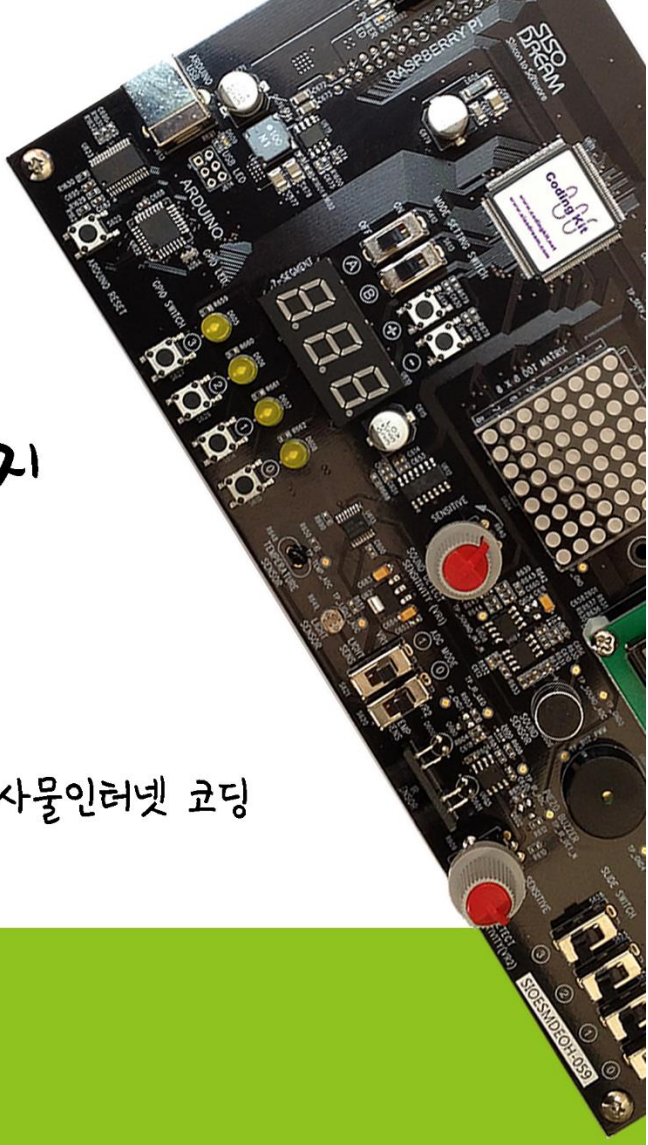
코딩키트



블록 코딩에서 사물인터넷 코딩까지
쉽고 재미있게 배울 수 있어요.

블록 코딩 / 아두이노 코딩 / 라즈베리파이 코딩 / 사물인터넷 코딩

코딩북2 - 블록코딩 스냅

아빠와 함께하는 곰곰이의 코딩 놀이

SISO DREAM
Silicon to Software

(주)시소드림 SISODREAM Inc.

- 본 문서의 저작권 -

발행일 : 2017 년 3 월 8 일 (초판)
발행처 : ㈜시소드림
연락처 : ck@sisodream.com / 031-757-7755

- 목 차 -

1. 아빠! 코딩이 뭐예요?

곰곰이가 아빠에게 코딩에 관하여 묻습니다.

아빠! 코딩이 뭐예요?

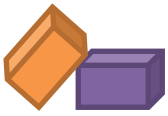
응~ 코딩은 컴퓨터에게 일을 시키는 거야!

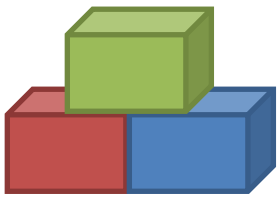
그런데, 컴퓨터는 사람 말을 알아듣지 못해. 그래서, 컴퓨터가 알아들을 수 있는 말을 사용해서 일을 시키지. 그것을 코딩이라고 해. 코딩 작업을 통해서 로봇을 움직이게 하고, 인터넷이나 게임을 할 수 있게 되는 거란다.

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!

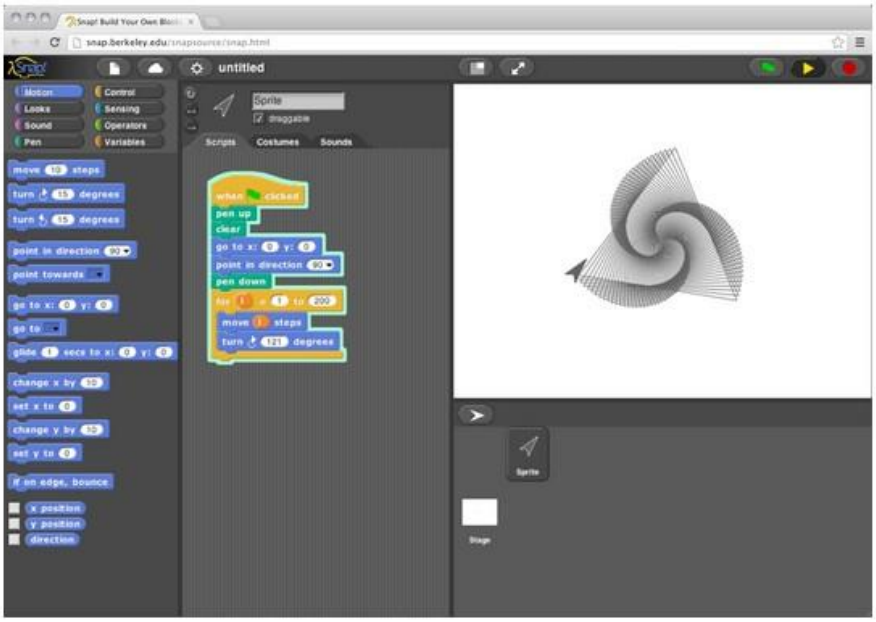
요즘에는 어린이들이 쉽게 코딩을 배울 수 있는 프로그램이 많이 있단다. 그런 프로그램으로는 스냅(Snap), 스크래치(Scratch), S4A(Scratch for Arduino), 엔트리(Entry) 등이 있지. 이 프로그램들은 모두 블록 놀이를 하듯이 코딩을 할 수 있게 해준단다.

코딩이 블록 놀이랑 같다면 정말 재미있을 것 같아요!

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

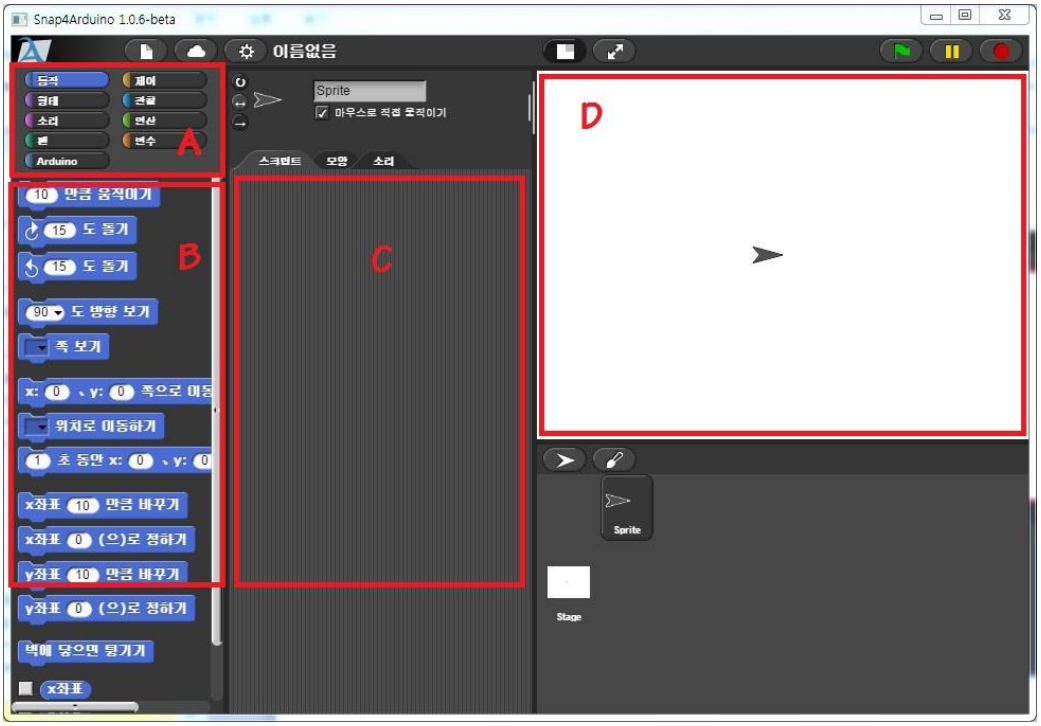
3-1. 스냅 아두이노 프로그램 살펴보기

블록코딩 프로그램인 스냅 아두이노(Snap4Arduino) 프로그램에 대해 간단히 살펴보겠습니다. 스냅 아두이노는 스크래치(Scratch) 프로그램과 사용법이 유사하면서도 스크래치보다 코딩키트 같은 소형 컴퓨터와 연결하는데 더 많은 기능을 제공합니다.



**** 스냅 아두이노 프로그램을 설치하는 과정은 "부록 C. 프로그램 설치 가이드"에 자세히 설명되어 있습니다. 아직 컴퓨터에 스냅 아두이노 프로그램이 설치되어 있지 않다면 지금 설치해 주세요. 아빠가 도와 주세요!!***

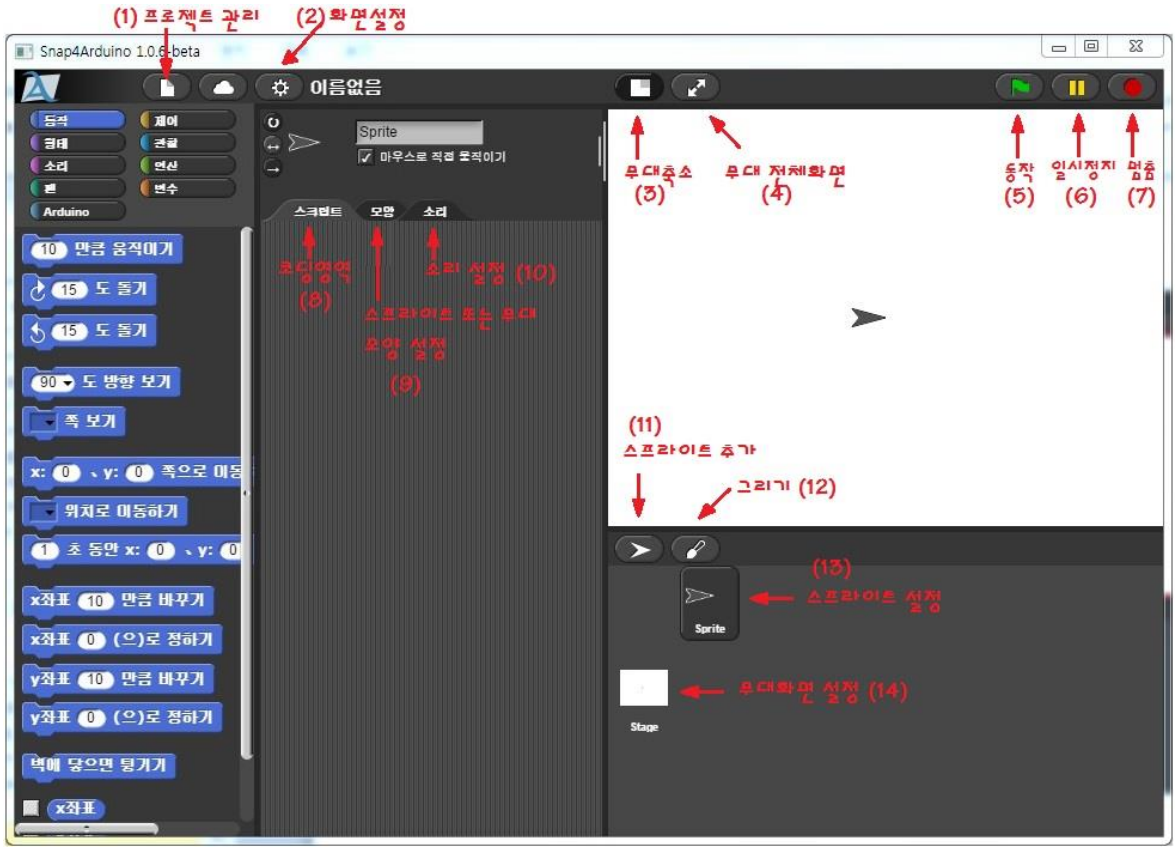
스냅 아두이노 프로그램은 크게 블록 영역(A+B), 스크립트 영역(C), 무대 영역(D)으로 나누어 집니다.



블록 영역(A+B)은 스크립트를 작성하기 위한 블록들이 있습니다. 이 블록들은 특성에 따라 동작, 제어, 형태, 관찰, 소리, 연산, 펜, 변수, 아두이노(Arduino)로 구분되어 있습니다. 이 블록들은 드래그-앤-드롭 방법으로 스크립트 영역(C)으로 가져다 놓으면 동작을 하게 됩니다. 스냅 아두이노에서 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 스냅 아두이노 프로그램의 동작 및 기능을 간단히 설명해드리겠습니다. 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) 프로젝트 관리 : 프로젝트를 저장하거나 이전에 만들어진 프로젝트를 가져올 때 사용합니다. 그리고 스냅 아두이노에서 제공되는 다양한 라이브러리나 소리, 모양들을 프로젝트로 가져올 때 사용합니다.

(2) 화면 설정 : 언어, 블록크기, 무대크기 등을 설정 할 수가 있습니다.

(3) 무대 축소 : 무대 화면을 작게 만듭니다. 반대로 스크립트 영역이 넓어집니다. 원래대로 되돌리기 위해서는 버튼을 다시 누릅니다.

(4) 무대 전체 화면 : 무대 화면을 전체 화면으로 만듭니다. 원래대로 되돌리기 위해 버튼을 다시 누릅니다.

(5) 동작 : 스크립트를 동작 시킵니다.

(6) 일시 정지 : 스크립트의 동작을 일시 정지 시킵니다.

(7) 멈춤 : 스크립트의 동작을 멈추게 합니다.

(8) 코딩 영역 탭 : 스크립트를 작성하는 창 입니다. 블록들을 이 창으로 드래그해서 순서대로 배치합니다. 이것이 블록 코딩입니다.

(9) 모양 탭 : 이 탭에서 스프라이트나 무대 배경에 사용할 그림들을 관리합니다.

(10) 소리 탭 : 이 탭에서 스크립트에서 사용할 소리 파일들을 관리합니다.

(11) 스프라이트 추가 : 스프라이트는 프로그램에서 동작의 주체입니다. 스프라이트를 추가할 때 사용합니다.

(12) 그리기 도구 : 스프라이트와 배경 화면 등을 그리기 도구로 직접 그릴 수 있습니다.

(13) 스프라이트 설정 : 이 버튼을 누르면 스크립트 영역이 해당 스프라이트의 작업 영역으로 설정됩니다.

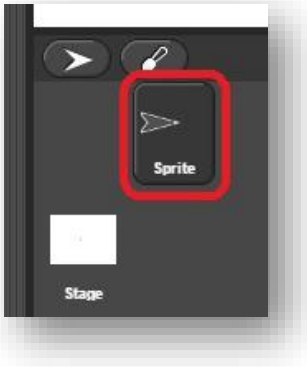
(14) 무대 화면 설정 : 이 버튼을 누르면 무대 화면을 만들거나 수정하는 작업을 할 수 있습니다.

3-2. 스냅 아두이노 맛보기

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

1 단계 : 바람개비 그림 올리기

① 스냅 아두이노 프로그램의 우측 하단에 있는 "Sprite" 버튼을 선택해주세요.

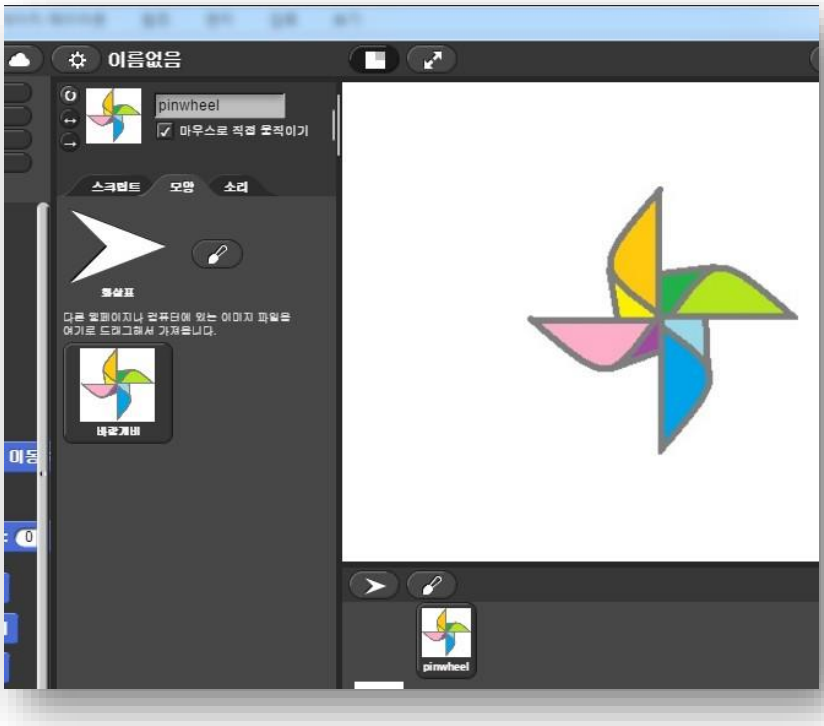


② 선택한 스프라이트의 이름을 변경해 보겠습니다. 프로그램의 중앙 상단에 다음 그림과 같이 스프라이트의 이름을 입력할 수 있는 창이 있습니다. 우리가 작업할 스프라이트의 이름을 "pinwheel" 이라고 수정해 볼까요? 이름을 수정한 후에 엔터를 눌러 입력을 끝내면, 스프라이트의 이름이 "pinwheel"로 변경된 것을 확인하실 수 있습니다.

③ 이제 화살표 모양의 "pinwheel" 스프라이트의 모양을 바람개비 그림으로 변경해 보겠습니다. 스프라이트의 이름을 입력했던 창 아래에 보시면 "모양" 탭이 있습니다. "모양" 탭을 선택해 주세요. 그러면, 화살표 그림이 나옵니다. 그리고 그 아래를 보시면 "이미지 파일을 여기로 드래그해서 가져옵니다."라고 적혀 있습니다. 그럼, 본 교재와 함께 제공된 바람개비 그림을 가져와 보겠습니다.

④ 윈도우 탐색기를 열고, 본서와 함께 제공되는 파일들 중에서 "바람개비.jpg" 파일을 드래그-앤-드롭 방식으로 "모양"탭 화면에 가져다 놓습니다.

⑤ 다음 그림과 같이 "pinwheel" 스프라이트의 모양이 바람개비 그림으로 바뀌었지요?



2 단계 : 스크립트 작성하기

① 이제, 바람개비를 돌려보겠습니다. 스크립트 영역에서 "스크립트" 탭을 선택해주세요. 그리고, 블록 영역에서 "제어" 버튼을 선택합니다.

② 블록 영역의 "제어" 블록들 모음에서 

블록을 스크립트로 드래그해서 가져옵니다.

③ 그리고, "제어" 블록들 모음에서 

블록을 스크립트로 드래그해서 가져와 다음 그림과 같이 

블록 아래에 끼워 넣습니다.

④ 이번에는, 블록 영역 상단에 있는 

버튼을 눌러주세요. 그리고, 동작 블록들 중에서 

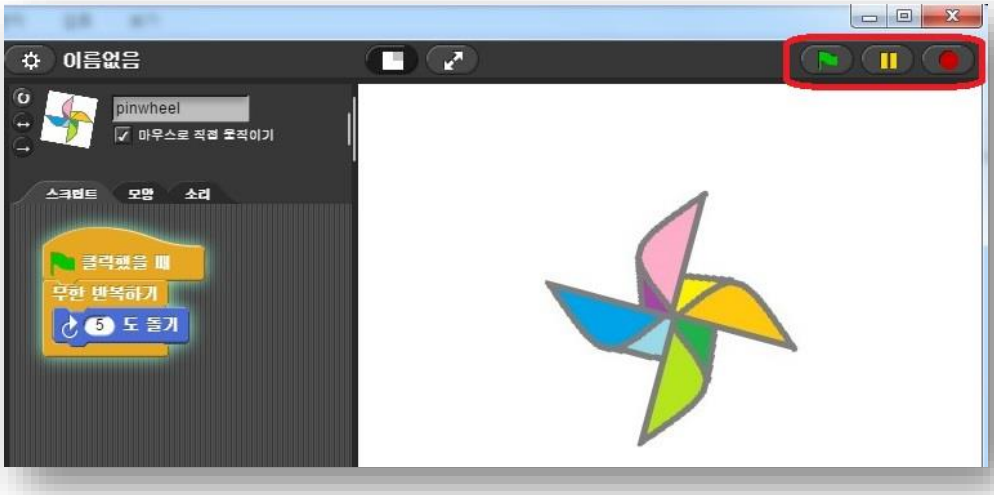
블록을 찾아서 스크립트로 가져와 

블록 안에 넣어 줍니다.

⑤ 바람개비가 도는 속도를 조절하기 위해 

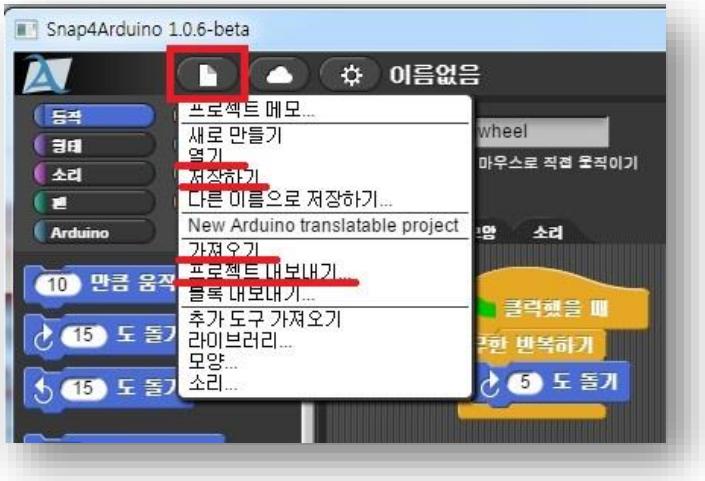
블록에 입력된 값을 5로 수정합니다. 바람개비를 좀 더 빨리 돌리게 하고 싶으면, 이 숫자를 줄여주면 됩니다. 반대로 바람개비를 천천히 돌리고 싶으면 숫자를 늘려주면 되겠지요?

⑥ 이제 코딩이 끝났습니다. 어떠셨나요? 그리 어렵지 않지요? 그럼 잘 만들었는지 동작을 시켜보겠습니다. 스냅 아두이노 프로그램의 우측 상단에 있는  버튼을 클릭해 보세요. 바람개비가 잘 돌아가지요? 동작을 잠깐 멈추게 하시려면,  버튼을 눌러주고, 끝내기를 하시려면  버튼을 눌려줍니다.



3 단계 : 스크립트 저장하기와 불러오기

① 작성한 스크립트를 저장하려면, 스냅 아두이노 프로그램의 좌측 상단에 있는 프로젝트 관리 메뉴 () 버튼을 눌러주세요. 프로젝트를 저장할 때는 "저장하기" 나 "프로젝트 내보내기" 메뉴를 사용합니다. 프로젝트를 가져올 때는 "열기"나 "가져오기" 메뉴를 사용합니다.



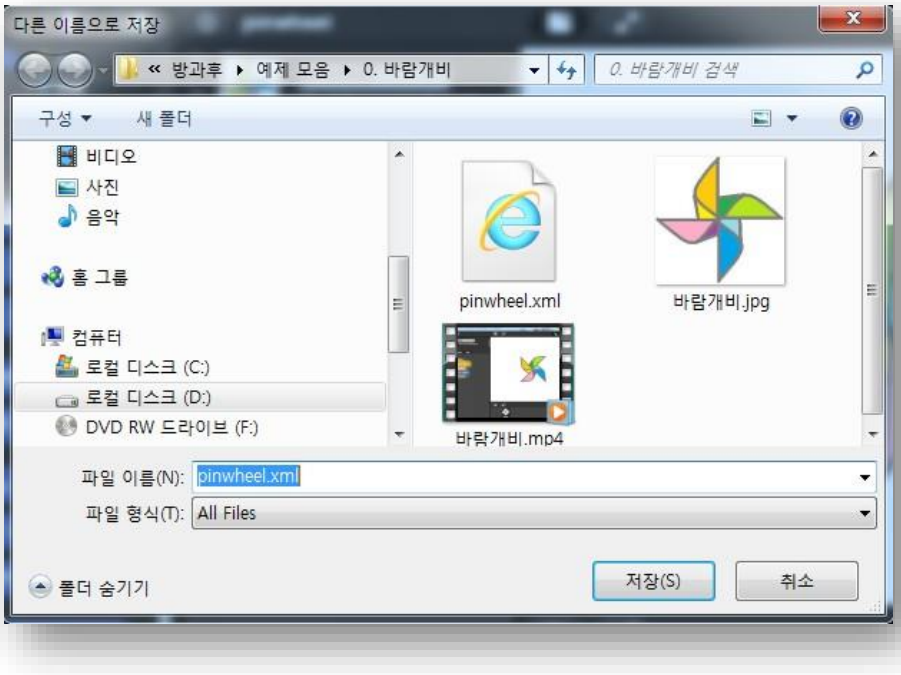
② "다른 이름으로 저장하기..." 메뉴를 선택하면 다음과 같은 화면이 나옵니다. 저장하고자 하는 이름을 적고 "저장하기" 버튼을 누릅니다. (프로젝트를 만들고 이름을 지정

하기 않고 "저장하기" 메뉴를 사용하면 "이름없음"으로 저장됩니다.)

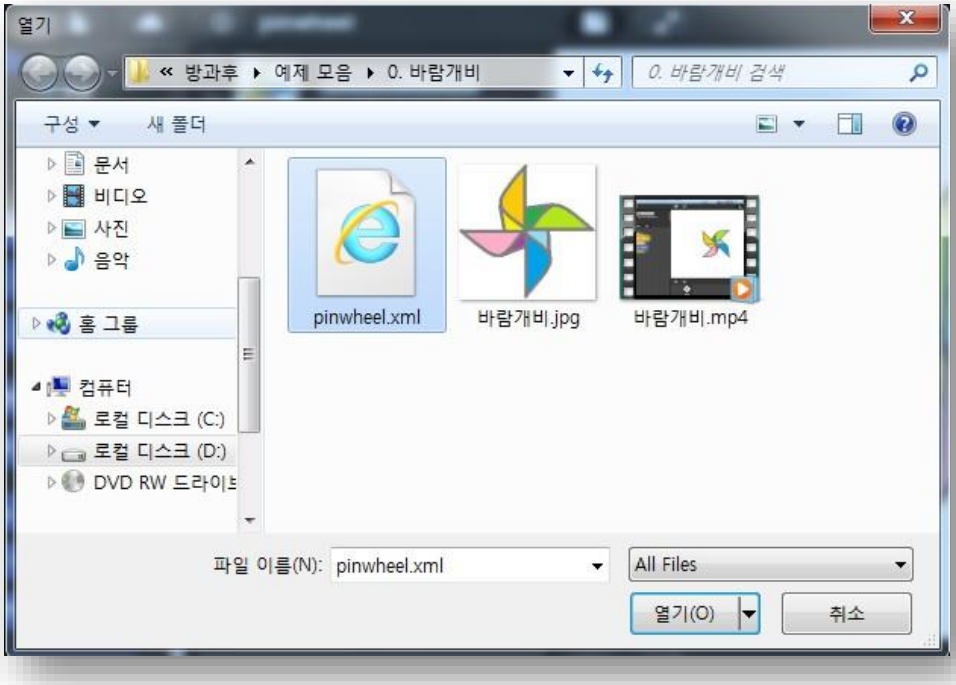
③ 위와 같이 저장된 프로젝트를 가져오고자 할 경우에는, 프로젝트 관리 메뉴에서 "열기" 메뉴를 선택합니다. 프로젝트명 리스트에서 원하는 프로젝트를 클릭하고 "열기" 버튼을 누르면, 프로젝트를 가져올 수 있습니다.

④ 또 다른 방법으로는 "가져오기..."와 "내보내기..." 메뉴를 사용할 수 있습니다. "내보내

기...” 메뉴를 선택하시면, 다음과 같이 윈도우 탐색기 창이 뜹니다. 원하는 위치와 파일명을 입력하고, “저장” 버튼을 눌러서 프로젝트를 저장합니다.



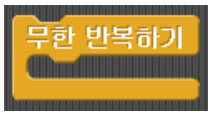
⑤ ④의 방법으로 외부에 저장된 프로젝트는 “가져오기...” 메뉴를 사용하여 프로젝트를 가져올 수 있습니다. “가져오기...” 메뉴를 선택하시고, 원하는 프로젝트 파일을 선택하고 “열기” 버튼을 누릅니다.



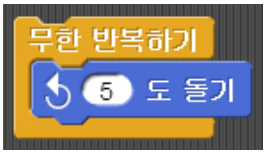
설명 더하기

✓ 

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.



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

✓ 스프라이트에서 쓰던 블록들을 지우는 방법으로는 블록들에서 마우스 오른쪽 버튼을 누르면 "삭제하기" 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 자동문으로 만들고...

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

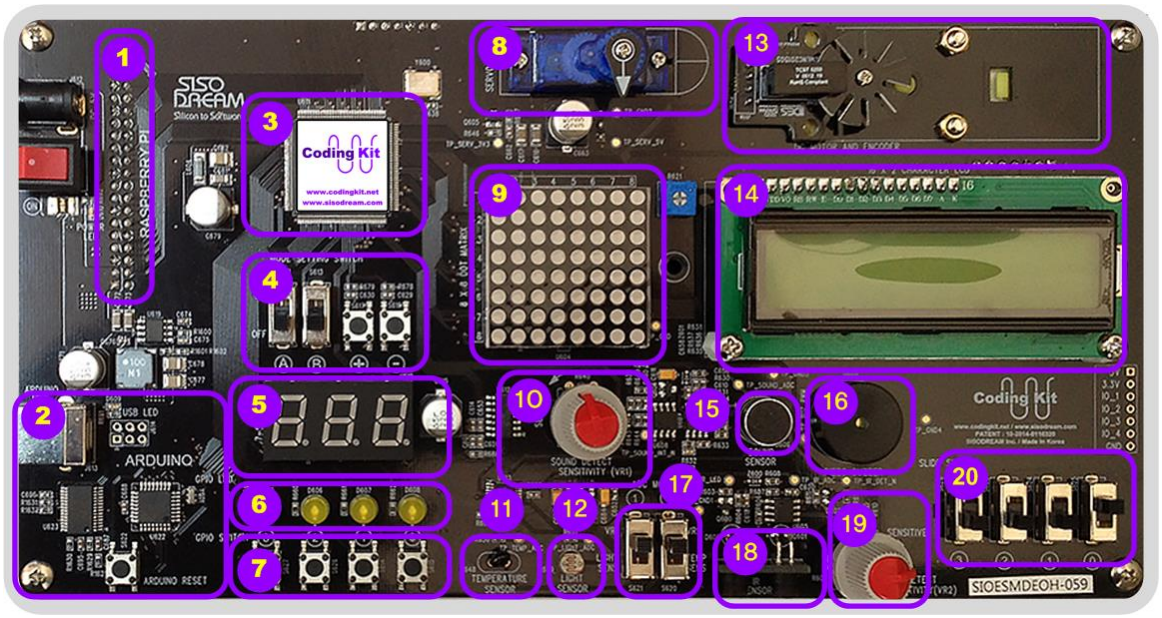
곰곰이는 아빠가 선물 해 주신 코딩키트 안에 작은 컴퓨터가 있다는 이야기를 듣고 놀랐습니다. 그리고 작은 컴퓨터와 다양한 장치들을 손쉽게 동작 시켜 볼 수 있다고 하니 빨리 코딩을 하고 싶은 생각이 들었습니다.

4-1. 코딩키트 살펴보기

코딩키트에는 로보트 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, 코딩키트에 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 로보트 팔 등을 움직일 수 있는 DC 모터, 서보 모터 등도 있습니다. 앞이로 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 코딩키트 보드에 대한 자세한 내용입니다.



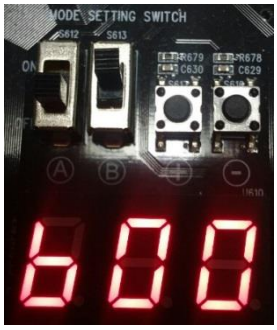
1. 라즈베리 파이 커넥터 (Raspberry Pi Connector)
2. 아두이노 (Arduino)
3. 스위칭 시스템 스위칭 칩 (Switching System, Switching Chip)
4. 스위칭 시스템 스위치와 버튼 (Switching System, Switch & Button)
5. 스위칭 시스템 세븐세그먼트 (Switching System, 7-Segment)
6. LED 4 개

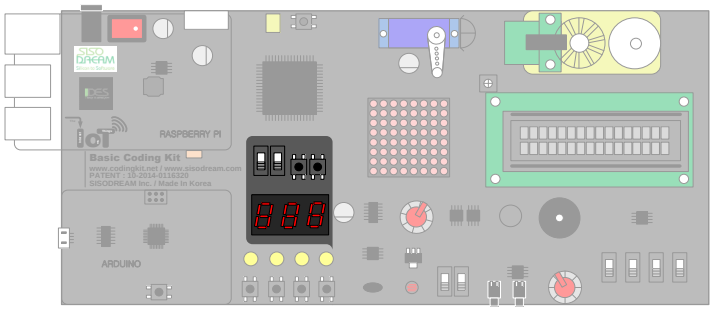
7. 버튼 4 개 (Button)
8. 서보 모터 (Servo Motor)
9. 도트매트릭스 (Dotmatrix)
10. 가변 저항 1 (Variable Resistor 1)
11. 온도 센서 (Temperature Sensor)
12. 밝기 센서 (Light Sensor)
13. DC 모터 (DC Motor) 와 인코더 (Encoder)
14. 캐릭터 LCD (Character LCD)
15. 소리 센서 (Sound Sensor)
16. 부저 (Buzzer)
17. ADC 모드 스위치
18. 적외선 센서 (IR Sensor)
19.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

4-2. 코딩키트 동작시켜보기

코딩키트에는 USB 케이블하고 아답터가 있습니다. USB 케이블은 PC 에, 아답터는 전원 콘센트와 연결해 주세요. 아답터의 콘센트 부분이 아닌 전원 케이블은 코딩키트에 연결해 주세요. 그리고 코딩키트의 전원 스위치를 켜주세요.

전원이 연결되면, 코딩키트의 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 올려 주세요.

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

코딩키트가 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. 어쩌죠?

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 LED 가 켜지고, 전원을 중단하면 LED 가 꺼진단다. 그리고 이 동작을 반복하면 LED가 깜박이게 되는 거지.

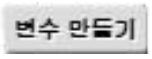
5-1. 코딩으로 LED를 켜보자

코딩으로 LED가 켜지도록 만들어 보겠습니다. 프로젝트 관리 메뉴에서 "새로 만들기"' 메뉴를 선택하시면 기존의 프로젝트는 지워지고 새로운 프로젝트가 만들어집니다.

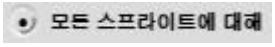
1 단계 : 변수 만들기

① 화면 좌측 상단에 있는 블록 모음들에서 

버튼을 클릭해주세요.

② 

버튼을 누르면, 다음 그림과 같이 변수 이름을 입력하라는 창이 뜹니다. 그러면, LED를 나타낼 변수 이름을 입력합니다. 여기서는 "LED1" 이라고 입력을 하겠습니다.

③ 

를 선택을 하고, 

을 눌러주세요

④ 그러면, 다음 그림과 같이 

변수가 만들어 집니다.

⑤ 변수 창에서 변수 에 0 저장하기 블록을 스크립트로 가져다 놓습니다.

⑥ 스크립트로 가져온 변수 에 0 저장하기 블록의 "변수" 다음의 첫 번째 입력 칸에 있는 아래 화살표를 마우스로 클릭합니다. 그러면, 다음 그림과 같이 변수를 선택 할 수 있게 됩니다. "LED1"을 선택합니다.

⑦ 그리고 두 번째 입력 창에는 숫자 "2" 를 씁니다. 코딩키트의 첫 번째 LED의 핀 번호가 2번이어서 "2" 를 쓴 것입니다. 핀 번호와 관련하여 자세한 내용은 코딩키트 교재의 "아두이노 파트"를 참고해 주십시요.

2 단계 : LED 전등을 켜는 동작 만들기

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

② 제어 창에서 클릭했을 때 블록을 스크립트로 가져와

블록 위에 놓습니다.

③ 화면 좌측 상단에 있는 블록 모음들에서 

버튼을 눌러주세요

④ 

창에 있는 

블록을 스크립트로 가져와 

블록 아래에 놓습니다.

⑤ 화면 좌측 상단에 있는 블록 모음들에서 

버튼을 클릭해주세요.

⑥ 

창에 있는 변수 

를 마우스로 드래그-앤-드롭을 하여 

블록의 첫 번째 빈칸에 가져다 놓습니다.

⑦ 화면 좌측 상단에 있는 블록 모음들에서 연산 버튼을 눌러 주세요.

⑧ 연산 창에 있는 참 블록을 마우스로 드래그-앤-드롭을 하여 set digital pin to 블록의 두 번째 빈칸에 가져다 놓습니다.

** snap4arduino 의 최근 출시 버전에서는 참, 거짓 이 다음과 같이 하나로 바뀌었고 스위치를 누르면 "참", "거짓"이 토글됩니다.*

⑨ set digital pin (

) 블록은 어떤 핀에 디지털 값을 전달할 때 사용합니다. 디지털 값이란 "참" 혹은 "거짓" 두 개의 값을 말합니다. 이 디지털 값들은 "0" 또는 "1", "High" 또는 "Low" 로도 표시합니다. 자세한 사항은 코딩키트 교재 "아두이노 코딩"편을 참고하세요. *(본 교재는 http://www.codingsite.net 사이트에서 무료로 다운로드 받아서 보실 수 있습니다.)*

3 단계 : 아두이노와 연결하여 동작 시켜보기

① 화면 좌측 상단에 있는 블록 모음들에서 Arduino 버튼을 눌러주세요

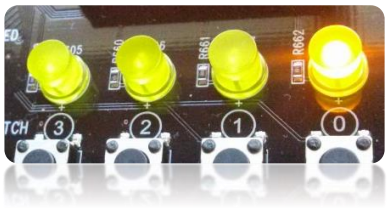
② 코딩키트에 전원과 COM 포트가 연결되어 있는 것을 확인하고, 세븐세그먼트는 A00로 맞춰줍니다.

③ Arduino 창에 있는 Connect Arduino 버튼을 눌러주세요.

④ 다음과 같은 메시지 창이 뜨면, 아두이노와 연결이 잘 된 것입니다.

⑤ 만약, 다음과 같이 아두이노와 연결할 수 없다는 메시지 창이 뜨면, 아두이노의 전원과 포트 연결이 잘 되어 있는지 확인해 주세요.

⑥ 아두이노와 연결이 잘 되었다면, 스냅의 우측 상단에 있는  을 눌러보세요. LED에 불이 켜진 것을 확인 할 수 있습니다.

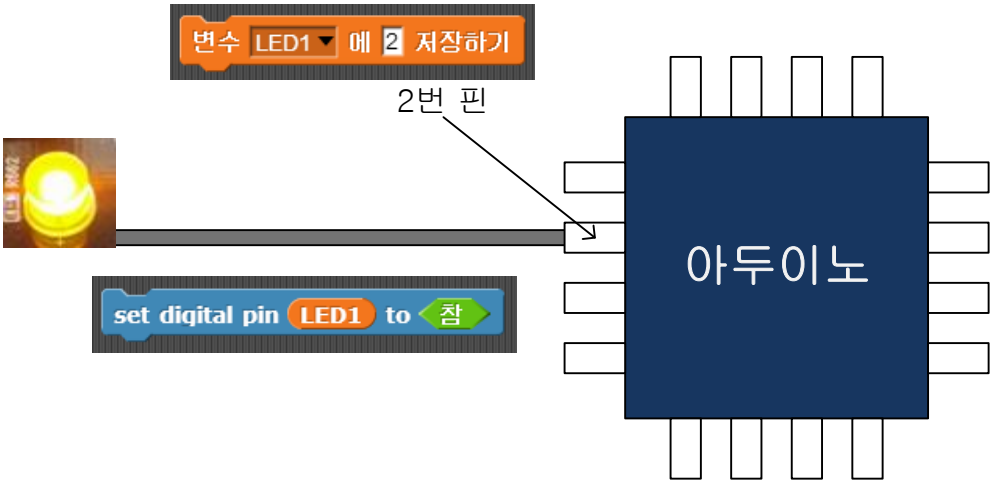


설명 더하기

✓ 변수블록은 이름을 가지고 값을 저장합니다. 위의 예제에서 변수 이름은 LED1 이고 "2" 값을 저장합니다.

✓ 이 "2" 값은 아래 그림과 같이 아두이노의 2번 핀을 의미합니다. 이 2번 핀이 LED 와 연결되어 있습니다. 그래서 LED1 변수에 2를 저장한 것입니다.

을 하면 2번 핀에 참 값을 전달하는 것이고 이 2번 핀에는 LED 가 연결되어 있으므로 LED 에 참 값을 전달해 준 것입니다. LED 에 참 값을 전달해 주면 LED 는 켜집니다.



생각하기 1

아래 그림은 위에서 만들었던 LED를 켜는 동작을 하는 스크립트 입니다. LED 는 계속해서 켜져 있습니다. 이제 LED를 끄는 동작을 하는 스크립트를 만들어 보세요. 힌트를 들이면 "참"과 반대되는 것을 이용하세요.

5-2. LED가 깜박이는 동작을 만들어 보자.

이번에는 LED가 깜박이는 동작을 자동으로 반복 하도록 만들어 볼까요? 4-2에서 만들었던 스크립트를 사용하겠습니다.

1 단계 : 블록 복사하기

① 5-1의 2단계에서 만든 스크립트 

블록에 마우스를 가져가서, 오른쪽 마우스를 클릭한 후 "복사하기"를 선택합니다.

② 그러면, 다음 그림처럼, 같은 블록이 하나 만들어집니다.

③ 복사된 

블록의 두 번째 칸에 있는 

블록을 

블록으로 바꾸어 줍니다. (화면 좌측 상단에 있는 블록 모음들에서 

버튼을 누르면, 

블록을 가져 올 수 있습니다.)

④ 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 

블록 모음을 선택하세요.

⑤ 

블록을 스크립트로 가져와, 

블록과 

블록 다음에 넣어 줍니다.

⑥ ⑤에서 만들어진 동작을  을 한번 눌러서 실행시켜 볼까요? LED가 한번 켜지고 꺼집니다.

2 단계 : 반복 동작 만들기

① 이번에는 ⑤에서 만든 동작을 반복하여, LED가 계속적으로 깜박이는 동작을 만들어 보겠습니다.  블록 모음에서  블록을 가져옵니다. 그리고, LED를 깜박이게 하는 블록들을  블록 안에 넣습니다.

② 스크립트를 다음 그림과 같이 만드셨나요? 그러면,  버튼을 눌러보세요. LED가 깜박이는 동작을 반복하는 것을 확인 하실 수 있습니다.

생각하기 2

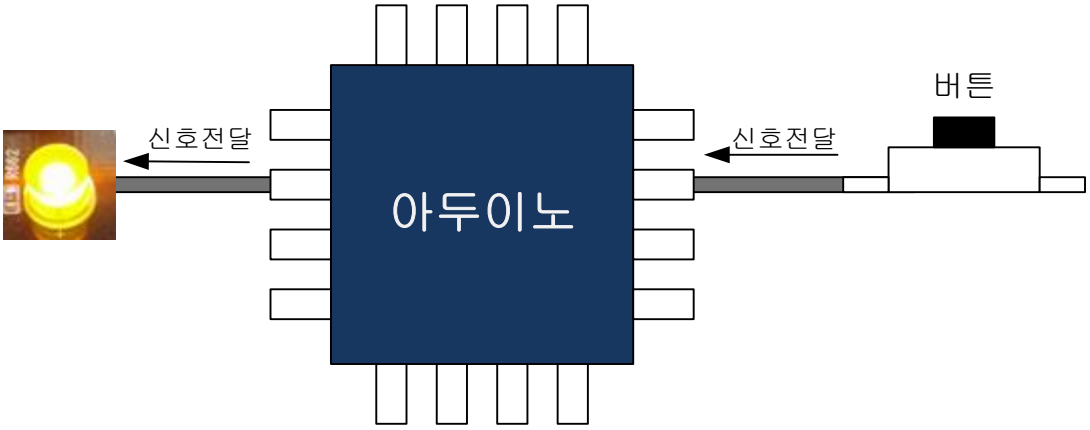
지금까지 LED 한 개를 깜박이도록 해 보았습니다. LED 두 개를 깜박이게 하고 싶다면, 어떻게 하면 좋을까요? (코딩 키트에서 LED의 핀 번호는 우측부터 2, 3, 7, 8번을 사용합니다.)

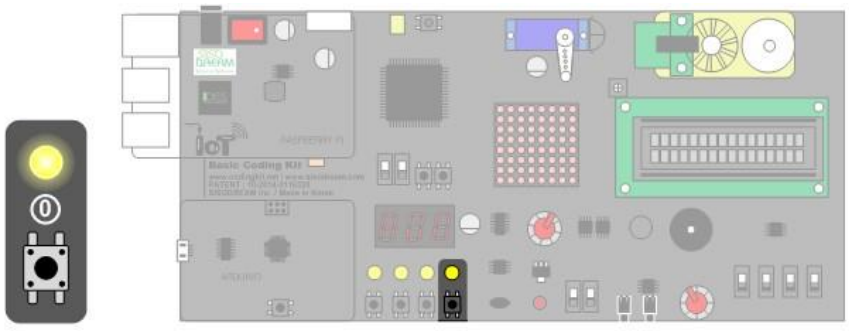
6. 초인종을 만들기

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 눌렸다는 신호가 들어오면 LED 를 켜도록 만드는 거야.





6-1. 버튼이 눌리면 LED 가 켜지도록 해보자.

버튼이 눌렸을 경우에 LED가 켜지는 동작을 만들어 보겠습니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 BUTTON, LED, BTN_PRESSED 변수를 아래 그림과 같이 만듭니다.

② 변수 에 0 저장하기 블록을 변수의 개수만큼 스크립트로 가져와서, BUTTON 변수에는 12, LED변수에는 2, BTN_PRESSED 변수에는 참 값을 설정합니다. 코딩키트의 4개의 버튼 중 오른쪽 첫 번째 버튼의 핀 값이 12이고, 코딩키트의 4개의 LED 중 오른쪽 첫 번째 LED의 핀 값이 2입니다.

2 단계 : 버튼의 상태에 따른 동작 구분하기

① 

블록 모음에서 
블록을, 

블록 모음에서 

블록을, 그리고 

블록 모음에서 

블록과 

블록을 스크립트로 가져옵니다.

③ 다음 그림과 같이 

블록의 빈 칸에 

블록을 넣어 버튼의 상태 값을 가져오게 합니다.

④ 

블록의 빈칸에 

블록과 

블록을 넣어 두 값이 같은지 비교합니다. 두 값이 같다면, 버튼이 눌려진 상태입니다.

⑤ ③의 결과에 따라 동작을 구분하는 블록을 만들기 위해 

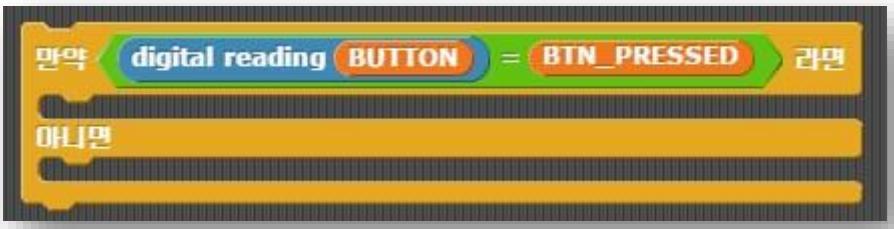
블록 모음에서 

블록을 가져옵니다.

⑥ ④ 에서 만든 

블록을 

블록의 첫 번째 빈칸에 넣어 버튼이 눌렸을 경우와 눌리지 않았을 경우의 동작을 구분하는 블록을 만듭니다.



⑦ 

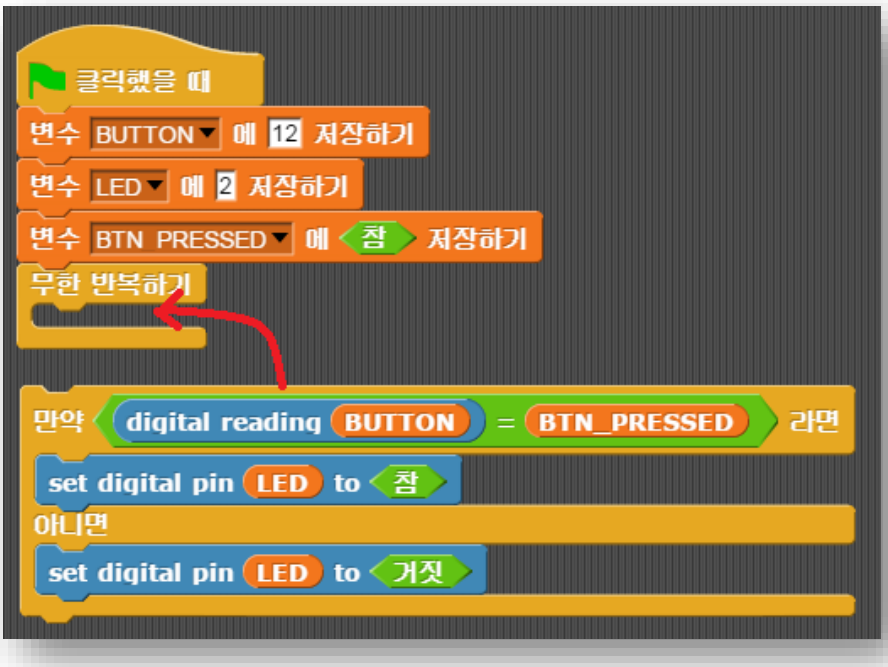
블록 모음에서 

블록 두 개를 스크립트로 가져와 다음 그림과 같이 LED를 켜는 블록과 끄는 블록을 각각 만듭니다.

⑧ 버튼이 눌리면 LED가 켜지고, 버튼이 눌려지지 않으면 LED가 꺼지도록 다음 그림과 같이 ⑦에서 만들어진 두 블록을 ⑥에서 만든 블록에 각각 넣어줍니다.

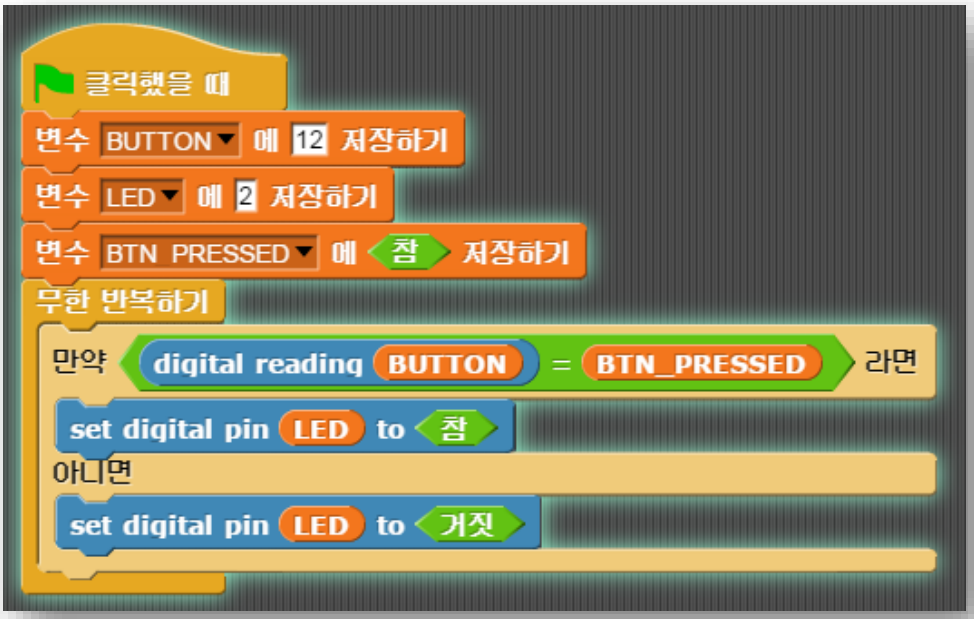
3 단계 : 반복 동작 만들기

① 버튼이 눌렸는지 눌리지 않았는지 반복하여 상태를 확인해야 합니다. 제어 블록에서 무한 반복하기 블록을 가져와 그 안에 2단계에서 만들어진 블록을 넣어 줍니다.



② 다음 그림과 같이 블록이 완성되었다면, 버튼을 눌러 실행시켜 보세요. 버튼이 눌리면 LED가 켜지고 버튼이 눌리지 않은 경우 LED가 꺼지는 결과를 확인할 수

있습니다.

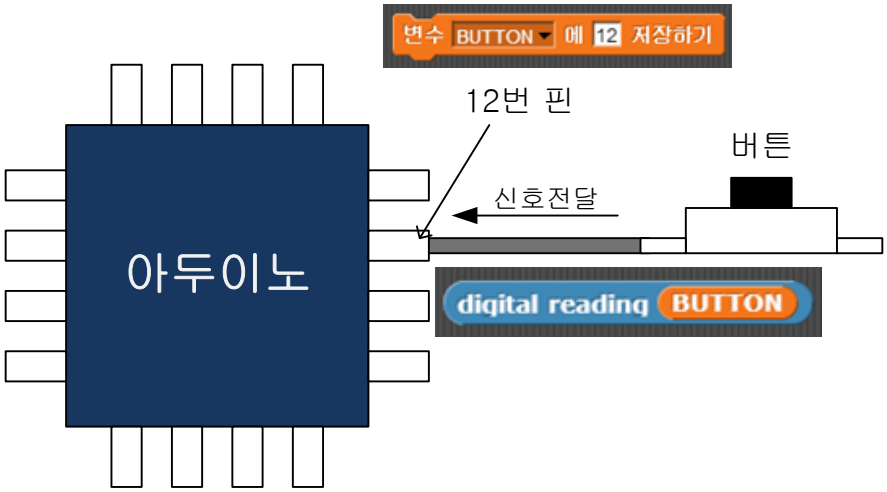


설명 더하기

✓ 

는 다음 그림과 같이 아두이노의 12번 핀에 버튼이 연결되어 있다는 뜻으로 

변수에 "12"를 저장합니다.



✓ 

은 디지털 값을 읽는 것입니다. 디지털 값이란 "참" 또는 "거짓" 둘만 있습니다. 그래서 

은 버튼으로부터 디지털 값을 읽으라는 것입니다. 버튼이 눌려졌을 때 "참" 이 읽히고 버튼이 눌리지 않았을 때 "거짓" 이 읽힙니다. 그래서 

과 같이 하면 버튼이 눌려진 것을 확인할 수 있습니다. 하지만 우리는 다음과 같은 코드를 이용하였는데요. 그렇게 하면 코드가 훨씬 더 보기 편하기 때문입니다.

생각하기 3

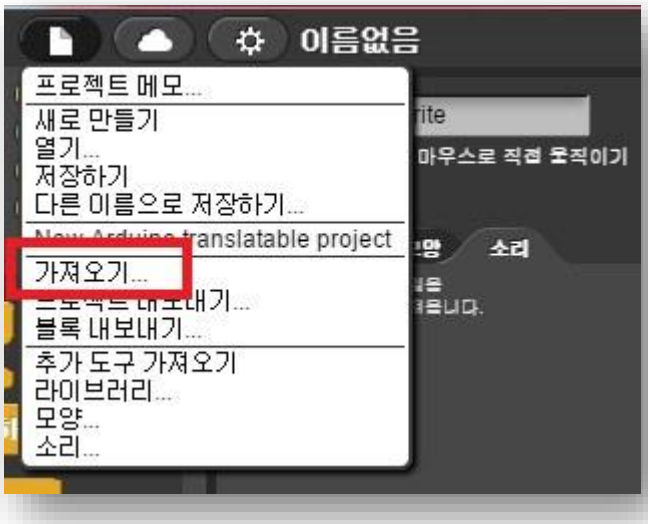
이번에는 버튼이 눌리면 LED 가 꺼지고 눌리지 않으면 켜지는 코딩을 해 보세요.

6-2. 버튼을 누르면 "딩동" 소리가 나게 해보자.

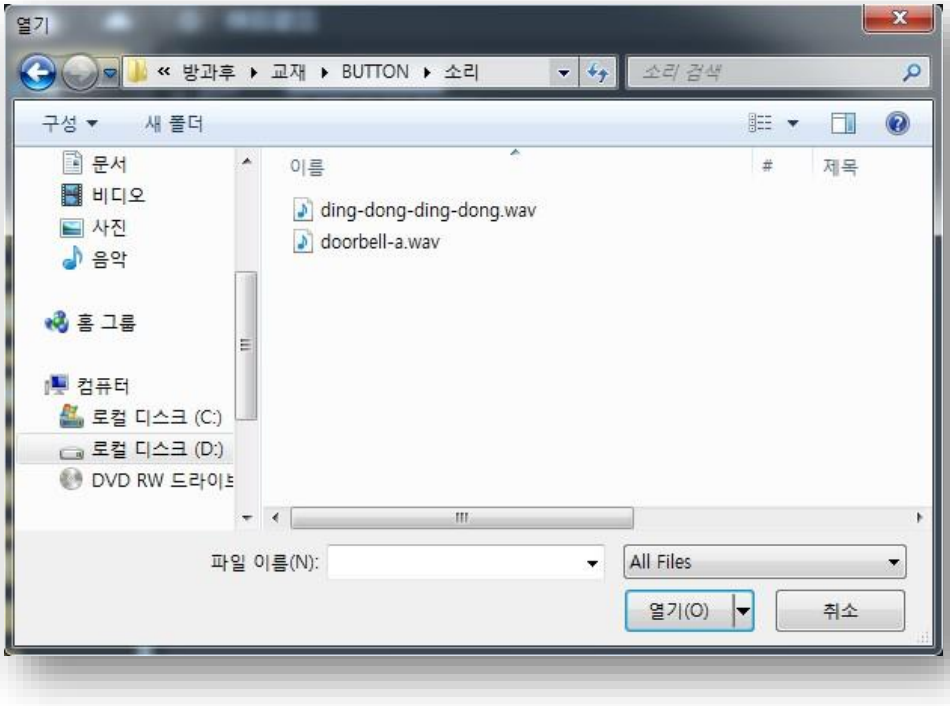
1 단계 : 소리 삽입하기

① 스크립트 화면의 상단을 보면, "스크립트", "모양", "소리" 탭이 있습니다. 여기에서 "소리" 탭을 누릅니다.

② "소리" 탭으로 컴퓨터에 있는 소리 파일을 드래그해서 가져와 소리 파일을 추가할 수 있습니다. 또는 프로그램 상단의  버튼을 눌러, "가져오기..."를 선택하여 소리 파일을 추가하는 방법이 있습니다.



③ "가져오기..." 를 선택하면 다음과 같이 파일을 선택하는 창이 뜹니다. 본서와 함께 제공되는 "doorbell-a.wav" 파일 또는 "ding-dong-ding-dong.wav" 파일을 선택하고 "열기" 버튼을 눌러 소리 파일을 가져옵니다.



④ 다음 그림은 "doorbell-a.wav" 파일과 "ding-dong-ding-dong.wav" 파일 두 개를 가져온 모습입니다.

2 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 BELL , BELL_PRESSED 변수를 만듭니다.

② 스크립트 탭으로 이동합니다.

③ 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져와 BELL 변수에 12 (코딩키트 0번 버튼의 핀 번호)를 저장하고, BELL_PRESSED 에는 참 을 저장합니다.

3 단계 : 버튼이 눌렸는지 확인하기

① 연산 블록 모음에서 = 블록을, Arduino 블록 모음에서 digital reading 블록을, 그리고 변수 블록 모음에서 BELL 블록과 BELL_PRESSED 블록을 스크립트로 가져옵니다.

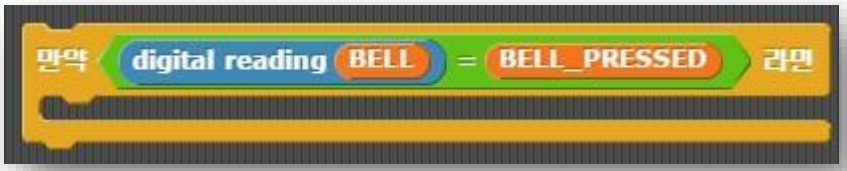
② 그림과 같이 digital reading 블록의 빈 칸에 BELL 블록을 넣어 버튼의 상태 값을 가져오게 합니다

③ = 블록의 빈칸에 digital reading BELL 블록과 BELL_PRESSED 블록을 넣어 두 값이 같은지 비교합니다. 두 값이 같다면, 버튼이 눌려진 상태입니다

④ 제어 블록 모음에서 만약 라면 블록을 가져와, 첫 번째 빈 칸에 ③에서 만든 digital reading BELL = BELL_PRESSED 블록을 넣어 줍니다.

⑤ 그러면, 버튼이 눌렸을 경우, 

블록 안이 실행이 됩니다. 이 부분에 초인종 소리를 넣으면 되겠지요?



4 단계 : 초인종 소리 내기

① 

블록 모음에서 

블록을 스크립트로 가져옵니다.

② 

블록의 첫 번째 빈칸의 아래 화살표를 누르면, 1 단계

에서 삽입된 소리 파일을 선택하여 사용할 수 있습니다.

③ "doorbell-a" 소리를 선택합니다.

④ 그리고, 

블록을 3단계의 ⑤에서 만든 버튼이 눌렸는지 확인하는 블록 안에 넣습니다.

⑤ 아래 그림과 같이 블록을 완성하셨나요? 이제 버튼이 눌리면 doorbell-a 소리가 나게 됩니다.

⑥ 버튼이 눌렸는지 반복적으로 확인을 해야 합니다. 제어 블록 모음에서 무한 반복하기 블록을 스크립트로 가져옵니다. 가져온 무한 반복하기 블록 안에 ⑤의 블록을 삽입합니다.

⑦ 다음 그림과 같이 블록이 완성되었다면, 코딩키트를 연결하고 버튼을 눌러 실행시켜 보세요. 버튼이 눌리면 "딩동~" 하는 소리가 나는지 확인해 보세요.

생각하기 4

소리가 다르게 나는 두 개의 벨을 만들어 보세요. 코딩키트에서 0번 버튼의 핀 번호는 12, 1번 버튼의 핀 번호는 13입니다.

생각하기 5

"안녕하세요" 라는 메시지를 녹음하여 벨 소리로 사용해 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!

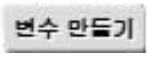
그럼 아빠랑 같이 만들어 볼까? 빠른 캐롤이 나오면 LED가 빠르게 깜박이고 느리게 나오면 느리게 깜박이게 하면 더 근사하겠구나.

7-1. 캐롤이 나오며 깜박이는 크리스마스 전등을 만들어 보자.

1 단계 : 변수 만들기

① 화면 좌측 상단에 있는 블록 모음들에서 

버튼을 클릭해주세요.

② 

버튼을 눌러서, LED1, LED2, LED3, LED4, SPEED 변수를 만듭니다.

③ 

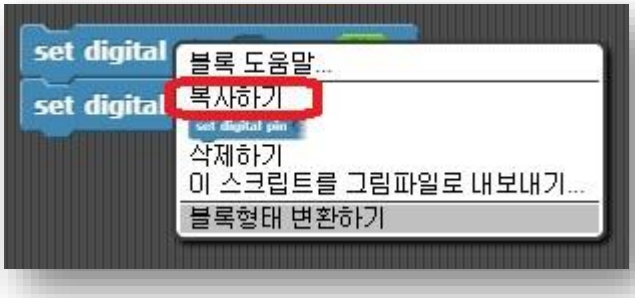
블록 모음에서 

블록을 스크립트로 가져와 변수 LED1, LED2, LED3, LED4에 각각의 LED 핀 값을 설정해 줍니다. (코딩 키트에 있는 4개의 LED의 핀 번호는 우측부터 차례대로 2, 3, 7, 8 번입니다.) 변수 SPEED는 LED가 켜지고 꺼지는 빠르기 입니다. 캐롤의 빠르기에 맞춥니다. 여기서는 0.3으로 하겠습니다.

2 단계 : LED가 번갈아 깜박이는 동작 만들기

① Arduino 블록 모음에서 set digital pin to 블록을 스크립트로 가져와 깜박이는 동작을 하는 블록을 만듭니다. (4-3절 참조)

② 만들어진 블록의 맨 위로에 마우스를 가져가 오른쪽 클릭을 합니다. "복사하기"를 선택하여 복사를 합니다.



③ 복사된 

블록의 첫 번째 빈칸을 LED1, LED2, LED3, LED4 변수로 순차적으로 넣어 줍니다. 그러면, LED1, LED3 이 참 값을 가질 때, LED2, LED4 은 거짓 값을 갖게 됩니다.

④ ③에서 만들어진 블록을 복사하여, LED1, LED3 이 거짓 값을 가질 때, LED2, LED4 이 참 값을 갖는 블록을 만듭니다.

⑤ ③과 ④에서 만들어진 블록을 반복하면, LED가 번갈아 가며 깜박이는 동작이 만들어

집니다. 깜박이는 동작의 시간 간격은 "1 단계 : 변수 만들기"에서 SPEED 변수에 저장해 두었습니다. 제어 블록 모음에서 1 초 기다리기 블록을 가져옵니다. 이 블록에 숫자 값 대신 SPEED 변수를 넣어 줍니다.

⑥ 제어 블록 모음에서 무한 반복하기 블록을 가져옵니다. 그리고, 위 ③ ~ ⑤ 과정에서 만들어진 블록들을 모두 무한 반복하기 블록 안에 넣어 스크립트를 완성합니다.

⑦ 다음 그림과 같이 완성 하셨나요?

⑧ Arduino 블록 모음에서 Connect Arduino 버튼을 눌러 코딩키트와 연결한 후, 버튼을 눌러 실행 시켜보세요. 4개의 LED들이 0.3초 간격으로 두 개씩 번갈아 깜박입니다.

3 단계 : 크리스마스 캐롤 들려주기

① 크리스마스 캐롤을 "소리" 탭으로 가져오도록 하겠습니다. 소리 파일을 가져오는 방법은 앞 절에서 설명하였습니다.

② 소리 블록 모음을 선택합니다. 을(를) 끝까지 소리내기 블록을 스크립트로 가져옵니다.

③ 을(를) 끝까지 소리내기 블록의 빈 칸의 아래 화살표를 클릭하면, ④에서 삽입한 음악 파일이 보입니다. 이 음악 파일을 선택합니다.

④ 제어 창에서 클릭했을 때 블록과 무한 반복하기 블록을 가져옵

니다.

⑤ 

블록을 

블록 안으로 넣습니다.

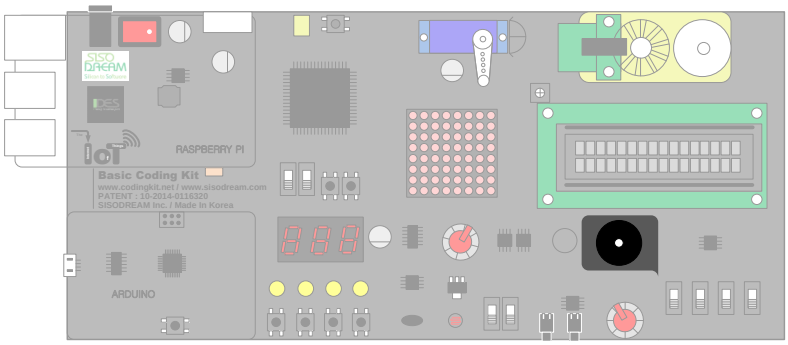
⑥ 다음 그림과 같이 블록이 완성이 되었다면,  를 클릭해 보세요. LED가 깜박이고, 캐롤이 동시에 흘러나오게 됩니다.

생각하기 6

버튼을 누르면 음악이 바뀌고 LED 동작 속도도 같이 변하는 크리스마스 전등을 만들어 보세요. 더 멋진 크리스마스 전등이 만들어 지겠죠? ^^

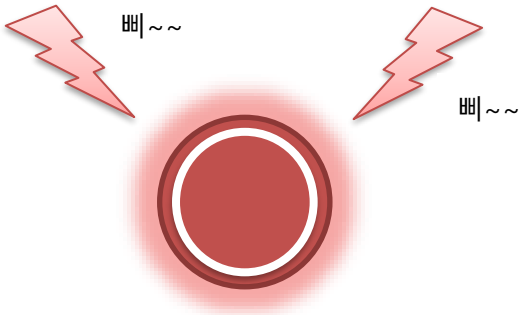
8. 도와주세요! 경보기

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 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 클릭하여 BUZZER 변수를 만듭니다.

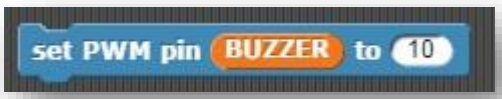
② 변수 블록 모음에 있는 변수 에 0 저장하기 블록을 스크립트로 가져와 BUZZER 에 5 값을 저장하는 블록을 만듭니다. 코딩키트에서 부저는 5번 핀에 연결되어 있습니다.

2 단계 : 부저 소리 내기

① Arduino 블록 모음에서 set PWM pin to 블록을 스크립트로 가져옵니다.

** snap4arduino 의 최근 출시 버전에서는* set PWM pin to *이 다음과 같이 하나로 바뀌었습니다.*

② set PWM pin to 블록의 첫 번째 빈칸에 BUZZER 변수 블록을 넣고 다음 빈칸에 10을 입력합니다. 부저의 소리 크기를 10으로 설정한 것입니다.



③ 부저의 소리는 1 초만 내도록 하려고 합니다. 제어 블록 모음에서 1 초 기다리기 블록을 가져옵니다.

④ 

블록에 마우스를 가져가 오른쪽 클릭을 하여 "복사하기" 를 선택합니다.

⑤ ④에서 복사된 

블록의 부저의 소리 크기를 10 대신 0 으로 설정하도록 수정합니다. 소리 크기가 0 이 되어 부저에서 소리가 나지 않습니다.

⑥ 위 과정에서 완성된 블록들을 다음 그림과 같이 하나로 연결합니다.  버튼을 눌러보세요. "삐~" 소리가 1초간 나고 멈춥니다.

생각하기 7

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

** PWM 신호에 대해서는 교재 중 "아두이노 코딩"편을 참고해 주세요.*

8-2. 경보기 만들기

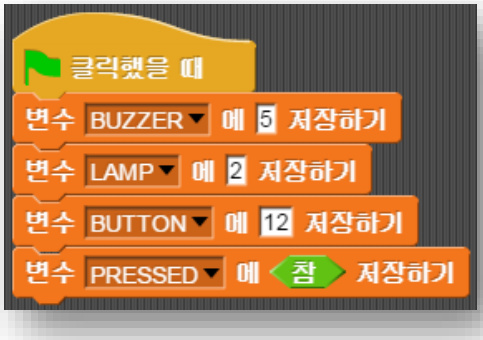
버튼이 눌리면, 부저 소리와 함께 비상등(LED)이 깜박이는 경보기를 만들어 보겠습니다. 지금

까지 배웠던 LED, 버튼, 부저를 모두 사용해 볼 수 있는 예제입니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 클릭하여 BUZZER, LAMP, BUTTON, PRESSED 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 변수의 개수만큼 스크립트로 가져옵니다. 아래 그림과 같이 BUZZER 변수에 5(부저의 핀 번호), LAMP 변수에는 2(LED의 핀 번호), BUTTON 변수에 12(버튼의 핀 번호), PRESSED 변수에는 참 값을 가지도록 설정 합니다.



2 단계 : 경보기 버튼이 눌렸는지 체크하기

① 연산 블록 모음에서 [ ] = [ ] 블록을, Arduino 블록 모음에서 digital reading 블록을, 그리고 변수 블록 모음에서 BUTTON 블록과 PRESSED 블록을 스크립트로 가져옵니다.

② digital reading 블록의 빈 칸에 BUTTON 블록을 넣어 버튼의 상태 값을 가져오는 블록을 만듭니다.

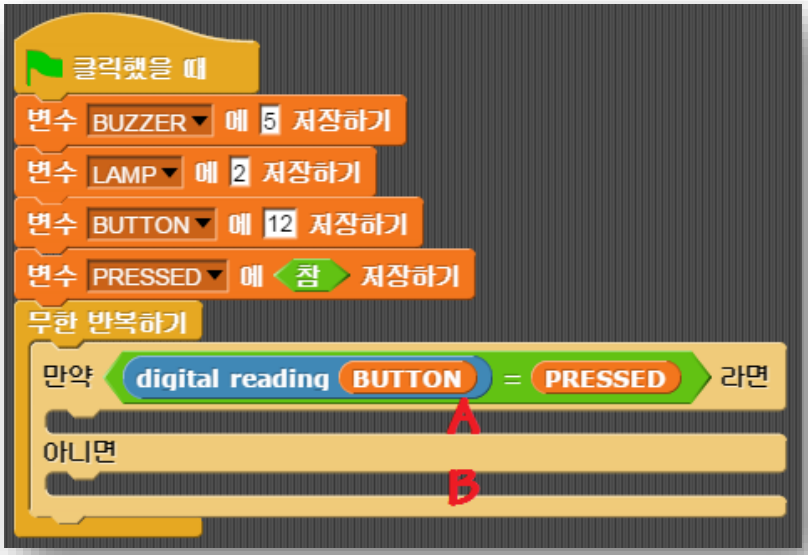
③ [ ] = [ ] 블록의 빈칸에 digital reading BUTTON 블록과 PRESSED 블록을 넣어 두 값이 같은지 비교하는 블록을 만듭니다. 두 값이 같다면, 버튼이 눌려진 상태입니다

④ 제어 블록 모음에서 

블록을 가져와 ③에서 만들어진 블록을 다음 그림과 같이 넣어 줍니다. 버튼이 눌렸는지 확인하는 블록입니다.

⑤ 그리고, 버튼이 눌렸는지 계속 확인을 해야 합니다. 제어 블록 모음에서 무한 반복하기 블록을 가져와 그 안에 버튼이 눌렸는지 확인하는 블록(④에서 만들어진 블록)을 넣어 줍니다.

⑥ 다음 그림과 같이 블록을 만드셨나요? 이제 버튼이 눌려지면 A 영역에 있는 블록이, 버튼이 눌려지지 않았을 경우에는 B 영역의 블록이 동작하는 블록을 만들었습니다.



3 단계 : 경보기 동작 시키기

① Arduino 에서 set PWM pin to 블록 두 개를 스크립트로 가져옵니다. 한 블록은 BUZZER 에 10 값을, 다른 블록에서는 0 값을 넣어 줍니다.

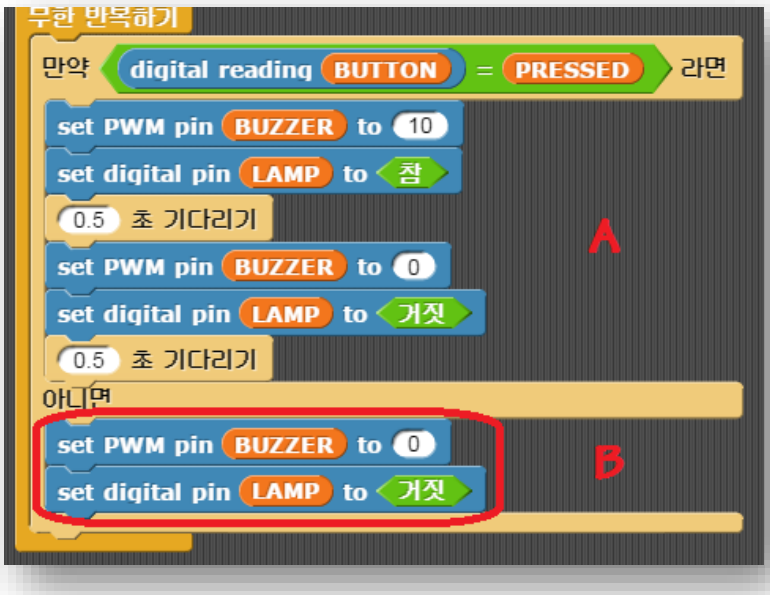
② 이번에는 Arduino 에서 set digital pin to 블록 두 개를 스크립트로 가져와 LAMP 에 참 값을 넣어 주는 블록과 거짓 값을 넣어 주는 블록을 만듭니다.

③ 제어 블록 모음에서 1 초 기다리기 블록을 두 개를 가져와 0.5초 기다리기 블록으로 수정합니다.

④ 0.5초 간격으로 부저 음이 울리면서 LED에 등이 켜지고, 부저음이 멈추면서 LED 등이 꺼지는 동작을 만들기 위해 ⑤, ⑥, ⑦번에서 만든 블록들을 다음의 순서로 나열합니다.

⑤ 위 ④번의 동작은 버튼이 눌렸을 경우에 실행되도록 합니다. 2 단계에서 만든 블록의 A 영역에 위 ④번에서 만든 블록을 넣어 줍니다.

⑥ B 영역에는 버튼이 눌리지 않았을 경우에 실행될 블록을 만들어 넣어 줍니다. 버튼이 눌리지 않았을 경우에는 부저음이 울리지 않고 LED등도 꺼진 상태를 유지하도록 다음 그림과 같이 블록을 만들어 줍니다.



⑦ 이제  버튼을 눌러서 잘 동작하는지 확인해 보세요. 코딩 키트의 첫 번째 버튼을 누르면 경보기가 동작합니다. 그리고 다시 버튼을 누르면 경보기가 멈출 것 입니다.

9. 밝기가 조절되는 무대 조명! 레디~ 액션!!

곰곰이는 디지털과 아날로그라는 말 들어봤니?

아~ 네! TV 광고에서 디지털 TV라는 말을 들어봤어요.

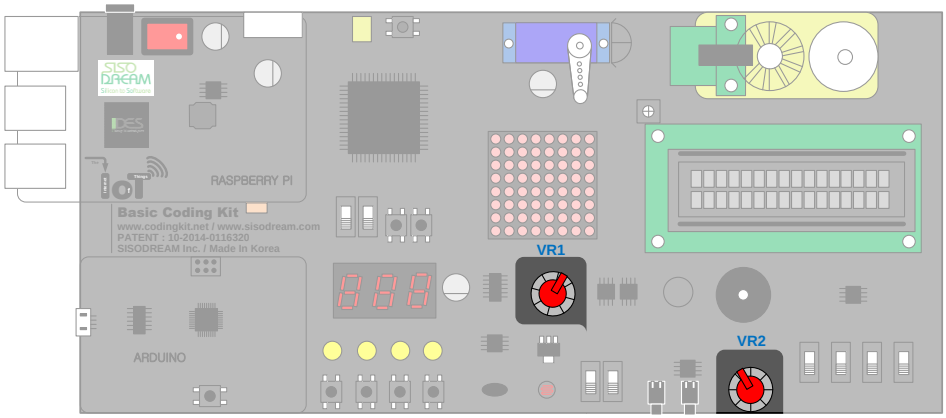
그래! 맞아! TV, 핸드폰 이런 전자 제품들이 모두 디지털 신호를 이용하여 만들어진 거란다. 우리가 지금까지 LED를 켜고 끄기 위해 "참" 또는 "거짓" 신호를 보냈었단다. 그리고 버튼이 눌렸는지에 대한 정보를 가져올 때도 "참" 또는 "거짓" 값으로 받아왔지. 이렇게 "참" 또는 "거짓" 이라는 서로 반대되는 2 가지 신호만을 사용하는 것을 **디지털 신호**라고 한단다. 이 "참" 또는 "거짓" 은 "0" 또는 "1", "ON" 또는 "OFF", "true" 또는 "false" 등으로도 표현한단다.

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.

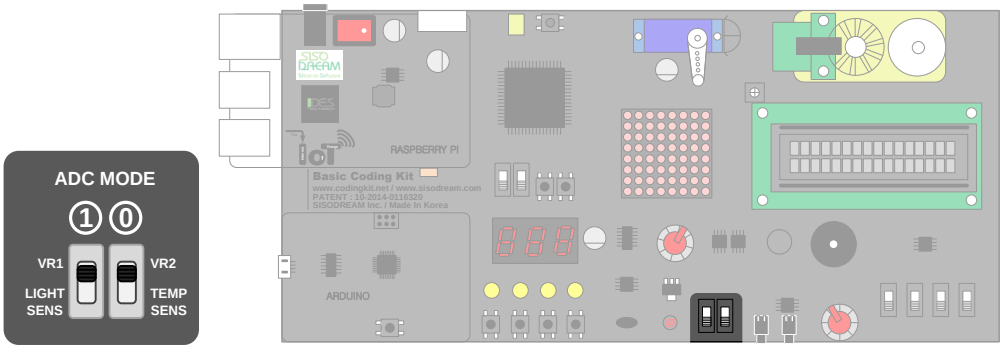
코딩키트에서 아날로그 신호를 다루어 볼 수 있는 가변 저항이라는 장치가 있단다. 이 장치는 아날로그 신호라서 0 에서부터 1023 까지의 값을 전달해 준단다. 배우들이 무대에 오를 때 조명이 서서히 밝아 지면서 등장하지? 가변 저항 장치의 아날로그 신호를 받아서 LED 의 밝기를 조절할 수 있는데, 이렇게 해서 무대 조명과 같은 효과를 낼 수 있단다.

9-1. 가변 저항으로 조명 밝기를 조절해 보자

코딩키트에 가변 저항(Variable Resistor : VR)이라고 하는 다이얼을 돌리는 장치 두 개가(아래 그림에서 VR1, VR2) 있습니다. 이 가변 저항으로 아날로그 신호를 전달해 보겠습니다.



가변 저항을 사용하기 위해 ADC 모드 스위치를 모두 위로 올려주세요.



4장에서 LED에 "참" 또는 "거짓" 값을 전달하여 LED를 켜고 끄는 연습을 해 보았습니다. 이번에는 LED에 가변저항 값을 전달하여 점점 밝아지거나 점점 어두워 지도록 해보겠습니다. 무대에서 조명이 점점 어두워 지거나 밝아지는 효과를 낼 때 사용할 수 있습니다.

1 단계 : 변수 만들기

① 화면 좌측 상단에 있는 블록 모음들에서 

버튼을 클릭해주세요.

변수 만들기 버튼을 눌러서 VR, LED, value 변수를 만듭니다.

② 변수 창에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. VR 변수에는 가변저항 핀 번호인 2를(두 개의 가변 저항 중 VR2를 사용), LED 변수에는 두 번째 LED 핀 번호인 3을, 그리고 value 변수에는 0값을 저장하는 블록을 만듭니다.

2 단계 : 가변 저항 값 받아오기

① 화면 좌측 상단에 있는 블록 모음들에서 Arduino 버튼을 클릭해주세요.

② Arduino 블록 모음에서 analog reading 블록을 스크립트로 가져옵니다. VR, 즉 가변 저항 값을 읽는 블록으로 만들어 줍니다.

③ 

블록을 스크립트로 가져와서 value에 읽어온 가변 저항 값을 저장하는 블록을 만듭니다. 

블록의 첫 번째 입력 칸은 value 변수를, 두번 째 입력 칸에는 

블록을 삽입합니다. 가변 저항의 다이얼을 돌리면 value에는 0~1023 사이의 값이 저장됩니다.

3 단계 : LED의 조도(밝기의 정도) 설정하기

① 

블록 모음에서 

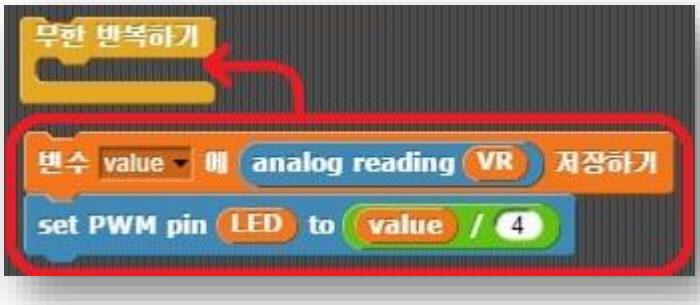
블록을 가져와 첫 번째 입력 칸에 LED 변수 블록을 넣어 줍니다. LED에 밝기의 정도를 알려 줄 것입니다. LED는 0~255 사이의 값을 설정할 수 있습니다.

② 

블록 모음에서 

블록을 스크립트로 가져옵니다. 가변 저항에서 읽어 온 값(0~1023)을 LED의 밝기를 조절하는 값(0~255)으로 변환하기 위해, value 에 저장된 값을 4로 나누어 LED에 전달합니다.

③ 위에서 완성한 블록들을  블록 안에 넣어 줍니다.



④ 다음 그림과 같이 블록이 완성되었다면,  버튼을 눌러 실행시켜 보세요. 가변저항의 다이얼(VR2)을 돌리면, LED의 밝기가 변화되는 것을 확인 하실 수 있습니다.

9-2. 가변 저항으로 LED 조명의 켜지는 개수를 조절해 보자

이번에는 가변 저항의 다이얼을 돌려서 LED 가 켜지는 개수를 조절하는 동작을 만들어 보겠습니다.

1 단계 : 변수 만들기

① 변수 블록 모음을 선택해주세요. 변수 만들기 버튼을 눌러서 VR, value, LED1, LED2, LED3, LED4, cnt 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. 변수 에 0 저장하기 블록을 사용하여 ①에서 만든 각각의 변수에 해당 장치의 핀 값을 설정해 줍니다. VR 변수는 "2", LED1 변수는 "2", LED2 변수는 "3", LED3 변수는 "7", LED4 변수는 "8" 값을 가지도록 만듭니다.

2 단계 : 가변 저항 값으로 켜지는 LED 수 결정하기

① 변수 value 에 가변 저항 값을 저장하는 블록을 만듭니다. (8-2절 2단계 참조)

② 변수 cnt 에 켜지는 LED의 개수(0~4)를 저장하려고 합니다. 0 은 LED 가 하나도 켜지지 않습니다. 켜지는 LED의 개수는 읽어 온 가변 저항 값(0~1023)을 256으로 나누기하여 계산합니다. (8-2절 3단계 참조)

③ 위 두 블록을 

블록 안에 넣습니다. 이제 가변 저항 값이 변할 때 마다, 켜지는 LED의 개수가 바뀌게 됩니다.

3 단계 : 조건에 따라 LED 켜기

① 연산 블록 모음에서 ▯ > ▯ 블록을, Arduino 블록 모음에서 set digital pin to 블록을 스크립트로 가져옵니다.

② ▯ > ▯ 블록의 첫 번째 입력 칸에 cnt 변수를 가져다 넣고, 두 번째 입력 칸에 "0"을 입력합니다.

③ set digital pin to 블록의 첫 번째 입력 칸에는 LED1 변수를, 두 번째 입력 칸에는 참 값을 넣어 줍니다.

④ 제어 블록 모음에서 

블록을 스크립트로 가져옵니다.

⑤ ① ~ ③ 과정에서 만든 

블록과 

블록을 

블록에 다음 그림과 같이 삽입합니다. cnt 가 0보다 클 경우, 첫 번째 LED에 불이 켜지는 블록을 완성하였습니다.

⑥ ①~⑤의 과정을 반복하여, cnt 가 1보다 클 경우 두 번째 LED에 불이 켜지는 블록, cnt 가 2보다 클 경우 세 번째 LED에 불이 켜지는 블록, cnt 가 3보다 클 경우 네 번째 LED에 불이 켜지는 블록도 만듭니다.

⑦ 다음 그림과 같이 LED1, LED2, LED3, LED4 에 거짓을 설정해주는 블록도 만들어 추가합니다. 모든 LED를 끈 상태에서 cnt에 저장된 값을 보고 LED 등을 하나씩 켜주게 됩니다.

⑧ 1~7 과정에서 만든 블록들을 모두 

블록 안에 넣어 줍니다.

⑨ 다음 그림과 같이 블록이 완성되었다면,  버튼을 눌러 실행시켜 보세요 가변저항의 다이얼을 돌리면, 켜지는 LED의 수가 변화되는 것을 확인 하실 수 있습니다.


만약 cnt > 1 라면
set digital pin LED2 to 참
만약 cnt > 2 라면
set digital pin LED3 to 참
만약 cnt > 3 라면
set digital pin LED4 to 참

10. 부릉~ 부릉~ 자동차!

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

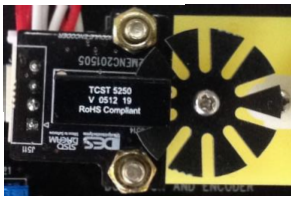
허허허! 아빠도 우리 곰곰이가 운전하는 차를 타면 정말로 뿌듯할 것 같구나. ^^

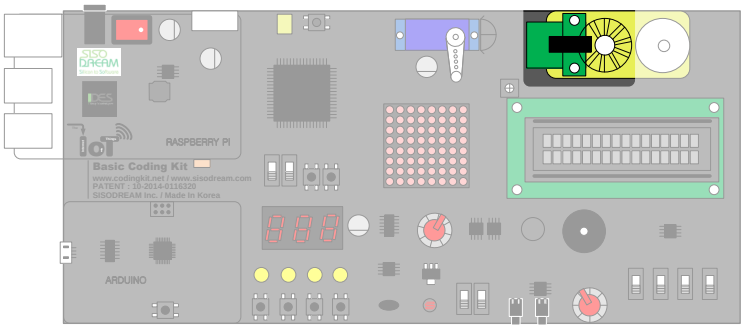
그런데 자동차는 어떻게 움직이는 거예요? 궁금해요.

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 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 MOTER_ON, GO, BACK 변수를 만듭니다.

② 변수 블록 모음에서 변수 ▾ 에 0 저장하기 블록을 스크립트로 가져와 MOTER_ON 변수에는 6, GO 변수에는 10, BACK 변수에는 11을 저장하도록 합니다. DC 모터를 동작시키기 위해 3개의 핀이 필요합니다. DC 모터 활성화를 위한 핀(6번 핀), DC 모터를 시계 방향으로 돌리는 신호를 주는 핀(10번 핀), DC 모터를 시계 반대방향으로 돌리는 신호를 주는 핀(11번 핀)을 사용하게 됩니다.

2 단계 : 모터 돌리기

① Arduino 블록 모음에서 set digital pin to 블록 2개를 스크립트

로 가져옵니다.

② MOTER_ON과 GO에 참 값을 넣어 주는 블록을 만듭니다. DC 모터를 활성화하고 정방향으로 돌도록 합니다.

③ DC 모터를 멈추게 하려면 MOTER_ON과 GO에 거짓 값을 넣어 주는 블록을 추가하면 됩니다.

④ ②와 ③에서 만든 블록 사이에 

블록을 가져다 놓습니다. 이 블록들이 동작이 되면, 1초 동안 모터가 돌고 정지합니다.

⑩ 다음 그림과 같이 블록이 완성되었다면,  버튼을 눌러 실행시켜 보세요. 모터가 시계방향으로 돌고 1초 후 멈춥니다.

생각하기 8

모터를 역방향으로 돌도록 만들어 보세요.

10-2. 엑셀과 브레이크를 만들자.

자동차의 엑셀을 밟으면 자동차의 속도가 빨라지고, 브레이크를 밟으면 자동차의 속도가 느려지면서 멈추지요? 코딩키트의 버튼과 모터를 이용하여 이러한 동작을 만들어 보려고 합니다. 버튼은 엑셀과 브레이크가 되고, 모터는 자동차 바퀴가 됩니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 ENGINE_START, GO, BACK, ACCELERATOR, BRAKE, PRESSED, speed 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다.(변수의 개수 만큼) ENGINE_START 에는 6, GO 에는 10, BACK 에는 11을 저장하는 블록을 만듭니다. (모터 활성화 핀 = 6, 정방향 회전 핀 = 10, 역방향 회전 핀 = 11) 그리고 ACCELERATOR 에 12, BRAKE 에는 13을 저장하는 블록을(코딩키트의 버튼들 중에 오른쪽에서 첫 번째 버튼이 엑셀, 두 번째 버튼이 브레이크) 만들어 줍니다. 엑셀 또는 브레이크 버튼이 눌렸는지 체크하기 위해 PRESSED 변수에 참 을 저장합니다. 그리고, speed 는 모터의 회전 속도 값

이 저장될 변수로, 처음 값으로 0을 저장합니다.

2 단계 : 엑셀과 브레이크 상태 읽어오기

① Arduino 블록 모음에서 set digital pin to 블록을 스크립트로 가져옵니다.

② 위에서 가져온 set digital pin to 블록을 사용하여 ENGINE_START 에 참 값을 넣어 주도록 하여 DC 모터를 회전하게 하는 블록을 만듭니다.

③ 엑셀과 브레이크 역할을 하는 버튼이 눌렸는지 값을 읽어 오기 위해 Arduino

블록 모음에서 

블록을 가져옵니다.

④ 

블록을 사용하여 다음과 같이 

와 

의 값을 읽어 오는 블록을 만듭니다.

⑤ 

블록 모음에서 

블록, 

블록 모음에서 

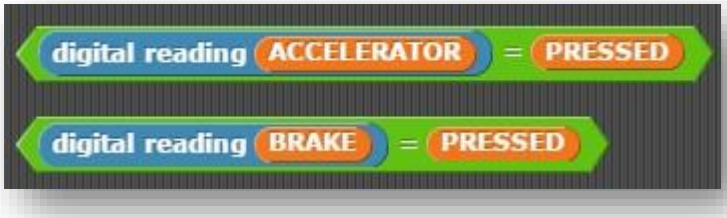
블록을 스크립트로 가져옵니다.

⑥ 

블록과 

블록에서 넘겨주는 값과 

의 값이 같은지 비교하는 블록을 만듭니다.

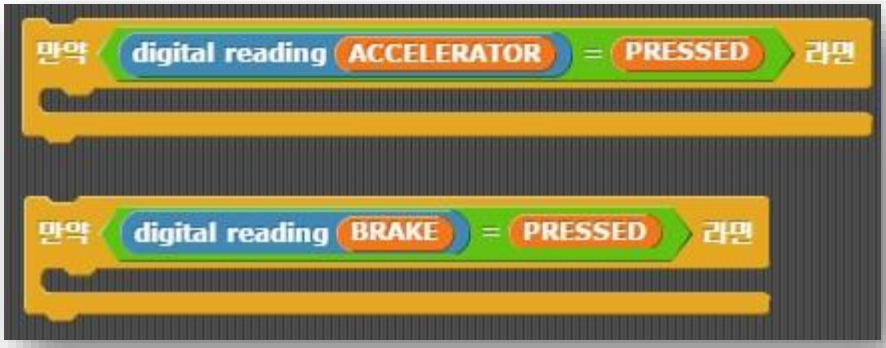


⑦ 제어 블록 모음에서 

블록을 스크립트로 가져옵니다. 위 ⑥에서 만든 블록을 

블록의 첫 번째 입력 칸에 넣습니다.

⑧ 엑셀 또는 브레이크에 해당하는 버튼이 눌렸을 경우에 실행되는 블록을 완성하였습니다.



3 단계 : 속도 조절하기

① 변수 블록 모음에서 

블록을 스크립트로 가

져옵니다. speed의 값이 20씩 커지게 하는 블록과 20씩 작아지게 하는 블록을 만듭니다.

② DC 모터는 0~255 값을 가질 수 있습니다. speed의 값을 최소 0에서 최대 220까지의 값으로 제한하려고 합니다. 제어 블록 모음에서 만약 라면 블록을, 연산 블록 모음에서 < 블록을, 변수 블록 모음에서 speed 블록을 스크립트로 가져옵니다.

③ < 블록의 좌측에는 speed, 우측에는 220을 입력합니다.

④ 위 ④에서 만든 speed < 220 블록을 만약 라면 블록의 조건 입력 칸에 넣어 줍니다.

⑤ ①에서 만든 

블록을 위 ④의 블록 안에 넣어 줍니다. 

의 값이 220보다 작은 경우 20씩 커지는 블록을 완성하였습니다.

⑥ ②~⑤의 과정을 반복하여, 다음과 같이 

의 값이 0보다 큰 경우 20씩 작아지는 블록을 완성합니다. (숫자가 20씩 커지거나 작아지므로 0보다 큰 경우는 20 이상인 경우이다)  블록 대신  블록을 가져와 사용합니다.

⑦ 2단계에서 만든 엑셀 또는 브레이크가 눌렸을 경우에 실행되는 블록 안에, 위 ⑤번 블록과 ⑥번 블록을 각각 다음 그림과 같이 넣어 줍니다. 그러면, 엑셀 버튼을 누르면 

가 20씩 증가되고, 브레이크를 누르면 20씩 감소되는 블록이 만들어 집니다.

⑧ 

블록을 가져와서 위 블록을 그 안에 넣어 줍니다. 버튼이 눌러지는 상태를 매번 확인하여 동작을 하게 됩니다.

⑨ 이제 엑셀 또는 브레이크 버튼을 누르면 속도 값이 증가하거나 감소하게 되었습니다.

그럼 이 속도 값에 따라 모터(자동차 바퀴)의 속도를 변화게 하는 블록을 만듭니다. 우선, 버튼을 입력하는 시간이 있으므로, 대기시간을 설정합니다. 

블록 모음에서 

블록을 스크립트로 가져와 

블록으로 수정하여 위 ⑧번 블록의 

블록 안에 넣어줍니다.

⑩ 다음으로 

블록 모음에서 

블록을 가져와서, 

에 

를 값으로 설정하는 블록을 만듭니다. 그러면, DC 모터가 정방향으로 

만큼의 속도로 돌게 됩니다. 이 블록도 ⑧ 번 블록의 

블록 안에 넣어줍니다.

⑪ 자! 지금까지 만든 블록을 다음 그림과 같이 연결하고,  버튼을 눌러 실행시켜 보세요. 엑셀 버튼을 누르면 모터가 점점 빠르게 돌고, 브레이크 버튼을 누르면 모터의 속도가 점점 느려지면서 나중에는 멈추게 됩니다.

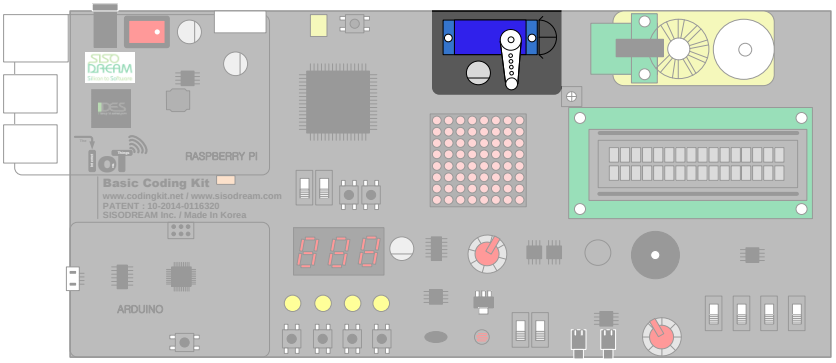

클릭했을 때
변수 ENGINE_START 에 6 저장하기
변수 GO 에 10 저장하기
변수 BACK 에 11 저장하기
변수 ACCELERATOR 에 12 저장하기
변수 BRAKE 에 13 저장하기
변수 PRESSED 에 참 저장하기
변수 speed 에 0 저장하기
set digital pin ENGINE_START to 참
set PWM pin GO to speed
무한 반복하기
만약 digital reading ACCELERATOR = PRESSED 라면
만약 speed < 220 라면
변수 speed 을(를) 20 만큼 바꾸기
만약 digital reading BRAKE = PRESSED 라면
만약 speed > 0 라면
변수 speed 을(를) -20 만큼 바꾸기
0.2 초 기다리기
set PWM pin GO to speed

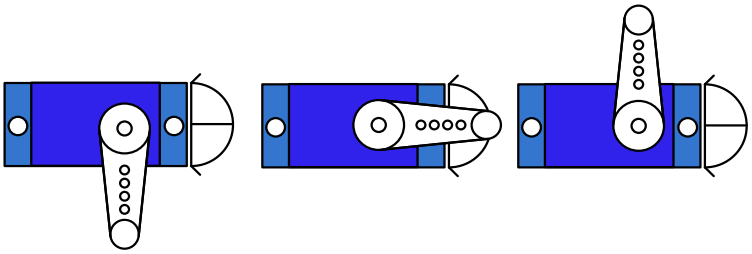
11. 끼익~ 끼익~ 로봇팔!

코딩키트에 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?





0 도 90 도 180 도

11-1.서보 모터를 움직여 보자.

간단하게 서보 모터를 움직이는 동작을 만들어 보겠습니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 SERVO_MOTOR 변수와 angle 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져와 SERVO_MOTOR에는 서보 모터에 연결 된 핀 값인 9를 저장하고, angle에는 각도 값으로 0을 저장합니다.

2 단계 : 서보 모터 움직이기

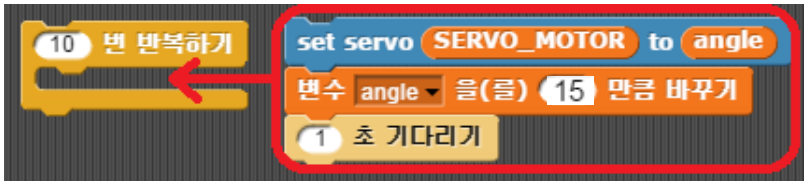
① Arduino 블록 모음에서 set servo to clockwise 블록을 스크립트로 가져옵니다.

② 다음 그림과 같이 SERVO_MOTOR 에 angle 값을 전달하는 블록을 만듭니다. 이 블록은 서보 모터를 angle 각도 만큼 움직이도록 합니다.

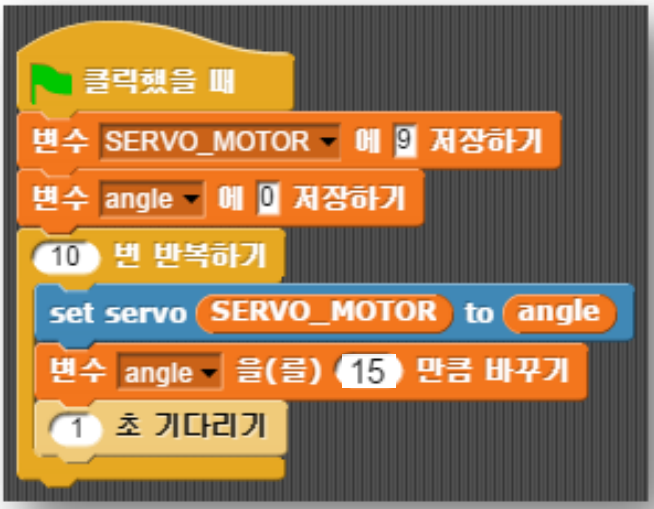
③ 변수 블록 모음에서 변수 을(를) 1 만큼 바꾸기 블록을 스크립트로 가져와서 angle 을 15 만큼 바꾸는 블록을 만듭니다.

④ 시간 간격을 1초을 두고 움직이도록 하기 위해 제어 블록 모음에서 1 초 기다리기 블록을 가져옵니다.

⑤ ② 에서 ④ 의 과정을 반복하면 서보 모터가 0도부터 20도씩 움직일 것입니다. 제어 블록에서 10 번 반복하기 블록을 가져와 그 안에 ②에서 ④에서 만든 블록을 넣어줍니다.



⑥ 다음 그림과 같이 블록들을 연결한 후,  버튼을 눌러 실행시켜 보세요. 서보 모터의 각도가 15도씩 변하면서 움직이는 모습을 확인하실 수 있습니다.



** 서보 모터가 어떤 각도에서 정확히 멈추어 있지 않고 약간씩 흔들일 수도 있습니다. 이것은 Sanp4Arduino 에서 주는 신호가 약간씩 지연되어 전달되기 때문입니다.*

** 서보 모터는 0도에서 180도까지 동작합니다. 이 이외의 범위에서 동작할 경우 서보 모터에 무리가 가니 주의 부탁드립니다. 180도에서 간혹 흔들릴 수 있으니 가급적 170도 정도까지 사용을 권장합니다.*

생각하기 9

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

11-2.원격조정 로봇 팔을 만들자

8장에서 배운 가변 저항으로 서보 모터의 각도를 조절하는 동작을 만들어 보겠습니다. 이번에도 VR2 가변 저항을 사용하겠습니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 VR, SERVO_MOTOR, value, angle 를 만듭니다.

② 변수 블록 모음에서 변수 VR 에 0 저장하기 블록을 스크립트로 가져옵니다. VR 변수에는 가변 저항에 연결된 핀 값 2를, SERVO_MOTOR 변수에는 서보 모터에 연결된 핀 값 9를 저장합니다.

2 단계 : 가변 저항으로 서보 모터 조정하기

① 

블록 모음에서 

블록을, 

블록 모음에서 

블록을 가져옵니다. 이 두 블록을 사용하여 아래 그림과 같이 

에 가변 저항 값을 저장하는 블록을 만듭니다.

② ① 의 블록을 다음 그림과 같이 가변 저항 값을 계속적으로 읽어올 수 있도록 

블록 안에 넣습니다. (8-2절의 "2단계 : 가변 저항 값 받아오기"를 참조)

③ 가변 저항 값은 0~1023값을 가지고, 서보 모터는 0도에서 180도까지 움직입니다. 가변 저항을 6로 나누어 서보 모터의 각도를 계산합니다. 

를 6로 나누어 

에 저장하는 블록을 만듭니다. (8-3절의 "2 단계 : 가변 저항 값으로 켜지는 LED 수 결정하기"를 참조)

④ Arduino 블록 모음에서 set servo to clockwise 블록을 스크립트로 가져옵니다. SERVO_MOTOR에 angle을 설정하는 블록을 만듭니다.

⑤ 위의 ②와 ③에서 만든 블록을 무한 반복하기 블록 안에 같이 넣어 줍니다. 다음 그림과 같이 블록들을 연결한 후, 버튼을 눌러 실행시켜 보세요. 가변 저항을 돌리면, 서보 모터도 돌아가는 것을 확인하실 수 있습니다.

12. 인공지능 전등 만들기

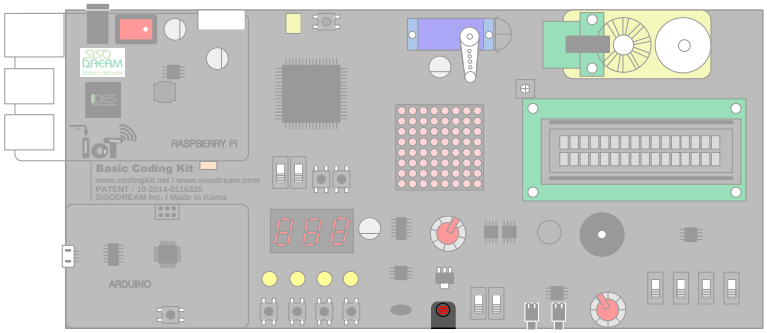
곰곰이와 아빠는 저녁에 마을 길을 산책하고 있었어요. 날이 어둑해져서 가로등불들이 하나, 둘 씩 켜지기 시작했습니다.

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.

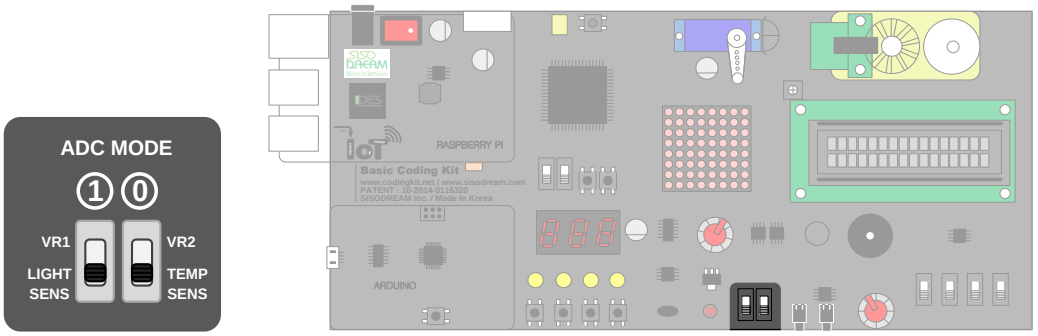
신기해요! 아빠! 인공 지능 전등이네요!

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



12-1.신기한 밝기 센서

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



1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 LIGHT와 value 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. LIGHT 변수에는 코딩키트의 밝기 센서와 연결된 핀 번호 3을 저장하고, value 는 밝기 값을 저장하기 위한 변수인데, 초기값으로 0을 저장합니다.

2 단계 : 밝기 센서 값 읽어 오기

① 변수 블록 모음에서 변수 에 0 저장하기 블록을, Arduino 블록 모음에서 analog reading 블록을 가져옵니다. 아래 그림과 같이 value 에 밝기 센서 값을 읽어와서 저장하는 블록을 만듭니다.

② ① 의 블록을 사용하여 밝기 센서의 값을 계속적으로 읽어올 수 있도록 무한 반복하기 블록 안에 넣습니다

③  버튼을 눌러 실행시켜 보세요. 밝기 센서의 값은 무대 창에 value 값으로 나타납니다. 밝기 센서에 검은색 박스로 덮어서 어둡게 하거나 전등을 비추어 밝게 하면서 밝기 값이 변화를 살펴보세요.

12-2.인공지능 전등을 만들자

어두워지면 LED 등이 켜지고 밝아지면 LED 등이 꺼지는 전등을 만들어 보려고 합니다. 위 12-1절에서 만들어진 스크립트에 블록을 추가하여 만들어 보겠습니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 LED 변수를 추가합니다.

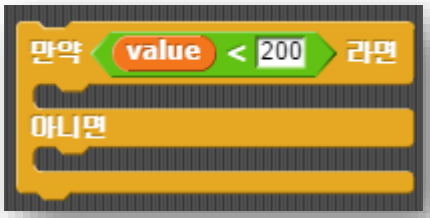
② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. LED 변수에 LED와 연결된 핀 번호 3을 저장하는 블록을 만들어 추가합니다.

2 단계 : 밝기 센서 값에 따라 전등 켜고 끄기

① 연산 블록 모음에서 < 블록과 변수 블록 모음에서 value 변수를 스크립트로 가져와서 밝기 센서의 값 value 가 200보다 작은지 묻는 블록을 만듭니다. (주변 밝기에 따라 밝기 센서 비교 값은 알맞게 조정해주세요)

② 제어 블록 모음에서 

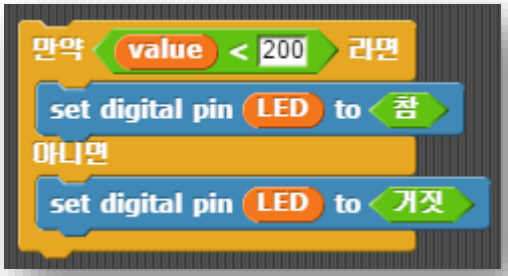
블록을 가져옵니다. 위 ①에서 만든 블록을 첫 번째 입력 칸에 넣어 줍니다. 밝기 값이 200 보다 작은 경우와 그렇지 않은 경우의 동작을 다르게 할 것입니다.



③ Arduino 블록 모음에서 

블록을 가져와 LED 에 참 값을 주는 블록과, 거짓 값을 주는 블록을 각각 만듭니다. LED 에 참 값을 주는 블록을 동작시키면 LED 등이 켜지고, 거짓 값을 주는 블록을 동작 시키면 LED 등이 꺼집니다.

④ 위 ③에서 만든 블록을 ②에서 만든 블록에 다음 그림과 같이 넣어 줍니다. 밝기 센서의 값 value 가 200보다 작을 경우(어두운 경우) LED 등이 켜지고, 200보다 큰 경우(밝은 경우) LED 등이 꺼지는 동작이 완성되었습니다. 조금 더 어두울 때 LED 등이 켜지게 하고 싶으면 밝기 기준 값을 내리면 됩니다.



⑤ ④에서 완성된 블록을 

블록 안에 넣어 주세요. 다음 그림과 같이 블록이 완성되었나요?  버튼을 눌러 실행시켜 보세요. 손가락으로 밝기 센서를 가려서 어둡게 하면 LED가 켜지고 다시 손가락을 떼면 LED가 꺼지는 동작을 확인해 보세요.

생각하기 10

어두운 정도에 따라 LED의 밝기가 조절되는 동작을 만들어 보세요. 날이 어두어 지면

LED가 점점 밝아지고, 날이 밝아지면 LED가 점점 어두워지다 꺼지게 해보세요. 9-1절에서 조명 밝기를 조절하는 동작을 만들어 보았지요? 이 동작을 응용해서 만들어 보세요.

생각하기 11

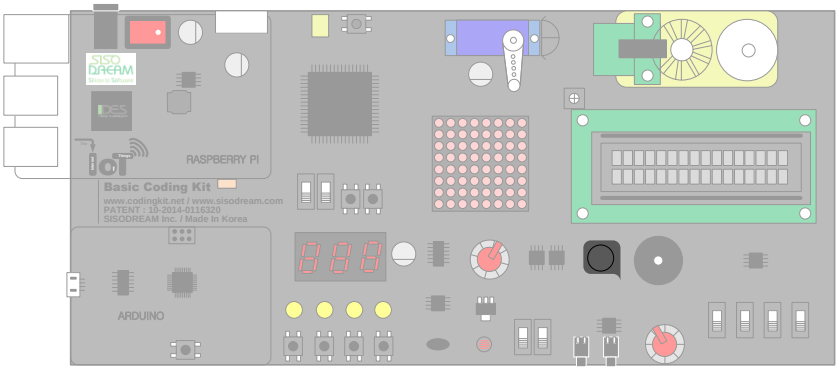
어두운 정도에 따라 LED 조명이 켜지는 개수를 조절해 볼 수도 있습니다. 9-2절을 응용해서 만들어 보세요.

13. 잠자는 고양이 깨우기

코딩키트에는 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.쉿 조용히.. 소리 센서

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 SOUND 와 value 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. SOUND 변수에는 코딩키트의 소리 센서와 연결된 핀 번호 1을 저장하고, value 는 소리 값을 저장하기 위한 변수인데, 초기값으로 0을 저장합니다.

2 단계 : 밝기 센서 값 읽어 오기

① 변수 블록 모음에서 변수 에 0 저장하기 블록을, Arduino 블록 모음에서 analog reading 블록을 가져옵니다. 아래 그림과 같이 value 에 소리 센서 값을 읽어와서 저장하는 블록을 만듭니다.

② 소리 센서의 값을 계속적으로 읽어올 수 있도록 ①의 블록을 

블록 안에 넣습니다

③  버튼을 눌러 실행시켜 보세요. 소리 센서의 값은 무대 창에 value 값으로 나타납니다. 소리 센서 가까이에서 큰 소리를 내거나 소리 센서를 톡톡 두드려서 value 값의 변화를 살펴보세요.

설명 더하기

소리 센서의 값은 순간적으로 높아졌다가 낮아집니다. 그래서 위의 코드를 실행하면서 손으로 소리 센서를 톡톡 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 아두이노 파트를 참고해 주세요.

생각하기 12

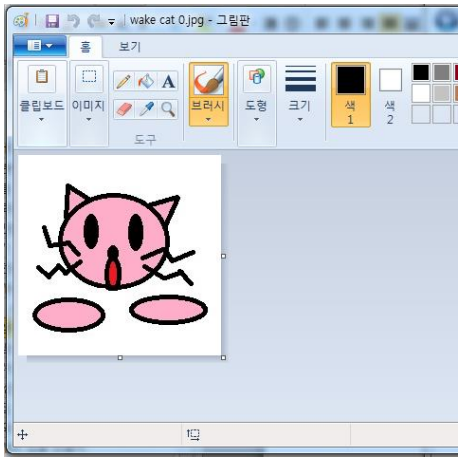
소리 센서를 사용해서 소리가 커지면 켜지는 LED 등의 수가 늘어나게 되는 소음 측정기를 만들어 보세요.

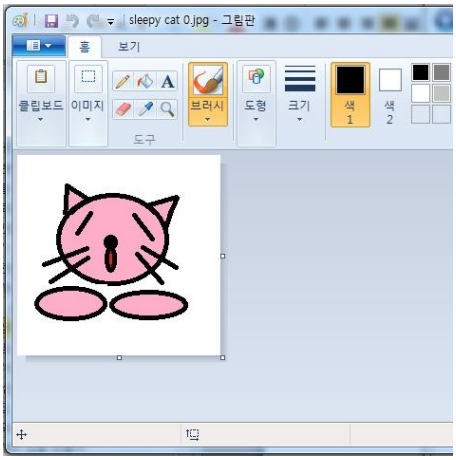
13-2.잠자는 고양이를 깨우기 놀이를 해보자

잠자는 고양이를 소리를 질러 깨우는 게임을 만들어 보려고 합니다. 위 12.1절에서 만들어진 스크립트에 블록을 추가하여 만들어 보겠습니다.

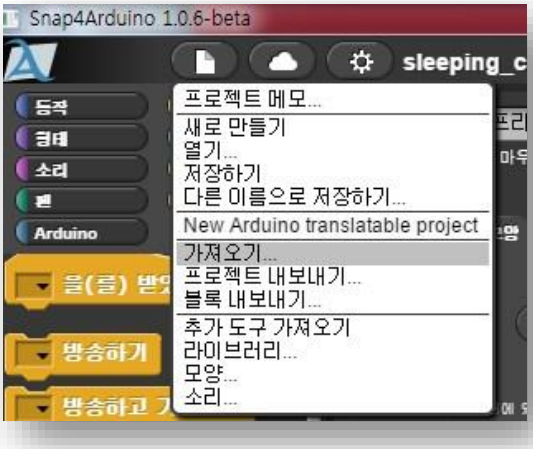
1 단계 : 그림 삽입하기

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. 그림 파일은 스크래치에서 사용했던 것을 그대로 사용하시면 됩니다.

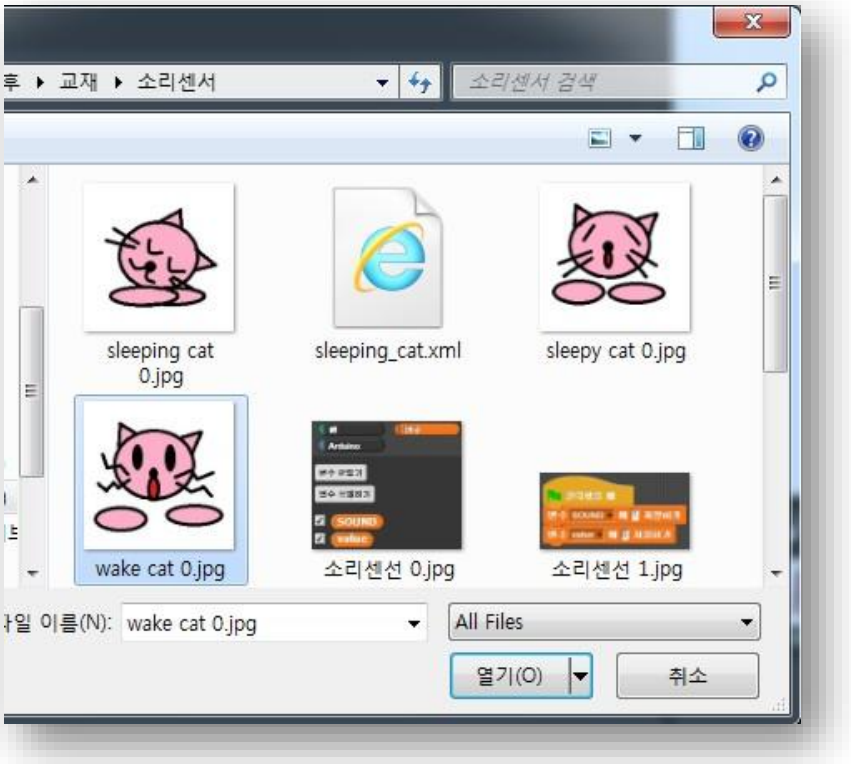




② Snap4Arduino 에서  버튼을 누른 후, "가져오기..." 메뉴를 선택합니다.



③ ①에서 그린 깨어있는 고양이"wake cat 0.jpg" 그림을 선택하고 "열기" 버튼을 누릅니다.



④ ②, ③ 과정을 반복하여 잠자는 고양이("sleepy cat 0.jpg") 그림도 가져옵니다.

⑤ "모양" 탭으로 가시면 그림이 삽입된 상태를 확인하실 수 있습니다.

2 단계 : 소리 센서의 값에 따라 그림 바꾸기

①  블록 모음에서  변수를,  블록 모음에서  블록을 스크립트로 가져와서, 소리 센서의 값이 900 보다 큰가를 비교하는 블록을 만듭니다. 비교 값은 바꾸셔도 됩니다.

②  블록 모음에서  블록을 스크립트로 가져옵니다. 위 ①에서 만든 블록을 첫 번째 입력 칸에 넣어 줍니다.

③ 블록 모음에서 블록을 스크립트로 가져옵니다.

④ 블록의 입력 칸의 아래 화살표를 누르면, 삽입된 그림들을 선택할 수 가 있습니다.

⑤ 다음 그림과 같이 잠자는 고양이 그림으로 바꾸는 블록과 깨어 있는 고양이 그림으로 바꾸는 블록 두개를 만듭니다.

⑥ 소리 센서의 값이 900보다 크면, 깨어 있는 고양이 그림으로 바꾸는 블록이 동작하고, 아니면 잠자는 고양이 그림으로 바꾸는 블록이 동작하도록 만듭니다.

⑦ 소리 센서의 값이 900을 넘어 가면 깨어 있는 고양이 그림이 되지만, 소리 센서 값이 내려가면 잠자는 고양이 그림으로 바로 바뀌게 됩니다. 제어 블록의 1 초 기다리기 블록을 사용하여 깨어 있는 고양이 그림이 나오는 경우 1초 동안 기다리도록 합니다.

⑧ 이제 13-1절 에서 만든 소리 센서의 값을 가져오는 블록에 위 ⑦의 블록을 넣어 주

기만 하면 됩니다. 

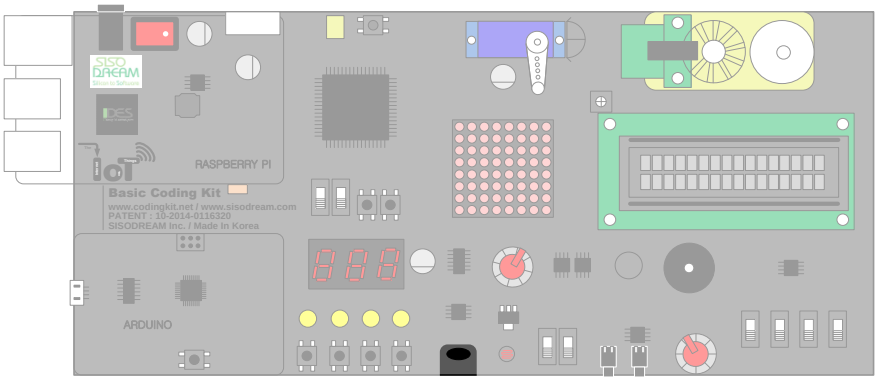
블록 안으로 위 ⑦에서 만든 블록을 다음 그림과 같이 넣습니다.

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 확인 하실 수 있습니다. 그리고 다시 잠자는 고양이 그림으로 바뀝니다. 귀가 아플 정도의 소리가 계속 들리지 않는다면요. ^^

14. 똑똑한 우리집 냉난방 시스템

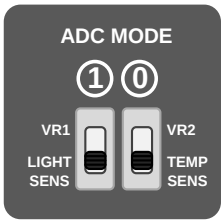
아~ 너무 더워요! 아빠! 이럴 땐 자동으로 에어컨이 켜졌으면 좋겠어요.

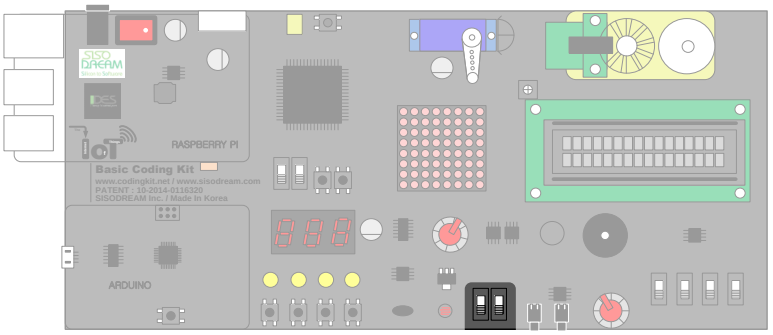
자동으로 더우면 에어컨이 켜지고 추우면 난방이 된다? 좋은 생각이구나! 그러고 보니 코딩키트에 온도 센서가 있단다. 온도 센서를 이용하면 가능할 것 같구나.



14-1.온도를 알려드려요. 온도 센서

코딩키트의 온도 센서를 이용하여 온도 값을 가져오는 블록을 만들어 보겠습니다. 온도 센서를 이용하기 위해서는 코딩키트의 ADC 모드의 ⓪번 스위치를 아래로 내려야 합니다.





1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 TEMPERATURE 변수와 value 변수를 만듭니다.

② 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. TEMPERATURE 변수에 코딩키트의 온도 센서와 연결된 핀 번호 2를 저장합니다.

2 단계 : 온도 센서 값 읽어 오기

① 변수 블록 모음에서 변수 에 0 저장하기 블록을, Arduino 블록 모음에서 analog reading 블록을 가져옵니다. 아래 그림과 같이 온도 센서 값을 읽어와서 value 에 저장하는 블록을 만듭니다.

② 온도 센서의 값을 계속적으로 읽어올 수 있도록 ①의 블록을 

블록 안에 넣습니다

③  버튼을 눌러 실행시켜 보세요. 온도 센서의 값은 무대 창에 value 값으로 나타납니다. 온도 센서를 손가락으로 쥐었다가 잠시 후 놓아 보세요. 온도 센서의 값 value가 올라갔다가 다시 내려가는 것을 확인 하실 수 있습니다.

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. 이번에도 위 14-1절에서 만들어진 스크립트에 블록을 추가하여 만들어 보겠습니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 STOVE 변수, FAN 변수, FAN_ON 변수를 추가합니다.

③ 변수 블록 모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. STOVE 변수에 코딩키트의 ① 번 LED와 연결된 핀 번호 2를 저장하고, FAN 변수에는 DC 모터 활성화를 시켜주는 핀 번호인 6을, FAN_ON 변수에는 DC 모터를 돌리는 신호를 주는 핀의 번호인 10을 입력하는 블록을 만듭니다.

2 단계 : 온도에 따라 동작시키기

① 연산 블록 모음에서  블록을 가져와 온도 센서의 값이 500보다 큰지 판단하는 블록을 만듭니다. 저는 500을 기준으로 하였으나 다른 값으로 하셔도 무방합니다. (위 14-1절에서 측정된 온도 센서의 값을 참고하여 기준 값을 정하면 됩니다.)

② 제어 블록 모음에서  블록을 가져와서 위 ①의 블록을 첫 번째 입력 칸에 넣어 줍니다. value 의 값이 500보다 큰 경우와 작은 경우의 동작을 구분하는 블록을 만들었습니다.

③ Arduino 블록에서 set digital pin to 블록을 가져와서, value 가 500보다 큰 경우 동작하는 블록을 만듭니다. 즉 온도가 높아서 선풍기를 돌리는 동작입니다. 난로는 꺼지고 선풍기가 돌아가야겠지요? STOVE 에는 거짓 을 넣어 주고, FAN 과 FAN_ON 에는 참 을 넣어 줍니다.

④ 이번에는 value 가 500보다 작은 경우 동작하는 블록을 만듭니다. 즉 온도가 낮아서 선풍기는 끄고, 난로를 켜는 동작입니다. STOVE 에는 참 을 넣어 주고, FAN 과 FAN_ON 에는 거짓 을 넣어 줍니다.

⑤ 다음 그림과 같이 위 ③과 ④에서 만든 블록을 ②의 만든 온도 값에 따라 동작을 구분하는 블록에 넣어 줍니다. value 가 500보다 크면 선풍기(모터)가 500보다 작으면 난로(LED)가 켜지게 되는 블록이 완성 되었습니다.

⑥ 지금까지 만든 블록을 14-1절에서 만든 온도 센서의 값을 가져오는 블록에 추가하여 줍니다.

⑦ 이제  버튼을 눌러 실행시켜 보세요. LED 등에 불이 들어왔나요? 온도 센서를 손으로 쥐어서 따뜻하게 해보세요. 온도 센서의 값이 500 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 500보다 높게 설정하여 동작시켜 보세요.

생각하기 13

실내 온도가 높아 지면 선풍기가 점점 세게 돌고, 실내 온도가 낮아지면 난로의 온도가 점점 올라가도록 만들어 보세요. 난로는 밝기 조절이 가능한 ①번 LED (핀 번호 3)를 사용하시면 됩니다. 밝기 조절 방법은 9-1절을 참조하시면 됩니다.

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!

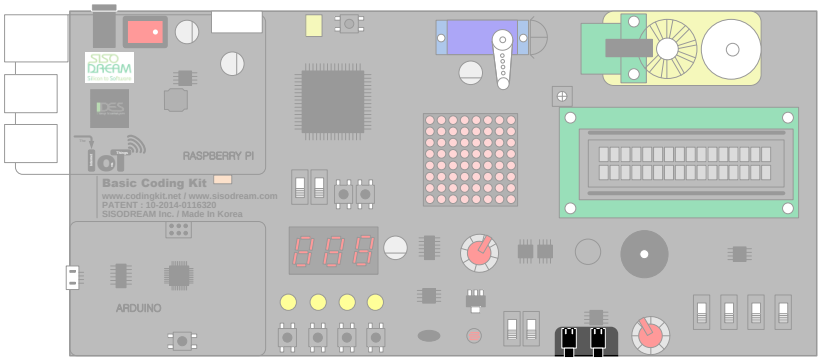
허허허! 정말 놀란 표정이구나! 지난번에 놀이 동산에 갔을 때, 곰곰이도 도깨비집에 들어갔다가 무서워서 혼났었지?

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

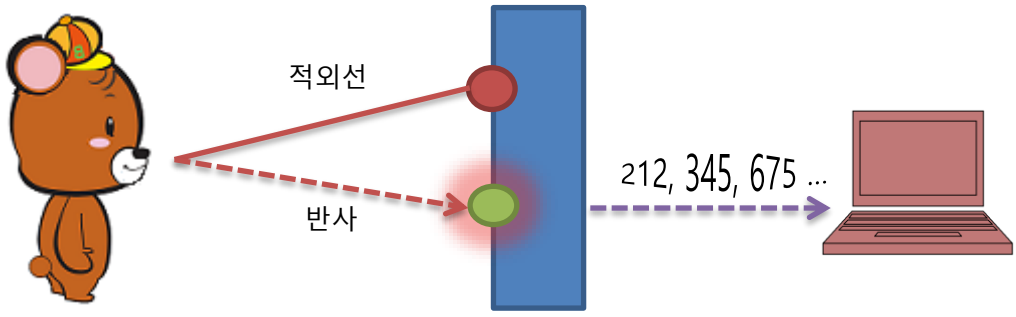
허허허! 실제로 도깨비집에 귀신이 사는 건 아니란다. 사람이 다가오는 것을 알아내는 센서가 있는데, 이 센서를 이용하여 사람이 다가올 때 인형들이 튀어 나오게 만드는 것이란다. 코딩키트에는 적외선 센서가 있는데, 이것을 이용하면, 사람이 다가오는 것을 알 수 있단다. 그럼 코딩키트의 적외선 센서를 이용하여 도깨비 상자를 만들어 볼까?

15-1.꼼짝마! 적외선 센서

코딩키트에는 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. 이렇게 반사된 신호 값을 가져오는 동작을 만들어 보겠습니다.



1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 IR , IR_LED , value 변수를 만듭니다.

② 제어 블록 모음에서 클릭했을 때 블록을 스크립트로 가져옵니다. 그리고, 변수 블록 모음에서 변수 에 0 저장하기 블록을 가져와서, IR 에는 수광부 핀 번호 0을, IR_LED 에는 발광부 핀 번호 4를 저장합니다. value 에는 적외선 센서의 전기 신호 값을 저장할 것입니다. 초기값으로 0을 저장합니다.

③ Arduino 블록 모음에서 set digital pin to 블록을 가져와서 IR_LED 에 참 값을 넣어 주는 블록을 만듭니다. 적외선 센서의 발광부가 동작을 하게 됩니다.

④ 다음으로 

블록 모음에서 

블록을 가져와서 

의 값을 읽어 오는 블록을 만듭니다. 이 블록은 수광부에서 감지한 전기 신호 값을 가져오게 됩니다.

⑤ 

블록 모음에서 

블록을 가져옵니다. 그리고, 위 ④에서 만든 블록에서 읽어오는 전기 신호 값을 

에 저장하는 블록을 만듭니다.

⑥ 

블록 모음에 있는 

블록을 가져와서 ⑤에서 만든 블록을 그 안에 넣어줍니다. 그러면, 수광부에 전달되는 전기 신호 값을 계속적으로 받을 수 있습니다.

⑦ 지금까지 만든 블록을 모두 연결해 주면, 다음 그림과 같이 완성됩니다. 이제 
버튼을 눌러 실행시켜 보세요. 물체를 적외선 센서에 가까지 가져갔다가 다시 떼면서 변화되는 

값을 확인해 보세요.

생각하기 14

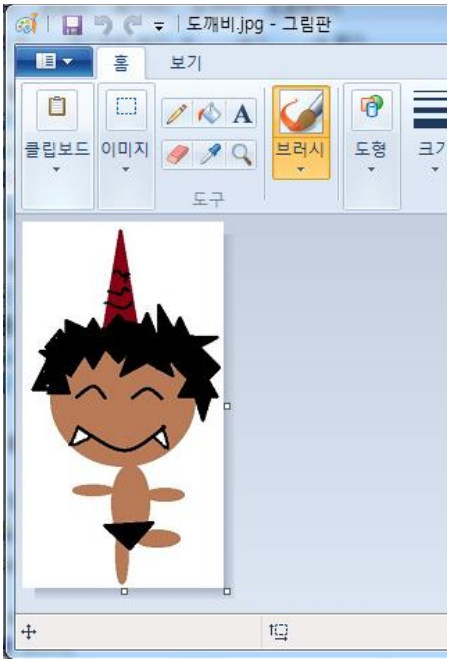
적외선 센서에 물체가 가가까이 다가가면 "삐~" 하는 부저가 울리도록 만들어 보세요.

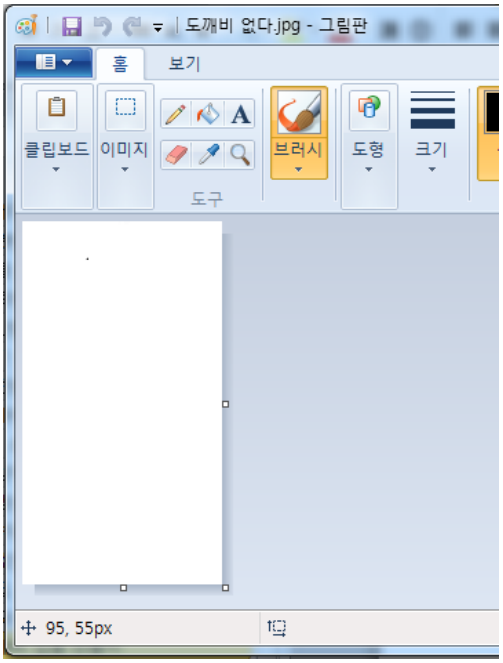
15-2.마법의 도깨비 집을 만들자

이번에는 적외선 센서에 물체가 가까이 다가오는 것이 감지되면 도깨비가 나타나는 무시무시한 마법의 집을 만들어 보겠습니다. 위 15-1절에서 만들어진 스크립트에 블록을 추가하여 만들어 보겠습니다.

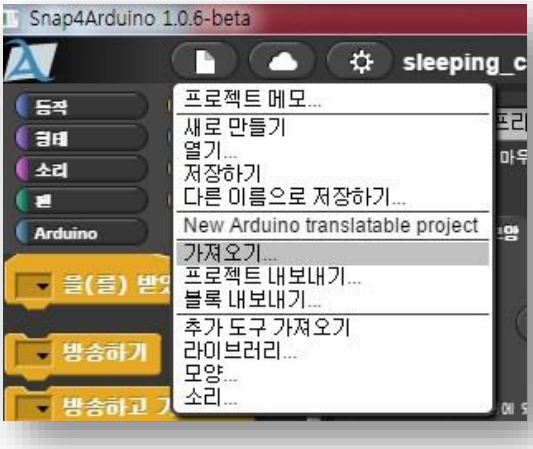
1 단계 : 그림 삽입하기

① 우선, 그림 툴을 이용하여, 도깨비 그림을 그립니다. 저는 윈도우에서 기본으로 제공되는 그림판 프로그램을 사용하였습니다. 도깨비 그림을 그려서 "도깨비.jpg"로 그림을 저장합니다. 그리고, 빈 그림을 "도깨비 없다.jpg"로 저장합니다. 스크래치에서 사용했던 그림을 그대로 사용해도 됩니다.

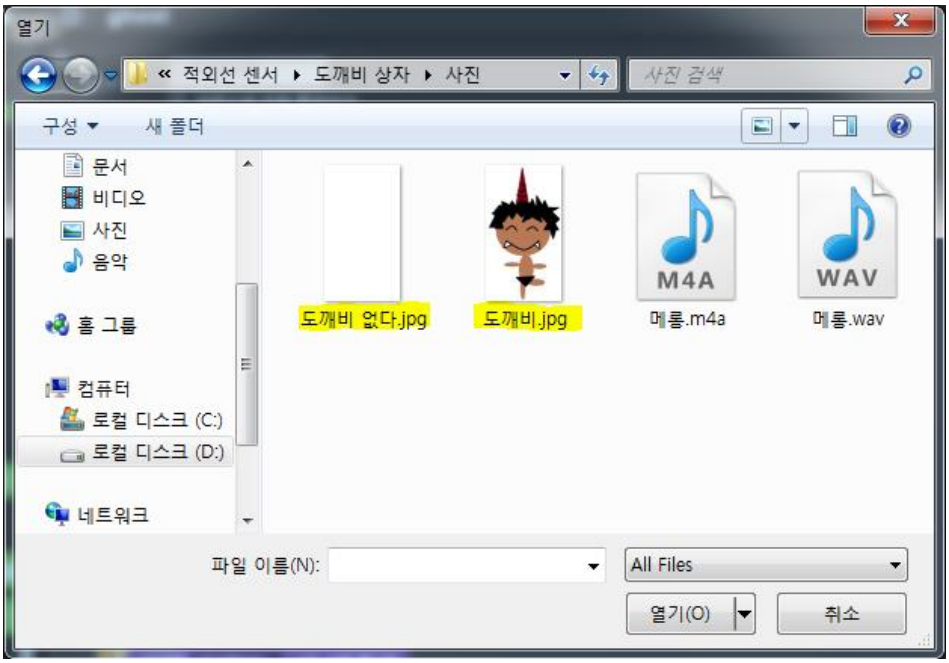




② Snap4Arduino 에서  버튼을 누른 후, "가져오기..." 메뉴를 선택합니다.



③ ①에서 저장한 "도깨비.jpg" 그림을 선택하여, "열기"버튼을 누릅니다.

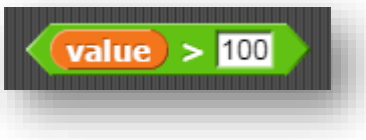


④ ②, ③ 과정을 반복하여 "도깨비 없다.jpg"그림도 가져옵니다.

⑤ "모양" 탭으로 가시면 아래와 같이 그림이 삽입된 상태를 확인하실 수 있습니다.

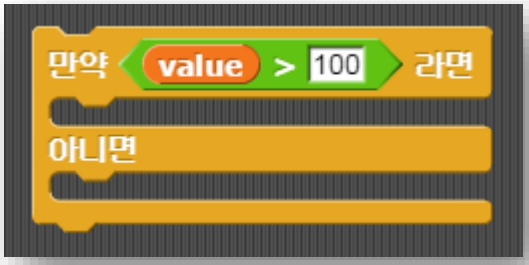
2 단계 : 적외선 센서의 값에 따라 그림 바꾸기

① 연산 블록 모음에서  블록을 스크립트로 가져와서, 적외선 센서의 값 value 가 100 보다 큰가를 비교하는 블록을 만듭니다. (적외선 센서 값은 물체가 가까이 오면 값이 증가합니다.)



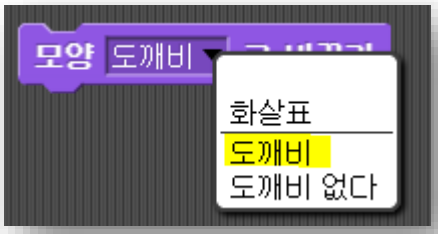
② 제어 블록 모음에서 

블록을 스크립트로 가져옵니다. 위 ①에서 만든 블록을 첫 번째 입력 칸에 넣어 줍니다.



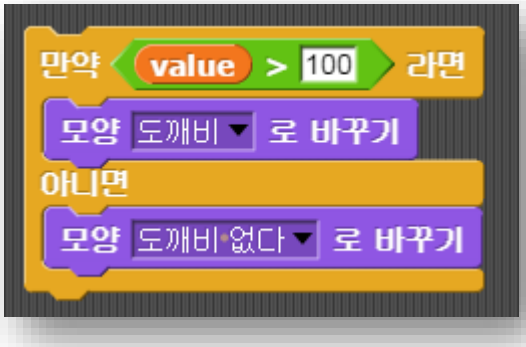
③ 블록 모음에서 

블록을 가져옵니다. 가져온 블록에서 아래 화살표를 누르면, 다음 그림과 같이 그림을 선택할 수 있습니다. "도깨비" 그림을 선택하겠습니다.



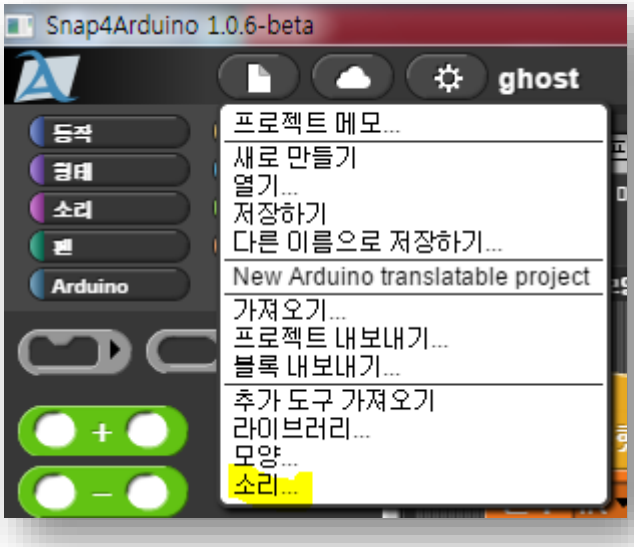
④ ③의 과정을 반복하여 "도깨비 없다" 그림으로 바꾸는 블록도 만듭니다.

⑤ ②에서 만든 블록에 위 ③, ④ 블록들을 아래 그림과 같이 넣어 줍니다. value가 100보다 크다면, 모양이 도깨비로 바뀌고, 100보다 작거나 같은 경우 모양이 도깨비 없다로 바뀌는 블록을 완성하였습니다.



3 단계 : 소리 삽입하기

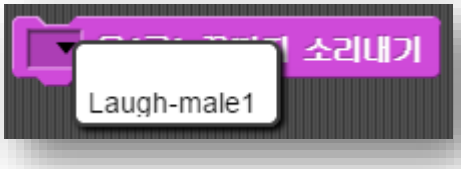
① 화면이 도깨비 모양으로 바뀔 경우, 으스스한 웃음소리도 나오게 하면 더 재미있겠지요? Snap4Arduino 에서  버튼을 누른 후, "소리..." 메뉴를 선택하세요.



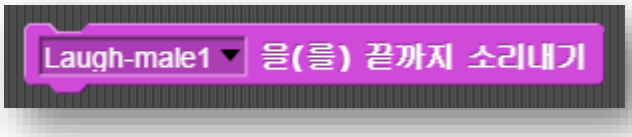
② 그러면, 다음 그림과 같이 소리 파일을 가져올 수 있습니다. 저는 "Laugh-male1.wav" 소리를 선택하겠습니다.

③ 소리 탭을 확인하시면, 다음 그림과 같이 선택한 소리 파일을 가져온 것을 확인 하실 수 있습니다. "재생" 버튼을 누르면 소리를 확인하실 수 있습니다.

④ 다시 "스크립트" 탭으로 옵니다. 소리 블록 모음에서 을(를) 끝까지 소리내기 블록을 가져옵니다. 이 블록에서 아래 화살표를 누르면 다음 그림과 같이 ②에서 가져온 소리파일을 블록으로 가져올 수 있습니다.



⑤ 위에서 "Laugh-male1"을 선택하면, 다음 그림과 같이 소리를 내는 블록을 만들 수 있습니다.

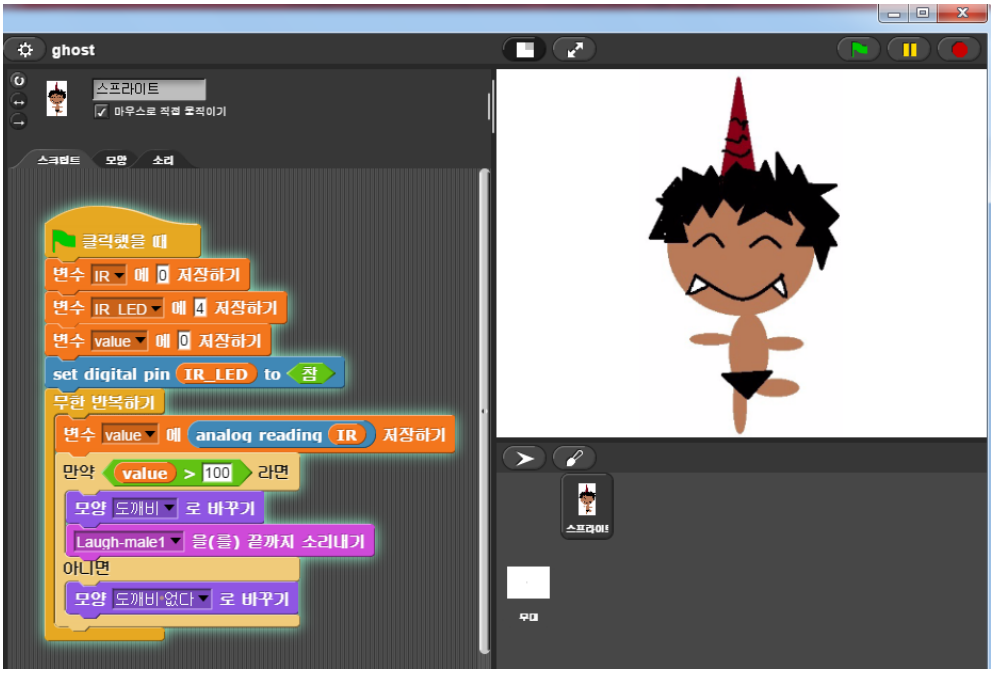


⑥ ⑤의 블록을 1 단계에서 만든 블록에서 

블록 다음에 끼워 넣습니다. 그러면, 화면이 도깨비로 바뀔 때, 웃음 소리도 같이 나오는 블록이 완성됩니다.

⑦ 다음 그림과 같이 15-1절에서 만들어진 스크립트에 위 블록을 삽입합니다.

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



16. 전광판 만들기

곰곰이네 삼촌이 분식집을 열었습니다. 곰곰이와 아빠는 축하를 해주러 삼촌네 분식집을 찾아 갔습니다. 처음 가는 동네이어서 찾아가기 어려울 것 같았지만, 삼촌네 분식집 간판이 눈에 띄는 전광판으로 되어 있어서 쉽게 찾아 갔습니다. 곰곰이는 좋아하는 떡볶이를 맛있게 먹고 돌아왔습니다.

삼촌이 만들어 주시는 떡볶이는 정말 최고예요! 우리 떡볶이 먹으러 삼촌네 자주 놀러 가요!

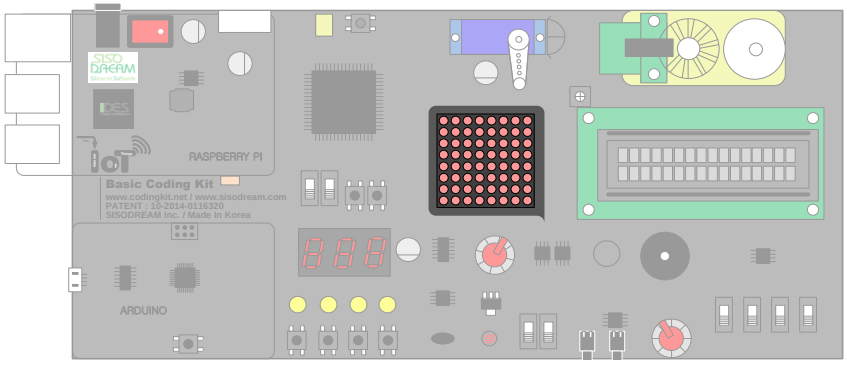
허허허! 그래~ 그래~ 알았다. 매운 음식은 잘 안 먹는 녀석이 삼촌 떡볶이는 참 잘 먹는구나.

삼촌네 떡볶이도 맛있었지만, 간판도 최고예요! 글씨가 나오고 하트도 나오면서 움직이니까 재미있었어요.

그래. 아빠도 간판에 자꾸만 눈이 가더구나. 그게 전광판이라고 하는 거란다. 코딩키트에 있는 도트매트릭스 같은 장치로 만드는 거란다.

16-1.도트매트릭스에 불을 켜보자.

도트매트릭스는 아래 그림과 같이 LED 를 여러 개를 촘촘히 모아둔 모양입니다. 야구장에 있는 전광판이 연상이 되지요?



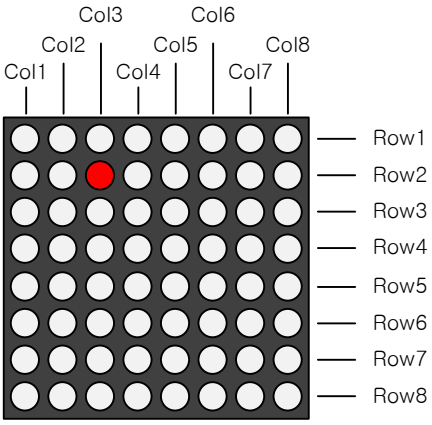
도트매트릭스에 있는 LED 를 하나 하나씩 별도로 켤 수 있기 때문에 원하는 문자나 그림을 표시할 수 있습니다. 도트매트릭스의 스위칭 ID는 b02 입니다. 스위치와 버튼을 이용하여 스위칭 ID 를 b01 로 바꾸겠습니다.

그런데, 이 LED 하나 하나가 다 아두이노의 핀에 연결되어 있는 것은 아닙니다. 도트매트릭스에는 가로 8개, 세로 8개 해서 64 개의(8x8=64) LED 가 들어 있는데, 아두이노에 가로줄 8 개와 세로줄 8 개가 8 핀에 연결이 되어 있고, 3번 핀을 On/Off 시켜서 가로줄과 세로줄을 선택하여 각각의 LED를 켜고 끄게 됩니다. 즉, 모두 9개의 핀으로 동작을 하게 하는 것입니다.

** 아두이노에서는 20개의 핀을 사용할 수 있습니다. 하지만, Snap4Arduino와 코딩키트의 통신을 위한 핀과 아날로그 핀을 제외하면 12개의 핀만을 사용할 수가 있습니다. 그래서 가로줄과 세로줄에 각각의 핀을 연결하여 사용할 수가 없습니다. 하지만 코딩키트에서는 도트매트릭스를 사용할 수 있도록 가로줄과 세로줄을 따로 선택하는 핀을 두어 도트매트릭스를 사용할 수 있도록*

하였습니다.

도트매트릭스는 가로줄(Row)이 켜지는 신호와 세로줄(Col)이 켜지는 신호가 교차하는 곳의 LED 를 켜 줍니다. 도트매트릭스의 가로줄과 세로줄은 2, 7, 10, 6, 4, 5, 12, 13 핀에 차례로 연결되어 있습니다. 아래 그림처럼, 세 번째 세로줄과 두 번째 가로줄이 교차하는 LED 를 켜주기 위해서 세로 10 번핀, 가로 7 번핀을 참으로 설정해 주면 됩니다.



1 단계 : 변수 만들기

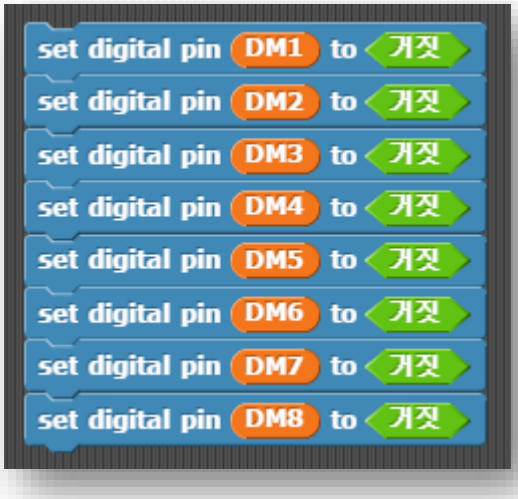
① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 DM1, DM2, DM3, DM4, DM5, DM6, DM7, DM8, DM_SEL 변수를 만듭니다.

② 제어 블록 모음에서 클릭했을 때 블록을 스크립트로 가져옵니다. 그리고, 변수 블록 모음에서 변수 에 0 저장하기 블록을 9개 가져옵니다. 아래 그림처럼, DM1 ~ DM8 변수에 도트매트릭스 핀 번호(2, 7, 10, 6, 4, 5, 12, 13)를 순서대로 저장하고, DM_SEL 변수에는 핀 번호 3을 저장합니다.

2 단계 : 도트매트릭스 초기화하기

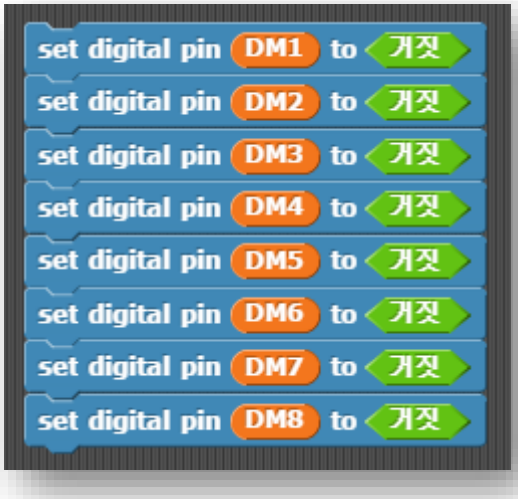
① Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수에 참 값을 넣어 주는 블록을 만듭니다. DM_SEL 이 참 값을 가지면, 가로줄(Row)을 설정하는 모드가 됩니다.

② 가로줄의 모든 핀들은 거짓 값을 설정해줍니다.



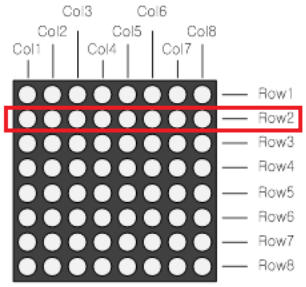
③ 이번에는 Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수에 거짓 값을 넣어 주는 블록을 만듭니다. DM_SEL 이 거짓 값을 가지면, 세로줄(Col)을 설정하는 모드가 됩니다.

④ 세로줄의 모든 핀들도 거짓 값을 설정해줍니다. 이제 도트매트릭스의 초기화가 끝났습니다.

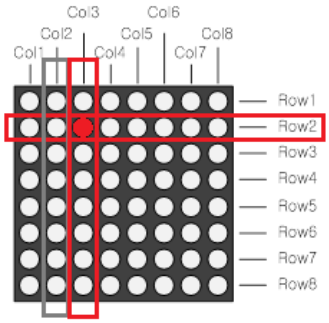


3 단계 : 도트매트릭스에서 1개의 LED 켜기

① Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수와 DM2 변수에 참 값을 넣어 주는 블록을 만듭니다. 가로줄 설정 모드 상태에서 두 번째 줄을 On 시키는 블록입니다.



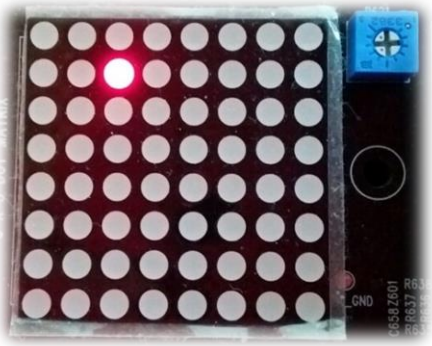
② 이번에는 Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수에 거짓 값을 넣어 주는 블록과 DM3 변수에 참 값을 넣어 주는 블록을 만듭니다. 세로줄 설정 모드 상태에서 세 번째 줄을 On 시키는 블록입니다. 그런데, 가로줄 설정 모드일 때의 상태가 남아 있습니다. 그래서 DM2 변수에 거짓 값을 넣어 주는 블록을 추가하여 세로 두 번째 줄은 Off 상태로 만들어 줍니다.



③ 다음 그림과 같이 블록을 완성 하셨나요? 그러면,  버튼을 눌러 실행시켜 보세요. 가로 두 번째 줄과 세로 세 번째 줄이 교차하는 LED가 켜졌는지 확인해 보세

요.


클릭했을 때
변수 DM1 에 2 저장하기
변수 DM2 에 7 저장하기
변수 DM3 에 10 저장하기
변수 DM4 에 6 저장하기
변수 DM5 에 4 저장하기
변수 DM6 에 5 저장하기
변수 DM7 에 12 저장하기
변수 DM8 에 13 저장하기
변수 DM_SEL 에 3 저장하기
set digital pin DM_SEL to 참
set digital pin DM1 to 거짓
set digital pin DM2 to 거짓
set digital pin DM3 to 거짓
set digital pin DM4 to 거짓
set digital pin DM5 to 거짓
set digital pin DM6 to 거짓
set digital pin DM7 to 거짓
set digital pin DM8 to 거짓
set digital pin DM_SEL to 거짓
set digital pin DM1 to 거짓
set digital pin DM2 to 거짓
set digital pin DM3 to 거짓
set digital pin DM4 to 거짓
set digital pin DM5 to 거짓
set digital pin DM6 to 거짓
set digital pin DM7 to 거짓
set digital pin DM8 to 거짓
set digital pin DM_SEL to 참
set digital pin DM2 to 참
set digital pin DM_SEL to 거짓
set digital pin DM3 to 참
set digital pin DM2 to 거짓



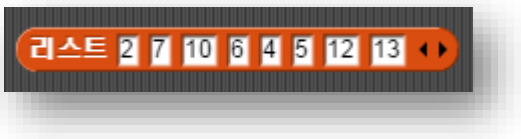
16-2.리스트와 도트매트릭스

도트매트릭스의 가로 8줄과 세로 8줄을 하나씩 처리하려고 하니, 블록의 개수가 너무 많아집니다. 이럴 경우에 변수 블록 모음에 있는 리스트 블록을 사용하면 좀 더 간단하게 만들 수 있습니다.

1 단계 : 변수 만들기

① 변수 블록 모음에서 변수 만들기 버튼을 눌러서 DM1 ~ DM8 변수 대신에 DMs 변수를 만듭니다. 이 변수에 핀 번호들이 모두 저장됩니다. 그리고, 이 핀 번호들을 가져와 사용하기 위해 count 변수를 만들고, DM_SEL 변수도 만들어 줍니다.

② 변수 블록 모음에 있는 리스트 블록을 스크립트 창으로 가져옵니다. 리스트 블록에 있는 오른쪽 화살표를 클릭하면, 입력 창이 하나씩 늘어납니다. 도트매트릭스 핀 번호(2, 7, 10, 6, 4, 5, 12, 13)를 저장할 수 있도록 입력 창을 8개를 만들고, 핀 번호를 순서대로 입력합니다.



③ 변수 블록 모음에서 변수 에 0 저장하기 블록을 가져옵니다. 리스트 블록에 입력된 도트매트릭스 핀 번호들을 DMs 변수에 저장합니다.

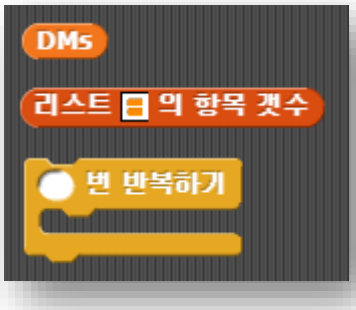
④ DM_SEL 변수에는 핀 번호 3을 저장합니다.

⑤ count 변수에 0을 저장합니다.

2 단계 : 도트매트릭스 초기화 하기

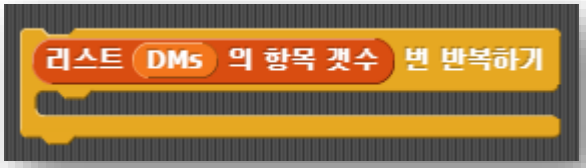
① Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수에 좀 값을 넣어 주는 블록을 만듭니다. 가로줄 설정 모드가 됩니다.

② 변수 블록 모음에서 DMs 변수와 리스트 의 항목 갯수 블록을, 제어 블록 모음에서 번 반복하기 블록을 스크립트로 가져옵니다.

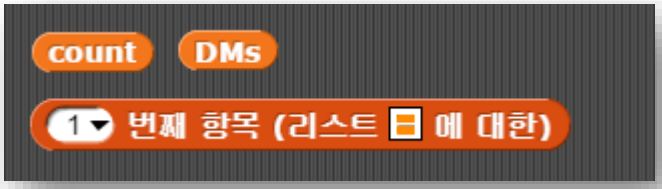


③ 리스트 의 항목 갯수 블록의 입력 칸에 DMs 변수를 넣어줍니다. 그러면, DMs 에 저장된 도트매트릭스 핀 번호들의 개수를 알 수 있습니다.

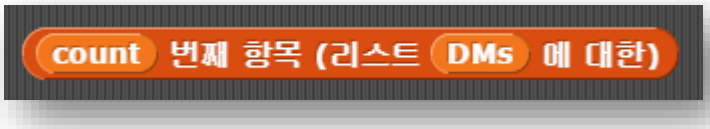
④ ③에서 만든 블록을 번 반복하기 블록의 첫 번째 입력 칸에 넣어 줍니다. 그러면, DMs 에 저장된 도트매트릭스 핀 번호들의 개수가 8개이므로, 8번 반복하기 블록이 됩니다. "8"이라는 숫자를 입력해서 8번 반복하기 블록으로 만들어도 되지만, 리스트에 저장된 값들의 개수에 변화가 있는 경우, 리스트 의 항목 갯수 블록을 사용하면, 일일이 수정하는 번거로움을 줄일 수 있습니다.



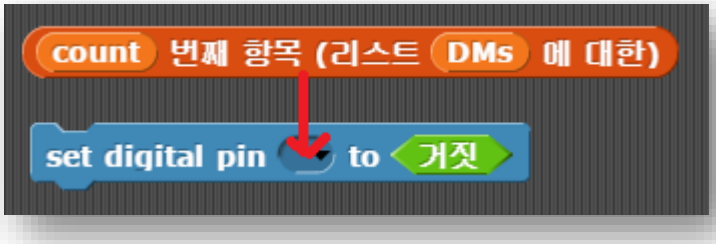
⑤ 변수 블록 모음에서 1 번째 항목 (리스트 에 대한) 블록과 count 변수, 그리고 DMs 변수를 스크립트로 가져옵니다.



⑥ 1 번째 항목 (리스트 에 대한) 블록의 첫 번째 입력 칸에는 count 변수, 그리고 두 번째 입력 칸에는 DMs 변수를 넣어 줍니다. 그러면, DMs 변수에 저장된 도트매트릭스 핀 번호들 중에서 count 번째 저장된 핀 번호를 가져옵니다.

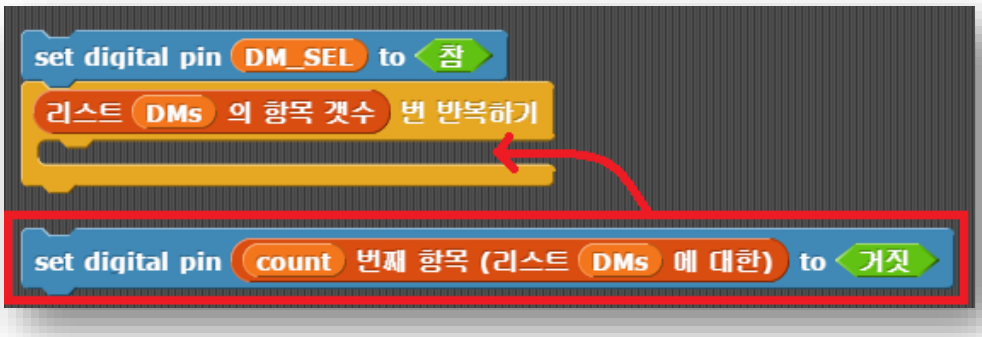


⑦ 이제 Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DMs 변수의 count 번째 핀에 거짓 값을 설정해 주는 블록을 만듭니다.



⑧ 8개의 핀에 모두 

값을 설정해 주어야 합니다. 위에서 만든 블록을 ④에서 만든 8번 반복하기 블록에 다음 그림과 같이 넣어 줍니다.



⑨ 그런데, 현재 

가 0 값을 가지고 있습니다. 

변수에 저장된 도트매트릭스의 모든 핀들에 

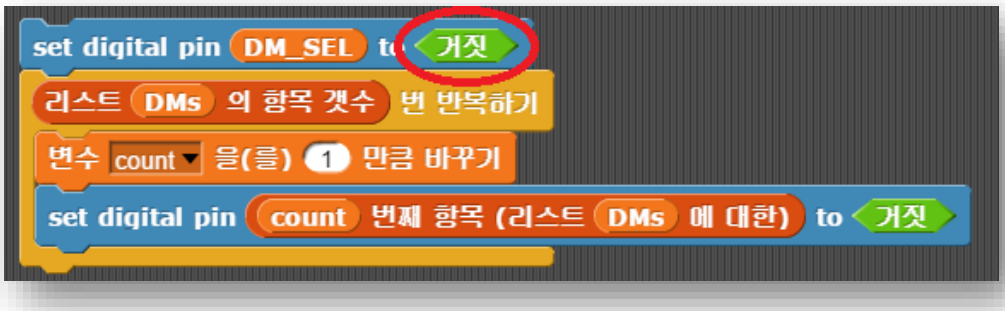
값을 설정할 수 있도록, 아래 그림과 같이 

값을 1씩 증가 시키는 블록도 같이 넣어 줍니다.

** snap4arduino 에서의 리스트의 인덱스는 0 번이 아닌 1 번부터 시작합니다. 대부분의 프로그램 언어에서는 리스트의 인덱스가 0 번부터 시작하지만 snap4arduino 에서는 1 번부터 시작한다는 것에 주의하세요.*

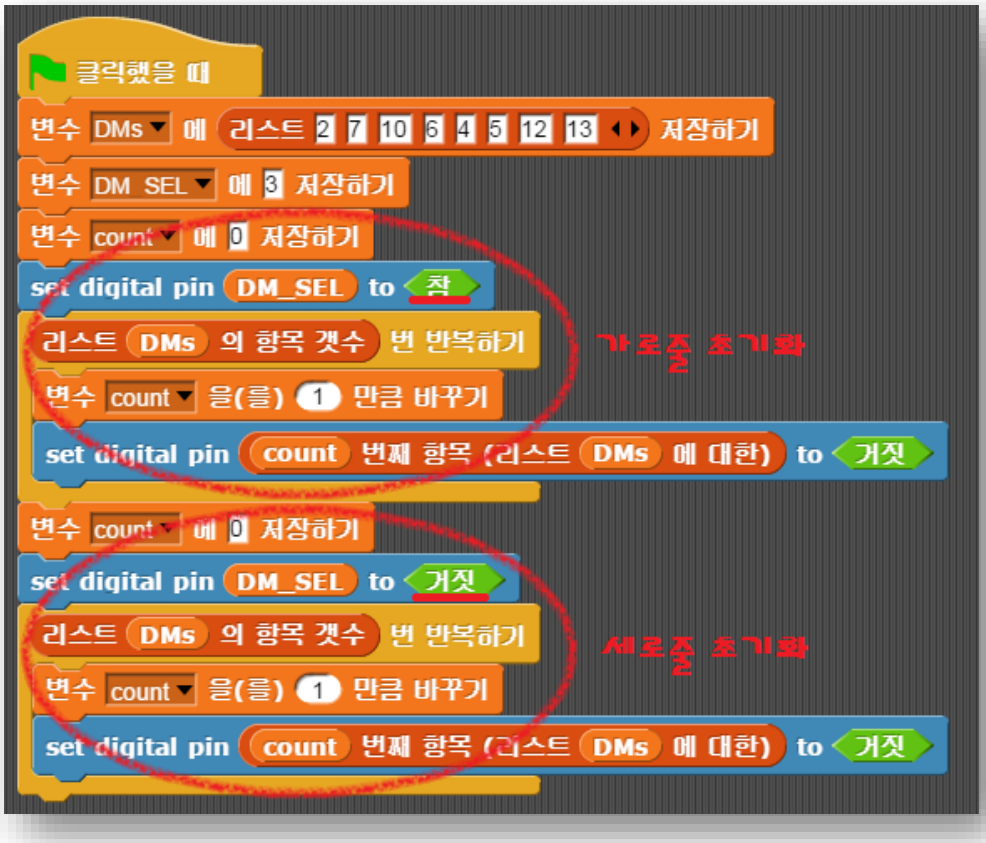
⑩ 다음 그림과 같이 블록을 완성 하셨나요? 이제, 가로줄에 대한 초기화가 끝났습니다.

⑪ 세로줄 초기화는 위 블록을 그대로 복사하여 사용하시면 됩니다. 위 블록을 복사한 후에 DM_SEL 변수에 거짓 값을 넣어 주도록 수정하세요.



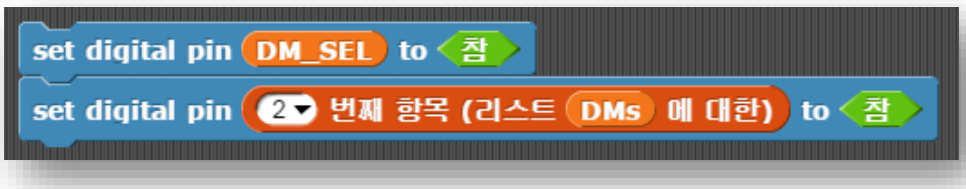
⑫ count 값을 다시 0으로 설정 해 주시는 것도 잊지 마세요.

⑬ 이제, 초기화 하는 작업이 모두 끝났습니다. 지금까지 잘 따라오셨다면, 다음 그림과 같이 블록이 완성됩니다. 16-1절에서 만든 도트매트릭스를 초기화 하는 블록과 비교해 보세요. 완성된 블록의 길이가 훨씬 짧아진 것을 확인하실 수 있습니다.



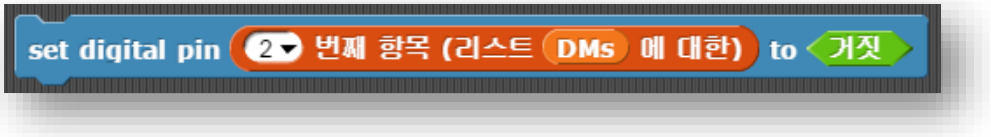
3 단계 : 도트매트릭스에서 1개의 LED 켜기

① Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수에 참 값을 넣어 주는 블록을 만듭니다. 그리고, 변수 블록 모음에서 1 번째 항목 (리스트 에 대한) 블록을 가져 와서 DMs 의 두 번째 저장된 핀에 참 값을 넣어 주는 블록을 만듭니다. 가로 두 번째 줄을 On 시키는 블록입니다.

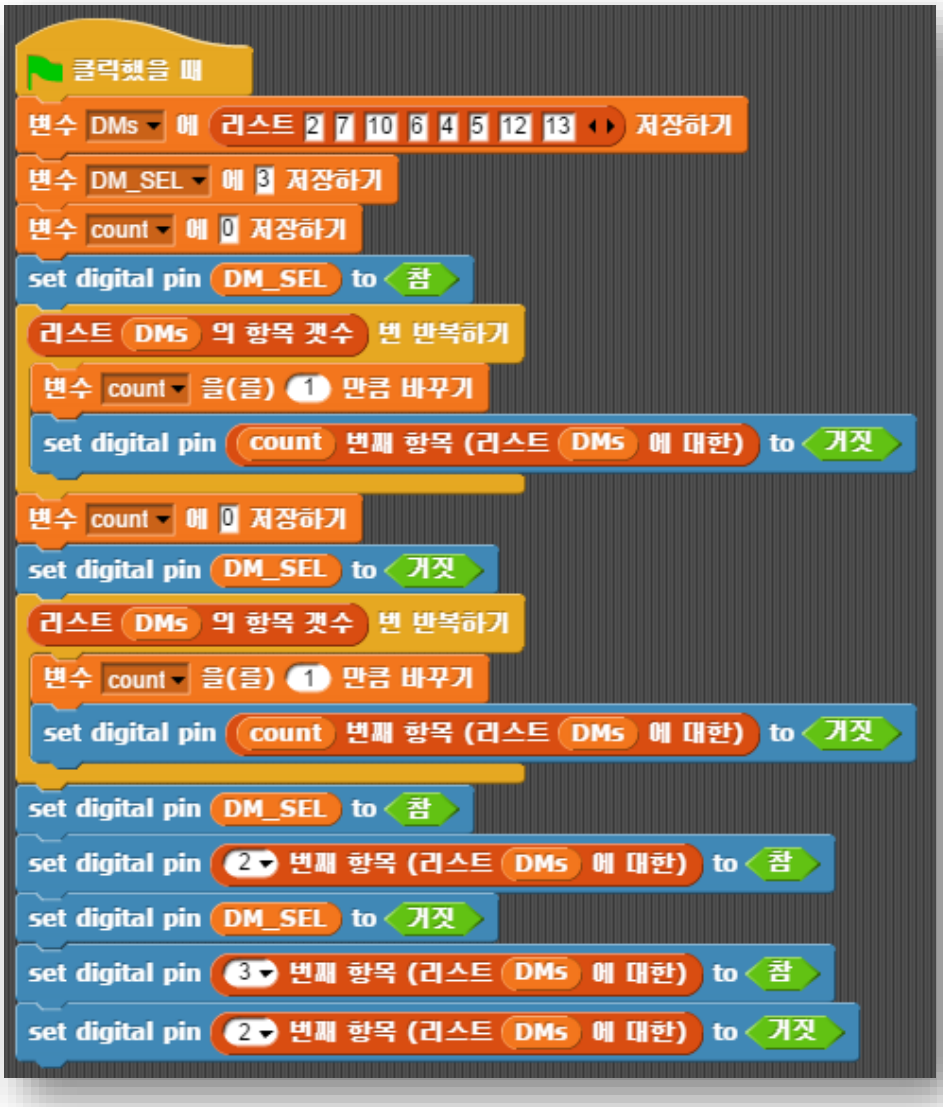


② 이번에는 세로 세 번째 줄을 On 시키는 블록을 만들겠습니다. 우선, Arduino 블록 모음에서 set digital pin to 블록을 가져와서 DM_SEL 변수에 거짓 값을 넣어 주는 블록을 만듭니다. 그러면, 세로줄 설정 모드로 바뀌게 됩니다. 그리고, 변수 블록 모음에서 1 번째 항목 (리스트 에 대한) 블록을 가져 와서 DMs의 세 번째 저장된 핀에 참 값을 넣어 주는 블록을 만듭니다. 이제 세로 세 번째 줄을 On 상태로 만들었습니다.

③ 그런데, 가로줄에서 설정 했던 정보가 그대로 남아 있기 때문에 세로 두 번째 줄이 On 상태로 남아 있습니다. 그래서 세로 두 번째 줄을 Off 상태로 만들어 주는 블록을 추가합니다. 변수 블록 모음에서 1 번째 항목 (리스트 에 대한) 블록을 가져 와서 DMs의 세 번째 저장된 핀에 거짓 값을 넣어 주는 블록을 만듭니다.



④ 다음 그림과 같이 블록이 완성 되었다면,  버튼을 눌러 실행시켜 보세요. 가로 두 번째 줄과 세로 세 번째 줄이 교차하는 LED가 켜졌나요? 그리고 16-1의 완성된 블록과도 비교해 보세요.



생각하기 15

도트매트릭스의 모든 LED에 불이 들어오게 하도록 블록을 만들어 보세요.

생각하기 16

도트매트릭스의 모든 LED의 불이 꺼지도록 블록을 만들어 보세요.

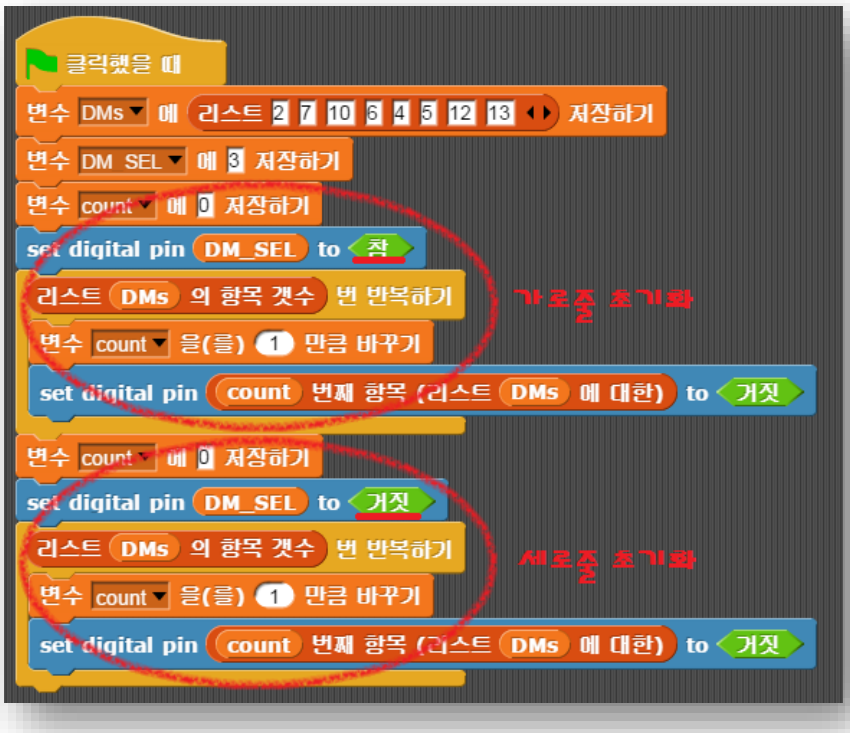
16-3.움직이는 전광판을 만들자

도트매트릭스의 LED 들이 하나 둘씩 천천히 켜지거나 꺼지게 만들면 어떨까요? 그러면, 도트매트릭스의 LED 들이 마치 움직이는 것처럼 보일 것입니다. "생각하기 15"와 "생각하기 16"에서 전광판 전체를 켜고 끄는 동작을 연습해 보았습니다. 이 동작을 연결하면 전체 도트매트릭스가 깜박이는 동작이 만들어 집니다. 이것을 응용하여 LED 등이 한 줄씩 켜지다가 다시 한 줄씩 꺼지는 동작을 만들어 보겠습니다.

** 도트매트릭스로 다양한 모양의 움직이는 그림을 만들 수 있습니다. 그런데, Snap4Arduino 와 아두이노의 통신 구조상 다소 어려움이 있습니다. 아두이노나 라즈베리파이로 도트매트릭스를 다루는 방법을 공부하시기를 권해드립니다. "코딩북 파트 3 아두이노 편" 또는 "파트 4 라즈베리파이 편"에서 방법을 찾으실 수 있습니다.*

1 단계 : 변수 만들기와 도트매트릭스 초기화

① 이 과정은 16-2절의 "1단계 : 변수 만들기"와 "2 단계 : 도트매트릭스 초기화" 과정과 동일합니다. 변수 만들기와 도트매트릭스 초기화 블록을 먼저 만들어 주세요.



② 이번에는 초기화할 때 도트매트릭스의 LED 를 모두 켜 주겠습니다. 그래서 다음과 같이 초기화 값을 

으로 해 줍니다.

2 단계 : 순차적으로 LED 점등/소등 시키기

① 

블록 모음에서 

블록을 가져와서 

변수에 

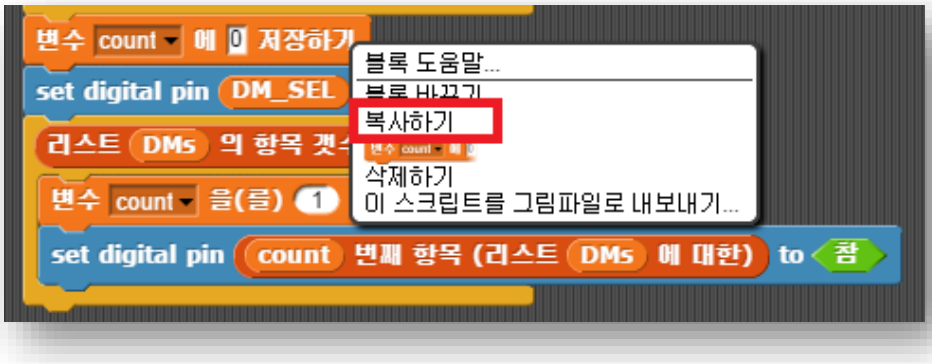
값을 넣어 주는 블록을 만듭니다. 세로로 LED 등이 한 줄씩 켜지고 다시 한 줄씩 꺼지는 동작을 만들 것 입니다.

② LED 등이 켜지고 꺼지는 동작을 반복하도록 만들려고 합니다. 

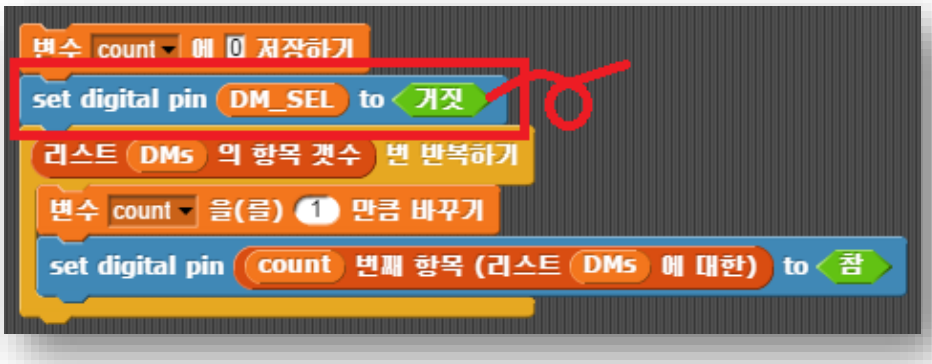
블록 모음에서 

블록을 가져옵니다.

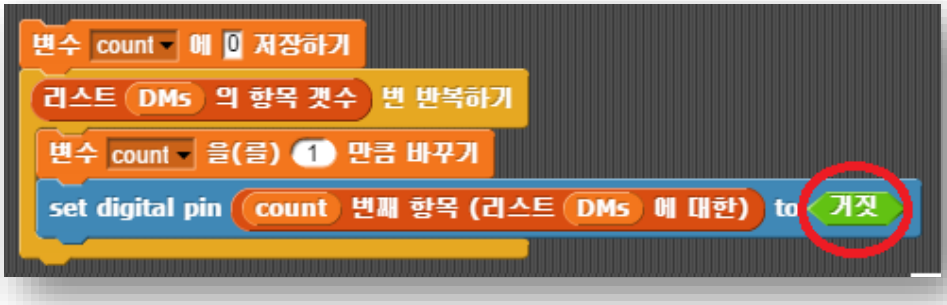
③ 도트매트릭스 초기화를 할 때 사용했던 블록들을 가져와 재사용하려고 합니다. 세로줄 초기화 블록들을 복사합니다.



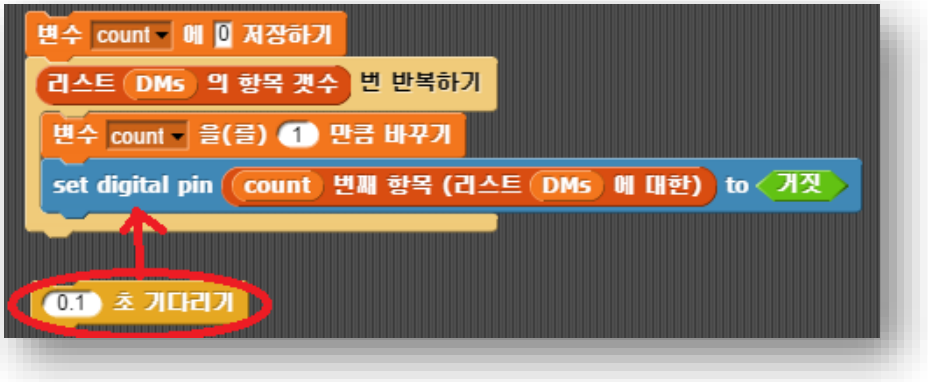
④ 복사된 블록들 중에서 세로줄 설정 모드가 중복되므로, 이 블록을 삭제합니다.



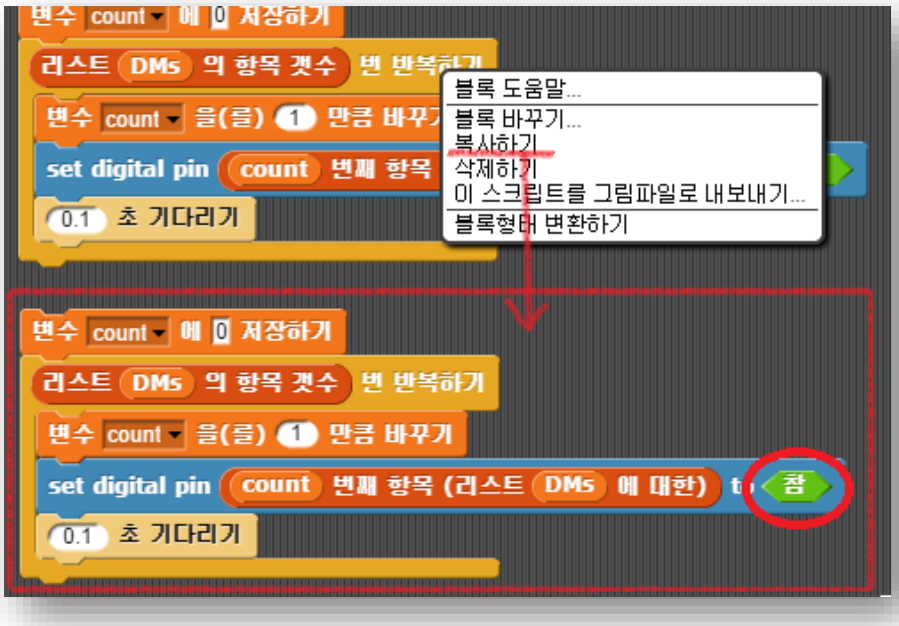
⑤ 초기화 단계에서 세로줄과 가로줄 모두 On 상태로 설정하였기 때문에, 도트매트릭스의 LED가 모두 켜진 상태에서 시작하게 됩니다. 세로 한 줄씩 LED를 순차적으로 끄는 동작부터 시작하도록, DMs 에 저장된 핀들에 값을 설정하는 블록을 수정합니다. 참 값 대신에 거짓 값으로 설정하도록 바꿉니다.



⑥ 제어 블록 모음에서 1 초 기다리기 블록을 가져옵니다. 다음 LED가 켜지는 시간 간격을 0.1초로 합니다.



⑦ LED가 순차적으로 꺼지는 블록을 만들었습니다. 이 블록을 복사하여, LED가 순차적으로 켜지는 블록을 만듭니다. 이번에는 DMs 에 저장된 핀들에 값을 참 값으로 설정하도록 바꿉니다.



⑧ ⑥과 ⑦에서 만든 LED가 순차적으로 꺼졌다 켜졌다 하는 동작을 

블록 안으로 넣어 줍니다.

다음 그림과 같이 블록을 완성하셨나요? 그러면,  버튼을 눌러 실행시켜 보세요. 도트매트릭스의 LED 등이 세로로 한 줄씩 켜지고 다시 한 줄씩 꺼지는 동작을 반복합니다.

생각하기 17

도트매트릭스의 LED 등이 가로로 한 줄씩 켜지고 다시 한 줄씩 꺼지는 동작을 반복하도록 만들어 보세요.

17. 전자 시계 만들기

곰곰이와 아빠는 오늘 오후 2시에 공원에 산책을 가기로 했습니다. 재미있는 동화책을 읽느라 곰곰이는 시간이 가는 줄 모르고 있었습니다.


곰곰아! 아빠랑 산책 갈 준비할까? 공원에서 친구를 만나기로 했었지? 서둘러 준비를 해야겠구나.


어? 벌써 약속 시간이 다 되었네요! 동화책이 너무 재미있어서 시간 가는 줄 몰랐어요. 옛날에는 친구랑 만날 약속을 어떻게 잡았을까요? 시계가 없으면, 시간을 모르잖아요.

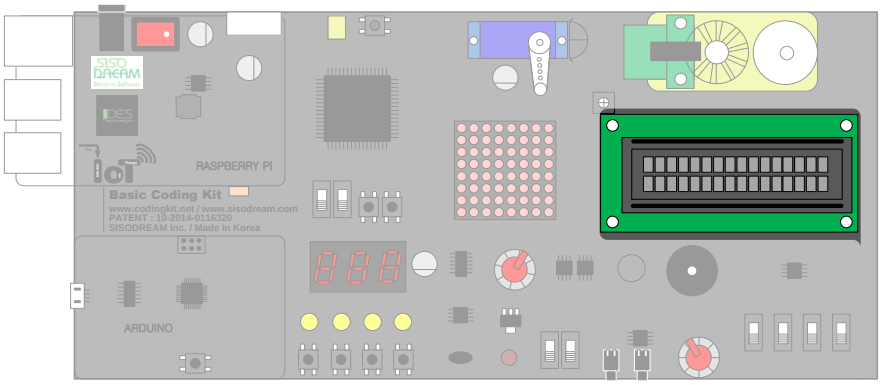

옛날에도 시계는 있었단다. 해시계, 물시계, 모래시계 등이 있었지. 그래도 지금처럼 정확한 시간을 알기는 어려웠을 거다.


아. 그렇군요! 신기해요! 시계는 혼자서도 똑딱똑딱 잘도 가잖아요. 코딩키트로 시계도 만들 수 있을까요?


그럼. 코딩키트에 있는 LCD 장치를 이용해서 디지털 시계를 만들어 볼 수 있지. 산책 다녀와서 한 번 만들어 보자꾸나.

17-1.화면으로 보여줘요! 캐릭터 LCD

캐릭터 LCD(Character Liquid Crystal Display)라는 것은 문자를 출력하기 알맞게 제작된 LCD 입니다. 코딩키트에는 16 개의 문자를 2 줄에 출력할 수 있는 캐릭터 LCD 가 장착되어 있습니다. 보통 16X2 캐릭터 LCD 라고 합니다.

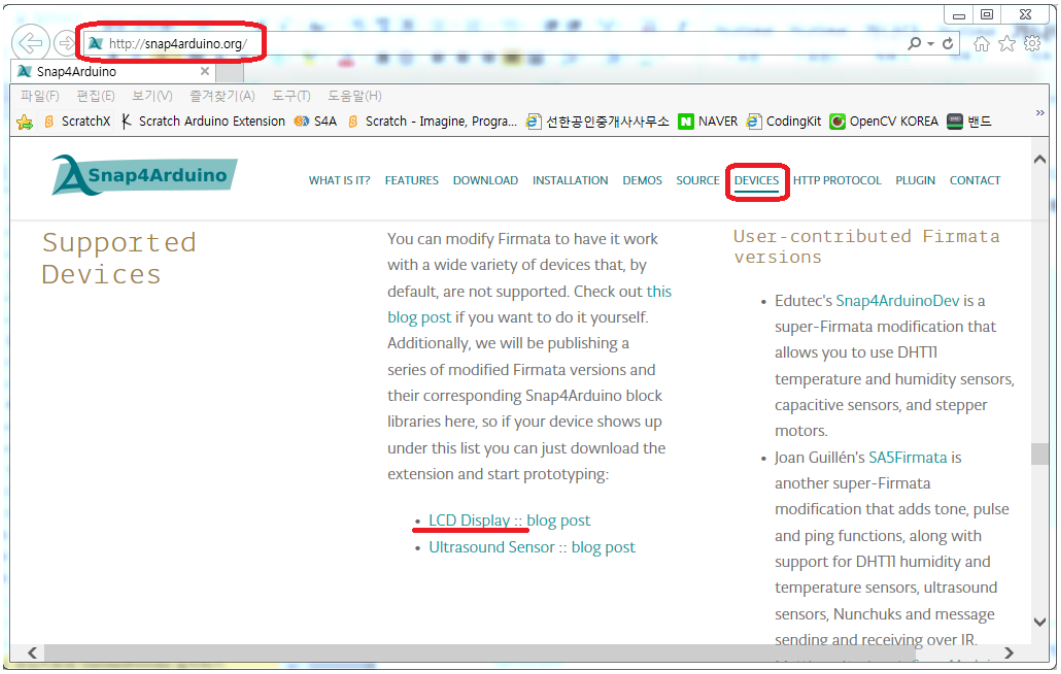


블록코딩에서 캐릭터 LCD를 쓰기 위해서는 스위칭 ID를 b01로 맞추어야 합니다.

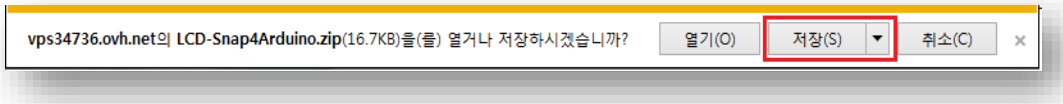
그럼, 캐릭터 LCD 에 메시지를 출력해 보는 코드를 만들어 보겠습니다. 우선, 캐릭터 LCD 를 사용하기 위해서는 코딩키트에 LCDFirmata 를 설치하셔야 합니다.

1 단계 : 코딩키트에 LCDFirmata 업로드하기

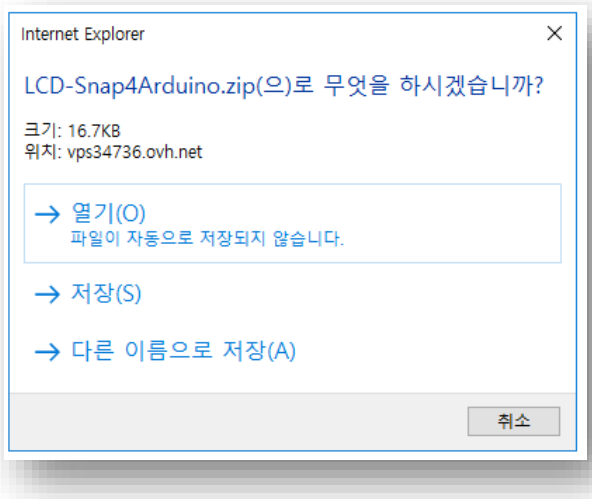
① Snap4Arduino 사이트(http://snap4arduino.rocks/)에 접속합니다. 상위 메뉴에서 "DEVICES"를 클릭합니다. 그러면, 아래와 같이 Supported Devices 화면이 나타납니다. 여기서 "LCD Display"를 클릭합니다.



② 그러면, 아래와 같이 LCD-Snap4Arduino.zip 파일을 다운 받을지 여부를 묻는 창이 나타납니다. "저장" 버튼 옆의 역삼각형 버튼을 눌러 "다른 이름으로 저장"을 선택합니다.

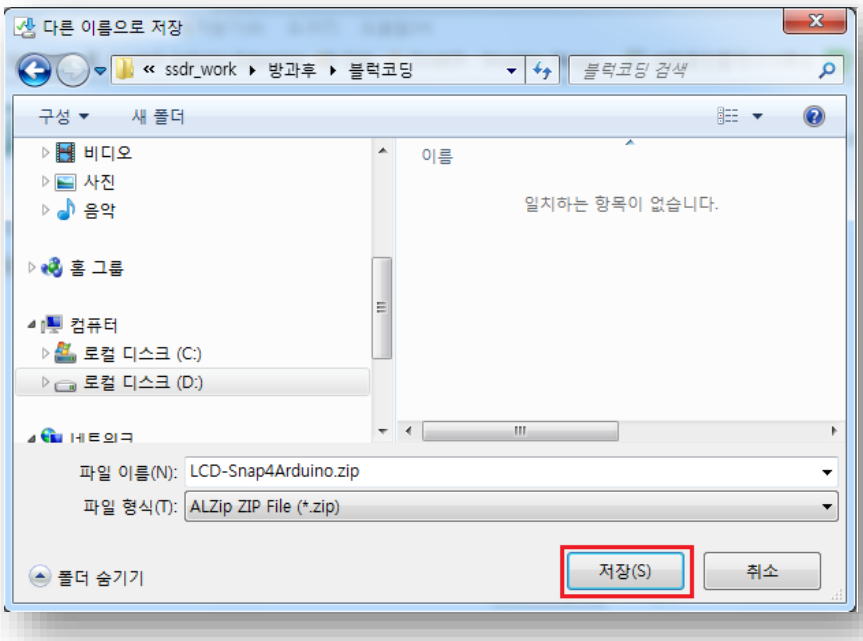


다음과 같은 메시지가 나올 수도 있습니다.



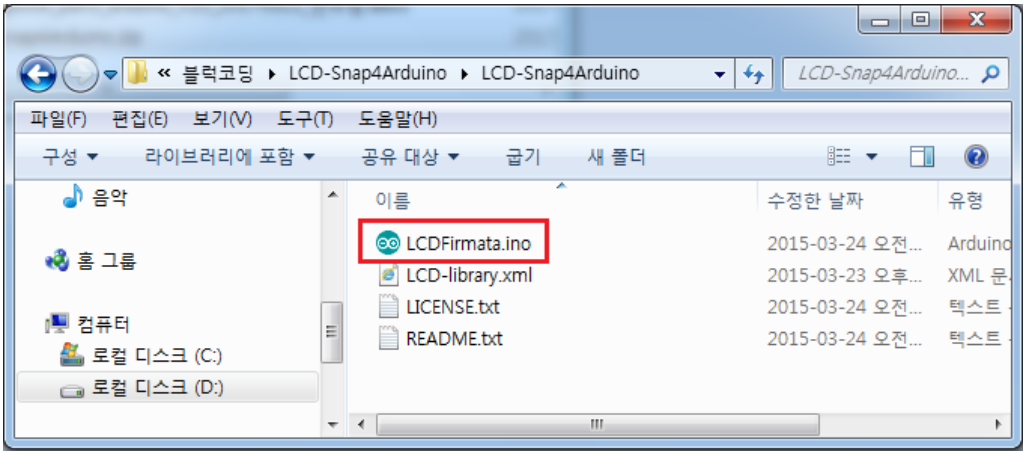
이 때도 "다른 이름으로 저장"을 선택합니다.

③ 그러면, 아래와 같이 LCD-Snap4Arduino.zip 파일을 저장하는 창이 나타납니다. 적당한 위치를 선택하고 "저장" 버튼을 클릭합니다.

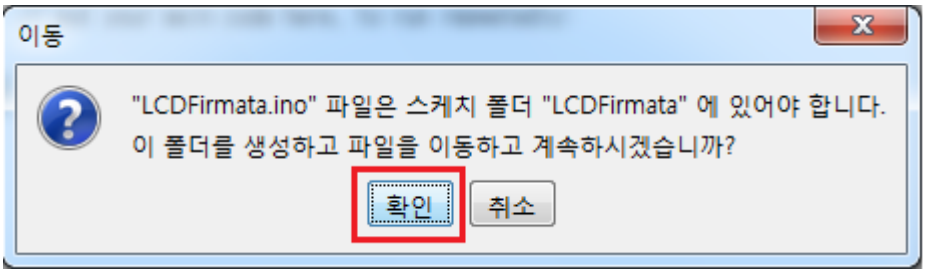


④ 저장한 파일의 압축을 풉니다.

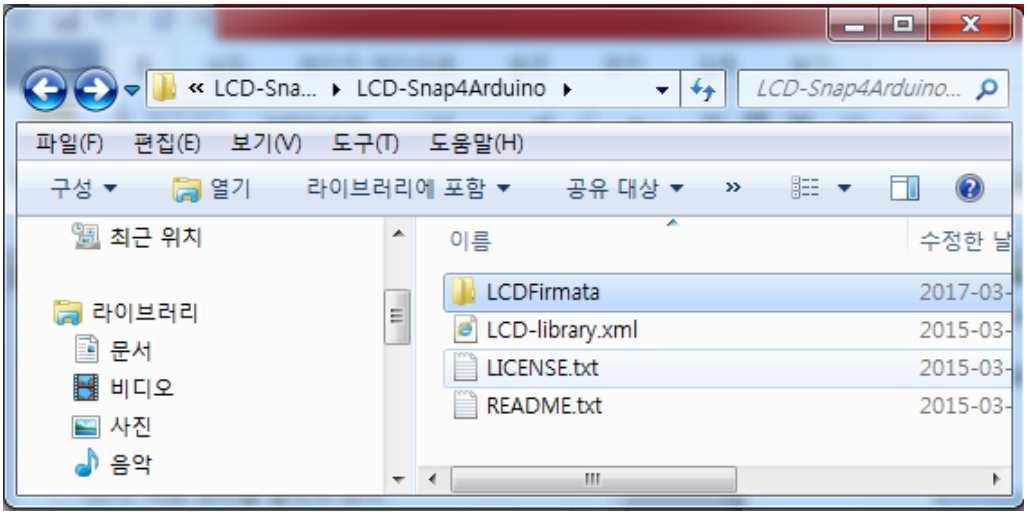
⑤ 압축을 풀면 다음 그림과 같은 파일들이 나옵니다. 여기서 "LCDFirmata.ino" 파일을 두 번 클릭합니다. (아두이노 프로그램을 실행해서 "파일" -> "열기" 메뉴를 선택하신 후 아래 파일을 선택하셔도 됩니다.)



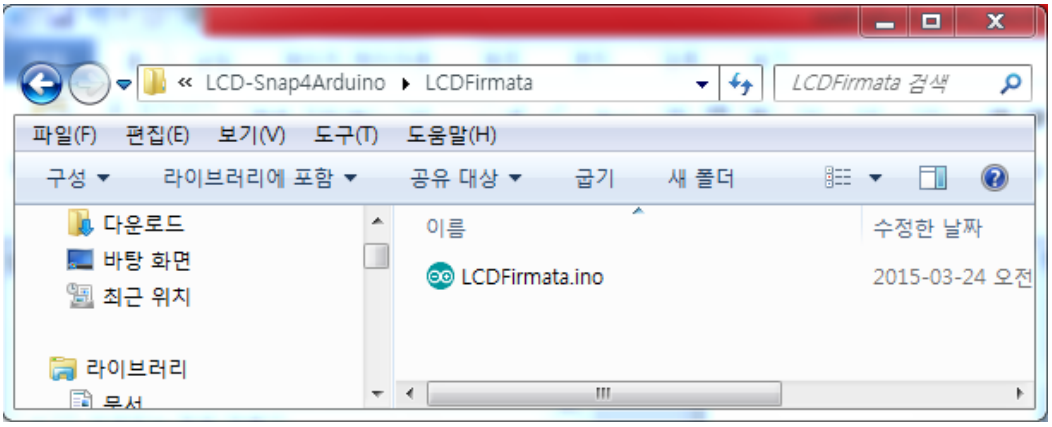
⑥ 그러면, 다음과 같은 메시지 창이 나타납니다. "LCDFirmata" 폴더를 만들고 그 폴더 안에 "LCDFirmata.ino" 파일을 이동시키겠다는 메시지 입니다. "확인" 버튼을 눌러주세요.



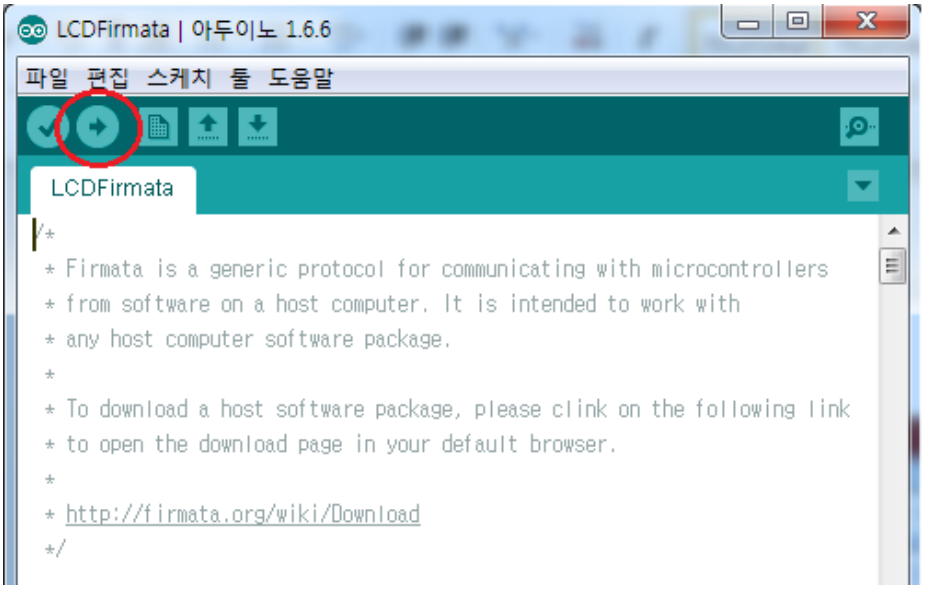
⑦ "LCDFirmata" 폴더가 생겼습니다. 이 폴더 안으로 들어가겠습니다. (LCDFirmata 아두이노 프로그램 창이 열리면 바로 업로드 하셔도 됩니다.)



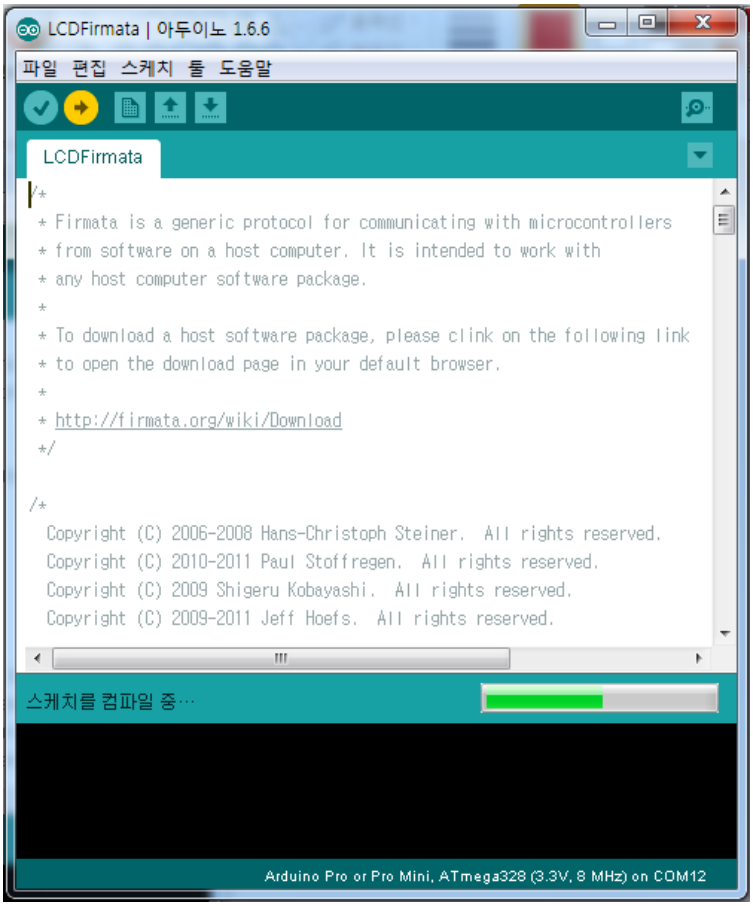
⑧ "LCDFirmata" 폴더 안에 이동된 "LCDFirmata.ino" 파일이 있습니다. 이 파일을 두 번 클릭하여 아두이노를 실행합니다.



⑨ 이제 아두이노 프로그램에서 "LCDFirmata.ino" 을 컴파일하여 코딩키트로 업로드합니다. 아래 빨간색 동그라미의 화살표 버튼을 누릅니다. (이때, 아두이노와 코딩키트가 연결된 상태인지를 확인하세요)



⑩ 그러면, 다음과 같이 컴파일이 진행됩니다.

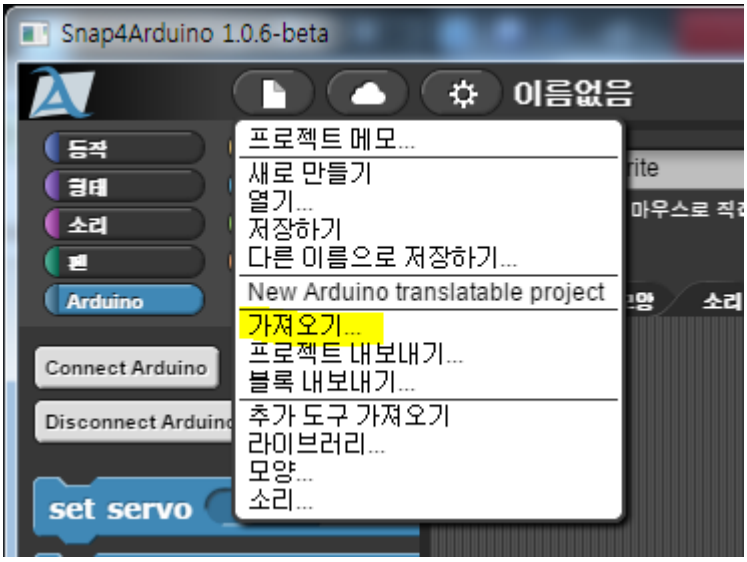


⑪ 컴파일이 성공적으로 진행되고, 아래 그림처럼 "업로드 완료"라는 메시지가 뜨면 업로드가 완료된 것입니다.

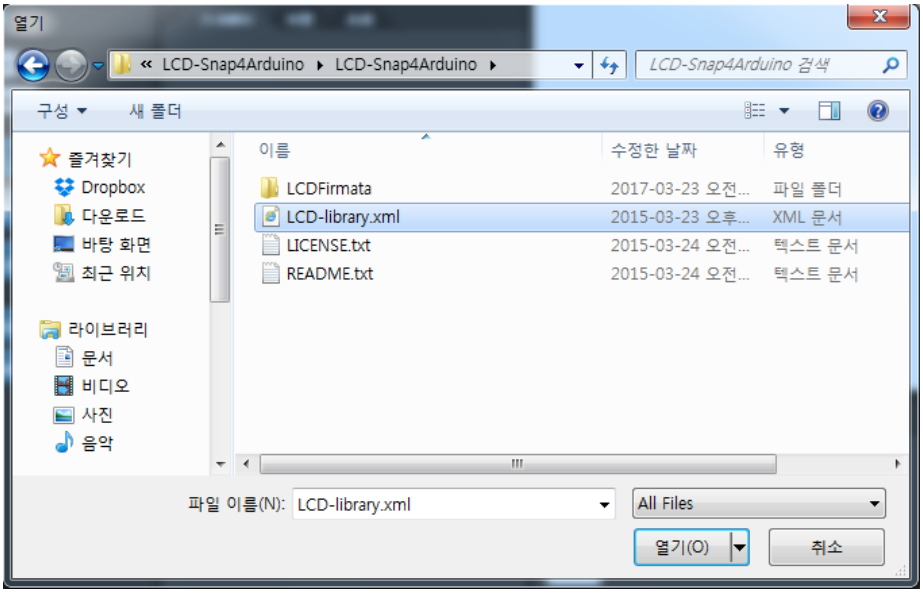
** snap4arduino 프로그램이 실행중에는 업로드가 안되고 포트를 사용중이라는 메시지가 나오는 경우가 있습니다. 이 때는 snap4arduino 프로그램을 종료하시고 다시 업로드를 해 주십시요.*

2 단계 : 새로운 LCD 블록들 가져오기

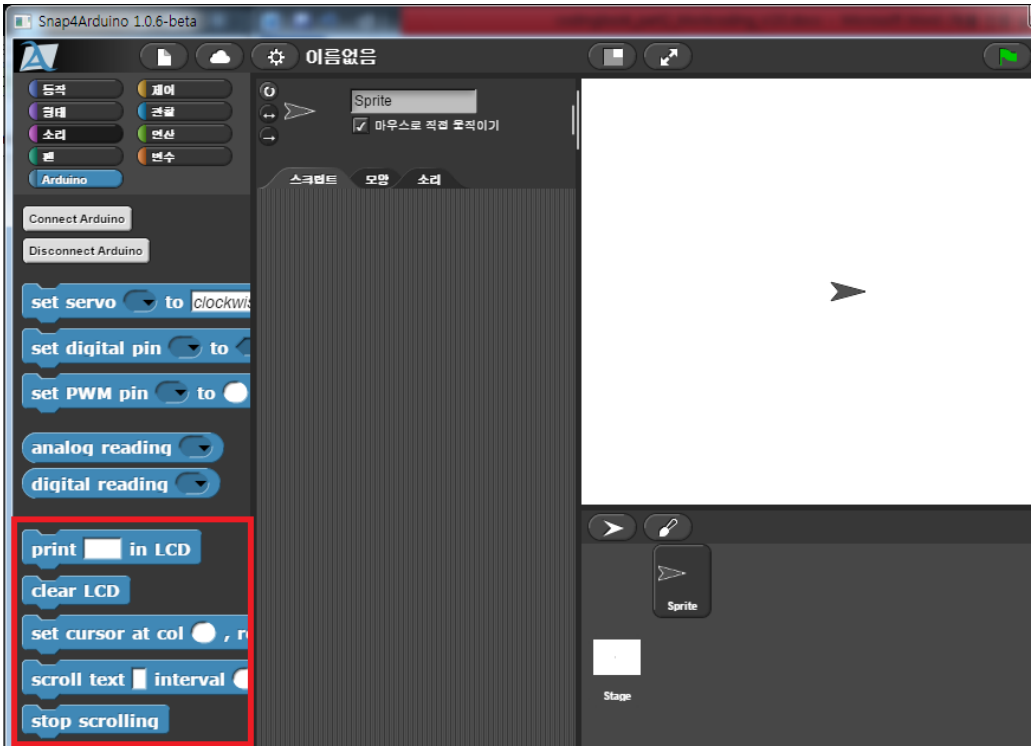
① "LCDFirmata.ino"가 코딩키트로 성공적으로 업로드가 되었다면, 아두이노 프로그램을 닫고 Snap4Arduino 프로그램을 실행 시킵니다. 프로그램의 상단의  버튼을 눌러, "가져오기..." 메뉴를 선택합니다.



② 위에서 다운 받은 LCD-Snap4Arduino.zip 파일에 있던 "LCD-library.xml" 파일을 선택하고, "열기" 버튼을 누릅니다.



③ Arduino 블록 모음으로 가보겠습니다. 다음 그림과 같이 새로운 블록들이 생긴 것을 볼 수 있을 것입니다. 이제 이 블록들을 사용하여 캐릭터 LCD에 문자들을 출력할 수 있습니다.



3 단계 : 변수 만들기

① 변수 블록모음에서 변수 만들기 버튼을 클릭하여 LCD 변수를 만듭니다.

② 제어 블록 모음에서 클릭했을 때 블록을, 변수 블록모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다.

블록을 사용하여, 

변수에 캐릭터 LCD 의 백라이트(Back Light) 핀 번호인 13 을 저장하도록 합니다. 코딩키트에 장착된 캐릭터 LCD 는 문자를 보여주고 이 문자가 밝게 보일 수 있도록 해 주는 백라이트가 있는 제품입니다.

4 단계 : 캐릭터 LCD 에 문자열 출력하기

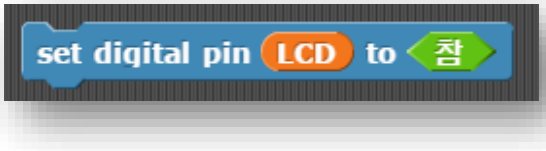
① 

블록 모음에서 

블록을 가져와 

변수에 

값을 설정하는 블록을 만듭니다.



② 

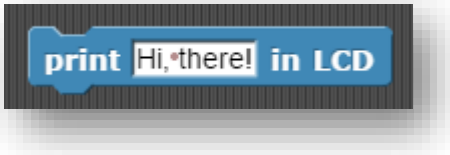
블록 모음에서 

블록을 가져와, 문자를 어느 위치에 출력할 지를 설정합니다. 첫 번째 행(row)의 맨 첫 번째 열(col)에 출력하기 위해, 각 행과 열 값으로 0 값을 입력합니다.

③ 출력할 문자들을 

블록에 입력해 줍니다. 이 블록도 

블록 모음에 있습니다. "Hi, there!" 이라고 입력해 보겠습니다.



④ 한 2 초 정도 출력상태를 유지하도록 할까요? 그럼, 

블록 모음에서 

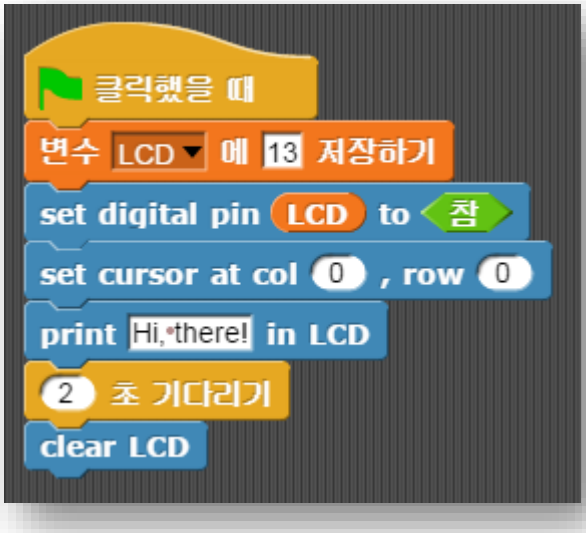
블록을 가져와 "2" 초를 기다릴 수 있도록 수정합니다.

⑤ 이제 다시 LCD 화면에 출력되었던 문자들을 지우겠습니다. 

블록 모음에서 

블록을 가져옵니다.

⑥ 위 ①~⑤에서 만들어진 블록들을 순서대로 연결하면 다음과 같이 완성이 됩니다.



⑦ 이제  버튼을 눌러 실행시켜 보세요. 캐릭터 LCD 화면에 "Hi, there!" 이라는 글자가 2 초간 출력되고 사라질 것입니다.

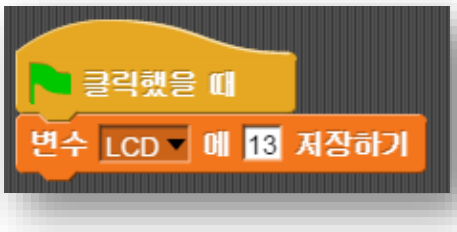
17-2.전자 시계를 만들자

전자 시계를 만들어 보겠습니다. 시간은 물론 날짜도 나오는 시계라면 좋을 것 같습니다. Snap4Arduino 에서 날짜와 시간을 알려주는 블록이 제공됩니다. 이 블록을 사용하여, 캐릭터 LCD 에 날짜와 시간을 화면으로 보여지도록 하겠습니다.

1 단계 : 변수 만들기

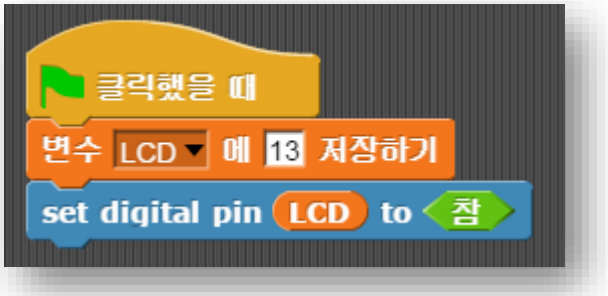
① 변수 블록모음에서 변수 만들기 버튼을 클릭하여 LCD 변수를 만듭니다.

② 제어 블록 모음에서 클릭했을 때 블록을, 변수 블록모음에서 변수 에 0 저장하기 블록을 스크립트로 가져옵니다. 변수 에 0 저장하기 블록을 사용하여, LCD 변수에 캐릭터 LCD 의 백라이트 핀 번호인 13 을 저장하도록 합니다.

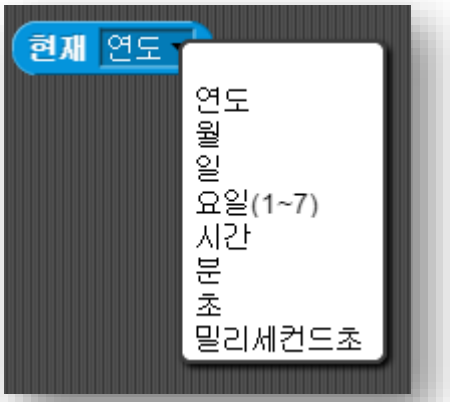


2 단계 : 날짜와 시간 출력하기

① Arduino 블록 모음에서 set digital pin to 블록을 가져와 LCD 변수에 참 값을 설정하는 블록을 만듭니다.

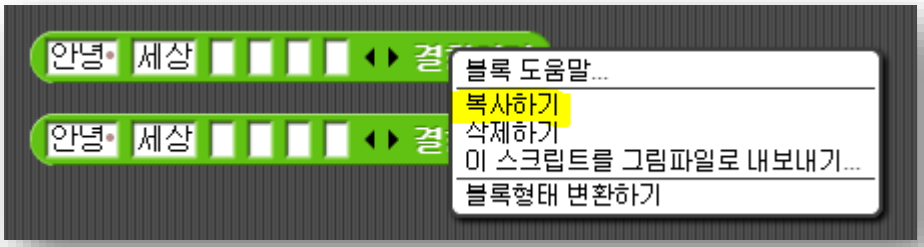


② 블록 모음에서 현재 일 을 가져옵니다. 이 블록에서 아래 화살표를 클릭하면, 다음과 같이 년, 월, 일, 시, 분, 초 등의 값을 얻어 올 수 있습니다.



③ 년, 월, 일, 시, 분, 초 값을 가져와야 하므로, 모두 6 개의 현재 일 블록을 가져와서, 해당 값들을 가져오도록 설정합니다.

④ 날짜와 시간 값들을 "DATE 년/월/일"과 "TIME 시간/분/초" 형식으로 만들려고 합니다. 연산 블록 모음에서 안녕 세상 결합하기 블록을 가져옵니다. 이 블록에 있는 왼쪽 화살표를 누르면 입력 칸이 줄고, 오른쪽 화살표를 누르면 입력 칸이 늘어납니다. 안녕 세상 결합하기 블록의 입력 칸이 모두 6 개가 되도록 오른쪽 화살표를 눌러줍니다. 같은 블록을 한 개 더 복사합니다.

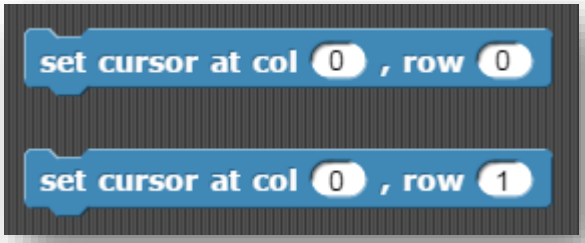


⑤ 두 안녕 세상 결합하기 블록에 입력되어 있는 "안녕"과 "세상" 글자는 지워줍니다. 맨 처음 입력 칸에 "DATE "와 "TIME "를 각각 입력해 줍니다. 년, 월, 일 값은 "/"로 구분하고, 시, 분, 초 값은 ":"로 구분 하도록 3 번째 5 번째 입력 칸에 각각 "/"과 ":" 문자를 넣어 줍니다. 그리고, 2 번째, 4 번째, 5 번째 입력 칸에는 ③에서 만들어진 날짜와 시간 블록들을 다음과 같이 넣어 줍니다.

⑥ 날짜와 시간을 나타내주는 블록이 다음 그림과 같이 완성되셨나요? 이제 날짜와 시간 값을 "DATE 년/월/일", "TIME 시/분/초" 형식으로 가져오게 됩니다.

⑦ LCD 에 날짜와 시간 값을 출력해 주기 위해서, Arduino 블록 모음에서 print in LCD 블록 두 개를 가져와서 위 ⑥에서 만든 블록들을 다음 그림과 같이 각각 넣어 줍니다. (만약 "새로 만들기"로 프로젝트를 새로 만드셨다면 이 블록들이 안 보일 것입니다. 이 때는 다시 한 번 더 "LCD-library.xml"을 "가져오기" 하셔야 합니다.)

⑧ 이제 위 날짜와 시간 값이 출력될 위치를 정해줍니다. Arduino 블록 모음에서 set cursor at col , row 블록을 두 개를 가져옵니다. 하나는 col 과 row 에 모두 "0" 값을 입력하고, 다른 하나는 col 에 "0" 값을, row 에 "1" 값을 입력합니다. col 은 칸을, row 는 줄을 의미합니다.

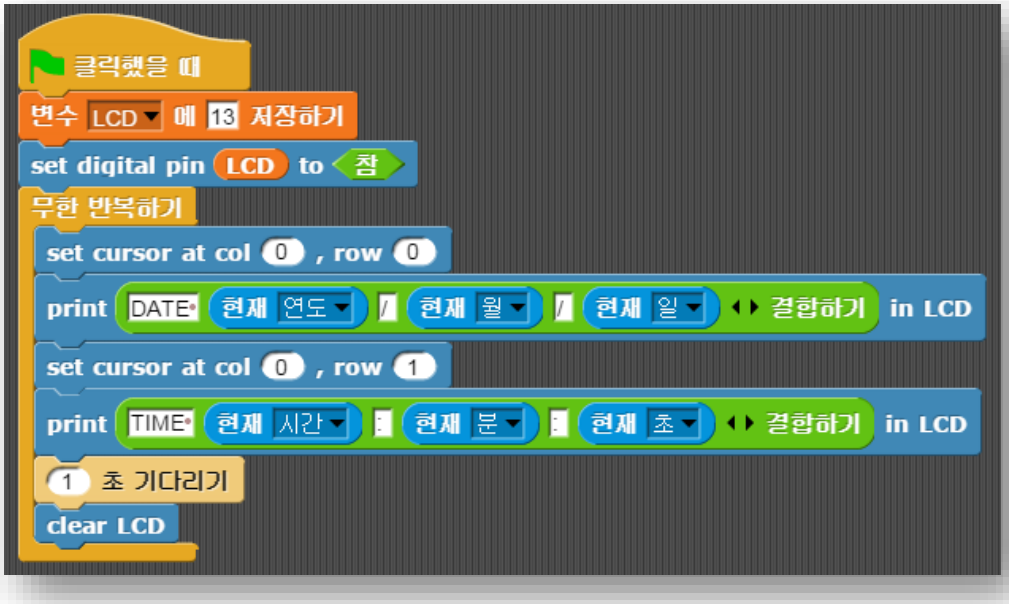


⑨ ⑥과 ⑦에서 만든 블록을 다음 순서로 연결합니다. "DATE 년/월/일"은 LCD 의 첫 번째 줄에, "TIME 시:분:초"는 두 번째 줄에 출력될 것입니다.

⑩ ①과 ⑧의 과정에서 만들어진 블록을 연결하면 다음 그림과 같습니다. 여기서  버튼을 눌러 실행을 시켜보세요. 현재 날짜와 시간이 표시되고 바로 동작이 멈출 것입니다.

⑪ 시간을 계속 반복하여 알려주어야 하므로, 제어 블록 모음에서 무한 반복하기 가져와서 날짜와 시간을 출력해주는 부분을 반복할 수 있도록 해 줍니다. 다시 버튼을 눌러 실행을 시켜 보세요. 잘 동작이 되나요? 처음에는 초 단위로 시간이 잘 가는 것처럼 보입니다. 그런데, 59 초에서 다음 0 초 1 초로 넘어갈 때 0 초, 1 초 가 아니고, 09 초, 19 초, 29 초로 변하는 것을 발견하실 수 있습니다. 이전 화면이 지워지지 않고 그대로 남아있기 때문입니다.

⑫ 이 문제를 해결하기 위해서 Arduino 블록 모음에서 clear LCD 블록을 가져옵니다. 그리고, 제어 블록 모음에서 1 초 기다리기 블록도 가져옵니다. 화면이 너무 빠른 속도로 지워지면 시간을 확인할 수 없게 되므로, 1 초마다 화면을 바꾸도록 합니다. 다음 그림과 같이 완성하셨나요? 그럼 버튼을 눌러 보세요. 시계가 잘 돌아가지요? ^^


클릭했을 때
변수 LCD 에 13 저장하기
set digital pin LCD to 참
무한 반복하기
set cursor at col 0 , row 0
print DATE 현재 연도 / 현재 월 / 현재 일 결합하기 in LCD
set cursor at col 0 , row 1
print TIME 현재 시간 : 현재 분 : 현재 초 결합하기 in LCD
1 초 기다리기
clear LCD


DATE 2017/4/3
TIME 10:37:34

부록 A. 스피드 가이드

1. LED등 켜기

A. 변수

B. 완성된 스크립트

2. LED등 깜박이기

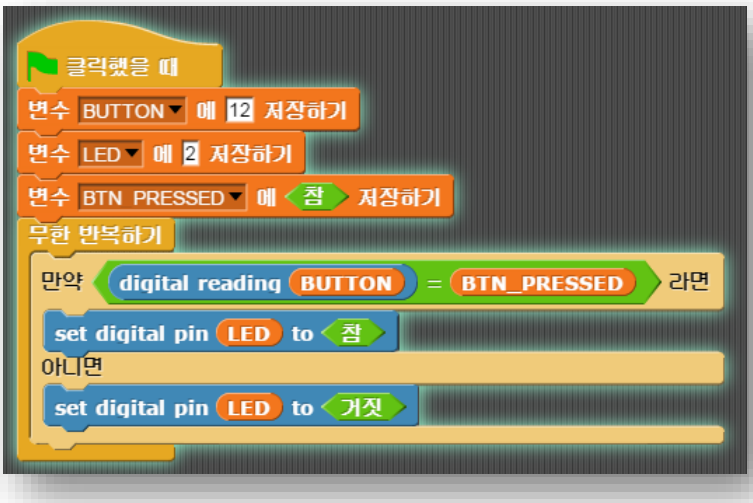
A. 변수

B. 완성된 스크립트

3. 버튼이 눌리면 LED 등 켜지기

A. 변수

B. 완성된 스크립트



4. 버튼이 눌리면 "딩동" 소리내기

A. 변수

B. 완성된 스크립트

5. 캐롤이 나오는 크리스마스 전등 만들기

A. 변수

B. 완성된 스크립트

6. 부저 소리 내기

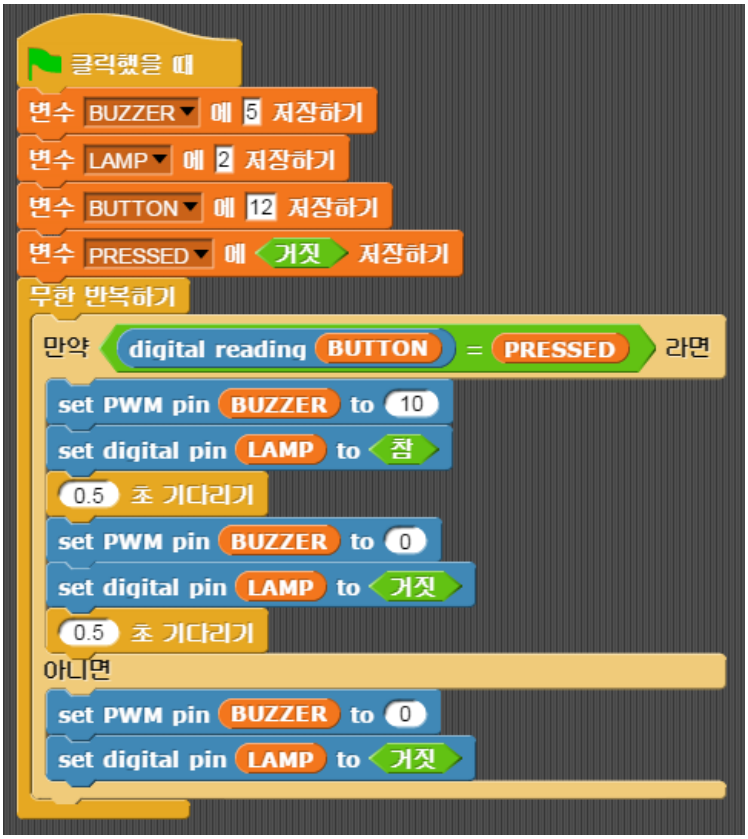
A. 변수

B. 완성된 스크립트

7. 경보기

A. 변수

B. 완성된 스크립트



8. 가변 저항으로 LED 밝기 조절하기

A. 변수

B. 완성된 스크립트

9. 가변 저항으로 LED 켜는 개수 조절하기

A. 변수

B. 완성된 스크립트

10. 바퀴를 움직여 보자

A. 변수

B. 완성된 스크립트

11. 엑셀과 브레이크를 만들자

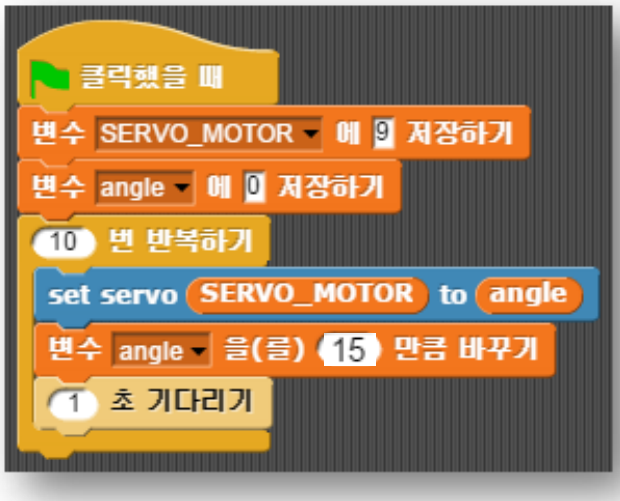
A. 변수

B. 완성된 스크립트

12. 서보 모터를 움직여 보자

A. 변수

B. 완성된 스크립트



13. 원격조정 로봇 팔을 만들자

A. 변수

B. 완성된 스크립트

14. 밝기 센서 값 읽어 오기

A. 변수

B. 완성된 스크립트

15. 인공 지능 전등을 만들자

A. 변수

B. 완성된 스크립트

16. 소리 센서 값 읽어 오기

A. 변수

B. 완성된 스크립트

17. 잠자는 고양이 깨우기

A. 변수

B. 완성된 스크립트

18. 온도 센서 값 읽어 오기

A. 변수

B. 완성된 스크립트

19. 더울 때는 선풍기, 추울 때는 난방기

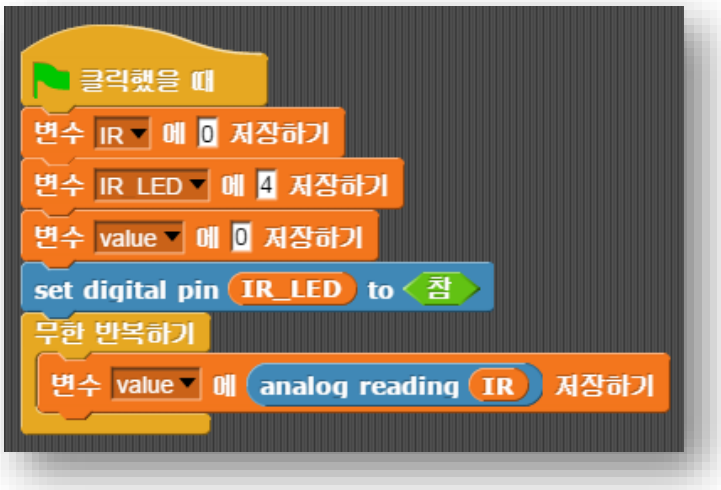
A. 변수

B. 완성된 스크립트

20. 꼼짝마! 적외선 센서

A. 변수

B. 완성된 스크립트



21. 도깨비집

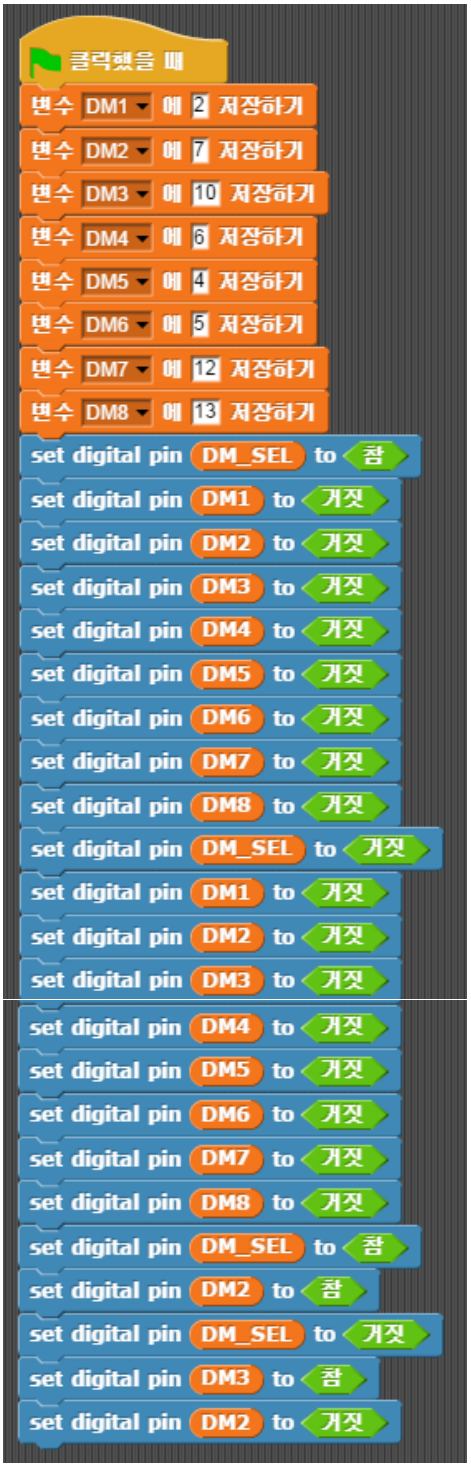
A. 변수

B. 완성된 스크립트

22. 도트매트릭스 LED에 불 켜기

A. 변수

B. 완성된 스크립트



23. 리스트로 도트매트릭스 다루기

A. 변수

B. 완성된 스크립트

24. 도트매트릭스의 LED를 순차적으로 켜고/끄기(세로줄)

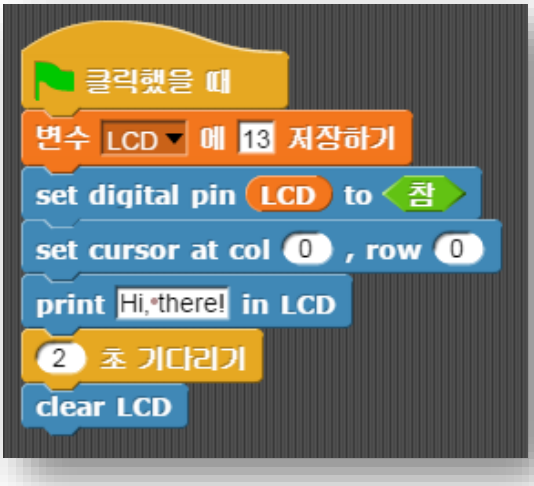
A. 변수

B. 완성된 스크립트

25. 캐릭터 LCD

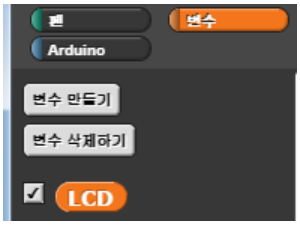
A. 변수

B. 완성된 스크립트

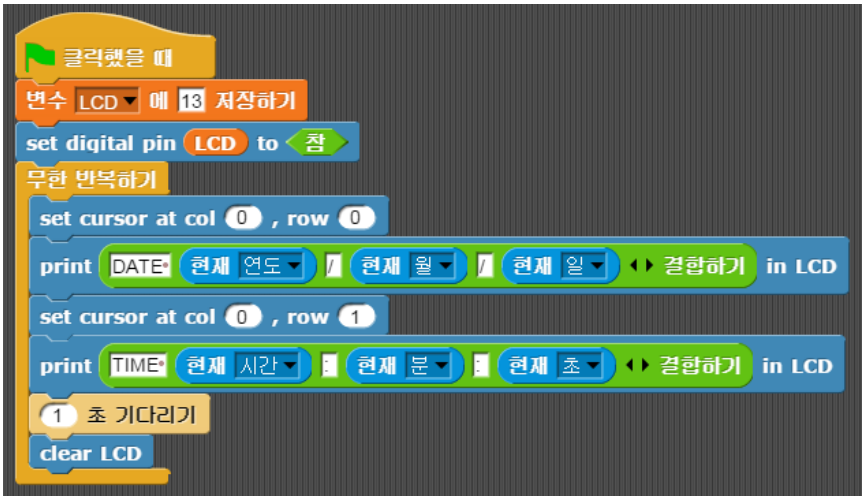


26. 전자시계를 만들자

A. 변수



B. 완성된 스크립트



부록 B. 생각하기 해답

생각하기 1. LED 끄기

A. 변수

B. 완성된 스크립트

생각하기 2. LED 두 개를 깜박이기

A. 변수

B. 완성된 스크립트

생각하기 3. 버튼이 눌리면 꺼지는 LED 등 만들기

A. 변수

B. 완성된 스크립트

생각하기 4. 소리가 다르게 나는 두 개의 벨 만들기

A. 변수

B. 완성된 스크립트

생각하기 5. "안녕하세요" 벨 만들기

A. 변수

B. 완성된 스크립트

생각하기 6. 버튼이 눌리면 음악이 바뀌는 크리스마스 전등

A. 변수

B. 완성된 스크립트

생각하기 7. 부저의 소리 높낮이를 변경해보기

A. 변수

B. 완성된 스크립트

생각하기 8. 모터를 시계 반대방향으로 돌리기

A. 변수

B. 완성된 스크립트

생각하기 9. 깃발을 좌우로 흔드는 로봇팔 만들기

A. 변수

B. 완성된 스크립트

생각하기 10. 밝기에 따라 조도가 조절되는 전등

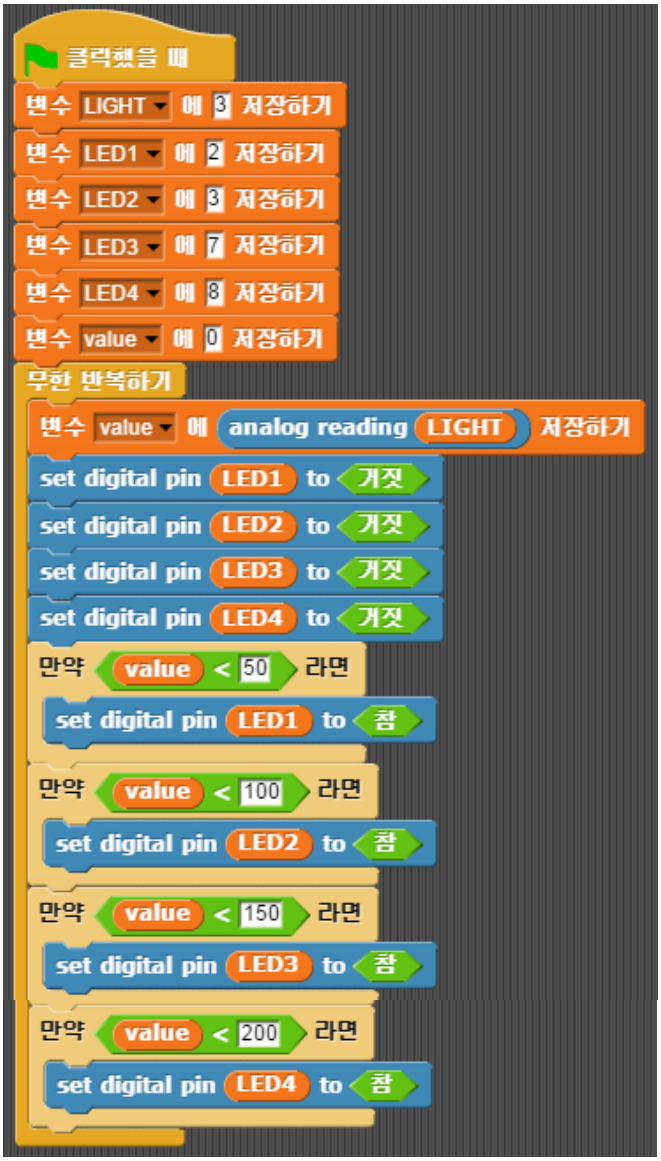
A. 변수

B. 완성된 스크립트

생각하기 11. 밝기에 따라 LED 조명이 켜지는 개수가 달라지는 전등

A. 변수

B. 완성된 스크립트



생각하기 12. 간단한 소음측정기

A. 변수

B. 완성된 스크립트

생각하기 13. 실내온도 자동조절장치

A. 변수

B. 완성된 스크립트

생각하기 14. 접근 감지 장치

A. 변수

B. 완성된 스크립트

생각하기 15. 도트 매트릭스의 모든 LED 점등하기

A. 변수

B. 완성된 스크립트

생각하기 16. 도트 매트릭스의 모든 LED 소등하기

A. 변수

B. 완성된 스크립트

생각하기 17. 도트 매트릭스의 LED 등을 순차적으로 켜고/끄기(가로줄)

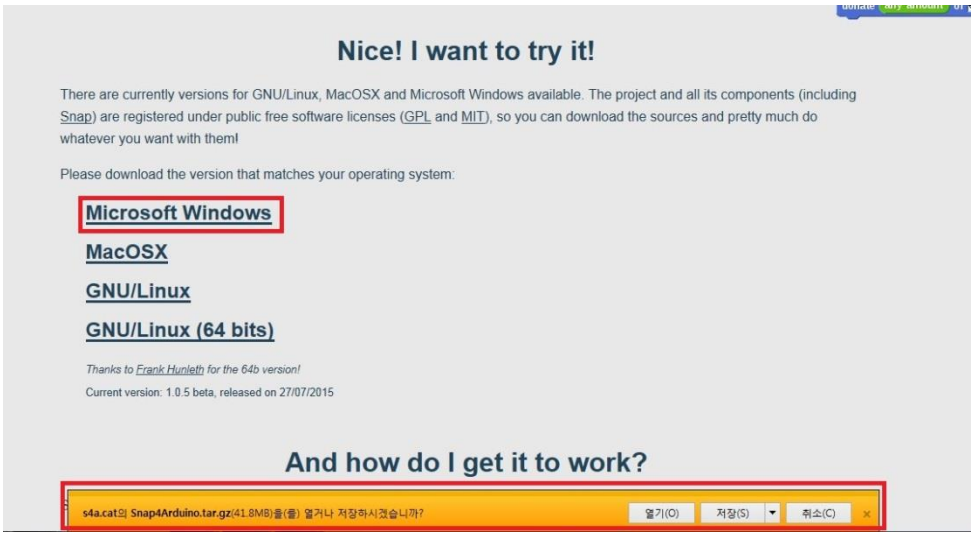
A. 변수

B. 완성된 스크립트

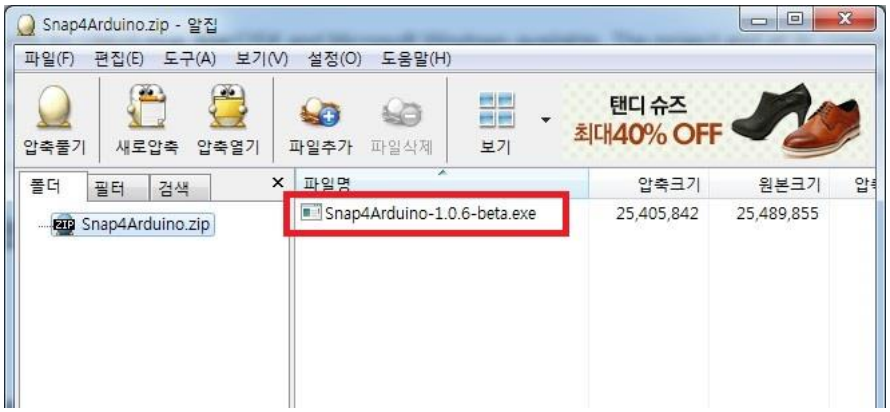
부록 C. 프로그램 설치 및 사용 가이드

1. 스냅 아두이노 프로그램 설치하기

스냅 아두이노(Snap4Arduino)는 http://s4a.cat/snap에서 무료로 자유롭게 다운받아서 사용할 수 있습니다. 자신의 컴퓨터 환경에 맞는 버전을 클릭하여 프로그램을 다운 받아 설치합니다. 본서에서는 윈도우 버전을 다운 받아서 사용해 보겠습니다.

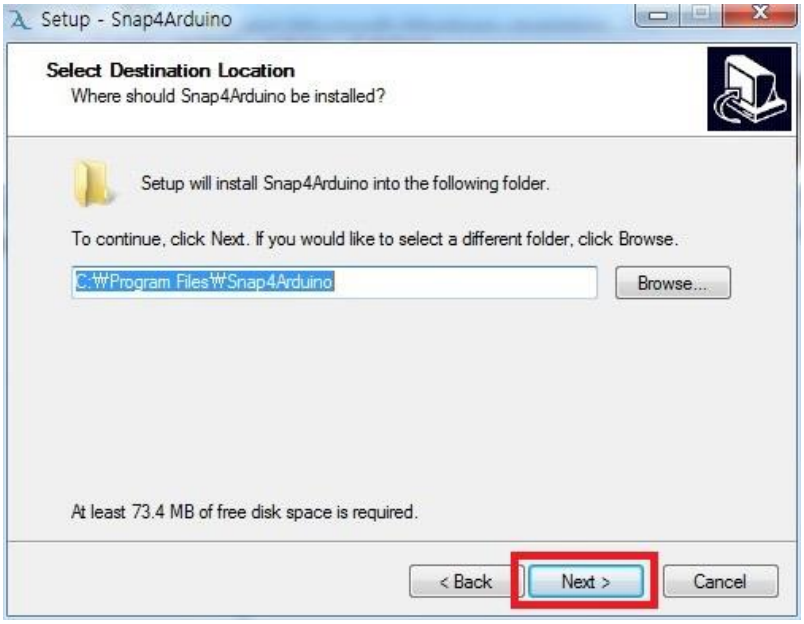


원하는 버전을 클릭하면, 위 그림과 같이 스냅 아두이노 프로그램의 설치 또는 저장 여부를 묻습니다. 열기를 눌러 바로 설치하거나, 저장 버튼 옆의 아래 화살표를 눌러 원하는 위치에 저장하여 설치합니다. 열기를 누르시거나, 다운 받은 파일을 두 번 클릭하면, 다음과 같이 압축을 푸는 창이 뜹니다. 그럼 "Snap4Arduino - 1.0.6beta.exe"를 선택하여 실행시켜 줍니다.

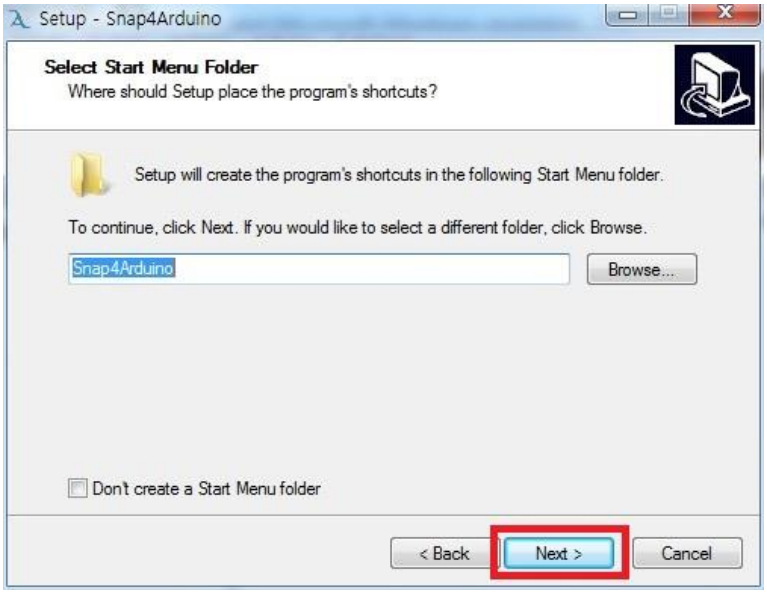


아래 그림과 같이 스냅 아두이노 설치가 시작됩니다. "Next" 버튼을 눌러 주세요.

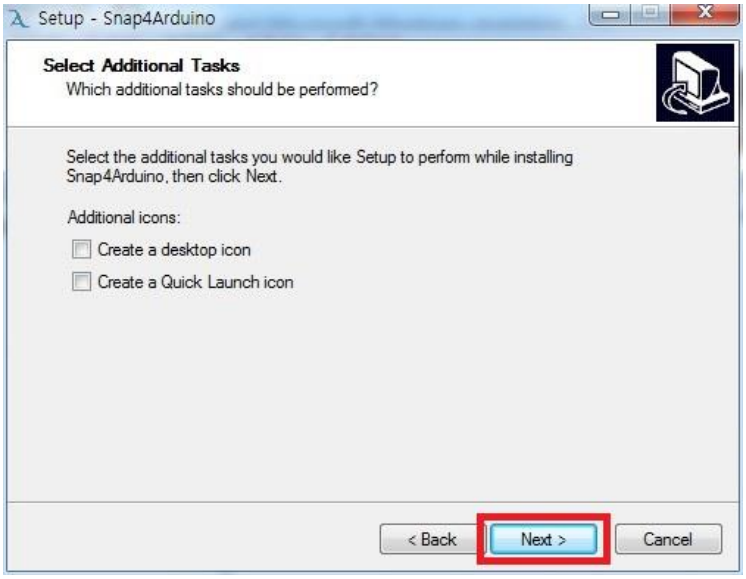
다음으로 프로그램을 설치 할 위치를 묻는 화면이 나옵니다. 설치를 원하시는 위치가 있다면 "Browse..."를 눌러서 위치를 선택하시고, 그렇지 않으면, "Next" 버튼을 눌러 설치를 진행합니다.



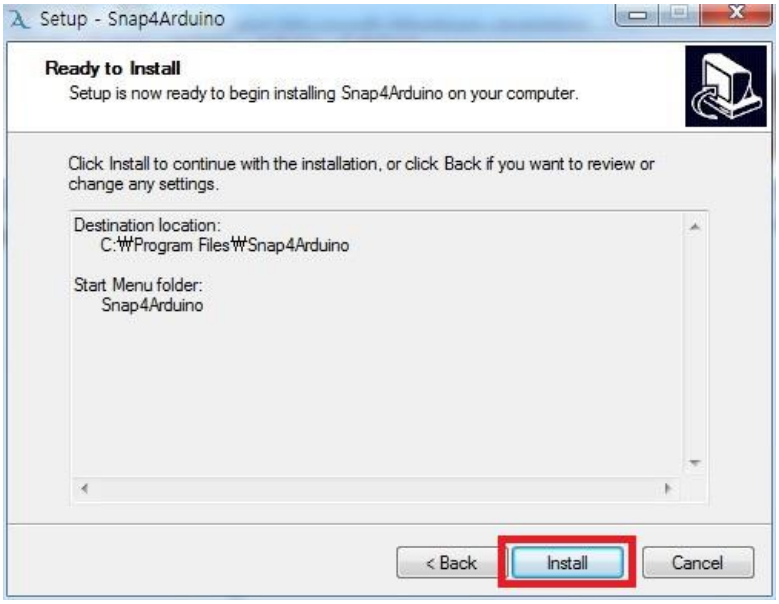
스냅 아두이노 프로그램을 시작 메뉴에 넣을 것인지 묻는 창이 나타납니다. 다른 폴더에 넣으시려면 "Browse..." 버튼을 눌러 선택하시고, 시작 메뉴에 넣지 않고 싶다면, "Don't create a Start Menu folder"를 체크하시면 됩니다. 그냥 디폴트로 설치를 원하신다면, "Next" 버튼을 눌러 진행하세요.



다음으로 추가적인 설정을 할 수 있는 창이 나옵니다. 바탕 화면에 아이콘을 생성하기를 원한다면, "Create a desktop icon"을 체크하시면 됩니다. 그리고, 빨리 가기 아이콘을 생성하기를 원하신다면 "Create a Quick Launch icon"을 체크하세요. 아니면, 그냥 "Next" 버튼을 눌러 진행하시면 됩니다.

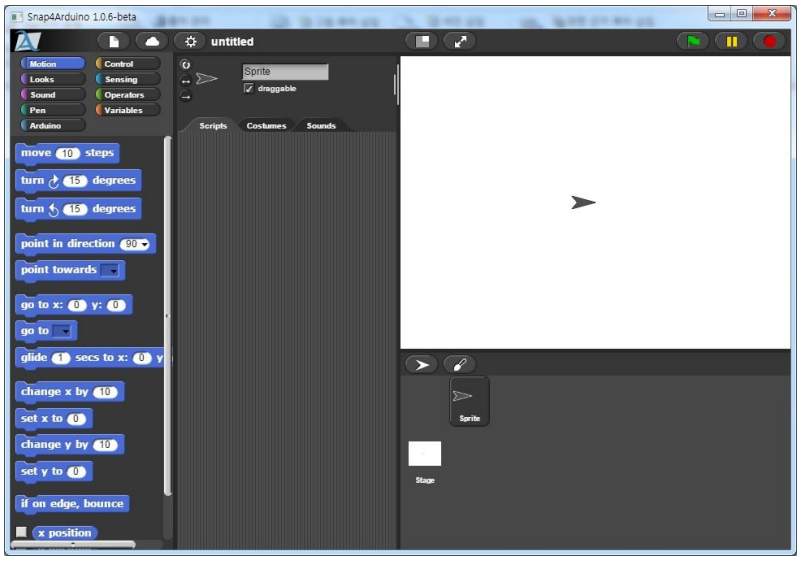


이제 프로그램을 설치할 준비가 끝났습니다. "install" 버튼을 누르시면, 프로그램이 설치가 됩니다.



여기까지 잘 따라오셨다면, 이제 "Finish" 버튼만 누르면 프로그램 설치가 완료됩니다. 설치 완료 후 스냅 아두이노를 실행시키기를 원하지 않는다면, 체크된 "Launch Snap4Arduino"를 해제하세요.

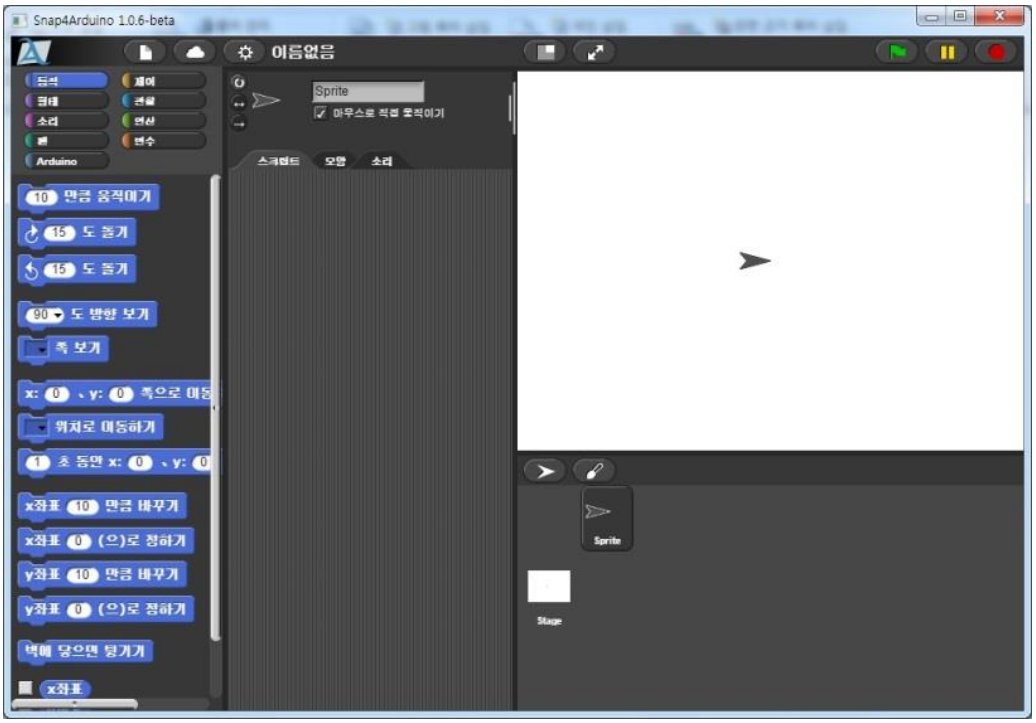
"Finish" 버튼을 누릅니다. "Launch Snap4Arduino"가 선택이 되어 있었다면, 스냅 아두이노가 아래와 같이 실행됩니다. 아니면, 프로그램 메뉴에서 Snap4Arduino를 찾으실 수 있을 것입니다. 스크래치 화면과 유사하지요? 사용 방법도 비슷합니다.



사용자가 보기 편하도록 사용하는 언어나, 글자 크기 등을 설정 할 수 있습니다. 다음 그림은 언어를 "한국어"로 설정하는 화면입니다. 톱니바퀴 모양의 그림이 그려진 버튼을 클릭하여,

"Language"를 선택하고, "한국어"를 선택하시면 화면이 한국어 표기로 바뀝니다. 글자 크기 등도 변경하여 사용하기 편한 상태로 설정해 보세요.

프로그램이 한글로 표기되니 훨씬 보기가 편해 지셨죠? ^^



2. 아두이노 프로그램 설치하기

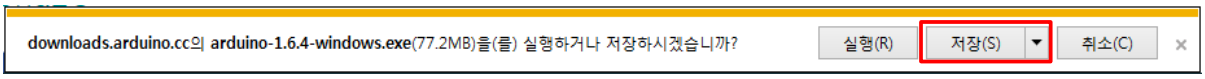
다음 사이트에서 프로그램을 다운 받습니다. (만약 컴퓨터에 아두이노 프로그램이 설치되어 있다면 이 부분은 지나치셔도 됩니다.)

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

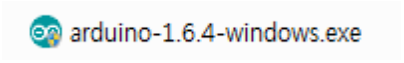
사이트를 방문하면 다음과 같은 화면이 보입니다. 여기서 Windows Installer 를 클릭합니다.

다음 화면이 나오면, JUST DOWNLOAD 를 클릭합니다.

그러면 다음과 같은 창이 브라우저 하단에 보입니다.



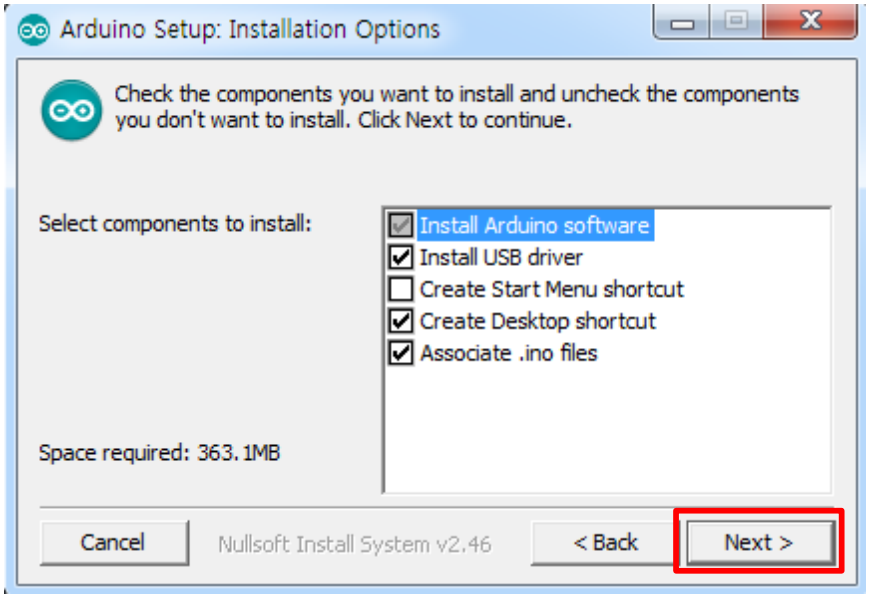
여기서 "저장" 버튼 중 오른쪽에 있는 아래쪽 방향 화살표를 누르면 "다른 이름으로 저장하기" 창이 나옵니다. 그러면 원하는 폴더를 선택하고 저장합니다. 저장한 폴더를 보면 다음과 같은 파일이 있습니다.



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

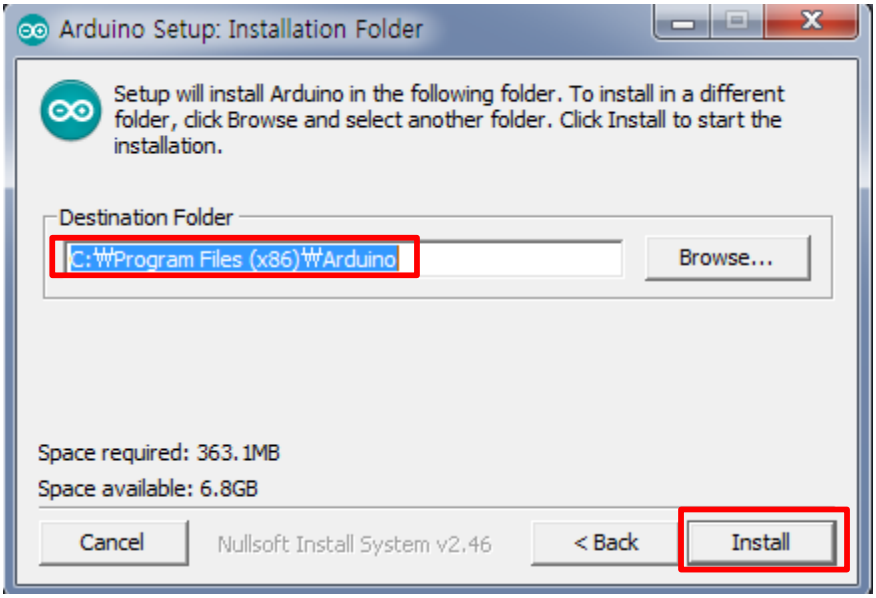
아두이노 프로그램을 설치하는 첫 화면은 다음과 같습니다.

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

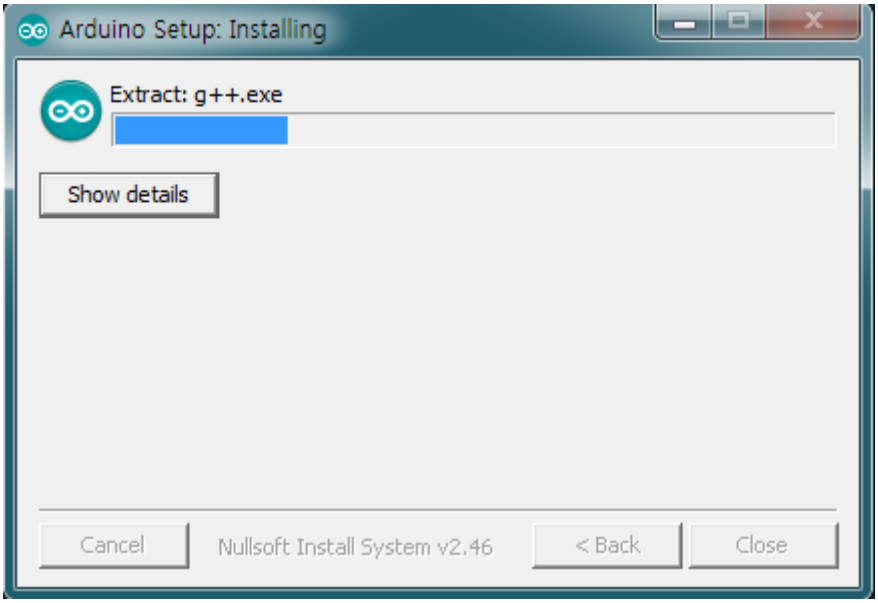


설치할 항목을 선택 합니다. 모든 것을 다 선택하셔도 되고, 위의 그림과 같이 선택하셔도 됩니다. "Next" 버튼을 클릭합니다.

다음으로, 설치할 폴더를 선택하라는 창이 나옵니다.

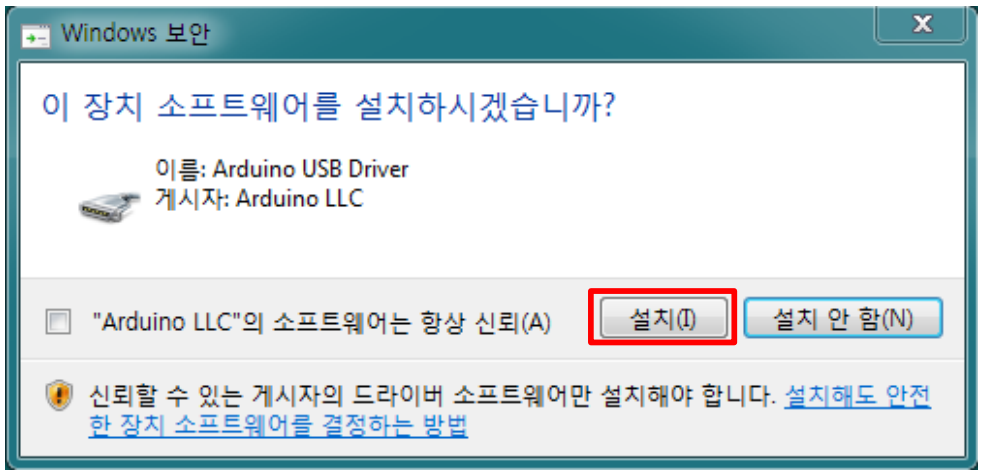


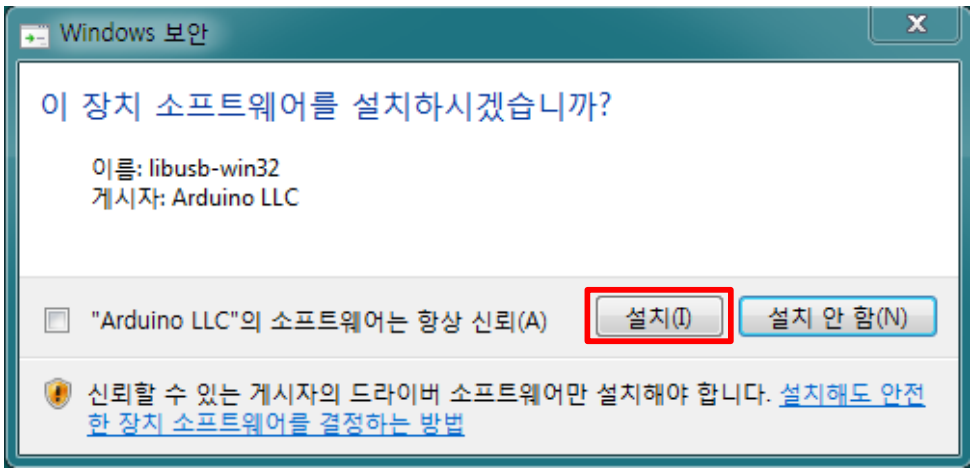
설치할 폴더를 쓰고 "Install" 버튼을 누릅니다. 이 때 기본 폴더를 선택하셔도 되고, "Browser..." 버튼을 이용하여 원하는 설치 폴더를 선택할 수 있습니다. 다음과 같은 설치 중인 화면이 보입니다. 약 2 ~ 3 분 정도 시간이 소요됩니다.



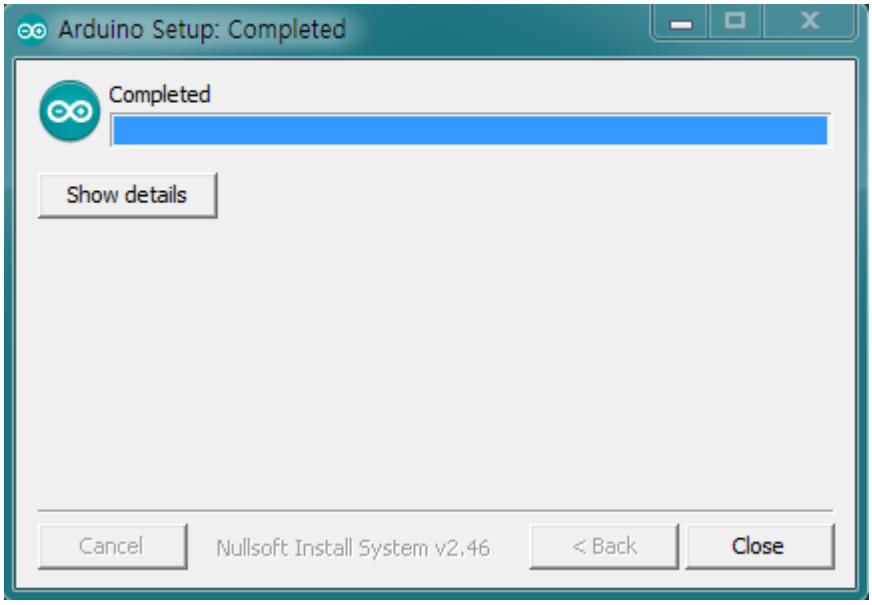
다음과 같이 중간에 설치를 물어 보는 창에서는 모두 "설치" 버튼을 클릭합니다. 이 창들은 모두 나올 수도 있고 이 중 한두 개만 나올 수도 있습니다.


Windows 보안
이 장치 소프트웨어를 설치하시겠습니까?
이름: Arduino USB Driver
게시자: Arduino srl
"Arduino srl"의 소프트웨어는 항상 신뢰(A)
설치(I)
설치 안 함(N)
신뢰할 수 있는 게시자의 드라이버 소프트웨어만 설치해야 합니다. 설치해도 안전한 장치 소프트웨어를 결정하는 방법


Windows 보안
이 장치 소프트웨어를 설치하시겠습니까?
이름: Arduino USB Driver
게시자: Arduino LLC
"Arduino LLC"의 소프트웨어는 항상 신뢰(A)
설치(I)
설치 안 함(N)
신뢰할 수 있는 게시자의 드라이버 소프트웨어만 설치해야 합니다. 설치해도 안전한 장치 소프트웨어를 결정하는 방법

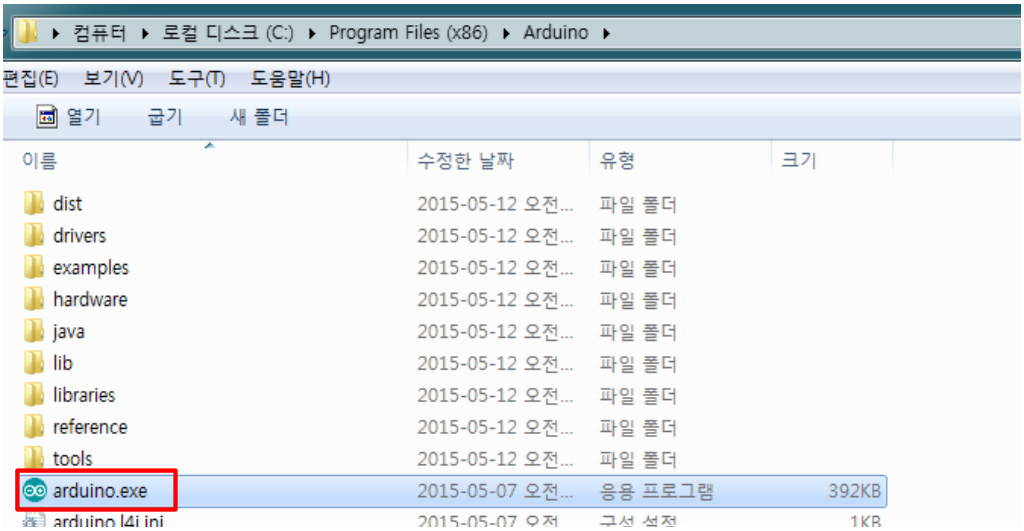

Windows 보안
이 장치 소프트웨어를 설치하시겠습니까?
이름: libusb-win32
게시자: Arduino LLC
"Arduino LLC"의 소프트웨어는 항상 신뢰(A)
설치(I)
설치 안 함(N)
신뢰할 수 있는 게시자의 드라이버 소프트웨어만 설치해야 합니다. 설치해도 안전한 장치 소프트웨어를 결정하는 방법

설치가 완료되면 다음과 같이 설치 완료 창이 표시됩니다.

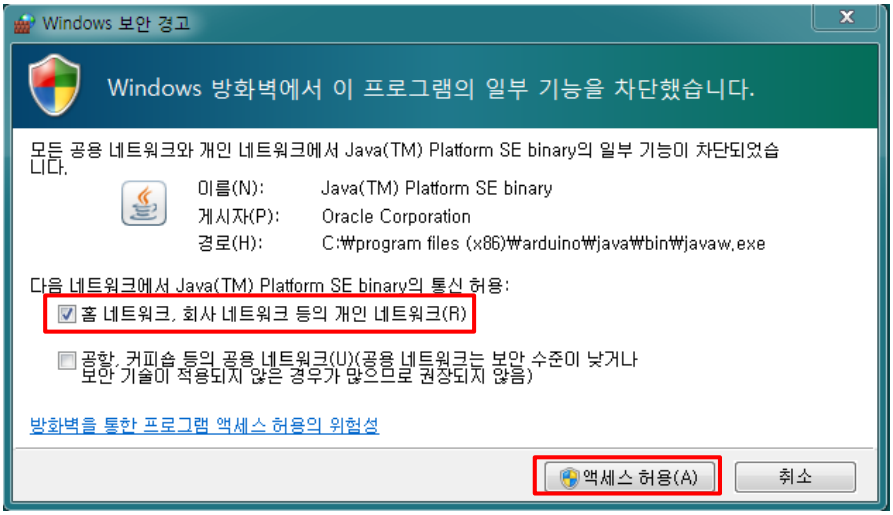


이제 바탕화면의 아두이노 아이콘을 더블 클릭하여 아두이노 프로그램을 실행합니다.

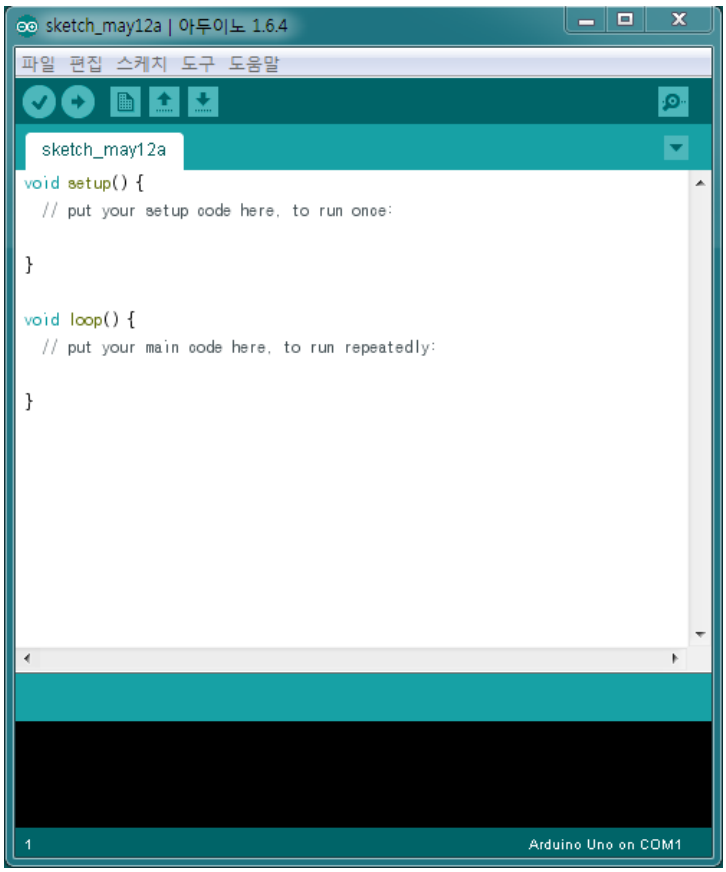
혹시 바탕화면에 아이콘이 없으면 설치 폴더로 가서 arduino.exe 를 실행합니다.



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

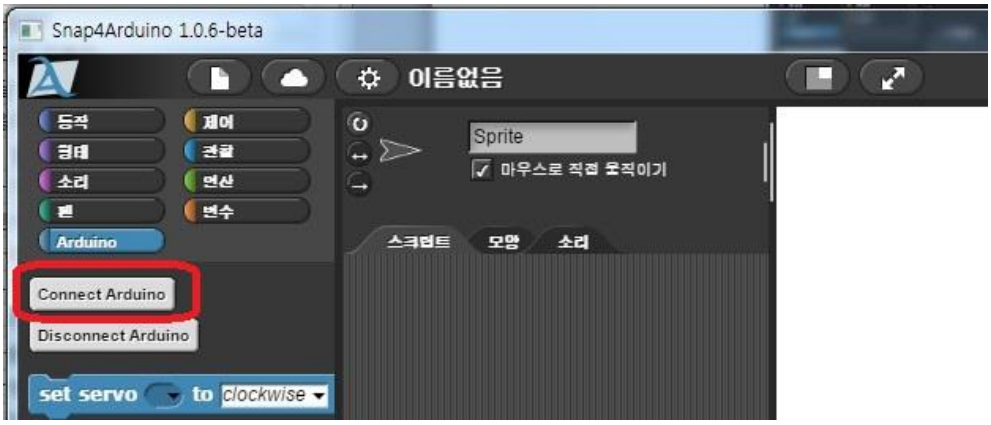


아두이노 프로그램 설치가 완료 되었습니다. 다음과 같이 아두이노 프로그램을 보실 수 있습니다.



3. 코딩키트와 연결하기

스냅 아두이노 프로그램과 아두이노 프로그램이 잘 설치가 되었다면, 이제 코딩키트와 연결해 볼까요? 스냅 아두이노 프로그램 화면의 좌측 상단에 보면 블록들의 모음들이 있습니다. Arduino 버튼을 클릭해 보시면, "Connect Arduino"라는 버튼이 있습니다. 이 버튼을 눌러볼까요?

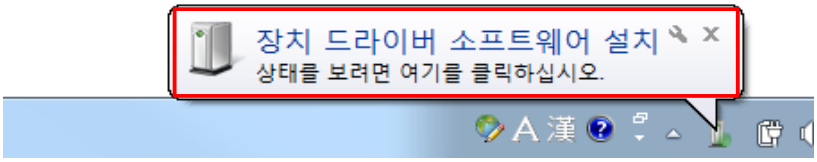


아마도, 다음과 같이 아두이노 보드를 찾을 수 없다는 메시지가 나올 것입니다.

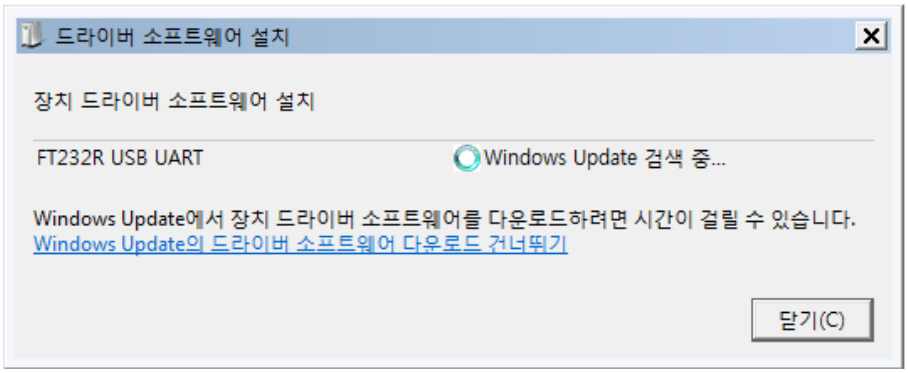
이 문제를 해결하기 위해 작업을 좀 해야 할 것 같습니다. 무엇부터 해야 할까요? 첫 번째로, 스냅 아두이노와 코딩키트와 물리적인 연결이 있어야 하겠죠? 다음 사진처럼, Snap4Arduino가 설치된 컴퓨터와 코딩키트를 USB 케이블로 연결합니다. 코딩키트에 전원 케이블도 연결해 줍니다.

그러면 세븐세그먼트가 "A00" 혹은 "r00", 또는 "b00" 값으로 깜박일 것입니다. 그리고 코딩키트와 PC를 처음 연결하는 것이라면, 다음과 같이 아두이노 드라이버 소프트웨어를 설치하기 위

한 준비를 할 것입니다.

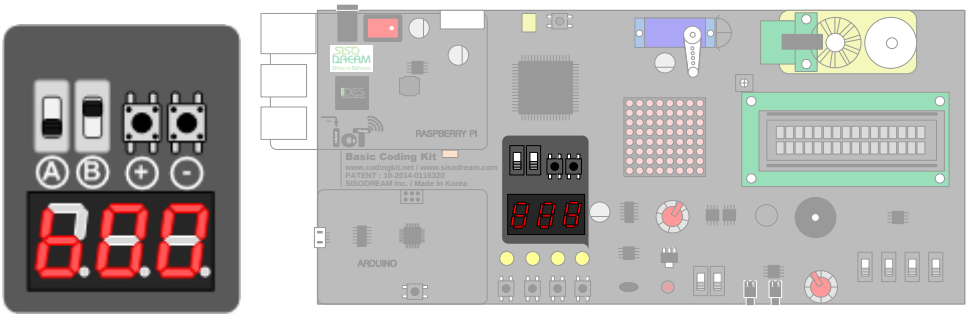


위의 그림의 붉은 박스 안을 클릭하면 다음과 같이 드라이버 소프트웨어를 자동으로 설치하는 화면을 보실 수 있습니다.



PC 의 설정은 이렇게 자동 이루어질 것입니다. 하지만 시간이 좀 오래 걸립니다. PC 설정이 이루어 지는 동안 코딩 키트의 스위칭 ID 를 설정해 주세요.

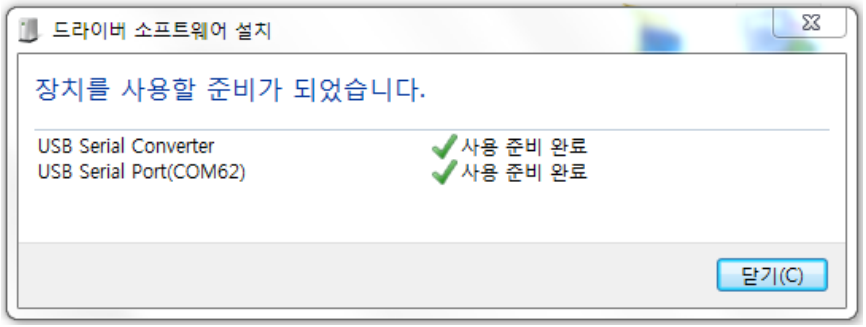
코딩 키트에 다음 그림과 같이 "MODE SETTING SWITCH" 가 있습니다. 스위치 A 는 아래로 내려주시고, B 스위치는 아래로 내립니다. 그리고, + 또는 – 버튼을 길게 1~2 초간 눌러주세요.



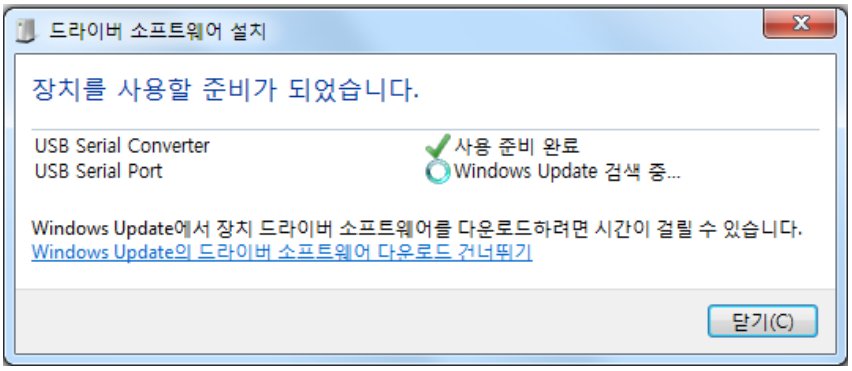
그러면, 세븐세그먼트의 깜박임이 멈춥니다. 세븐세그먼트에 표시되는 스위칭 ID 가 A00 상태인지 확인하세요. + 또는 – 버튼을 누르면 스위칭 ID 를 변경할 수 있습니다. 지금부터 우리가 공부할 스냅 아두이노의 코딩 놀이를 위한 스위칭 ID 는 "b00" 입니다.

** 코딩키트의 스위칭 시스템에 자세히 알고 싶으시다면, 교재 중 코딩키트 소개편을 참고하세요.*

이제 코딩키트의 설정은 완료가 되었고 PC 에서 아두이노 드라이버 소프트웨어 설치가 완료되었는지를 확인합니다. 다음과 같이 "드라이버 소프트웨어 설치"에서 모두 사용 준비 완료가 표시되면 드라이버 설치가 완료된 것입니다.



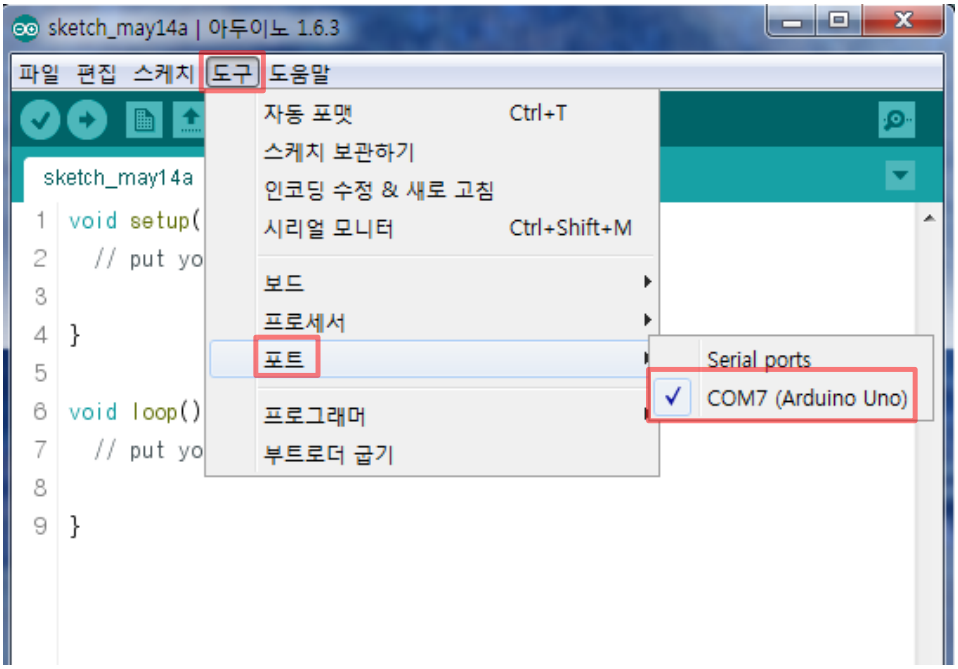
위의 그림과 같지 않고 다음과 같이 아직도 설치 중이라면 조금 더 기다립니다.



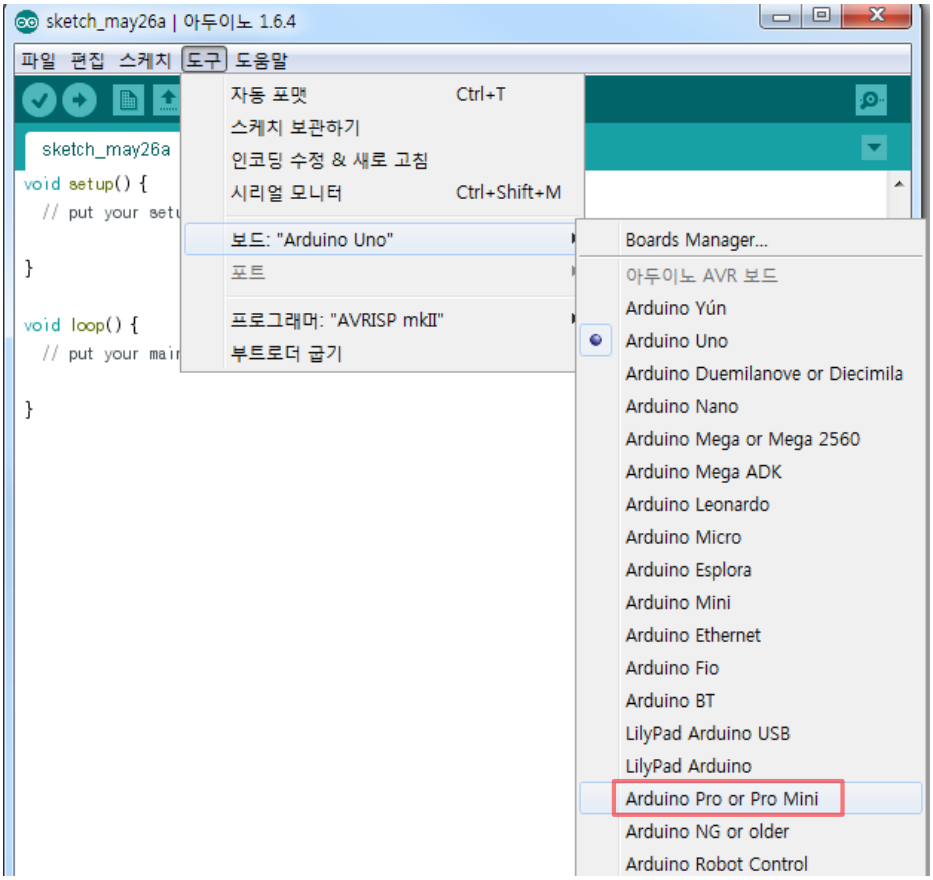
설치가 완료 되면 다음과 같이 아두이노 프로그램의 "도구 → 포트" 메뉴에서 해당 COM 포트를 확인할 수 있다. 이 포트를 선택해 줍니다. 그림에서는 COM7 이지만 다른 포트로 잡혀 있다면 다른 포트를 선택하여 주십시요.

Note : 위의 설명은 Windows7 을 기준으로 설명하였습니다. Windows8 에서의 사용은 코딩 사이트의 관련 게시물을 참고해 주십시요. 코딩 키트는 Windows7 과 Windows8 만을 지원합니다.

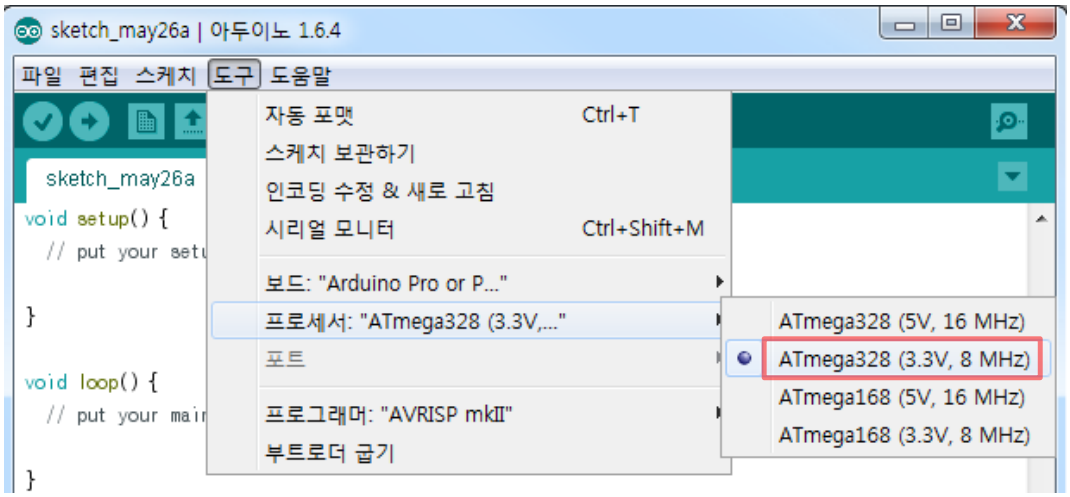
Note : 만약 한참을 기다려도 드라이버 소프트웨어가 자동으로 설치되지 않을 경우 코딩 사이트에 문의해 주십시요.



다음 그림과 같이 "도구 → 보드 → Arduino Pro or Pro Mini" 를 선택합니다.

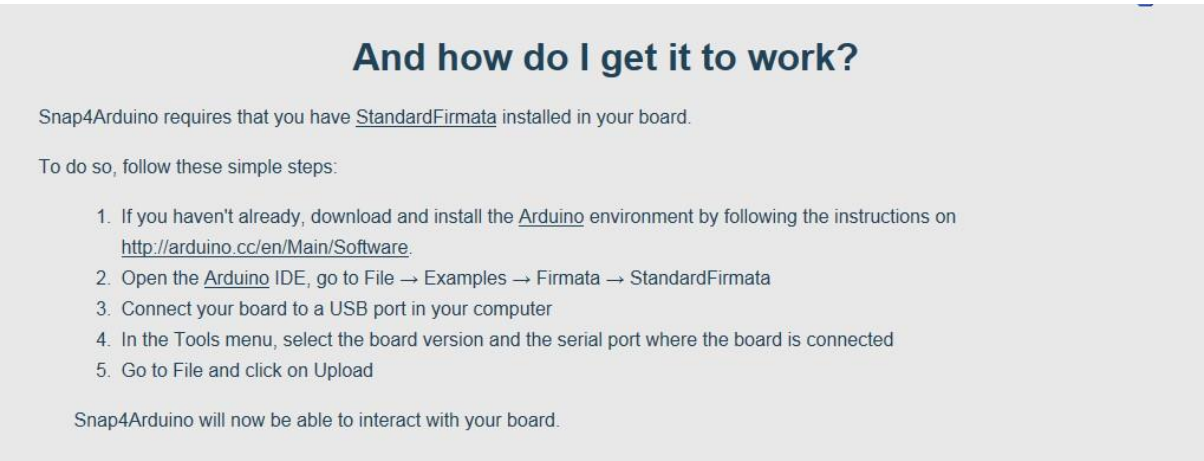


다음 그림과 같이 "도구 → 프로세서 → ATmega328 (3.3V, 8MHz)" 를 선택합니다.



4. 코딩키트에 StandardFirmata 설치하기

여기까지 준비가 되셨다면, 코딩키트에 StandardFirmata를 설치하시면 됩니다. 자, 그럼 앞에서 스냅 아두이노를 다운받았던 사이트로(http://s4a.cat/snap) 다시 돌아가보겠습니다. StandardFirmata를 설치하는 방법이 자세히 설명되어 있습니다.



And how do I get it to work?

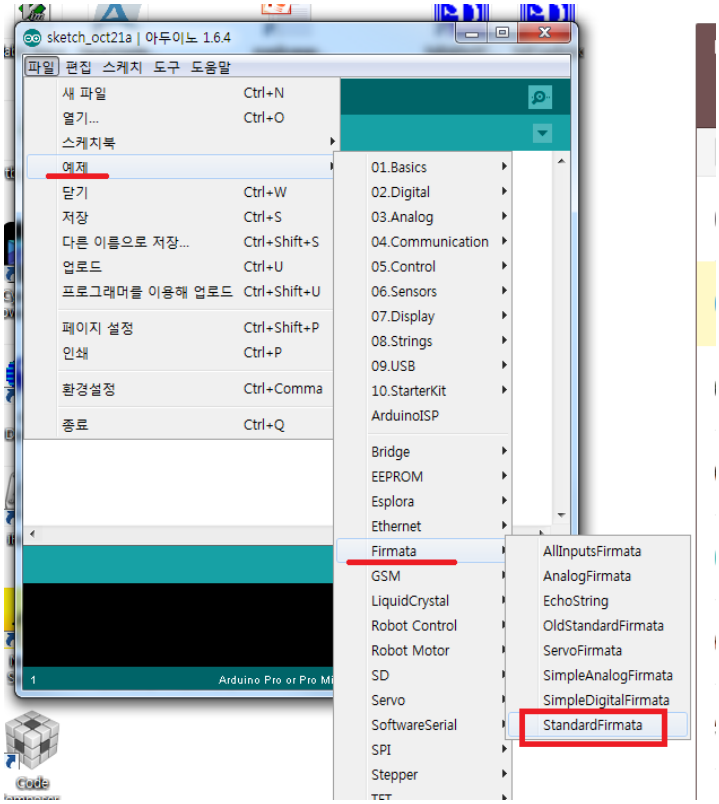
Snap4Arduino requires that you have StandardFirmata installed in your board.

To do so, follow these simple steps:

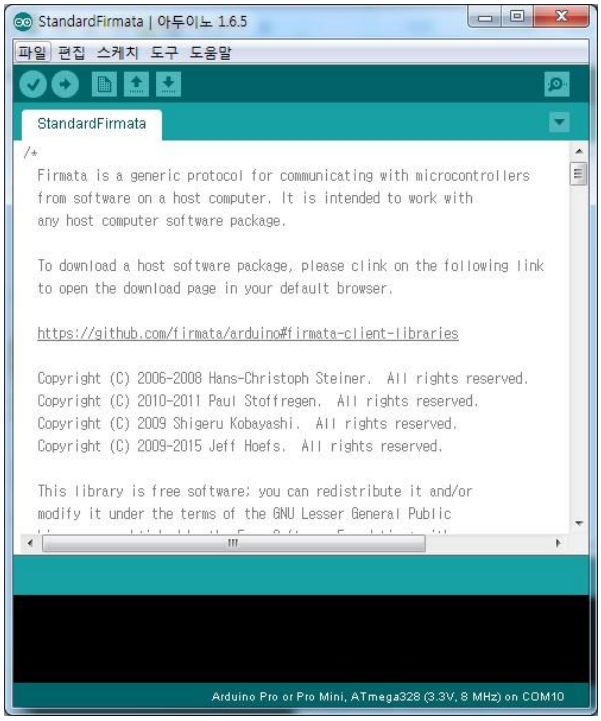
1. If you haven't already, download and install the Arduino environment by following the instructions on http://arduino.cc/en/Main/Software.
2. Open the Arduino IDE, go to File → Examples → Firmata → StandardFirmata
3. Connect your board to a USB port in your computer
4. In the Tools menu, select the board version and the serial port where the board is connected
5. Go to File and click on Upload

Snap4Arduino will now be able to interact with your board.

아두이노 프로그램을 실행합니다. 다음으로 "파일" 메뉴에서 "예제" → "Firmata" → "StandardFirmata"를 선택합니다.

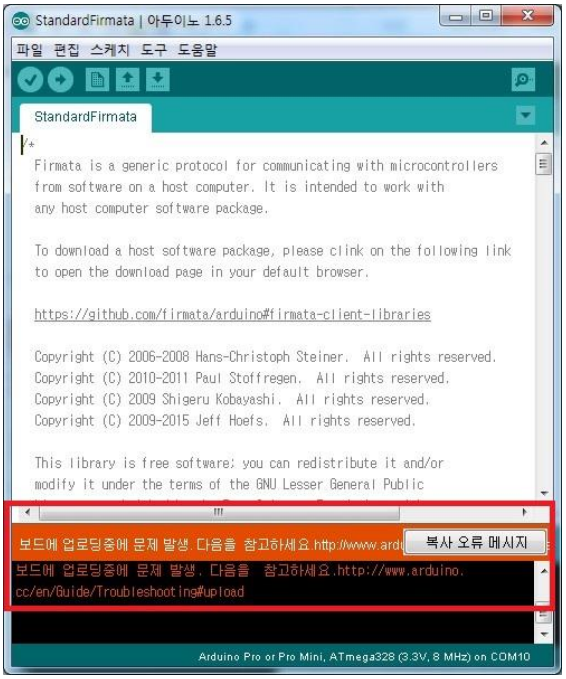


그러면, 다음과 같이 StandardFirmata.ino 파일을 열 수 있습니다.

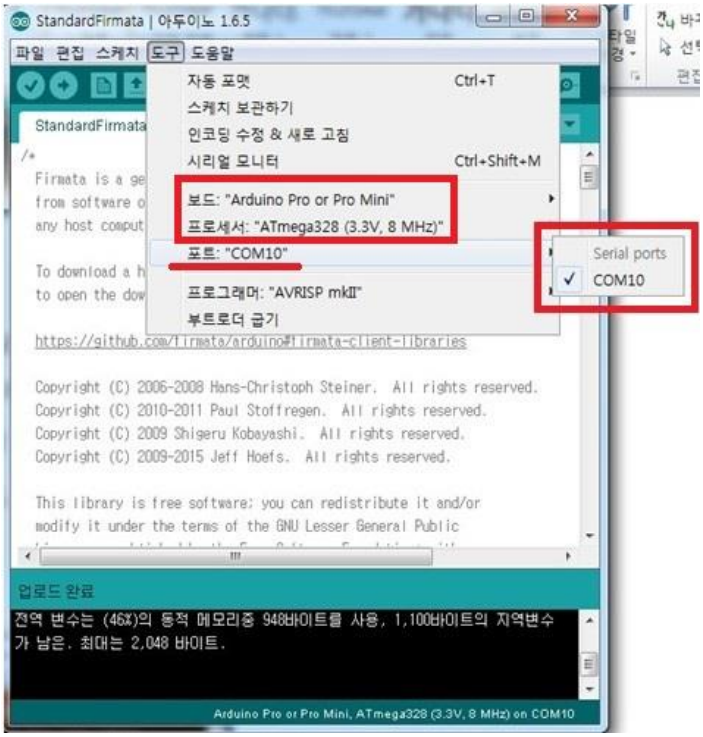


다음으로, 화살표 그림이 그려진 동그란 버튼을 눌러 StandardFirmata 를 코딩키트로 업로드 시킵니다.

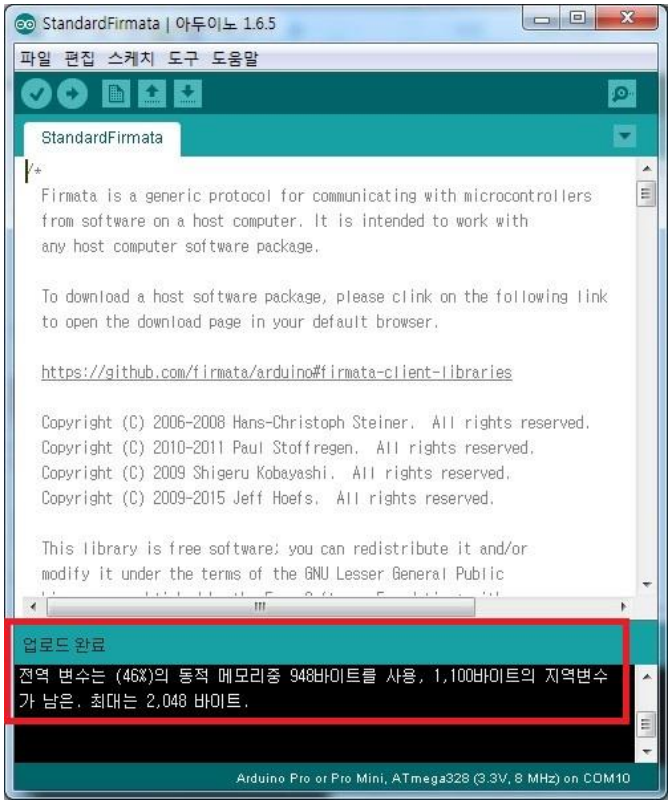
혹시 아래 그림과 같은 복사 오류 메시지가 나오셨나요? 그러면, 코딩키트와 컴퓨터의 연결을 살펴봐주세요. 코딩키트의 전원이 켜져 있어야 하고, 컴퓨터의 USB 포트와 코딩키트가 USB 케이블로 연결되어 있어야 합니다.



"도구" 메뉴에서 보드 버전과 시리얼 포트의 설정도 확인해 주세요.



정상적으로 StandardFirmata 가 코딩키트로 업로드 되었다면, 다음과 같이 업로드 완료 화면을 확인하실 수 있습니다.



다시 스냅 아두이노 프로그램으로 돌아가겠습니다. "Arduino"블록으로 가서 "Connect Arduino" 버튼을 눌러주세요.

지금까지 잘 따라오셨다면, 다음과 같은 메시지가 화면에 뜹니다. "An Arduino board has been connected. Happy prototyping!"이라는 메시지를 받으셨다면, 코딩키트와 성공적으로 연결이 되었다는 의미입니다.

정말 연결이 잘 되었는지 확인해 볼까요? "set digital pin __ to __" 블록을 스크립트 화면으로 드래그 앤 드롭을 하여 가져다 놓습니다. 화살표가 그려진 빈칸을 마우스로 클릭해보세요. 그러면, 연결된 코딩키트에서 사용 가능한 디지털 핀들 목록을 보실 수 있습니다.

이제 모든 준비가 끝났습니다. 그럼, 코딩키트를 이용한 재미있는 하드웨어 프로그래밍 세계로 떠나볼까요? ^^

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	Switching ID 핀 번호					
디바이스	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

V2.1 : 2018. 1. 24

-. 66 페이지 : 추가

** snap4arduino 의 최근 출시 버전에서는* 

이 다음과 같이 하나로 바뀌었습니다.

-. 104~105 페이지 : 변경

서보 모터의 각도를 20도에서 15도로 변경

-. 107 ~ 108 페이지 : 변경

서보 모터의 값을 value/5 에서 value/6 으로 변경

-. 153 페이지 : 변경

스크립트 그림 변경 – DM_SEL 신호에 대한 초기화가 빠져 있었음

-. 171 페이지 : 변경

Snap4Arduino 사이트 주소 변경 : http://snap4arduino.rocks/