg" 그림이 "모양" 탭에 추가 되고, 아두이노 보드 그림이었던 스프라이트가 자동차 그림으로 변경됩니다.

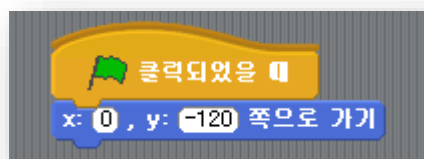


## 2 단계 : 자동차 방향 제어하기

- ① **제어** 블록 모음에서 **클릭되었을 때** 블록을 스크립트로 가져옵니다.
- ② **동작** 블록 모음에서 **x: 0, y: 0 쪽으로 가기** 블록을 스크립트로 가져옵니다. 이 블록은 무대 화면 상의 스프라이트의 위치를 지정해 줄 수 있습니다.



- ③ **x: 0, y: 0 쪽으로 가기** 블록에 x는 숫자 "0"을 y는 숫자 "-120"을 입력합니다.



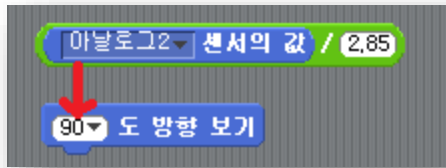
- ④ 핸들, 즉 가변 저항을 돌리면 무대에 보여지는 자동차의 방향이 바뀌는 동작을 추가하겠습니다. **동작** 블록 모음에서 **아날로그 센서의 값** 블록과 **90도 방향 보기** 블록을 스크립트로 가져옵니다.



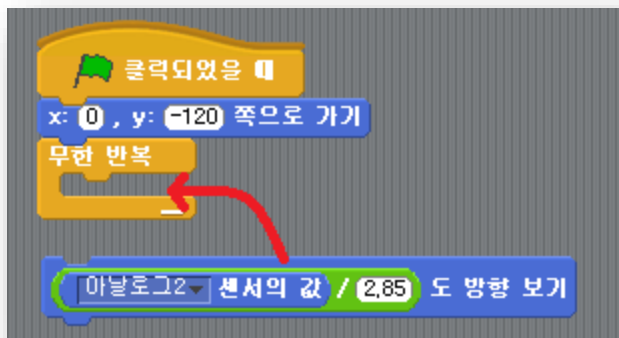
- ⑤ 아날로그2 센서의 값, 즉 VR2 값으로 자동차(스프라이트)의 방향을 조절하겠습니다. VR2 값은 0~1023의 값을 가지므로 0~360(원의 각도)의 값을 가지도록 2.85로 나누어 사용할 것 입니다. 수식 계산을 위해 **연산** 블록 모음에서 **0 / 0** 블록을 스크립트로 가져옵니다. 그리고, 두 번째 입력란에 숫자 "2.85"를 입력합니다.



- ⑥ **0 / 0** 블록의 첫 번째 입력 칸에 **아날로그2 센서의 값** 블록을 넣어 줍니다. 그리고, 이 블록을 **90도 방향 보기** 블록의 첫 번째 칸에 넣어 줍니다.





- ⑦ **제어** 블록 모음에서 **무한 반복** 블록을 스크립트로 가져옵니다. 위 ⑥에서 만들어진 블록을 **무한 반복** 블록 안에 넣습니다. 여기서 버튼을 누르고, 가변 저항의 다이얼을 오른쪽 또는 왼쪽으로 돌려보세요. 가변 저항의 움직임에 따라 자동차의 진행 방향이 바뀌는 것을 확인하실 수 있습니다.

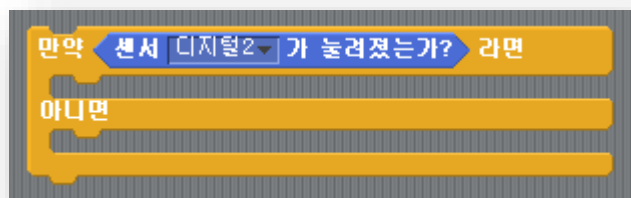





### 3 단계 : 자동차 움직이기

- ① **동작** 블록 모음에서 **센서 디지털2가 눌러졌는가?** 블록과 **제어** 블록 모음에서 **만약...그러면** / **아니면** 블록을 스크립트로 가져옵니다.



- ②  블록의 첫 번째 입력 칸에  블록을 넣어 줍니다.



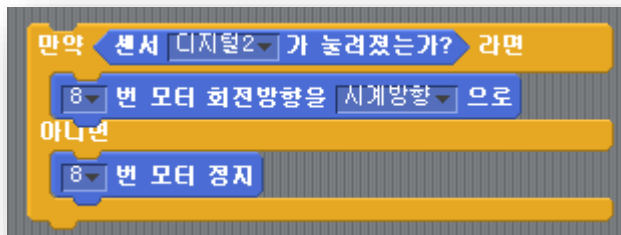
- ③ 디지털2 즉, ①번 버튼이 눌리면 바퀴(모터)가 돌아가도록 합니다. 그리고, ①번 버튼이 눌리지 않았을 때는 바퀴(모터)가 돌아가는 것이 멈추도록 합니다.  블록 모음에서  블록과  블록을 스크립트로 가져옵니다.



- ④ **8번 모터 회전방향을 시계방향으로** 블록과 **8번 모터 정지** 블록을



블록 안에 다음 그림과 같이 넣어 줍니다. 이제 버튼이 눌리면 모터가 돌아갈 것입니다.



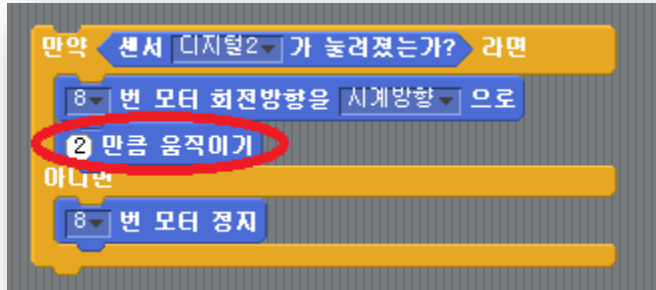
- ⑤ 모터가 돌아갈 때, 무대 화면의 자동차 그림도 앞으로 전진하도록 만들어 줍니다.

**동작** 블록 모음에서 **10만큼 움직이기** 블록을 가져와, 첫 번째 입력 칸에 각각 숫자 "2"를 입력합니다.

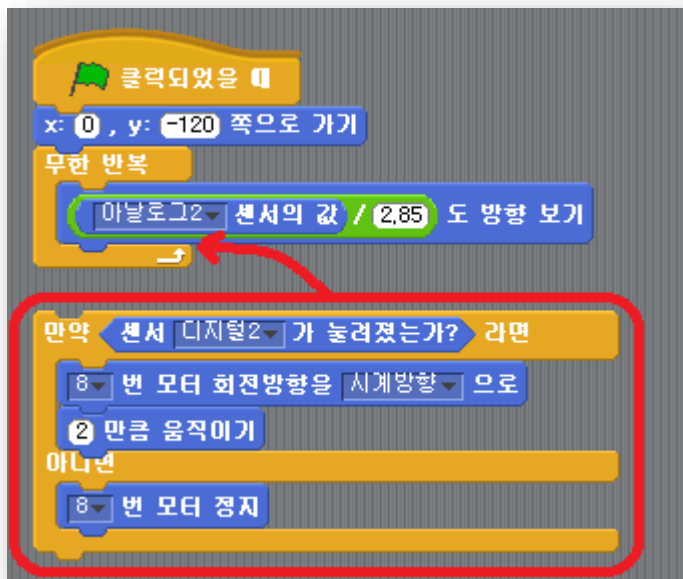





- ⑥ 위 ⑤의 블록을 다음 그림과 같이 **8** 번 모터 회전방향을 시계방향으로 블록 아래에 넣어 줍니다.



- ⑦ 2단계에서 완성된 블록에 위 ⑥의 블록을 다음 그림과 같이 넣어 줍니다.



- ⑧ 다음 그림과 같이 블록을 완성하였다면,  버튼을 눌러 보세요. 그리고, 0번 버튼을 누르고 가변 저항(VR2)을 움직여 보세요. 자동차가 원하는 방향으로 잘 움직이나요?

```

    클릭되었을 때
    x: 0, y: -120 쪽으로 가기
    무한 반복
    아날로그2 센서의 값 / (2.85) 도 방향 보기
    만약 센서 디지털2 가 눌러졌는가? 라면
    8 번 모터 회전방향을 시계방향 으로
    2 만큼 움직이기
    아니면
    8 번 모터 정지
  
```



### 🔗 생각하기 7

자동차에 후진 기능을 추가하여 만들어 보세요.

# 11. 끼익~ 끼익~ 로봇팔!



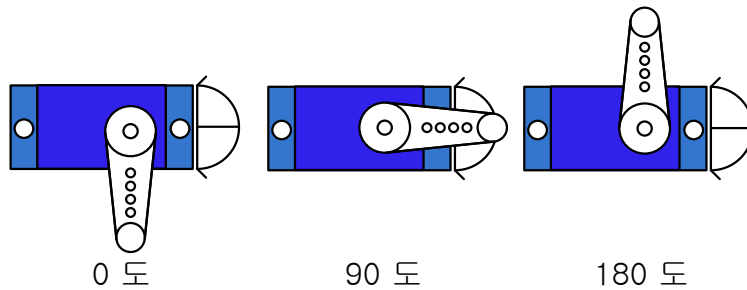
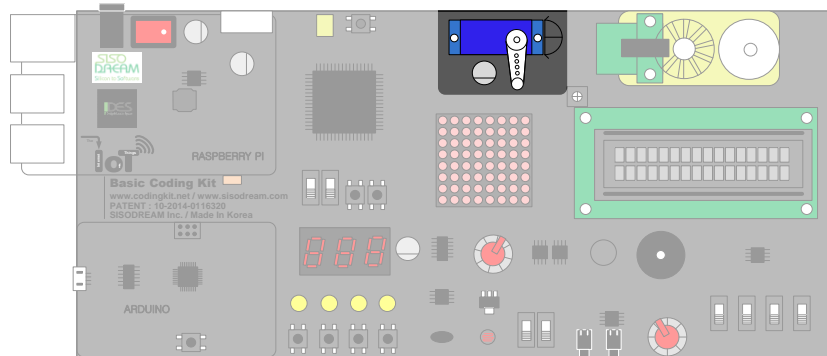
코딩킷에 또 다른 모터가 하나 더 있단다. 바로 서보 모터라는 거야. 이 모터는 로봇을 만들 때, 관절 등에 주로 사용하는 모터이지.



와! 로봇이라고요?



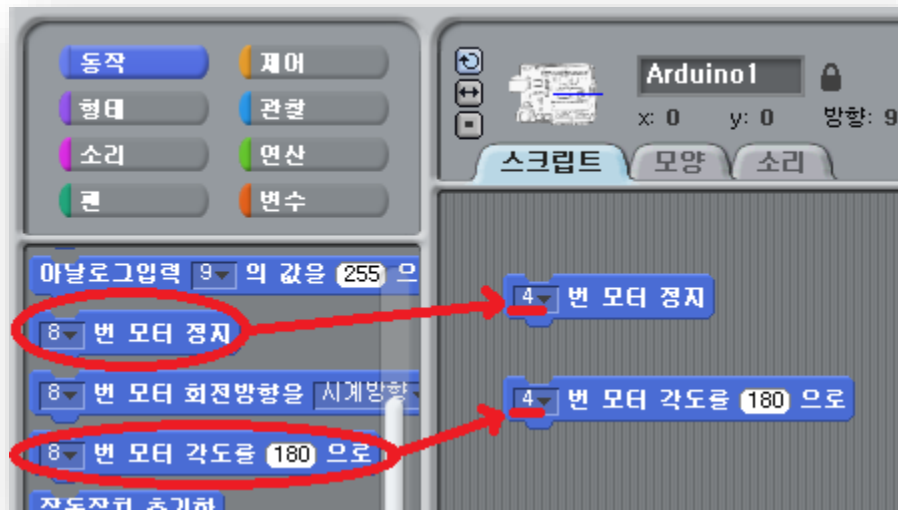
그래! 로봇! 우리 곰곰이가 로봇을 좋아하지? 로봇 팔의 관절에 서보 모터를 장착하면, 일정한 각도로 움직인단다. 서보 모터를 가지고 로봇 팔을 움직이는 것처럼 코딩을 해보면 재미있을 것 같은데, 한번 만들어 볼까?



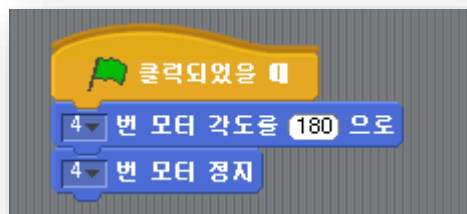
## 11-1. 서보 모터를 움직여 보자.

간단하게 서보 모터를 움직이는 동작을 만들어 보겠습니다.

- ① **동작** 블록 모음에서 **8번 모터 정지** 블록과 **8번 모터 각도를 180도로** 블록을 스크립트로 가져옵니다. 코딩 키트의 서보 모터는 4번 모터입니다. 4번 모터로 변경해 주세요.




- ② **제어** 블록 모음에서 **클릭되었을 때** 블록을 스크립트로 가져와, 위 블록들을 순서대로 배열합니다.



- ③ 위 ②는 동작이 되지 않을 것입니다. 서보 모터가 180도 방향으로 돌아가기 전에 모터가 정지해 버리기 때문입니다. 서보 모터가 움직일 수 있는 시간을 주기 위해

**제어** 블록 모음에서 **1 초 기다리기** 블록을 가져와 **4 번 모터 정지** 블록 위에 놓습니다. 그리고, 숫자 "1" 대신에 "0.1"을 입력합니다. 서보 모터가 움직이는데 0.1초면 충분합니다. 서보 모터가 충분히 움직이지 못 했다는 생각이 들면 이 값을 좀 더 키워 주세요.



- ④ 서보 모터의 각도를 변경하면서,  버튼을 눌러 보세요. 그러면, 서보 모터가 정해진 각도로 위치를 변경한 후, 멈출 것 입니다.

서보 모터가 어떤 각도에서 정확히 멈추어 있지 않고 약간씩 흔들일 수도 있습니다. 이것은 S4A에서 주는 신호가 약간씩 지연되어 전달되기 때문입니다.

### 생각하기 8

깃발을 좌우로 흔드는 로봇팔을 만들어 보세요. 서보모터의 위치를 0도와 90도를 왔다 갔다 하게 합니다.

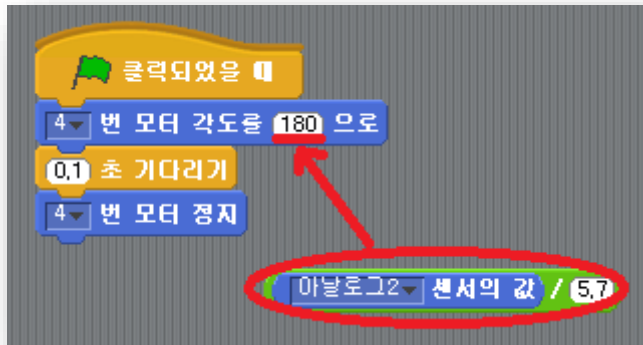
## 11-2.원격조정 로봇 팔을 만들자


이전의 8 장에서 배운 가변 저항으로 서보 모터의 각도를 조절하는 동작을 만들어 보겠습니다. 위 11-1 절에서 만들어진 스크립트에 블록을 추가하여 만들어 보겠습니다.

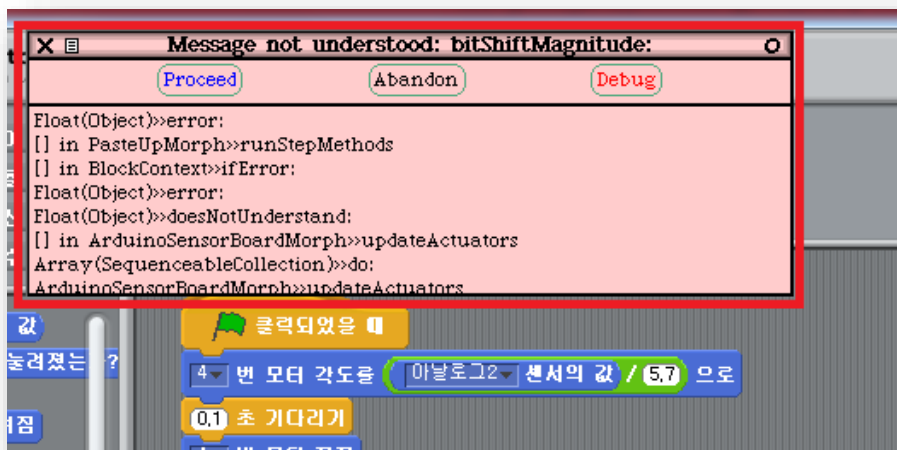
- ① 가변 저항을 사용할 것이므로, **동작** 블록 모음에서 **아날로그0 센서의 값** 블록을 스크립트로 가져옵니다. 첫 번째 입력 칸의 화살표를 눌러서, "아날로그2" 로 변경합니다.
- ② 코딩 키트의 서보 모터는 0~180도 까지 움직입니다. 가변 저항은 0~1023까지의 값을 전달하므로, 5.7로 나누어서 0~180사이의 값을 전달하도록 만듭니다. 수식 계산을 위해 **연산** 블록 모음에서 **/** 블록을 스크립트로 가져옵니다. **/** 블록의 첫 번째 입력 칸에는 **아날로그2 센서의 값** 블록을 넣고, 두 번째 입력 칸에는 숫자 "5.7"을 입력합니다.





- ③ **4번 모터 각도를 180 으로** 블록의 두 번째 입력 칸에 위 ②의 블록을 넣어 주세요.



- ④ 이 상태에서  버튼을 눌러 실행을 시키면, 아래 화면과 같이 에러 화면이 나옵니다. 5.7로 나눈 결과 소수점이 발생하기 때문입니다.




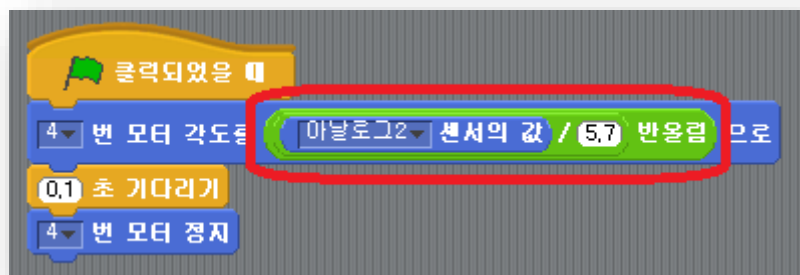
- ⑤ 이 문제를 해결하기 위해서 결과 값을 반올림하도록 합니다.  블록 모음에서  블록을 스크립트로 가져옵니다.





- ⑥ **반올림** 블록의 입력 칸에 **아날로그2 센서의 값 / 5.7** 블록을 넣어 줍니다.





- ⑦ 반올림된 값이 서보 모터에 전달되도록 수정하였습니다.  버튼을 눌러 실행을 시켜보세요. 에러가 나지 않을 것입니다.

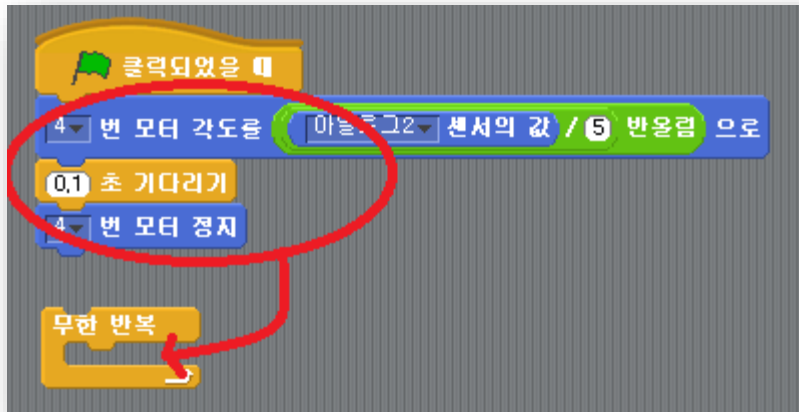



- ⑧ 그런데, 위 ⑦의 상태에서는 가변 저항을 돌려보면, 서보 모터가 돌아가지 않습니다. 서보 모터가 한번 움직이고 멈추기 때문입니다. 가변 저항 값을 계속 받아서 서보

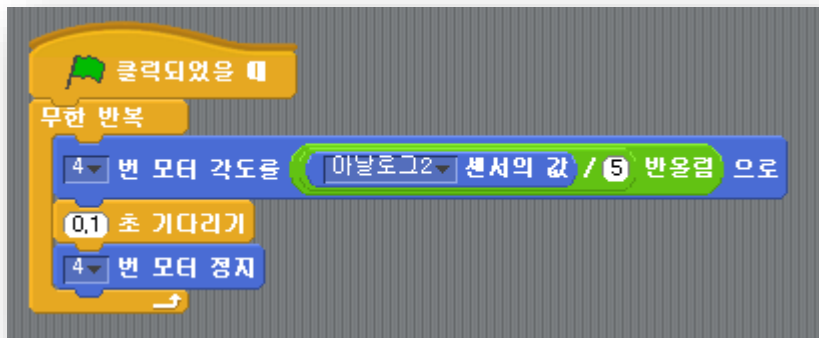
모터를 돌리기 위해서는  블록이 필요합니다.  블록 모음에

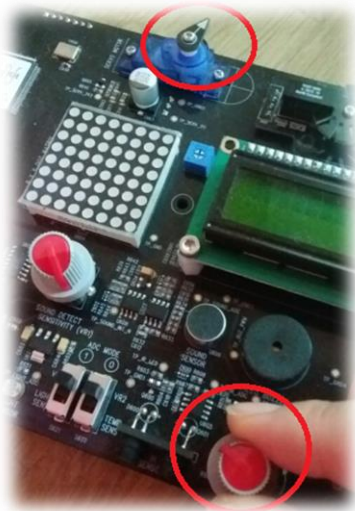



서  블록을 스크립트로 가져옵니다. 그리고, 위 ⑦에서 만들어진 블록을  블록 안에 넣습니다.



⑨ 이제  버튼을 눌러 실행시켜 보세요. 가변 저항을 돌리면, 서보 모터도 돌아가는 것을 확인하실 수 있습니다.





 을 눌러 실행을 멈추어도 서보 모터가 약간씩 계속 움직이는 경우가 있습니다. 이런 움직임은 다음과 같은 스크립트를 실행시켜 없앨 수 있습니다.



## 🔗 생각하기 9

위 스크립트에서는 서보모터가 가변 저항의 방향과는 반대로 움직일 것입니다. 서보 모터의 방향과 가변 저항의 방향을 같게 하려면 어떻게 해야 할까요?

## 12. 인공지능 전등 만들기

곰곰이와 아빠는 저녁에 마을 길을 산책하고 있었어요. 날이 어둡해져서 가로등불들이 하나, 둘 켜지기 시작했습니다.



어? 가로등불이 자동으로 켜지네요?



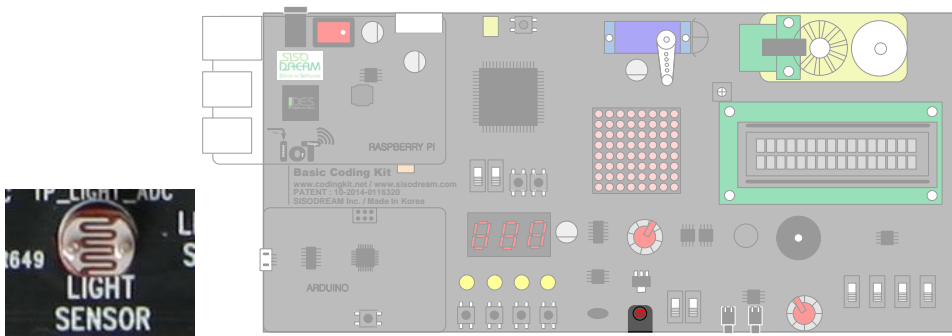
응, 날이 어두워지면 저절로 가로등불이 켜지도록 만든 거야. 매번 사람이 가서 가로등을 켜면 얼마나 불편하겠니? 그래서 밝기 센서를 이용해서 날이 어두워지면 자동으로 가로등에 불이 켜지고 날이 밝아지면 다시 꺼지게 만든 것이란다.



신기해요! 아빠! 인공 지능 전등이네요!

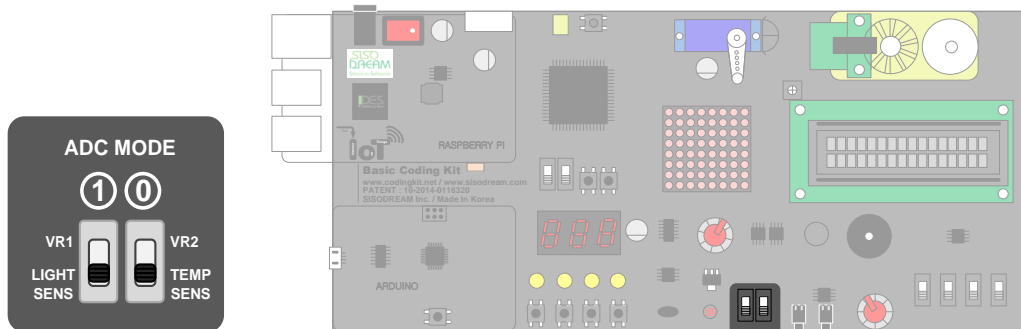


허허허! 그렇구나! 코딩킷에도 밝기 센서가 있단다. 우리도 집에 가서 인공 지능 전등을 만들어 보면 되겠구나.

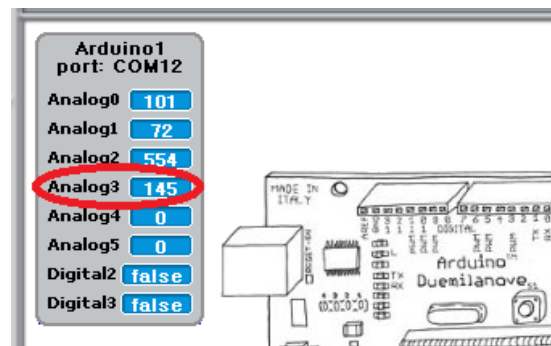
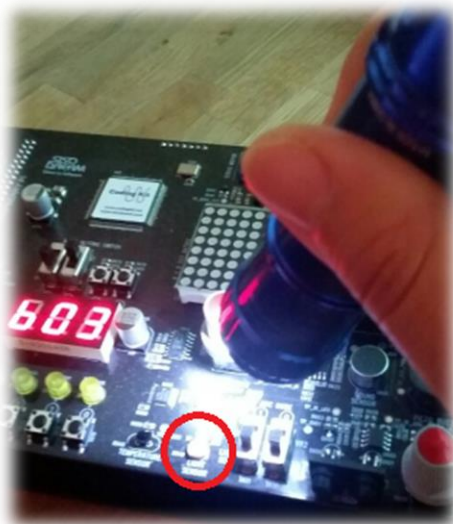


## 12-1.신기한 밝기 센서

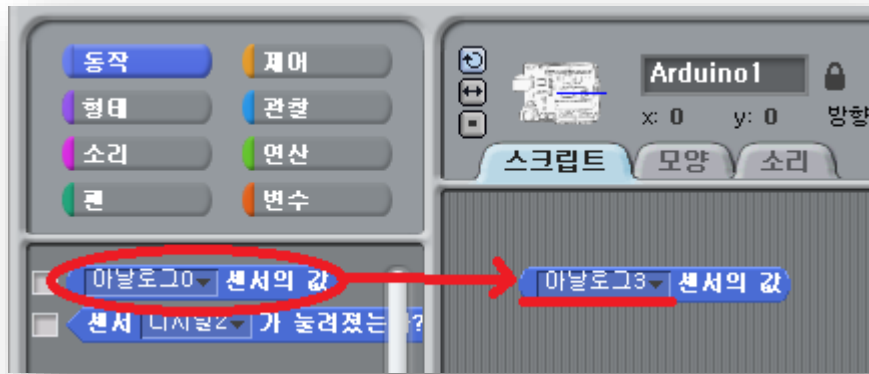
코딩킷의 밝기 센서를 사용해서 현재 실내의 밝기 정도를 알아볼까요? 밝기 센서를 이용하기 위해서 코딩킷의 ADC 모드의 ①번 스위치를 아래로 내려야 합니다.



밝기 센서는 ADC 모드 스위치의 왼쪽에 있습니다. 이 센서를 손가락으로 가리거나, 후레쉬를 사용하여 밝게 빛을 비춰보세요. 그러면, 무대 화면에 있는 상태 표시 정보의 "Analog3"의 값이 변하는 것을 보실 수 있습니다. 이 값이 밝기 센서에서 전달해 주는 값입니다. (코딩킷의 ADC 모드의 ①번 스위치가 아래로 내려가 있으면, Analog3은 가변 저항 값 대신 밝기 센서의 값을 전달해줍니다.)



밝기 센서의 값을 사용하려면 어떤 블록을 사용해야 할지 감이 오시나요? 눈치 채셨을 것 같네요. **등적** 블록 모음에서 **아날로그0 센서의 값** 을 가져와 사용하면 됩니다. 아래 화살표를 눌러서 "아날로그0" 대신에 "아날로그3"을 선택하세요.



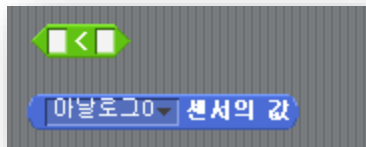
## 12-2. 인공지능 전등을 만들자

어두워지면 LED 등이 켜지고 밝아지면 LED 등이 꺼지는 전등을 만들어 보려고 합니다.

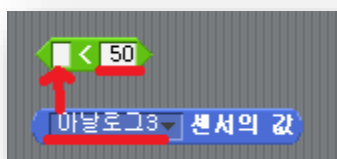
- ① **제어** 블록 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.
- ② **등적** 블록 모음에서 **디지털 13의 출력 꺼짐** 과 **디지털 13번 출력 켜짐** 블록을 스크립트로 가져옵니다.



- ③ **연산** 블록 모음에서 **◀<** 블록을, **등적** 블록 모음에서 **아날로그0 센서의 값** 블록을 스크립트로 가져옵니다.



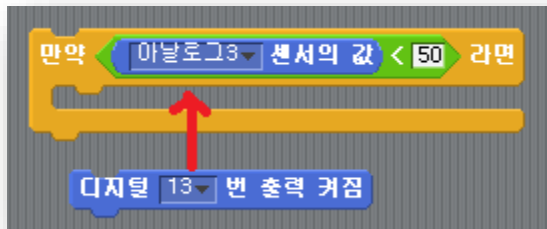
- ④ **아날로그0 센서의 값** 블록의 아래 화살표를 눌러서 "아날로그3"을 선택한 후, **◀<** 블록의 첫 번째 칸에 넣어 줍니다. 그리고, **◀<**의 두 번째 칸에는 숫자 "50"을 입력합니다.



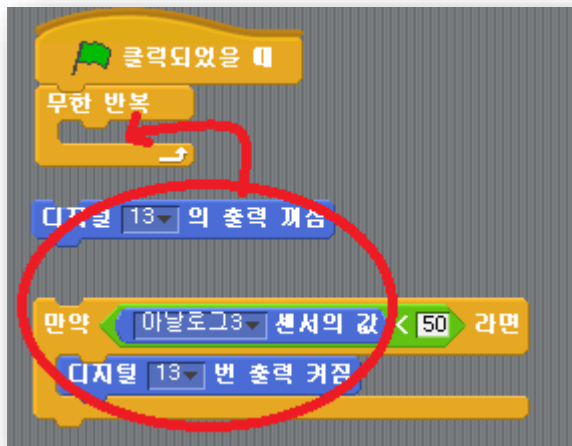
- ⑤ **제어** 블록 모음에서 **만약...그러면** 블록을 스크립트로 가져옵니다. **만약...그러면** 블록의 첫 번째 입력 칸에 위 ④에서 만든 블록을 넣어 줍니다.




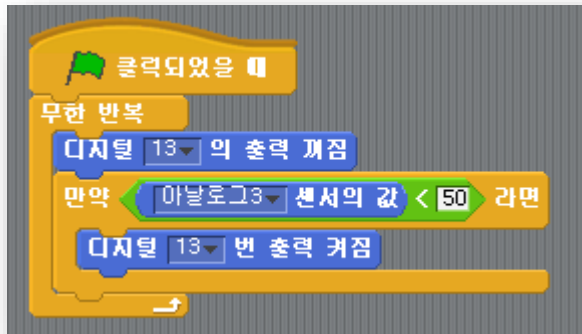
- ⑥ “아날로그3 센서의 값” 즉, 밝기 센서의 값이 50 보다 작을 경우에 LED가 켜지도록 합니다. 위 ⑤의 블록에 **디지털 13번 출력 켜짐** 블록을 삽입합니다.



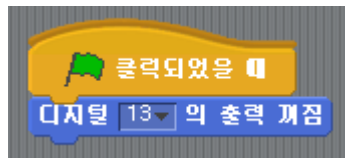
- ⑦ 이제 위에서 만든 블록들을 모두 **무한 반복** 블록 안에 넣어 줍니다.



- ⑧ 다음과 같이 블록이 완성 되셨나요?  버튼을 눌러 실행시켜 보세요. 밝기 센서를 손으로 가려서 어둡게 만들어 보세요. LED 불이 켜지나요?



한 번 켜진 LED 는 안 꺼지죠? 이것을 끄려면 다음과 같은 스크립트를 실행시키세요.



## 🚩 생각하기 10

어두운 정도에 따라 LED 조명이 켜지는 개수를 조절해 볼 수도 있습니다. 위에서 만든 스크립트를 응용해서 만들어 보세요.



## 13. 잠자는 고양이 깨우기



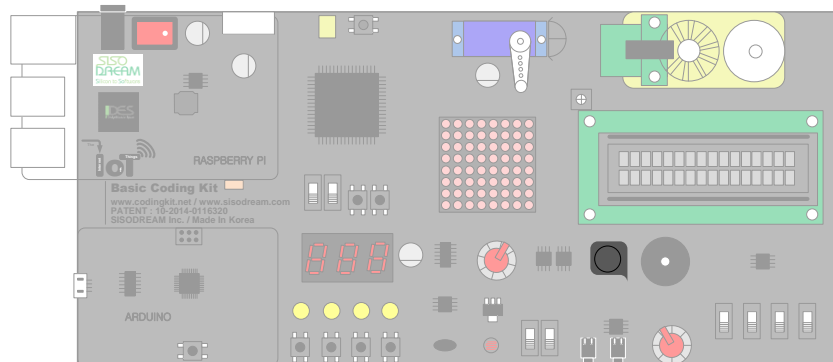
코딩킷에는 소리 센서도 있단다. 소리 값을 수치로 알려주는 거지. 소리 센서로 소음 측정기를 만들 수 있단다.



소음 측정기를 만들어서, 목소리 큰 사람 뽑기 대회에서 사용하면 좋겠어요!

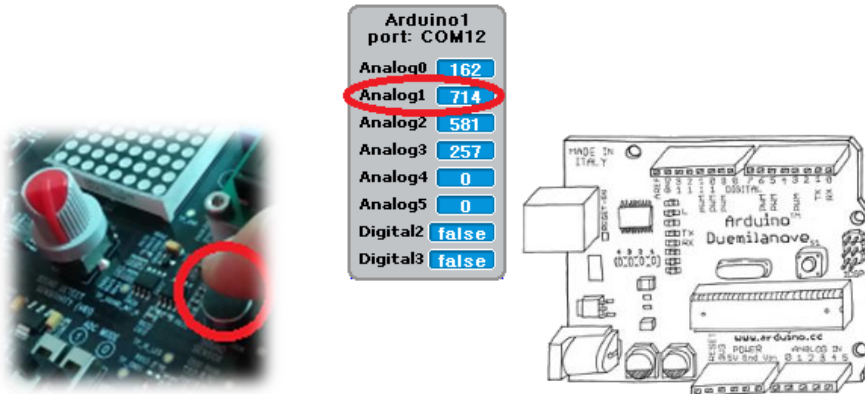


허허허! 재미있는 생각이구나!

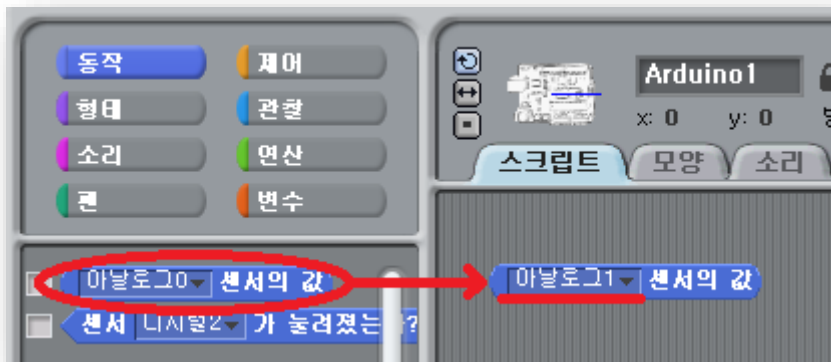


### 13-1.셋 조용히.. 소리 센서

소리 센서는 마이크와 같은 것입니다. 외부의 소리가 크면 클수록 소리 센서의 값은 올라 갑니다. 소리 센서는 가변 저항(VR1)의 오른쪽 옆에 있습니다. 이 센서에 가까이 대고 크게 소리를 내거나, 손가락으로 센서를 툭툭 쳐보세요. 그러면, 무대 화면에 있는 상태 표시 정보의 "Analog1"의 값이 변하는 것을 보실 수 있습니다. 이 값이 소리 센서에서 전달해 주는 값입니다.



소리 센서의 값을 사용하려면 **동작** 블록 모음에서 **아날로그0** 센서의 값 을 가져와 사용하면 됩니다. 아래 화살표를 눌러서 "아날로그0" 대신에 "아날로그1"을 선택하세요.



### 🐞 설명 더하기

소리 센서의 값은 순간적으로 높아졌다가 낮아집니다. 그래서 손으로 소리 센서를 툭툭

건드렸을 때 매번 소리 센서의 값이 높아지지는 않습니다. 그래서 일반적으로 소리 센서의 값을 여러 번 읽어 와서 그 값들의 평균을 내어 사용합니다. 혹은 여러 번 읽어서 최대 값을 사용하기도 합니다. 좀 더 자세한 사항은 코딩북의 아두이노 파트를 참고해 주세요.

## 생각하기 11

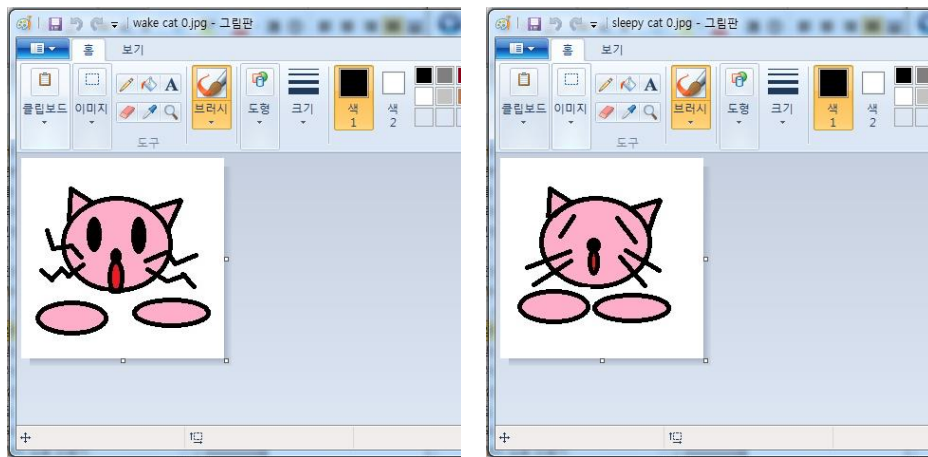
소리 센서를 사용해서 소리가 커지면 커지는 LED등의 수가 늘어나게 되는 소음 측정기를 만들어 보세요.

## 13-2. 잠자는 고양이를 깨우기 놀이를 해보자

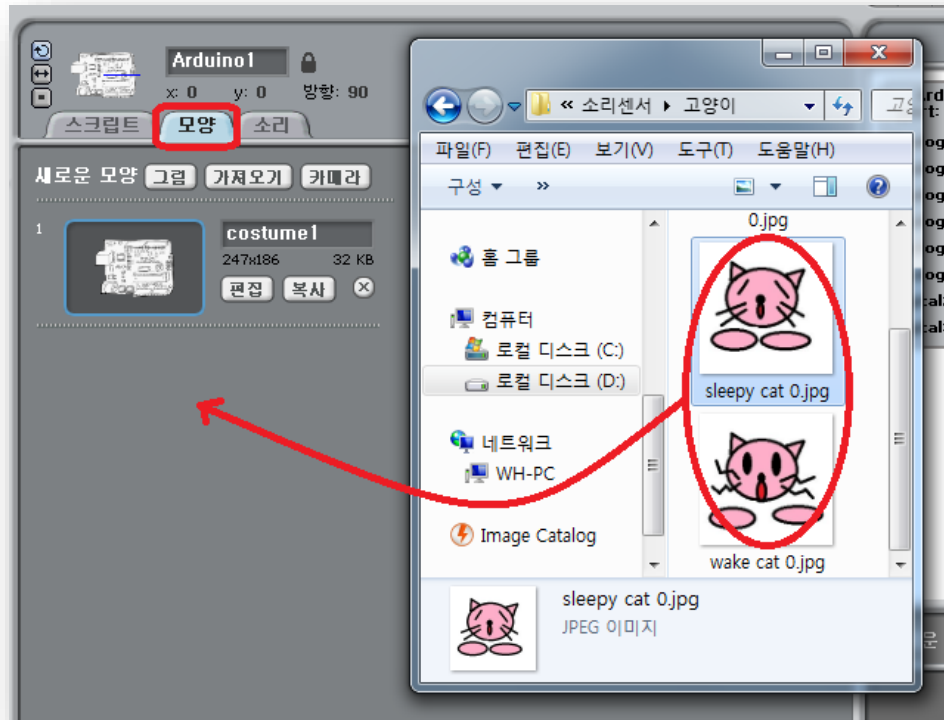
잠자는 고양이를 소리를 질러 깨우는 게임을 만들어 보려고 합니다.

### 🌀 1 단계 : 그림 삽입하기

- ① 그림 툴을 이용하여, 잠자는 고양이 그림과 깨어 있는 고양이 그림을 그립니다. 저는 윈도우에서 기본으로 제공되는 그림판 프로그램을 사용하였습니다. 각각 "wake cat 0.jpg"와 "sleepy cat 0.jpg"로 저장합니다. "...WS4AW그림\고양이" 폴더에서 가져오기 해도 됩니다.



- ② "모양" 탭을 클릭합니다. 탐색기에서 "wake cat 0.jpg"와 "sleepy cat 0.jpg" 파일을 찾아, "모양" 탭 안으로 마우스로 끌어다 놓습니다.



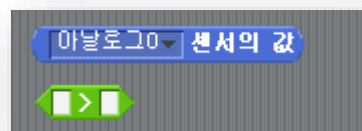
- ③ 아래 그림과 같이 “모양” 탭에 그림이 삽입됩니다.





## 2 단계 : 소리 센서의 값에 따라 그림 바꾸기

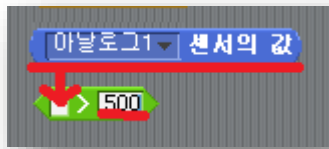
① "스크립트" 탭을 선택합니다. **제어** 블록 모음에서 **클릭되었을 때** 블록과 **무한 반복** 블록을 스크립트로 가져옵니다.

② **연산** 블록 모음에서 **▶▶▶** 블록을, **등적** 블록 모음에서 **아날로그 ▶ 센서의 값** 블록을 스크립트로 가져옵니다.




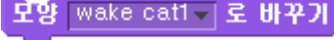
③ **아날로그 ▶ 센서의 값** 블록의 아래 화살표를 눌러서 "아날로그1"을 선택한 후,

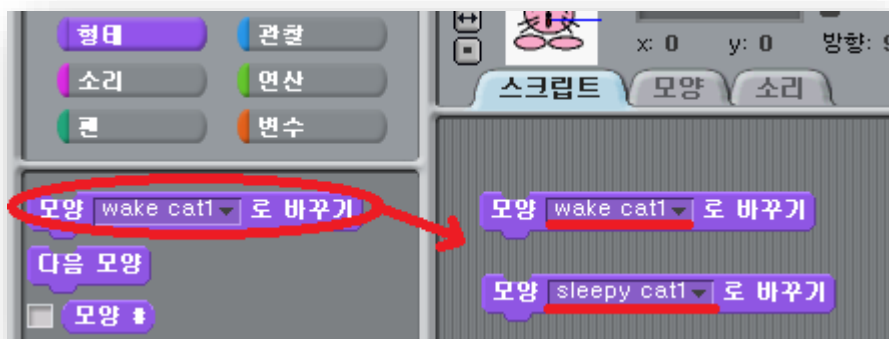
 블록의 첫 번째 칸에 넣어 줍니다. 그리고,  의 두 번째 칸에는 숫자 "500"을 입력합니다.



④  블록 모음에서  블록을 가져옵니다. 첫 번째 입력 칸에 위 ③의 블록을 삽입합니다.

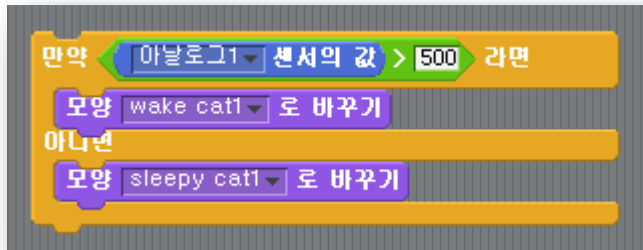


⑤  블록 모음에서  블록 두 개를 가져옵니다. 아래 화살표를 눌러서 각각 "wake cat1"과 "sleepy cat1" 모양을 선택합니다.

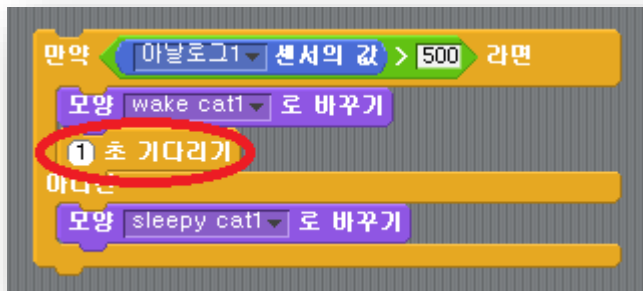


⑥ 소리 센서의 값이 500 보다 크면 깨어있는 고양이 그림, 500 보다 작으면 잠자는 고양이 그림

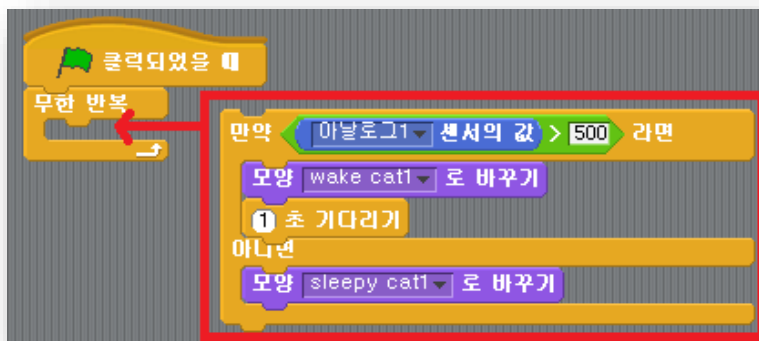
양이 그림이 나오도록 위 ④의 블록의 해당 칸에 위 ⑤의 블록을 다음 그림과 같이 넣어 줍니다.




⑦ 제어 블록의 1 초 기다리기 블록을 사용하여 깨어 있는 고양이 그림이 나오는 경우 1초 동안 기다리도록 합니다.

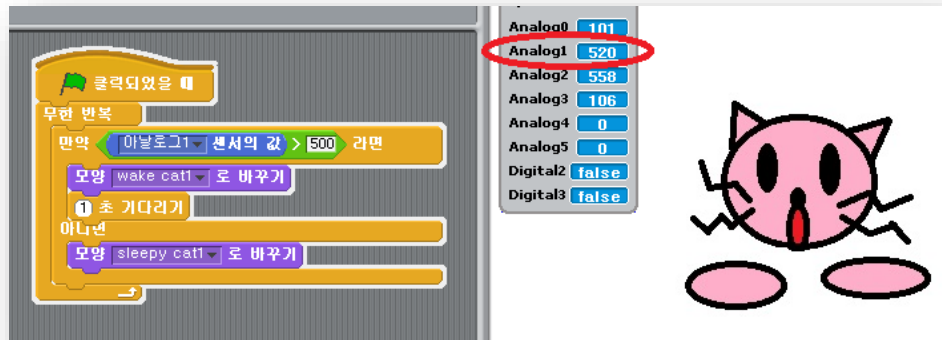


⑧ 이제 무한 반복 블록 안으로 위 ⑦의 블록을 다음 그림과 같이 넣어 줍니다.





- ⑨ 이제  버튼을 눌러 실행시켜 보세요. 무대 화면에 잠자는 고양이 그림이 나오죠? 귀가 아플 정도로 크게 소리를 질러 보세요. 깨어 있는 고양이 그림으로 바뀌는 것을 확인하실 수 있습니다. 그리고 다시 잠자는 고양이 그림으로 바뀝니다. 귀가 아플 정도의 소리가 계속 들리지 않는다면요. ^^



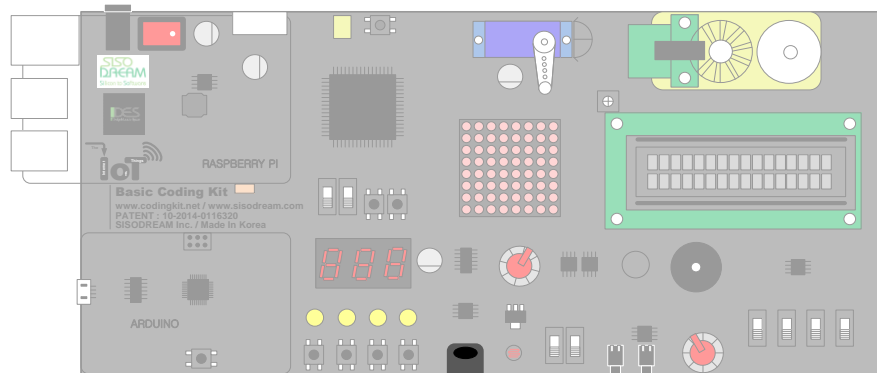
## 14. 똑똑한 우리집 냉난방 시스템



아~ 너무 더워요! 아빠! 이럴 땐 자동으로 에어컨이 켜졌으면 좋겠어요.

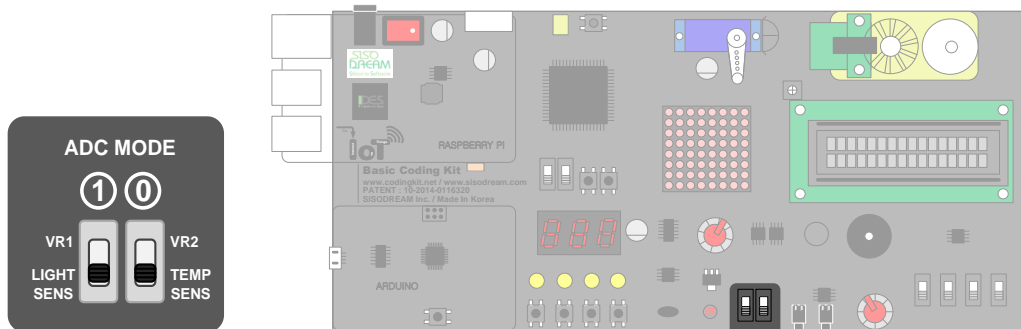


자동으로 더우면 에어컨이 켜지고 추우면 난방이 된다? 좋은 생각이구나! 그리고 보니 코딩킷에 온도 센서가 있단다. 온도 센서를 이용하면 가능할 것 같구나.

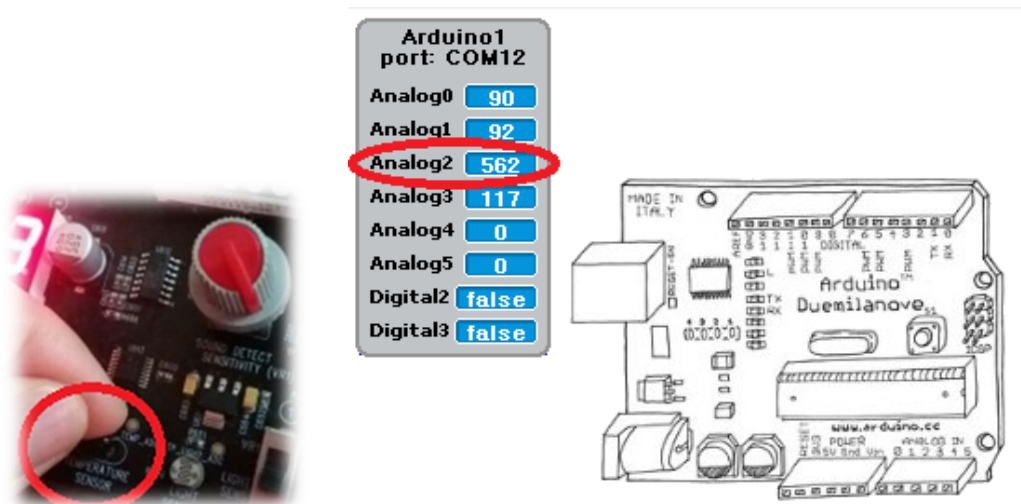


## 14-1. 온도를 알려드려요. 온도 센서.

코딩킷의 온도 센서를 이용하기 위해서는 코딩킷의 ADC 모드의 ①번 스위치를 아래로 내려야 합니다.




온도 센서는 밝기 센서의 왼쪽에 있습니다. 이 센서를 엄지와 집게 손가락으로 잡아서 따뜻하게 만들어 주세요. 그러면, 무대 화면에 있는 상태 표시 정보의 "Analog2"의 값이 변하는 것을 보실 수 있습니다. 이 값이 온도 센서에서 전달해 주는 값입니다. (코딩킷의 ADC 모드의 ①번 스위치가 아래로 내려가 있으면, Analog2는 가변 저항 값 대신 온도 센서의 값을 전달해줍니다.)

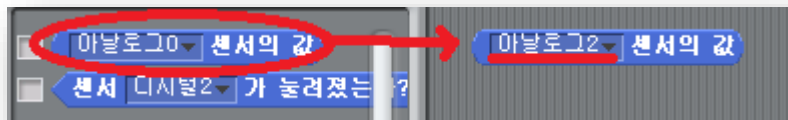





## 14-2.더울 때는 선풍기를, 추울 때는 난로를 켜자

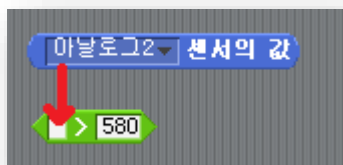
온도 센서가 알려주는 온도 값을 이용하여 더울 때는 선풍기가 돌아가고 추울 때는 난로가 켜지는 동작을 만들어 볼까요? 선풍기 대신 모터를 사용하고, 난로 대신 LED 등을 사용하여 만들어 보겠습니다. (5장 LED와 10장 모터에 대한 설명을 참조하세요)

- ① **제어** 블록 모음에서  블록과  블록을 스크립트로 가져옵니다.

- ② **동작** 블록 모음에서  블록을 가져옵니다. 아래 화살표를 눌러서 "아날로그2"를 선택합니다.



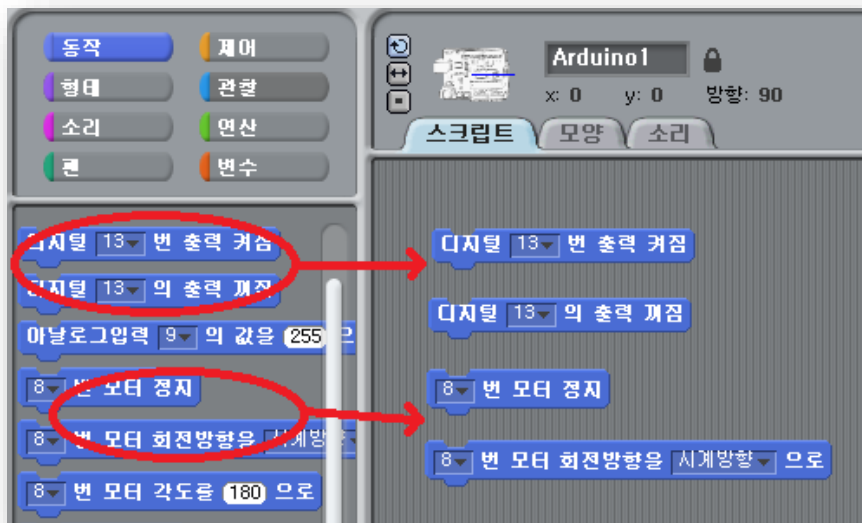
- ③ **연산** 블록 모음에서  블록을 스크립트로 가져옵니다.  블록의 첫 번째 입력 칸에는  블록을 넣고, 두 번째 입력 칸에는 580을 입력합니다. (이 수치는 적절하게 조정하여 주세요.)



- ④ **제어** 블록 모음에서  블록을 가져와서 위 ①의 블록을 첫 번째 입력 칸에 넣어 줍니다.




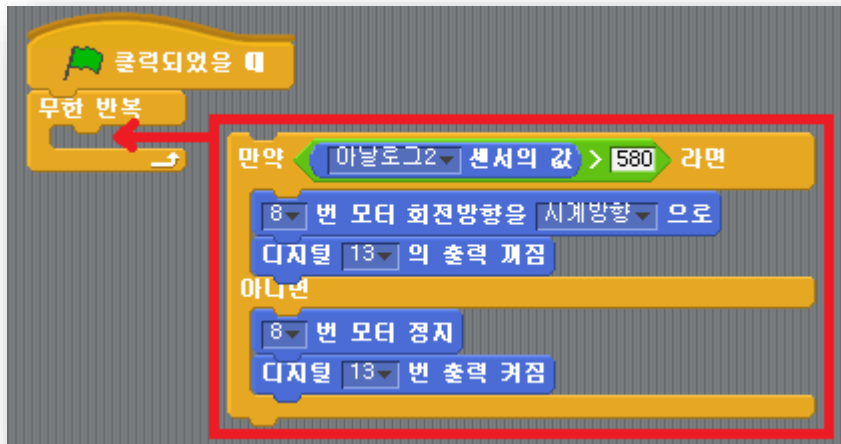
- ⑤ **등적** 블록 모음에서 **디지털 13 번 출력 켜짐** 과 **디지털 13 번의 출력 꺼짐** 블록을 스크립트로 가져옵니다. 이 블록들은 난로가 켜지고 꺼짐을 나타내기 위해 사용할 것입니다. **8 번 모터 회전방향을 시계방향** 으로 블록과 **8 번 모터 정지** 블록도 스크립트로 가져옵니다. 이 블록들은 선풍기가 켜지고 꺼지는 것을 나타내기 위해 사용할 것입니다.




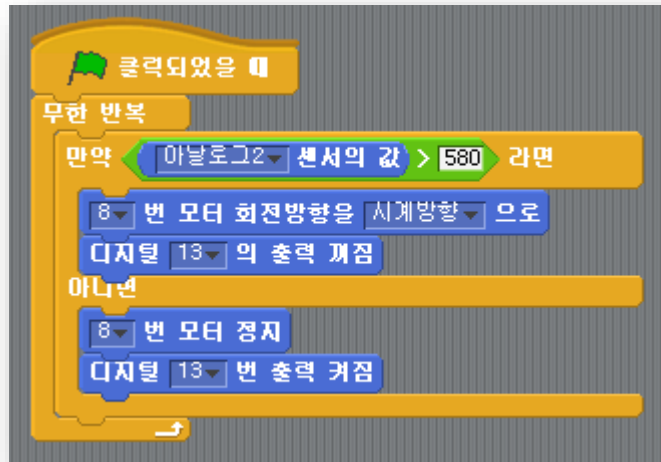
- ⑥ 온도 센서가 580보다 커지면 LED는 꺼지고, 모터가 돌아가도록 합니다. 온도가 올라가면 난로는 꺼지고 선풍기가 돌아가겠지요? 그리고, 온도 센서 580보다 작으면 모터는 정지하고, LED는 켜지도록 합니다. 온도가 내려가서 센서 값이 580 보다 작으면 선풍기는 꺼지고 난로가 켜지게 됩니다.



⑦ 온도를 계속 체크해야 하므로, 위 ⑥의 블록을  블록 안에 넣습니다.



⑧ 이제  버튼을 눌러 실행시켜 보세요. LED 등에 불이 들어왔나요? 온도 센서를 손으로 쥐어서 따뜻하게 해보세요. 온도 센서의 값이 580 이상으로 올라가면, LED 등이 꺼지고 모터가 돌아가는 것을 확인 하실 수 있습니다. 만약에 처음부터 LED가 꺼지고 모터가 돌아가고 있다면, 실내 온도가 높아서 그런 것 입니다. 이런 경우에는 동작을 시험하기 위해서 온도의 비교 값을 580보다 높게 설정하여 동작을 시켜 보세요.



## 15. 마법의 도깨비 집

곰곰이는 아빠와 TV를 보고 있습니다. TV에서는 도깨비가 불쑥 튀어나와서 개그맨들이 놀라서 도망가는 프로그램을 하고 있습니다.



크크크! 저 아저씨 정말 깜짝 놀랐나 봐요! 저렇게 갑자기 인형이 튀어 나오면 정말 무섭겠어요!



허허허! 정말 놀란 표정이구나! 지난번에 놀이 동산에 갔을 때, 곰곰이도 도깨비집에 들어갔다가 무서워서 혼났었지?



네, 정말 무서웠어요. 그런데 어떻게 사람들이 다가오는 줄 알고 인형들이 튀어 나오는 거예요? 정말 도깨비집에 귀신이 사는 건 아닐까요?

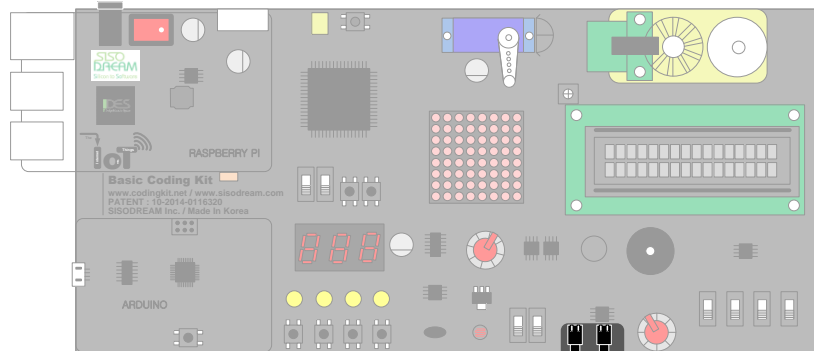


허허허! 실제로 도깨비집에 귀신이 사는 건 아니란다. 사람이 다가오는 것을 알아내는 센서가 있는데, 이 센서를 이용하여 사람이 다가올 때 인형들이 튀어 나오게 만드는 것이란다. 코딩키트에는 적외선 센서가 있는데, 이것을 이용하면, 사람이 다가오는 것을 알 수 있단다. 그럼 코딩키트의 적외선 센서를 이용하여 도깨비 상자를 만들어 볼까?

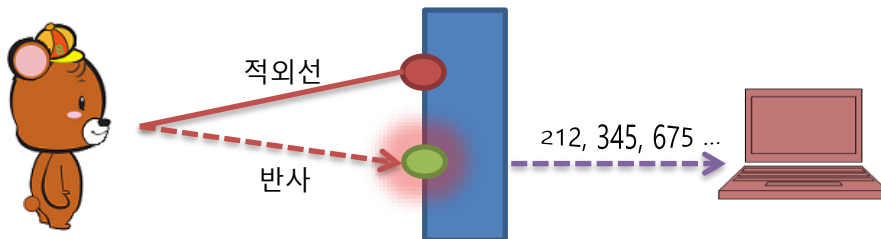


## 15-1. 꼼짝마! 적외선 센서

코딩킷에는 적외선(Infrared Rays) 센서가 있습니다. 적외선 센서를 이용하면 주변에 물체가 있는지를 알 수 있는데, 물체가 가까이 있으면 있을수록 적외선 센서의 전기 신호 값이 커지게 됩니다.



적외선 센서는 발광부와 수광부가 있습니다. 발광부는 적외선을 발사하고, 수광부는 발사된 적외선이 물체에 맞고 반사되어 오는 신호를 감지합니다.



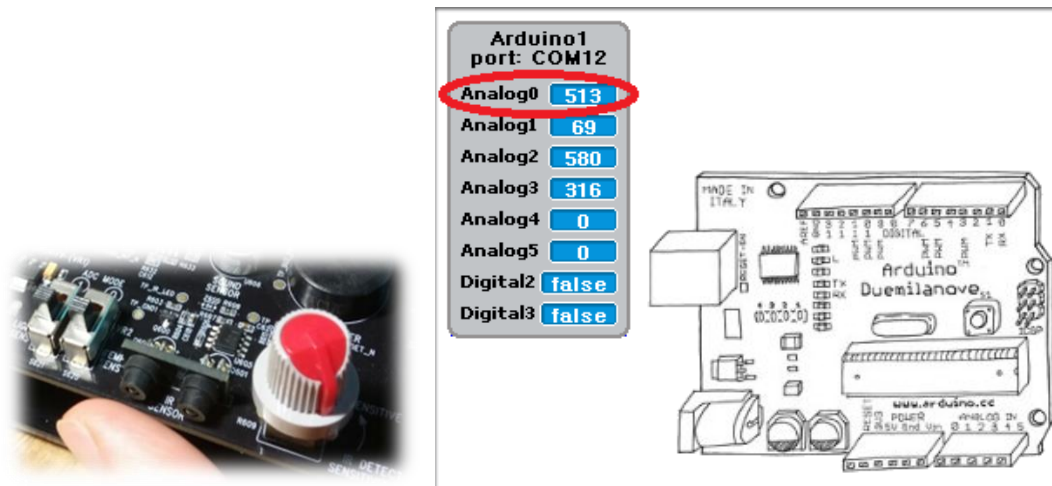
적외선 센서를 사용하려면, 7-SEGMENT 를 b04로 변경해야 합니다. ⊕ 또는 ⊖ 스위치 버튼을 눌러서 7-SEGMENT 를 b04로 변경한 후에, 2~3초간 눌러주세요.



우선 적외선 센서를 사용하기 위해서는 적외선 LED를 On 시켜주어야 합니다. **모판** 블록 모음에서 **디지털 13번 출력 켜짐** 블록을 스크립트로 가져옵니다. 스크립트로 가져온 **디지털 13번 출력 켜짐** 블록을 마우스로 한번 클릭해 줍니다. 그러면, 이 블록이 동작하게 됩니다.



이제 적외선 센서에 물체를 가까이 가져가 보세요. 그러면, 무대 화면에 있는 상태 표시 정보의 "Analog0"의 값이 점점 커지는 것을 보실 수 있습니다. 이 값이 적외선 센서에서 전달해 주는 값입니다.



적외선 센서에 가까이하는 물체를 흰색으로 하면 더 정확한 값이 측정됩니다. 적외선은 흰색에서 가장 잘 반사됩니다. 흰색 종이를 사용하면 좋습니다. 너무 가까이 가면 이 값이 다시 줄어듭니다.

이것은 적외선을 반사하는 각도가 잘 안 맞게 되어 값이 줄어드는 것입니다.

## 🔗 생각하기 12

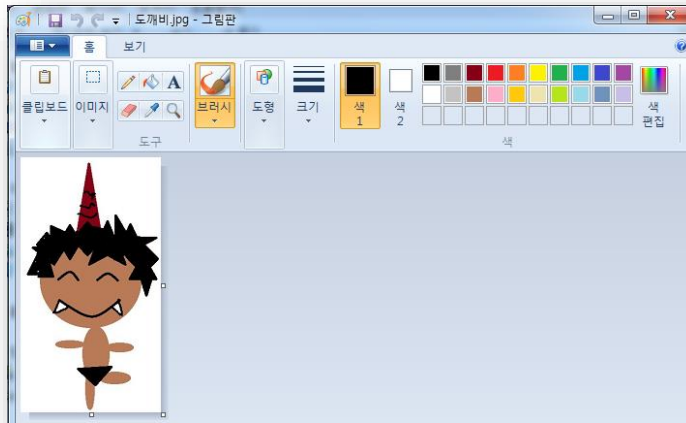
적외선 센서에 물체가 가까이가 다가가면 “삐~” 하는 부저가 울리도록 만들어 보세요.

## 15-2.마법의 도깨비 상자를 만들자

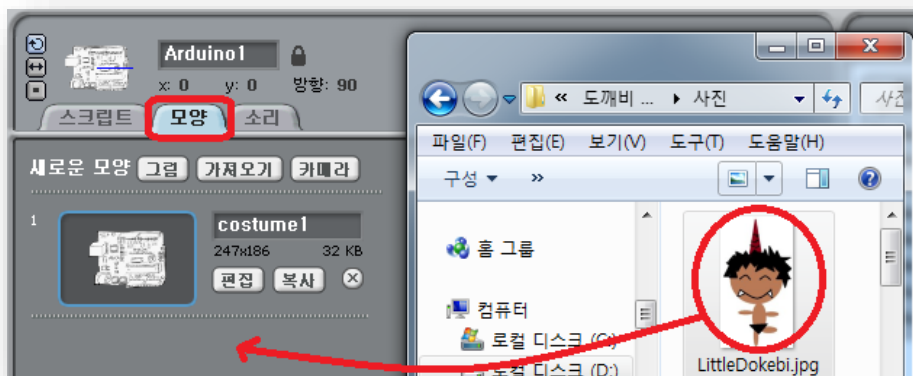
이번에는 적외선 센서에 물체가 가까이 다가오는 것이 감지되면 도깨비가 나타나는 무시무시한 마법의 상자를 만들어 보겠습니다.

### 1 단계 : 그림 삽입하기

- 우선, 그림 툴을 이용하여, 도깨비 그림을 그립니다. 저는 윈도우에서 기본으로 제공되는 그림판 프로그램을 사용하였습니다. 도깨비 그림을 그려서 "LittleDokebi.jpg"로 그림을 저장합니다. "...WS4A\그림\도깨비" 폴더에서 가져오기 해도 됩니다.



- S4A에서 "모양" 탭을 누릅니다. 탐색기를 열어서, 위 ①에서 저장한 "LittleDokebi.jpg" 그림을 선택하고, "모양" 탭 안에 끌어다 놓습니다.

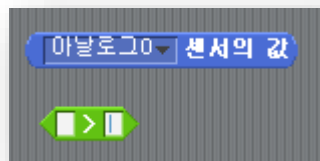


- ③ “모양” 탭에 “LittleDokebi” 그림이 추가 되었습니다.



## 2 단계 : 적외선 센서의 값에 따라 그림 바꾸기

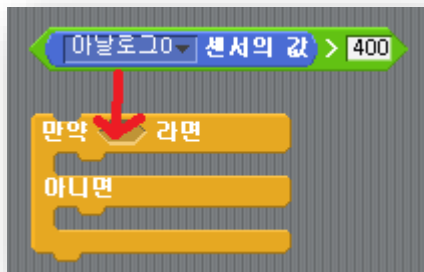
- ① **동작** 블록 모음에서 **아날로그 센서의 값** 블록을, **연산** 블록 모음에서 **▶▶** 블록을 가져옵니다.



- ② **▶▶** 블록의 첫 번째 입력 칸에 **아날로그 센서의 값** 블록을 넣고, 두 번째 입력 칸에는 숫자 “400”을 입력합니다. 적외선 센서의 값이 400보다 크면 물체가 가까이 온 것으로 감지하려고 합니다.



- ③ 제어 블록 모음에서 **제어** 블록을 가져와서 위 ②의 블록을 첫 번째 입력 칸에 넣어 줍니다.



- ④ 형태 블록 모음에서 **형태** 블록 모음에서 **모양 LittleDokebi로 바꾸기** 블록을 가져와 위 ③의



블록 안에 다음 그림과 같이 넣어 줍니다. “아날로그0 센서의 값”이 400보다 작을 경우, 모양을 “costume1” 으로 바꾸도록 변경합니다. 적외선 센서에서 물체가 멀어지면, 도깨비 그림 대신 원래의 화면을 보여주게 됩니다.



### 3 단계 : 소리 삽입하기

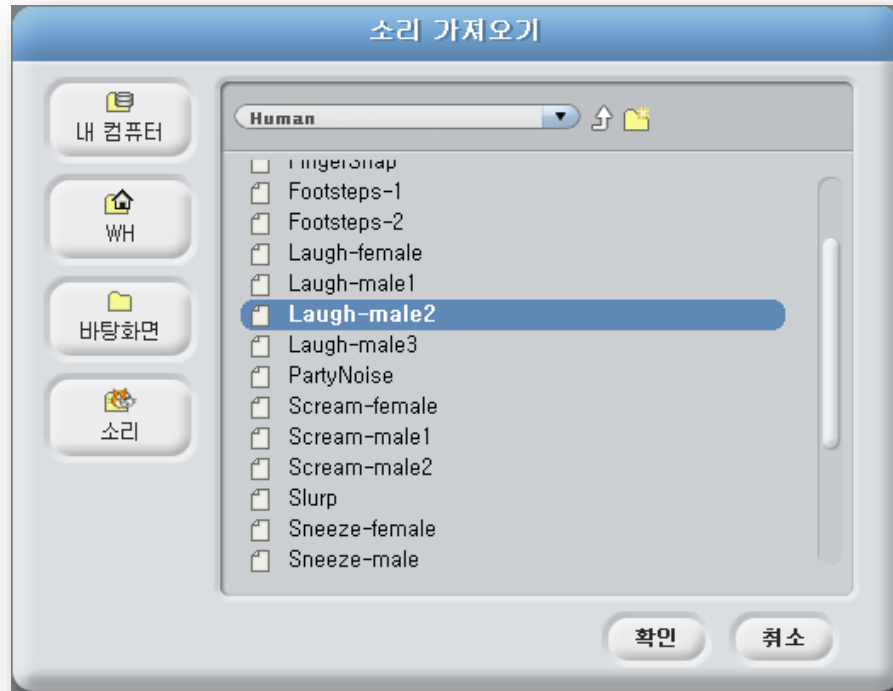
- ① 화면이 도깨비 모양으로 바뀔 경우, 으스스한 웃음소리도 나오게 하면 더 재미있겠지요? S4A에서 "소리" 탭을 누르고, "가져오기" 버튼을 누릅니다.



- ② 그러면, 다음 그림과 같이 소리 파일을 가져올 수 있습니다.



- ③ 저는 "Human" 폴더에 있는 "Laugh-male2" 소리를 선택하겠습니다. 선택하면 바로 소리가 나니 조심하세요. "확인" 버튼을 눌러 줍니다.



- ④ “소리” 탭에 “Laugh-male2” 소리가 삽입되었습니다. “재생” 버튼을 누르면 소리를 확인하실 수 있습니다.



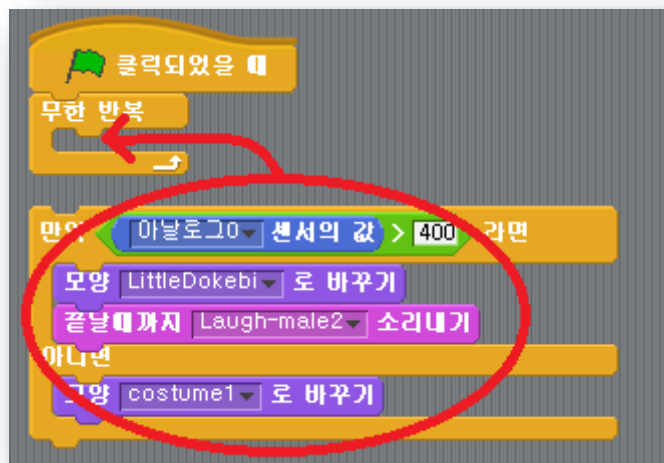
- ⑤ 이제, “스크립트” 탭을 선택합니다. **소리** 블록 모음에서 **끝날때까지 Laugh-male2 소리내기** 블록을 가져옵니다. 그리고, 무대 화면이 도깨비 모양으로 바뀔 때 소리가 나오도록 **모양 LittleDokebi 로 바꾸기** 블록 아래에 넣어



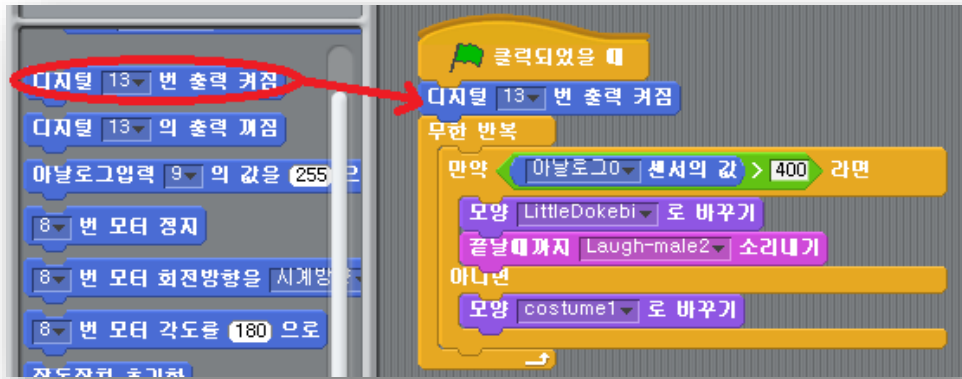
줍니다.




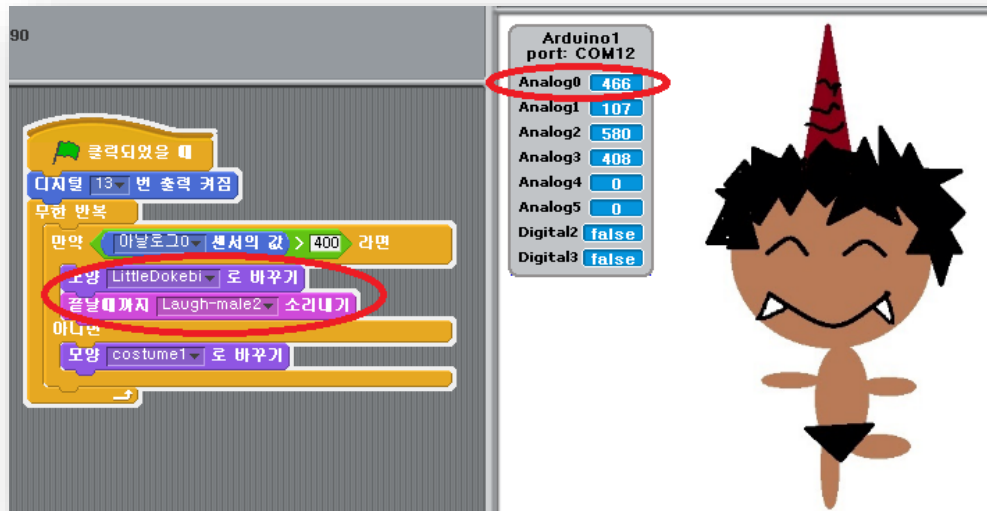
- ⑥ 제어 블록 모음에서 클릭되었을 블록과 무한 반복 블록을 스크립트로 가져옵니다. 그리고 위 ⑤의 블록을 무한 반복 블록 안에 넣어 줍니다.



- ⑦ 한가지 빼먹은 것이 있습니다. 적외선 LED를 On 시켜 주어야 적외선 센서가 제대로 동작을 합니다. 동작 블록 모음에서 디지털 13번 출력 커짐 블록을 스크립트로 가져와서, 클릭되었을 블록 아래에 넣어 줍니다.

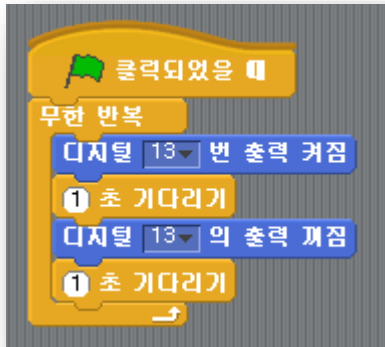


- ⑧ 이제  버튼을 눌러 실행시켜 보세요. 물체를 적외선 센서에 가까이 다가가게 해 보세요. 화면에 도깨비 그림이 나타나면서 웃음소리가 들릴 것입니다. 그리고, 물체를 적외선 센서에서 떨어뜨리면, 도깨비 그림이 사라질 것입니다.

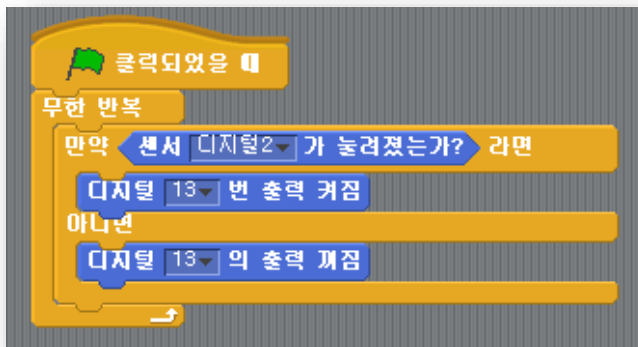


## 부록 A. 스피드 가이드

### 1. LED 깜박이기



### 2. 버튼이 눌리면 LED 켜지게 하기



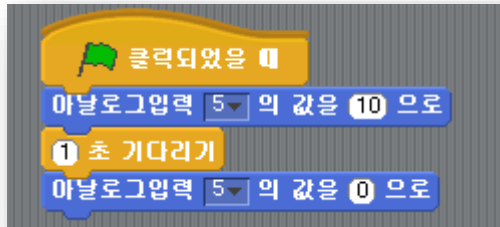
### 3. 초인종 만들기



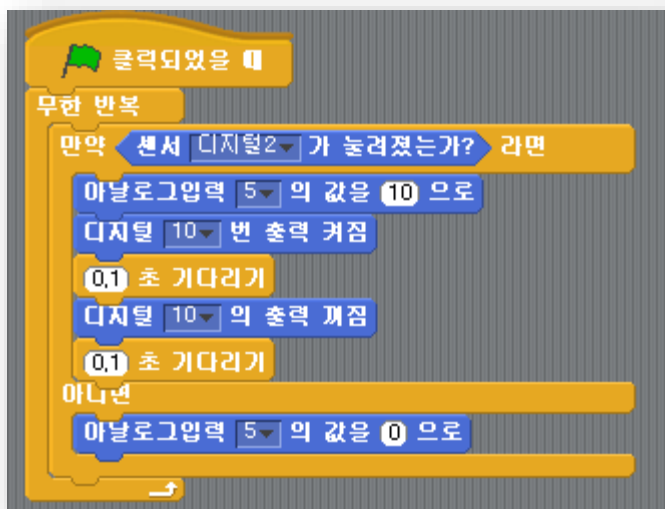
### 4. 크리스마스 전등 만들기



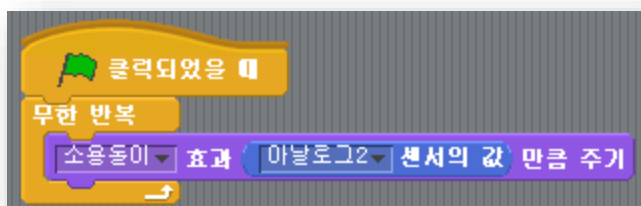
## 5. 부저 소리 내기



## 6. 경보기 만들기



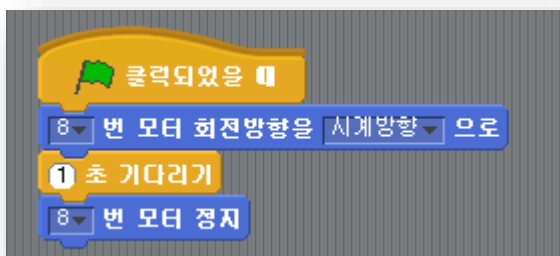
## 7. 가변 저항으로 화면 조절하기



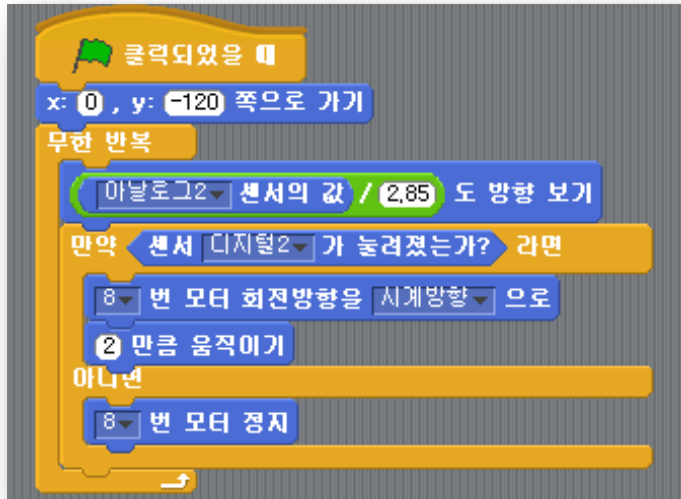
## 8. 가변 저항으로 음량 조절하기



## 9. 모터 움직이기



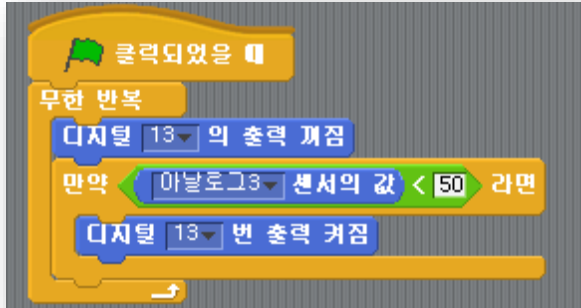
## 10. 자동차를 만들자



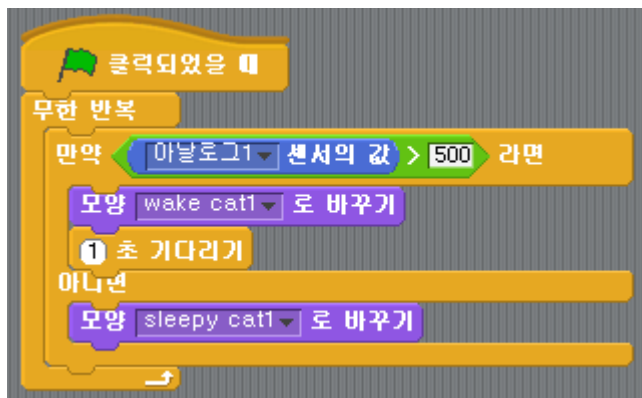
## 11. 서보 모터 움직이기



## 12. 인공지능 전등 만들기

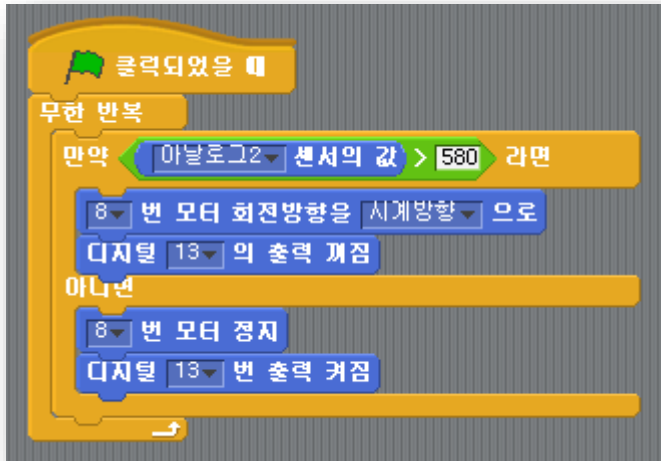


## 13. 잠자는 고양이 깨우기





## 14. 자동 온도 조절 시스템

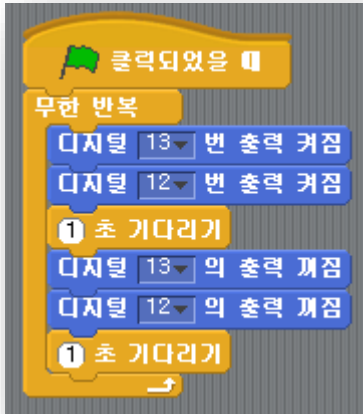


## 15. 도깨비 상자 만들기

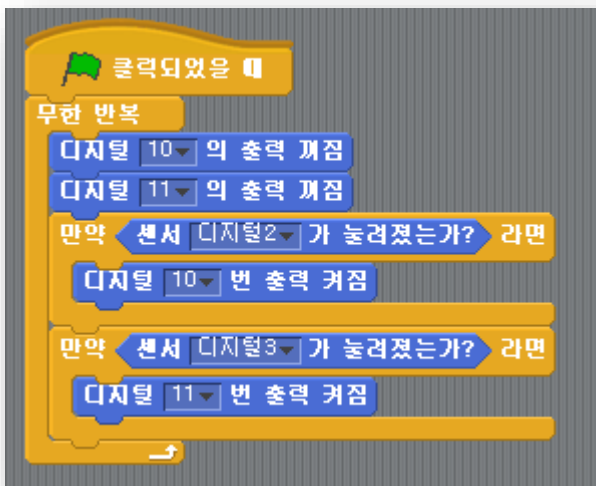


## 부록 B. 생각하기 해답

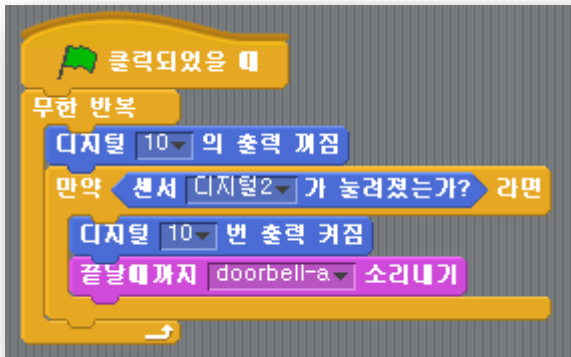
### 생각하기 1. LED 두 개를 깜박이기



### 생각하기 2. 버튼이 눌리면 해당 LED가 켜지게 하기



### 생각하기 3. LED 가 켜지는 초인종



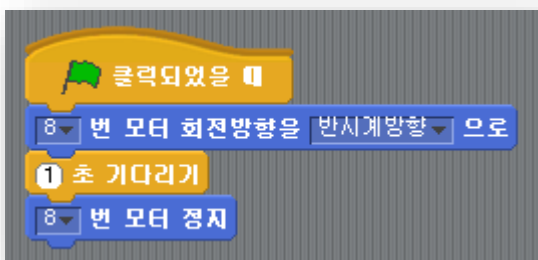
### 생각하기 4. 크리스마스 전등 2



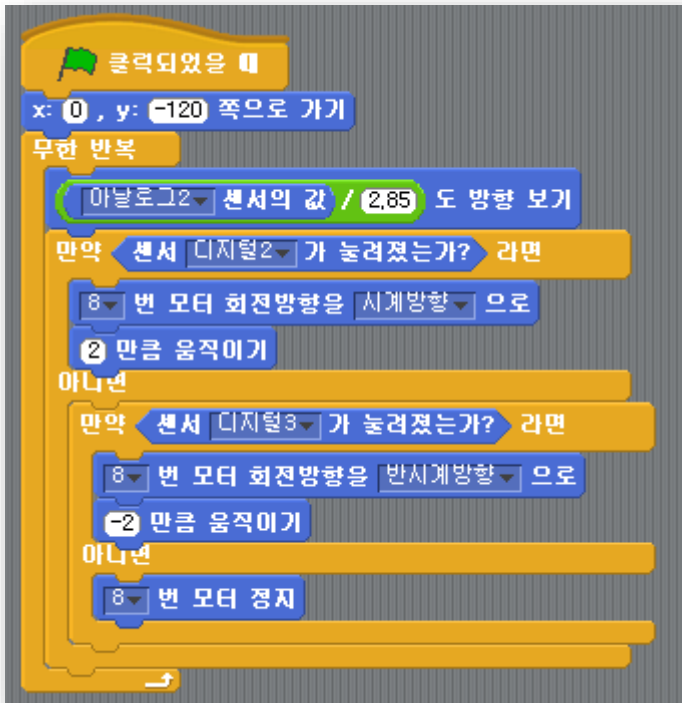
## 생각하기 5. 부저 소리 변화주기



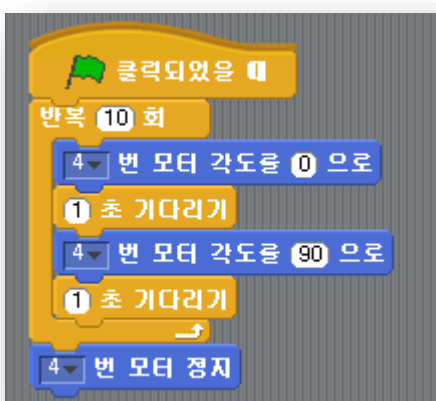
## 생각하기 6. 모터를 반시계 방향으로 돌리기



## 생각하기 7. 후진기능이 추가된 자동차



## 생각하기 8. 깃발을 흔드는 로봇



### 생각하기 9. 가변저항에 따라 움직이는 서보모터

```

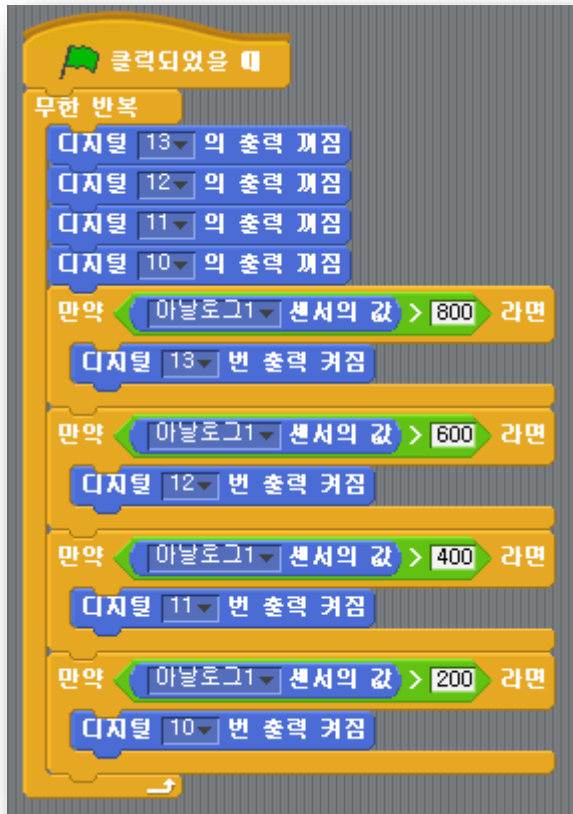
    클릭되었을 때
    무한 반복
    4번 모터 각도를 200 - (아날로그2 센서의 값 / 5) 반올림 으로
    0.1 초 기다리기
    4번 모터 정지
  
```

### 생각하기 10. 밝기에 따라 켜지는 조명 개수 조절하기

```

    클릭되었을 때
    무한 반복
    디지털 13의 출력 꺼짐
    디지털 12의 출력 꺼짐
    디지털 11의 출력 꺼짐
    디지털 10의 출력 꺼짐
    만약 (아날로그3 센서의 값 < 150) 라면
      디지털 13번 출력 켜짐
    만약 (아날로그3 센서의 값 < 75) 라면
      디지털 12번 출력 켜짐
    만약 (아날로그3 센서의 값 < 50) 라면
      디지털 11번 출력 켜짐
    만약 (아날로그3 센서의 값 < 25) 라면
      디지털 10번 출력 켜짐
  
```

## 생각하기 11. 소음 측정기



## 생각하기 12. 접근 감지 센서



## 부록 C. 프로그램 설치 및 사용 가이드

### 1. S4A(Scratch for Arduino) 소개 및 설치

S4A는 스크래치(Scratch)에서 오픈 소스 하드웨어 플랫폼인 아두이노에 프로그래밍이 가능하도록 만든 프로그램입니다. S4A는 아두이노와 연결된 여러 센서들과 제어 시스템을 관리할 수 있도록 새로운 블록을 제공합니다. S4A는 프로그래밍 세계로 사람들을 끌어들이고, 아두이노 프로그래머들에게는 보드의 다양한 이벤트를 관리할 수 있는 기능을 가지는 상위 레벨의 인터페이스를 제공하기 위한 목적으로 만들어졌습니다.


S4A는 Snap4Arduino와 유사한 프로그램으로 <http://s4a.cat/>에서 무료로 다운 받을 수 있습니다. S4A 프로그램을 다운 받기 위해서, Downloads 메뉴를 클릭하세요.





다음과 같이 설치 과정에 대한 간단한 설명과 함께 사용자의 환경에 맞는 S4A 프로그램을 다운 받을 수 있도록 링크가 되어 있는 것을 보실 수 있습니다.

## Download and Install



1 Download and install **S4A**

2 Install our firmware into your **Arduino**

3 Enjoy programming your **world!**

Installing S4A requires you to install software both in your PC and your Arduino board. Here you'll find the detailed steps to get it up and running.

### Installing S4A into your computer

S4A works in the three major consumer operating systems. Download and install the one that fits your configuration:

- Windows
- Mac
- Linux (Debian)
- Linux (Fedora) (version 1.5)
- Raspbian (Debian for RaspberryPi) (version 1.5)

설치 과정을 요약하면 다음의 3 단계로 설명이 됩니다.

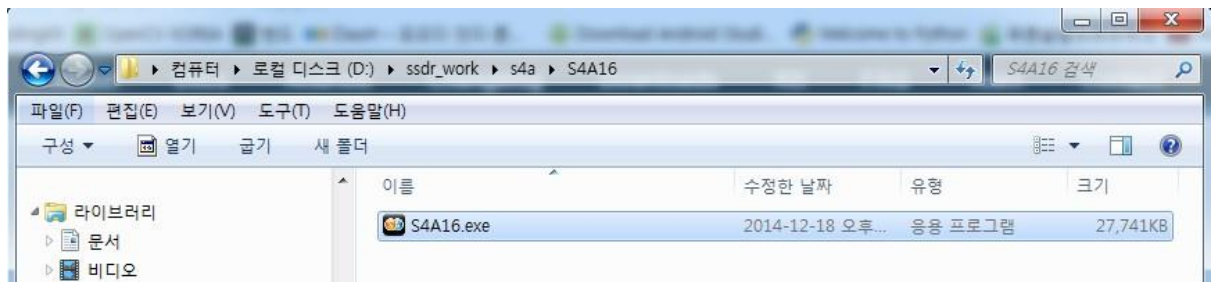
- ① S4A를 다운로드 받아서 설치를 합니다.
- ② 아두이노 보드로 펌웨어를 업로드합니다.
- ③ S4A로 프로그래밍을 즐기세요.

첫 번째 단계로, 사용하시는 컴퓨터 환경(운영체제)에 맞는 버전을 선택하여 다운로드를 받아 설치하세요. 저는 윈도우 버전을 다운받아서 설치하도록 하겠습니다.

윈도우 버전의 S4A를 선택하면, 다음과 같이 파일을 저장하라는 메시지가 뜹니다. 원하는 장소에 파일 다운 받아서 압축을 풀어주세요.



다운받은 S4A16.exe 파일을 실행시켜 설치하여 주세요.



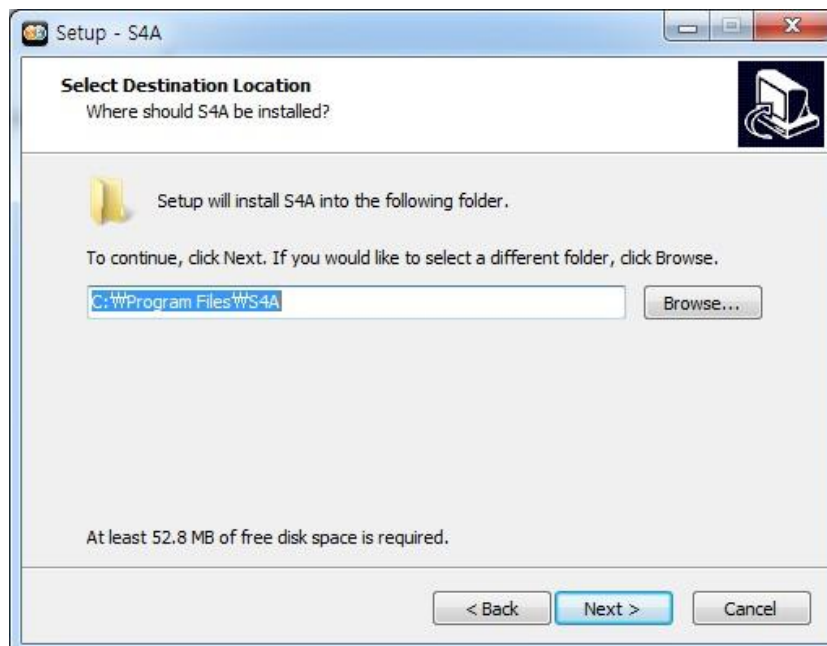
S4A16.exe를 실행시키면, 다음과 같이 설치 시작 화면이 나옵니다. 설치를 원하시면, "Next"버튼을 눌러주세요.



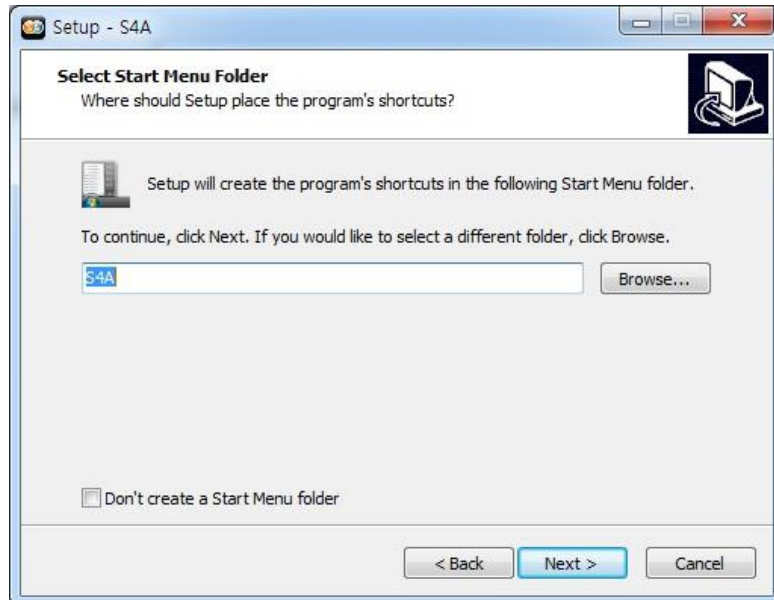
라이선스에 동의하는지 묻는 창이 나타납니다. 동의하시면, "I accept the agreement"를 선택하고, "Next" 버튼을 누릅니다.



S4A를 설치할 경로를 선택합니다. "Browse.." 버튼을 누르면, 원하는 설치 경로를 선택하실 수 있습니다. 선택하셨으면, "Next" 버튼을 누릅니다.



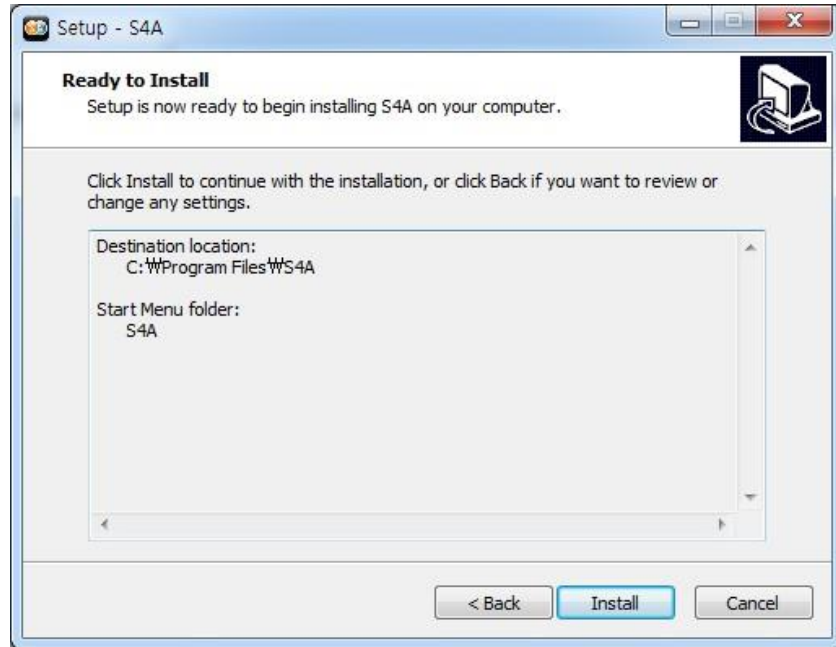
S4A 프로그램의 바로가기 생성할 시작 메뉴 폴더를 선택합니다. "Browse.." 버튼을 눌러서 원하시는 위치를 선택하실 수 있습니다. 선택하셨으면, "Next" 버튼을 눌러주세요. 시작 메뉴 폴더에 S4A 프로그램의 바로가기를 생성하는 것을 원하지 않는다면, "Don't create a Start Menu folder" 를 체크해 주시면 됩니다.



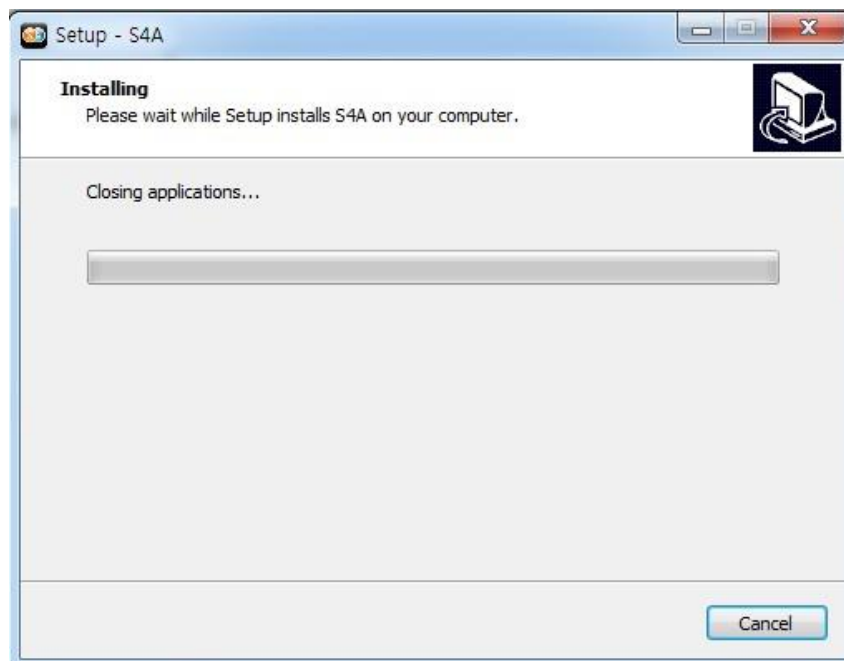
바탕화면에 S4A 프로그램의 아이콘을 만들기를 원하는 경우, "Create a desktop icon" 을 선택해주세요. "Next" 버튼을 눌러 다음 단계로 진행합니다.



이제 설치 준비를 마쳤습니다. 컴퓨터에 S4A를 설치하기를 원한다면, "Install" 버튼을 눌러주세요.



설치를 시작합니다. 설치가 완료 될 때까지 기다려주세요. 설치를 중단하고 싶으시다면, "Cancel" 버튼을 누르세요.



설치가 완료 되었습니다. S4A를 지금 시작하는 것을 원하지 않으시면, "Launch S4A"에 체크를 해제하시면 됩니다. "Finish" 버튼을 눌러주세요.



위 화면에서 "Launch S4A"를 선택하셨다면, 설치를 마친 후, 다음과 같이 S4A가 실행됩니다. 설치가 잘 되셨나요? S4A 프로그램의 오른쪽을 봐주세요. 보드 검색을 하고 있지요? S4A에서 아두이노 보드를 찾을 수 있도록 S4A 펌웨어를 아두이노 보드에 업로드 시키는 작업을 해 줘야 합니다. 이 작업을 위해 우선 S4A 프로그램을 종료시킵니다.

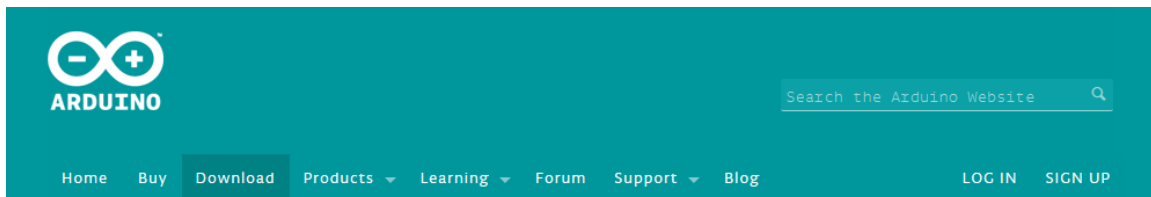


## 2. 아두이노 프로그램 설치하기


다음 사이트에서 프로그램을 다운 받습니다. (만약 컴퓨터에 아두이노 프로그램이 설치되어 있다면 이 부분은 지나치셔도 됩니다.)

<http://www.arduino.cc/en/Main/Software>

사이트를 방문하면 다음과 같은 화면이 보입니다. 여기서 Windows Installer 를 클릭합니다.



### Download the Arduino Software



**ARDUINO 1.6.4**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows Installer**  
Windows ZIP file for non admin install

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
Linux 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums](#)

ARDUINO SOFTWARE

**HOURLY BUILDS**

LAST UPDATE  
12 May 2015 1:46:57 GMT

Download a preview of the incoming release with the most updated features and bugfixes.

Windows  
Mac OS X (Mac OS X Lion or later)  
Linux 32 bit, Linux 64 bit

ARDUINO 1.0.6 / 1.5.x / 1.6.x

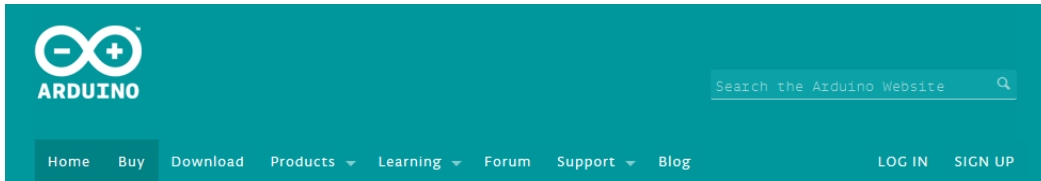
**PREVIOUS RELEASES**

Download the previous version of the current release, the classic Arduino 1.0.x, or the Arduino 1.5.x Beta version.

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.



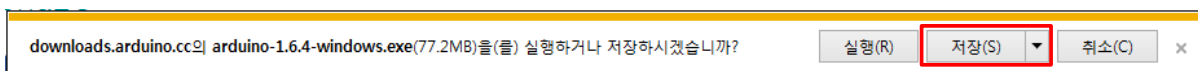
다음 화면이 나오면, JUST DOWNLOAD 를 클릭합니다.



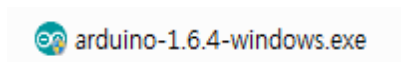
## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)

그러면 다음과 같은 창이 브라우저 하단에 보입니다.



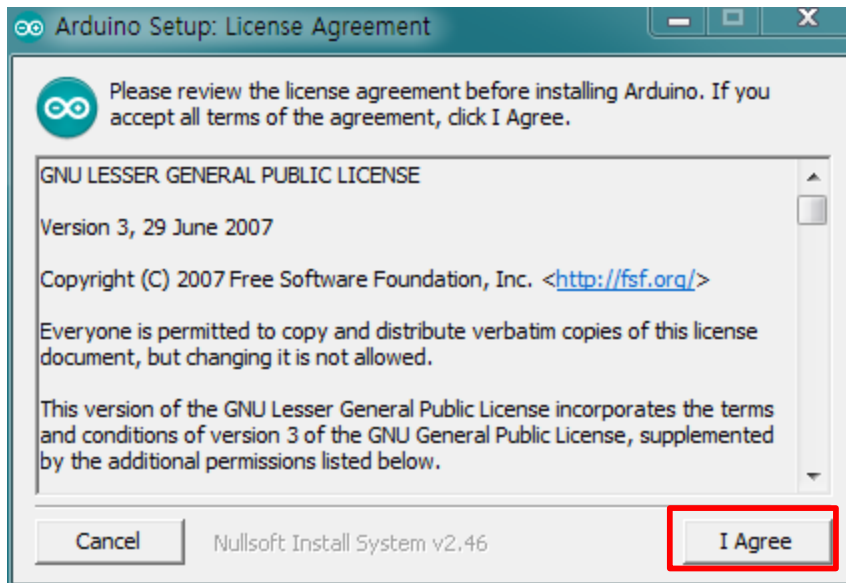
여기서 "저장" 버튼 중 오른쪽에 있는 아래쪽 방향 화살표를 누르면 "다른 이름으로 저장하기" 창이 나옵니다. 그러면 원하는 폴더를 선택하고 저장합니다. 저장한 폴더를 보면 다음과 같은 파일이 있습니다.



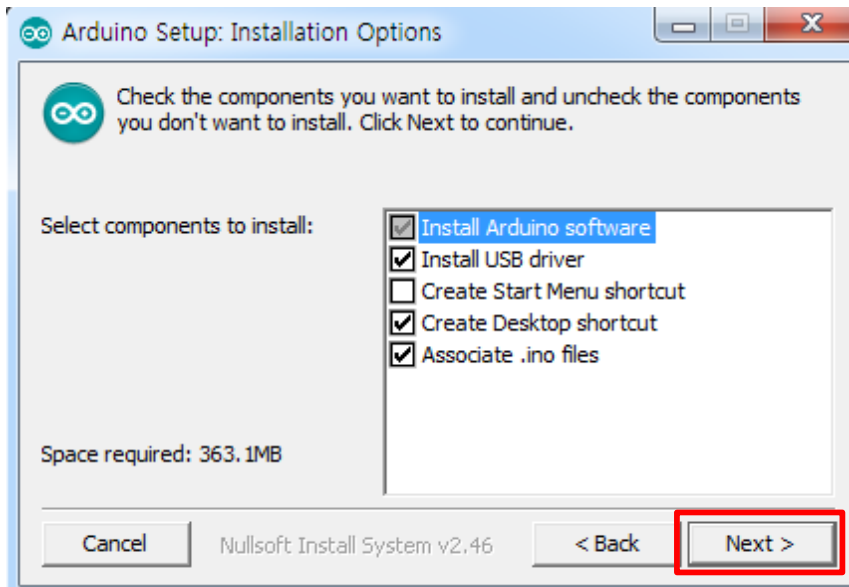
이 파일을 더블 클릭하여 실행합니다. 이제부터 아두이노 프로그램을 설치하는 것입니다.



아두이노 프로그램을 설치하는 첫 화면은 다음과 같습니다.

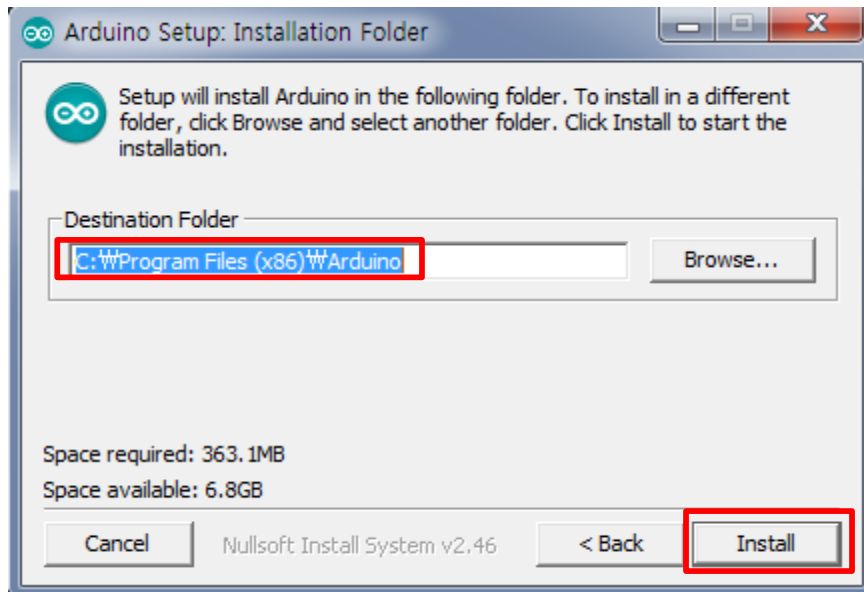


붉은색 박스 안의 "I Agree" 버튼을 클릭합니다. 그러면 다음과 같이 화면이 보입니다.

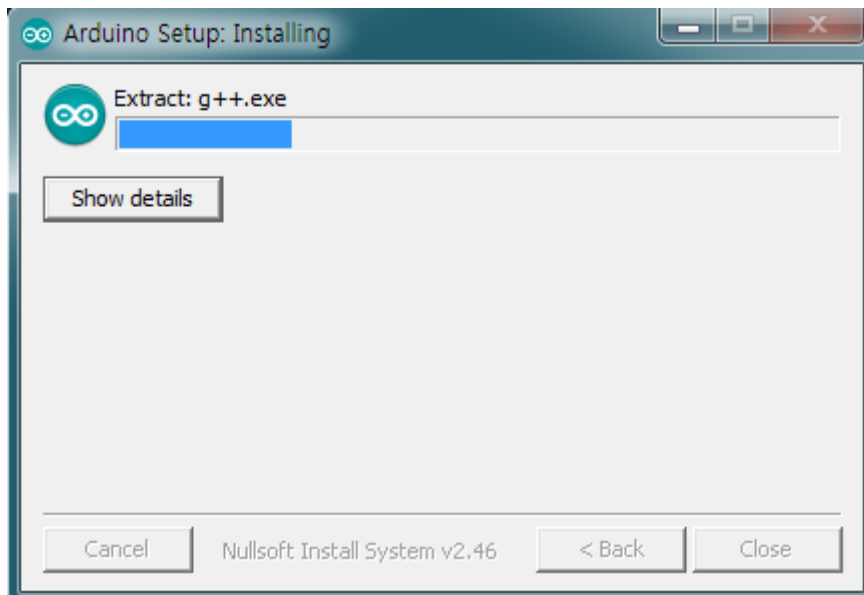


설치할 항목을 선택 합니다. 모든 것을 다 선택하셔도 되고, 위의 그림과 같이 선택하셔도 됩니다. "Next" 버튼을 클릭합니다.

다음으로, 설치할 폴더를 선택하라는 창이 나옵니다.



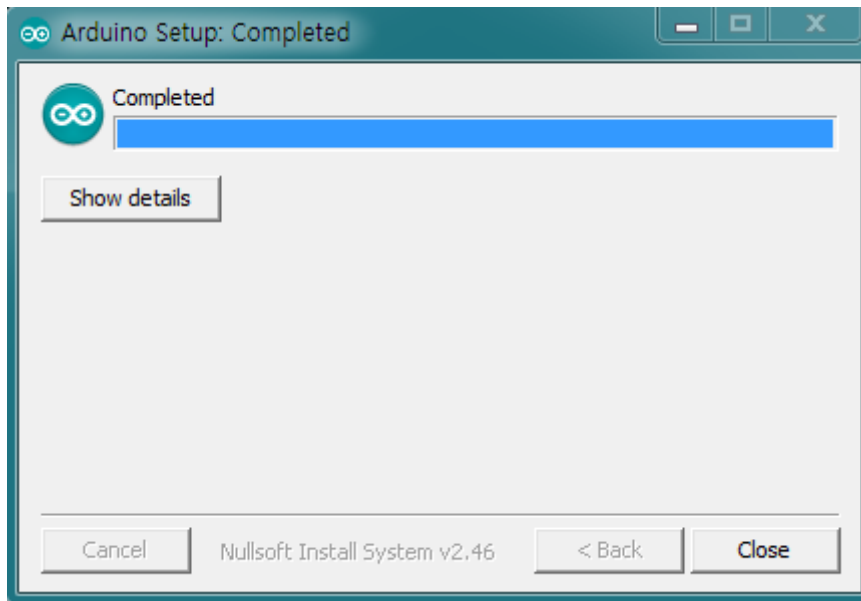
설치할 폴더를 쓰고 "Install" 버튼을 누릅니다. 이 때 기본 폴더를 선택하셔도 되고, "Browse..." 버튼을 이용하여 원하는 설치 폴더를 선택할 수 있습니다. 다음과 같은 설치 중인 화면이 보입니다. 약 2 ~ 3 분 정도 시간이 소요됩니다.



다음과 같이 중간에 설치를 물어 보는 창에서는 모두 "설치" 버튼을 클릭합니다. 이 창들은 모두 나올 수도 있고 이 중 한두 개만 나올 수도 있습니다.



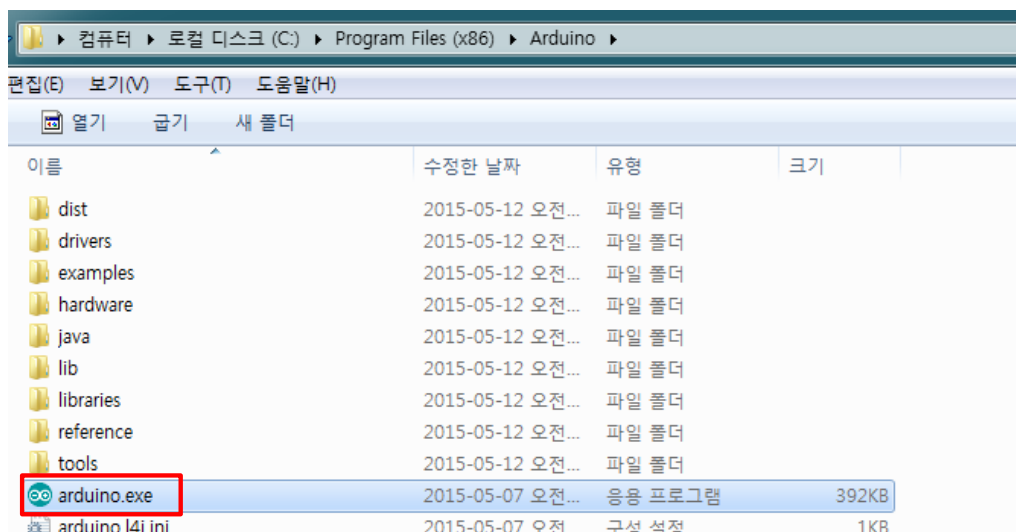
설치가 완료되면 다음과 같이 설치 완료 창이 표시됩니다.



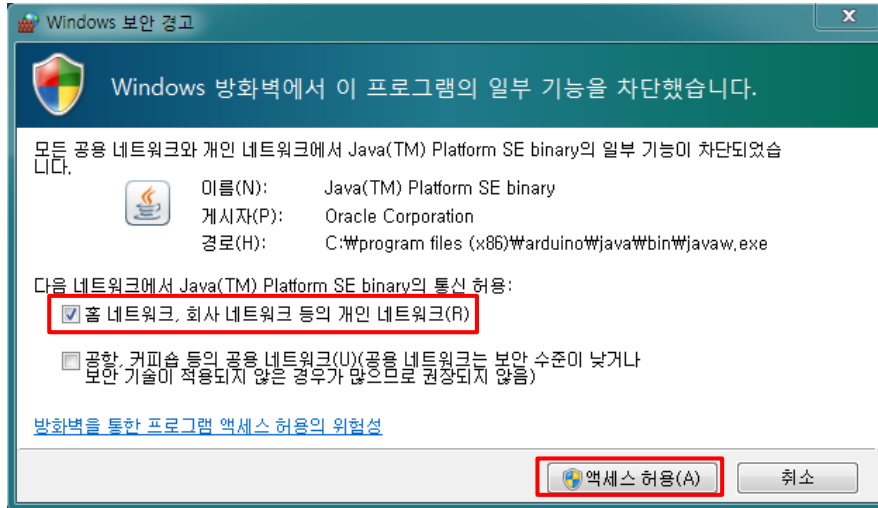
이제 바탕화면의 아두이노 아이콘을 더블 클릭하여 아두이노 프로그램을 실행합니다.



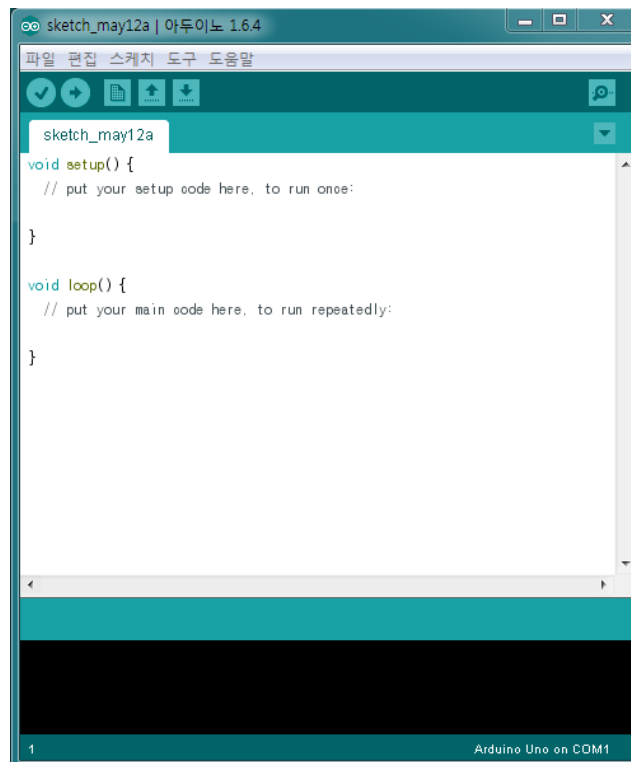
혹시 바탕화면에 아이콘이 없으면 설치 폴더로 가서 arduino.exe 를 실행합니다.



실행 중에 다음과 같은 창이 보이면 "홈 네트워크, 회사 네트워크 등의 개인 네트워크" 만을 체크하고, "액세스 허용" 버튼을 누릅니다.



아두이노 프로그램 설치가 완료 되었습니다. 다음과 같이 아두이노 프로그램을 보실 수 있습니다.



### 3. 코딩킷에 S4AFirmware16 설치하기

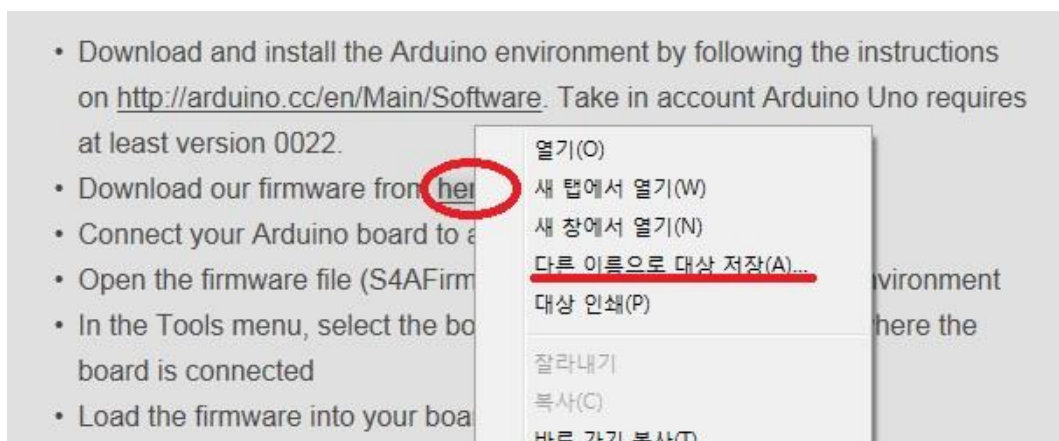
이전 장에서 아두이노 프로그램 설치를 잘 마무리 하셨다면 다음 단계인 펌웨어 업로드 과정을 실행하셔야 합니다. S4A 설치 프로그램을 다운 받았던 웹페이지로 돌아갑니다. <http://s4a.cat/>으로 가셔서 "Downloads" 를 누릅니다.

#### Installing the Firmware into your Arduino

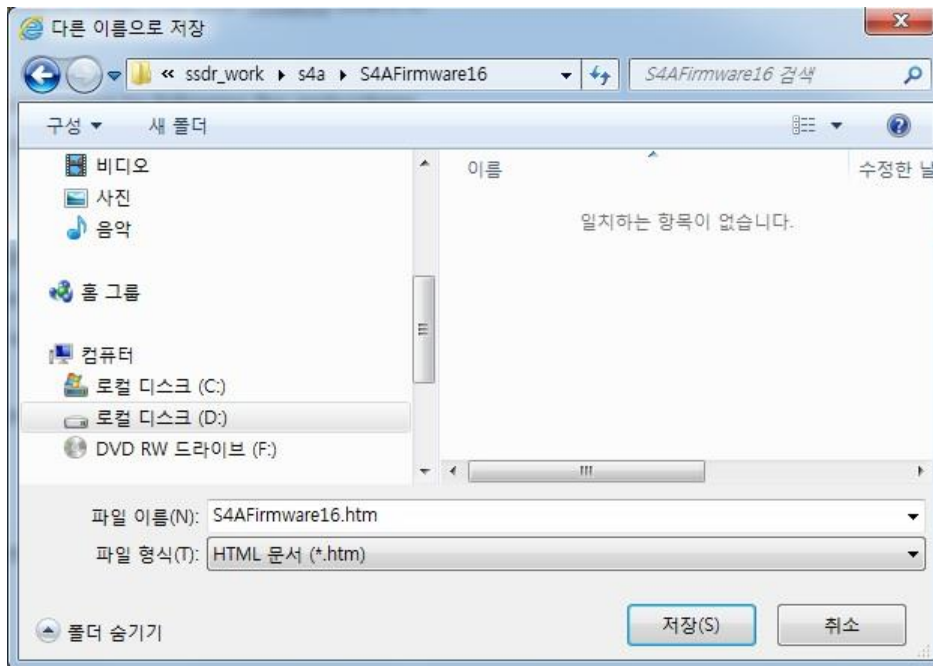
This firmware is a piece of software you need to install into your Arduino board to be able to communicate with it from S4A.

- Download and install the Arduino environment by following the instructions on <http://arduino.cc/en/Main/Software>. Take in account Arduino Uno requires at least version 0022.
- Download our firmware from [here](#)
- Connect your Arduino board to a USB port in your computer
- Open the firmware file (S4AFirmware16.ino) from the Arduino environment
- In the Tools menu, select the board version and the serial port where the board is connected
- Load the firmware into your board through File > Upload

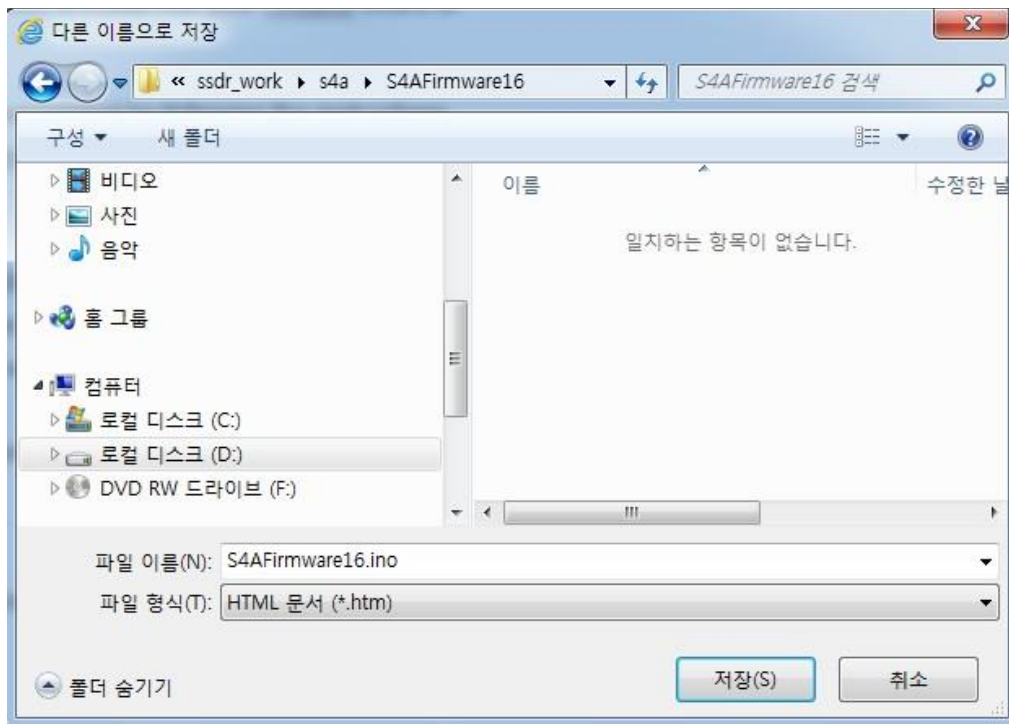
아두이노 개발 환경이 설치되었다면 펌웨어를 다운로드 합니다. "here"에 마우스를 가져가 오른쪽 클릭을 하여 "다른 이름으로 대상 저장(A)..."을 선택하시면 링크된 펌웨어를 다운로드 하실 수 있습니다.



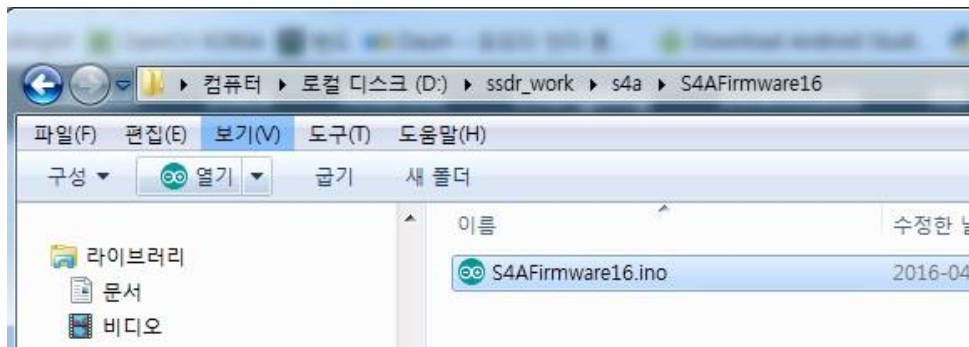
S4A 펌웨어를 저장하는 화면이 나옵니다. "S4AFirmware16.htm" 파일을 저장하라고 나옵니다. S4A 펌웨어를 저장할 위치를 선택하세요.



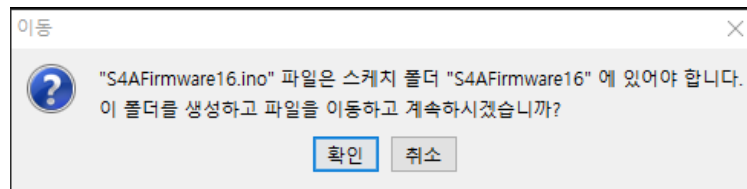
펌웨어 이름을 "S4AFirmware16.htm" 에서 "S4AFirmware16.ino"로 변경하여 저장합니다.



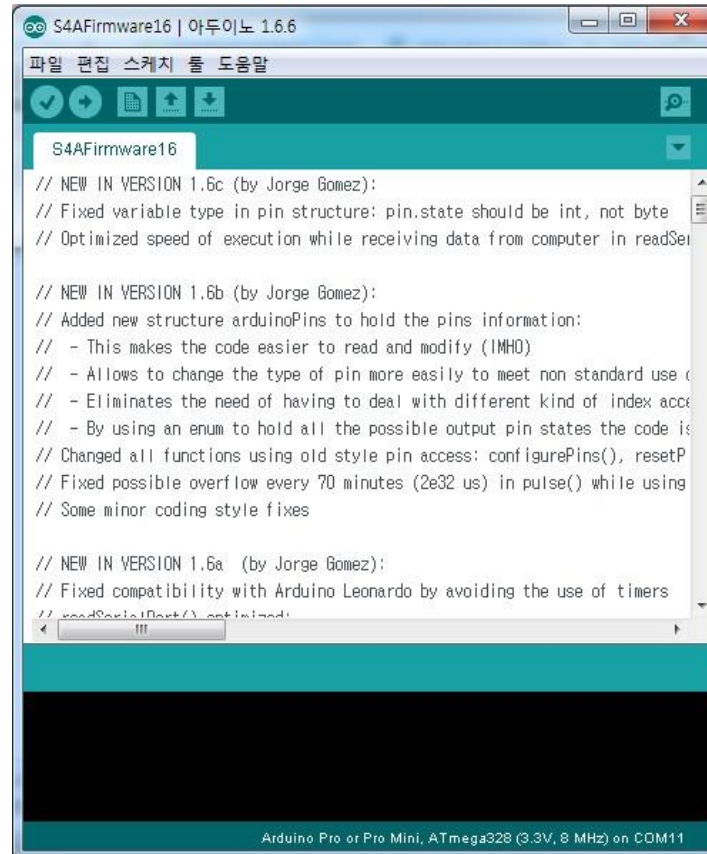
S4A 펌웨어를 다운 받으셨다면, "S4AFirmware16.ino"를 두 번 클릭하여 실행시킵니다.



다음과 같은 메시지가 나오면 "확인" 을 누릅니다. 아두이노 코드들은 항상 같은 이름의 폴더 안에 있어야 합니다.

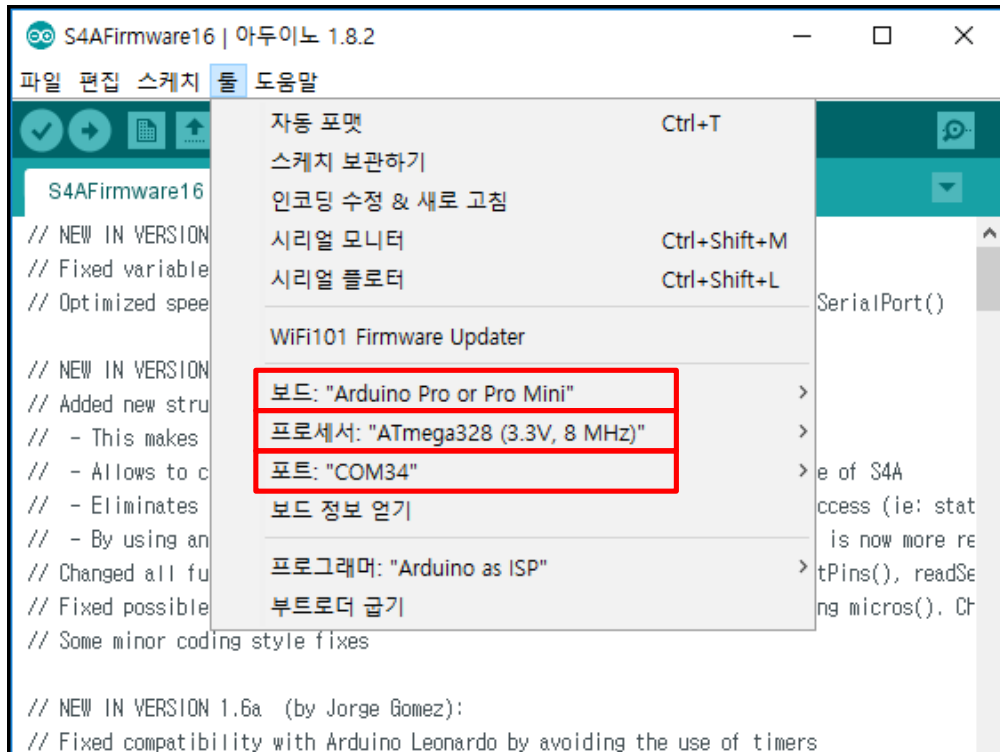


다음과 같이 아두이노 프로그램이 실행됩니다.

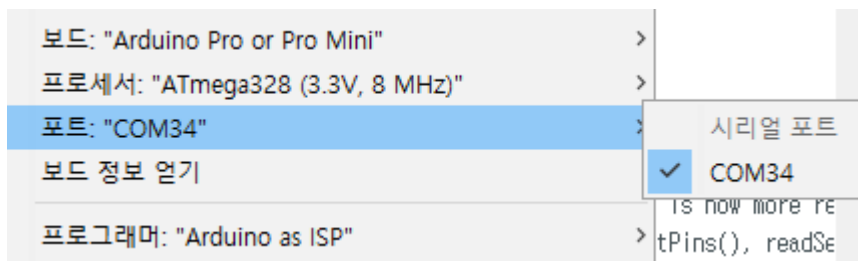




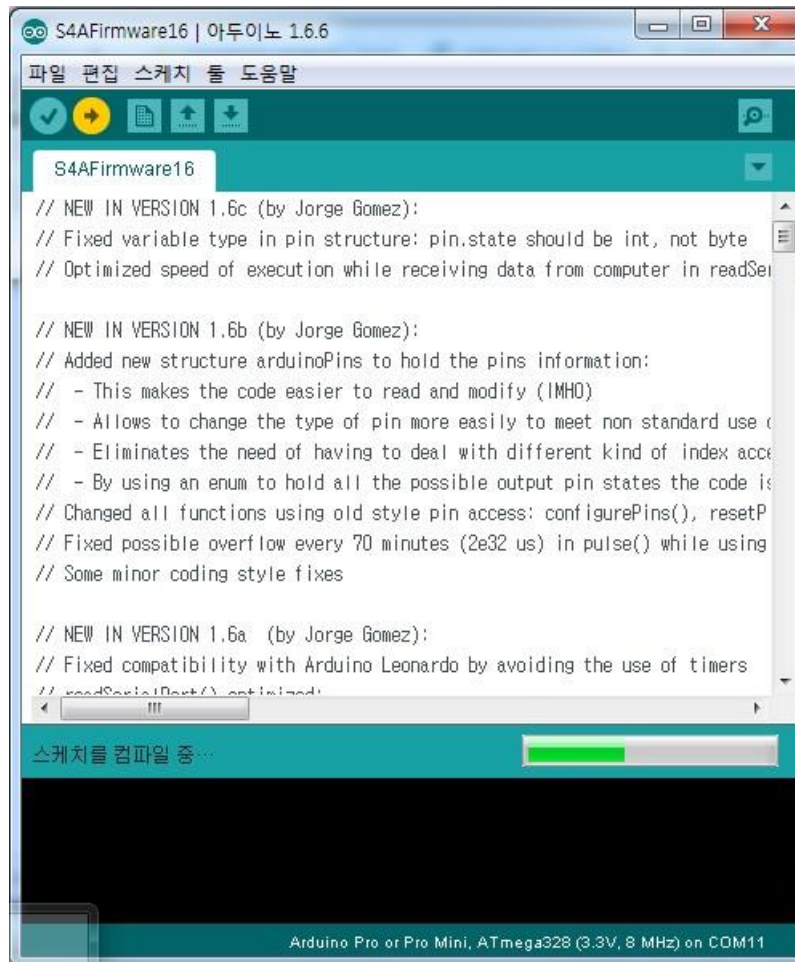
이제 코딩킷의 전원을 올리시고 USB 케이블로 컴퓨터와 연결합니다. 그리고 펌웨어를 업로드 하시기 전에 다음 그림과 같이 "툴" 메뉴의 "보드", "프로세서"를 그림과 같이 선택해 주셔야 합니다. "포트"는 현재 코딩킷이 연결되어 있는 "포트"를 선택해 줍니다.



"포트" 메뉴를 클릭하시면 다음 그림과 같이 "COM" 과 숫자가 보일 것입니다. 대부분 하나가 보일 것이고 그것을 선택해 주시면 됩니다.



화살표 버튼을 누르면, 스케치를 컴파일 하여 아두이노 보드로 펌웨어를 업로드 합니다.



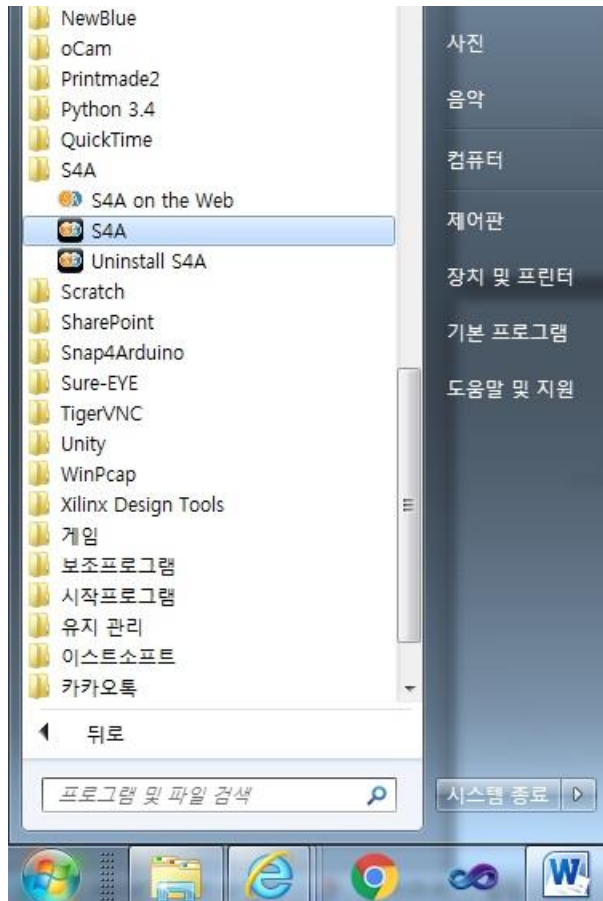
업로드 과정을 무사히 마쳤다면, 다음과 같은 화면을 확인하실 수 있습니다. 스케치를 컴파일 하는 과정에서 "typedef" 선언에 대한 경고 메시지가 나옵니다. 이것은 S4A 펌웨어 코드의 문제이니, 무시하셔도 상관없습니다.



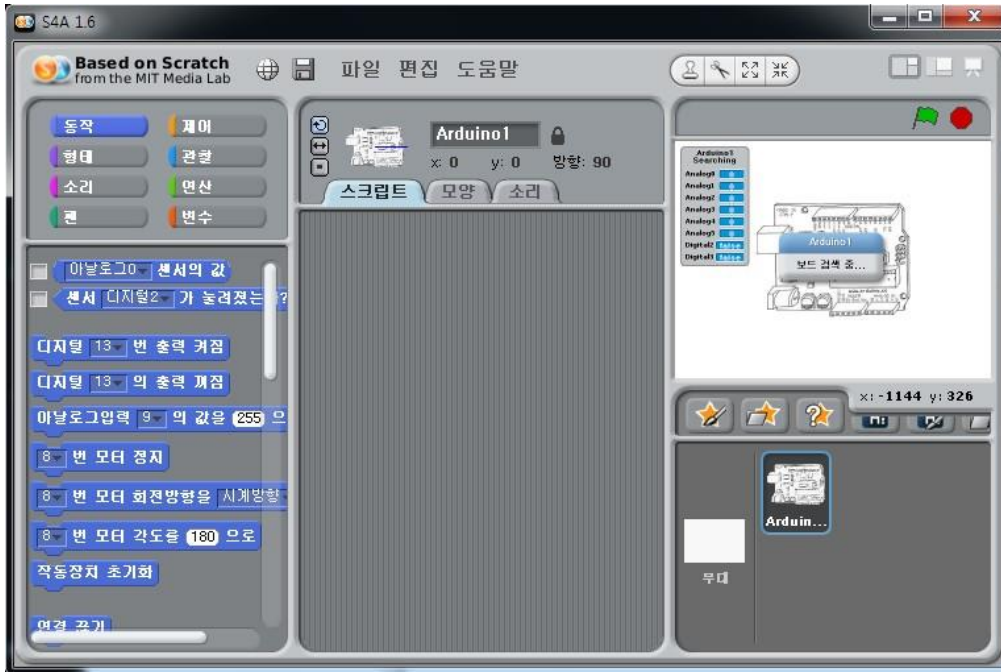
## 4. 코딩킷과 S4A 연결하기

이제 준비는 다 되었습니다. S4A 프로그램을 실행시켜볼까요?

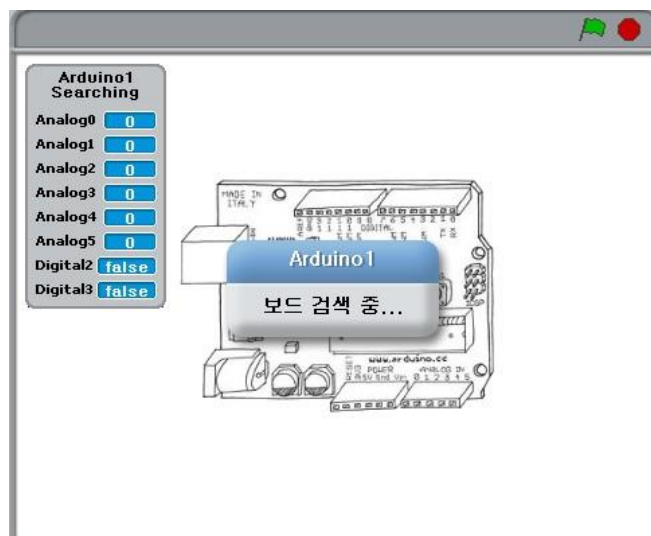
프로그램 시작 메뉴에서 S4A 프로그램이 설치된 경로를 찾아 S4A 프로그램을 시작합니다. 설치하실 때, 경로를 기억하세요.



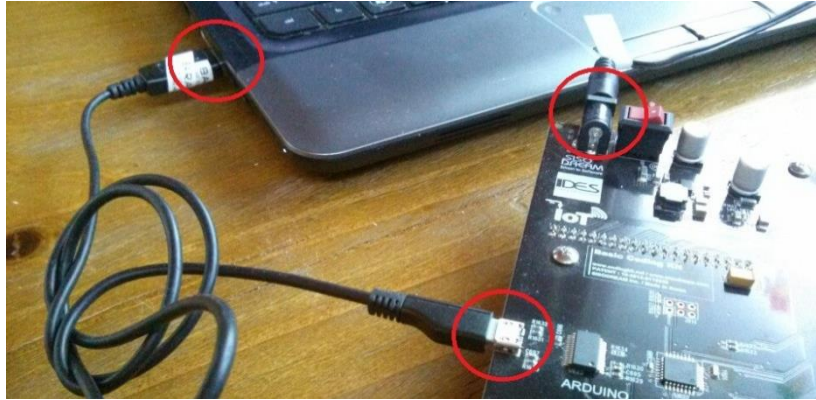
다음 화면과 같이, S4A 프로그램이 실행되었나요?



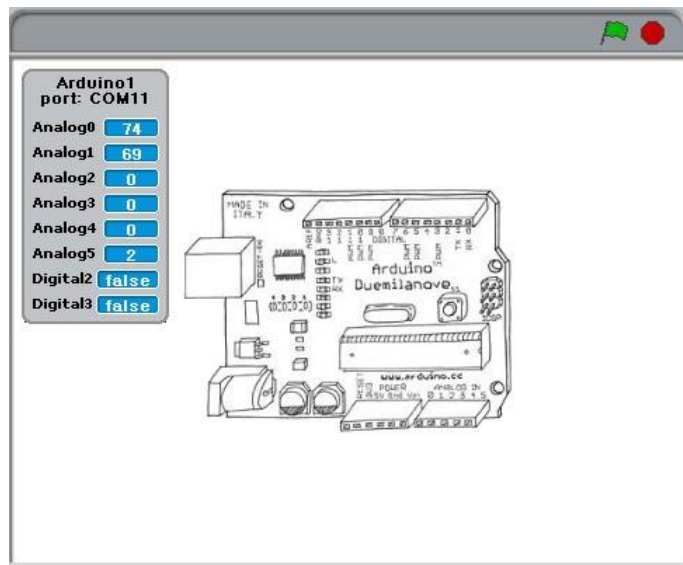
오른쪽 상단 창을 보시면, 다음과 같이 아두이노 보드를 검색하는 화면이 나옵니다.



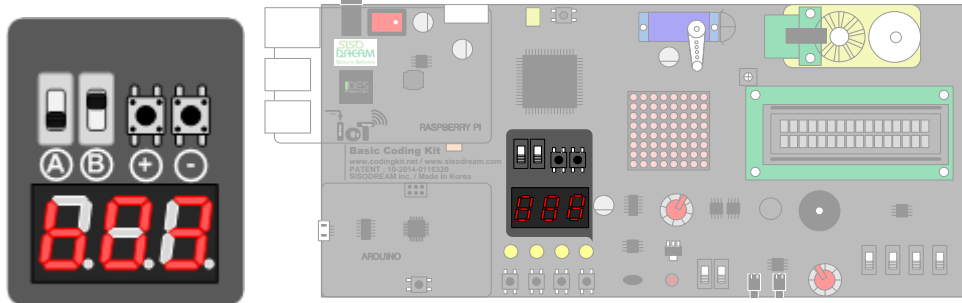
보드 검색이 성공적으로 이루어지기 위해 작업을 좀 해야 할 것 같습니다. 무엇보다 해야 할까요? 첫 번째로, S4A와 코딩킷과 물리적인 연결이 있어야 하겠죠? 다음 사진처럼, S4A가 설치된 컴퓨터와 코딩킷을 USB 케이블로 연결합니다. 코딩킷에 전원 케이블도 연결해 줍니다.



지금까지의 설치 과정을 잘 마쳤다면, 다음과 같이 보드 검색이 성공적으로 됩니다. 아래 그림에서는 COM 포트 11에 아두이노가 연결이 되었네요. 그리고 각 장치들의 값들이 읽혀지는 상태를 확인하실 수 있습니다.



코딩 키트에 다음 그림과 같이 "MODE SETTING SWITCH" 가 있습니다. 스위치 A는 아래로 내려주시고, B스위치는 위로 올려주세요.



코딩킷의 세븐세그먼트를가 "b00" 값으로 깜박이고 있을 것입니다. + 또는 - 버튼을 길게 1~2초간눌러 주시면, 세븐세그먼트의 깜박임이 멈춥니다.

우리는 S4A를 사용하여 블록코딩을 할 것입니다. + 또는 - 버튼을 눌러 "b03"으로 변경해 주세요. 그리고 다시 + 또는 - 버튼을 길게 1~2초간 눌러서, 세븐세그먼트의 깜박임이 멈추도록 해 줍니다.

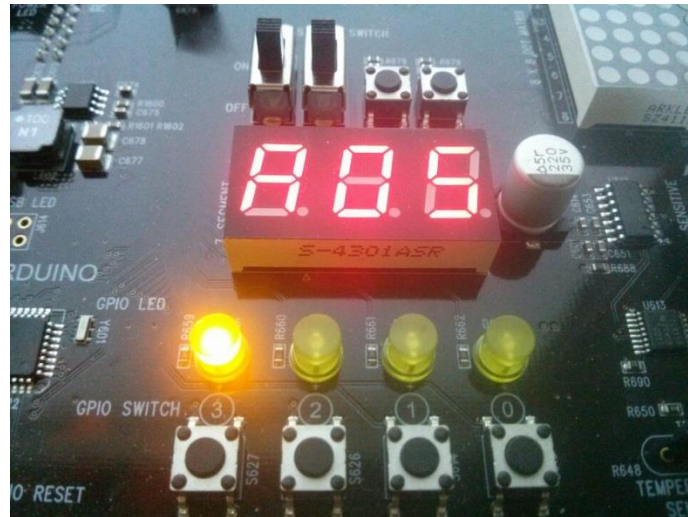
\* 코딩킷의 스위칭 시스템에 대해서 자세히 알고 싶으시다면, 코딩킷 교재중 "코딩북1 - 코딩킷 소개" 교재를 참고하세요.

이제 모든 준비 과정을 다 마쳤습니다. 그러면, 잘 동작하는지 확인해 볼까요? "동작" 블록 모음에서 "디지털 13번 출력 켜짐" 블록과 "디지털 13번 출력 꺼짐" 블록을 스크립트로 가져옵니다.





“디지털 13 번 출력 켜짐” 블록을 마우스로 두 번 클릭하여 실행시켜 보세요. 다음 화면과 같이 LED 에 불이 들어오는 것을 확인할 수 있습니다. LED 의 불을 끄고 싶다면 “디지털 13 번 출력 꺼짐” 블록을 마우스로 두 번 클릭하여 실행시키시면 됩니다.



여기까지 잘 따라오셨다면, S4A로 코딩을 즐기실 준비가 되신 것입니다! 수고하셨습니다. ^^



## 부록 D : 스위칭(Switching) ID

코딩키트는 스위칭 시스템을 이용하여 여러가지 연결을 설정할 수 있습니다. 이렇게 연결된 각각의 설정은 스위칭 ID 로 표시합니다. 다음은 코딩키트의 블록코딩에서 스위칭 ID 에 따른 디바이스 연결 표입니다. 표에서 붉은색으로 표시된 숫자는 파란색으로 표시한 스위칭 ID 에서 각각의 디바이스에 연결된 아두이노의 핀 번호입니다. 여러분은 블록코딩에서 이 번호로 코딩을 하시면 됩니다.

Block Coding 디바이스	Switching ID 핀 번호					
	b00	b01	b02	b03	b04	b05
LED 0	2			10	10	10
LED 1	3			11	11	11
LED 2	7			12		
LED 3	8			13		
Button 0	12	6	13	2	2	
Button 1	13	7	A4	3		
Button 2		8				
Button 3		9				
Buzzer	5			5	5	5
DC Motor Enable	6			8	12	12
DC Motor Forward	10			8	6	6
DC Motor Backward	11			8	9	9
IR Sensor LED	4		A5		13	13
Servo Motor	9			4	4	4
Dotmatrix			*2			
Character LCD EN, RS, Data		*1				
Character LCD Back Light		13				
Switch 0		10			3	2
Switch 1						3
Serial Monitor	0,1		0,1	0,1	0,1	0,1

위의 표에서 Character LCD EN, RS, Data 와 Dotmatrix 의 아두이노 연결 핀 번호는 다음과 같습니다.

*1 : Char LCD	*2 : Dotmatrix
LCD_EN : 11	DM1 : 2
LCD_RS : 12	DM2 : 7
LCD_D0 : 5	DM3 : 10
LCD_D1 : 4	DM4 : 6
LCD_D2 : 3	DM5 : 4
LCD_D3 : 2	DM6 : 5
	DM7 : 12
	DM8 : 13
	DM_SEL : 3

# Release Note

V2.0 : 2017. 7. 28

- 첫번째 버전 Release