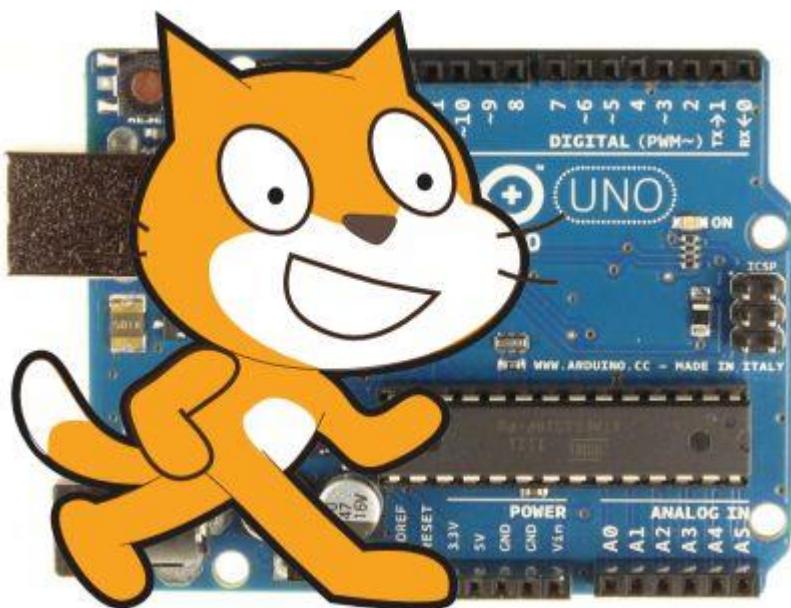
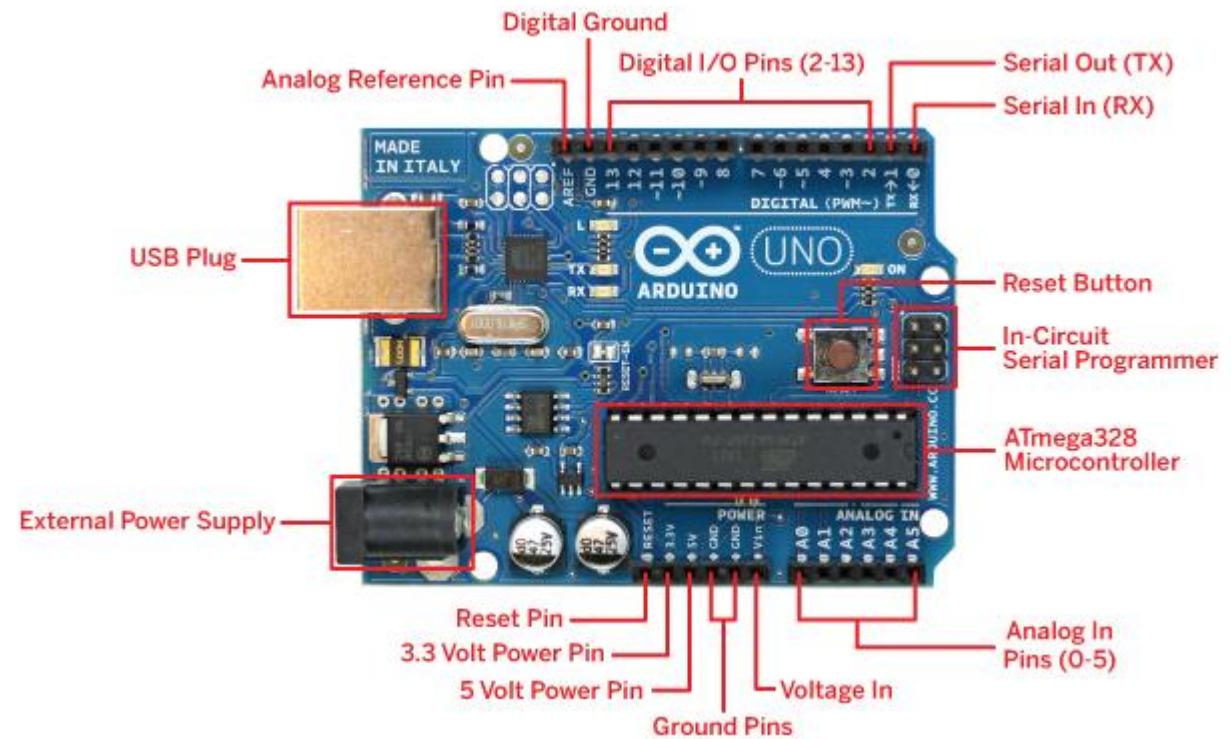
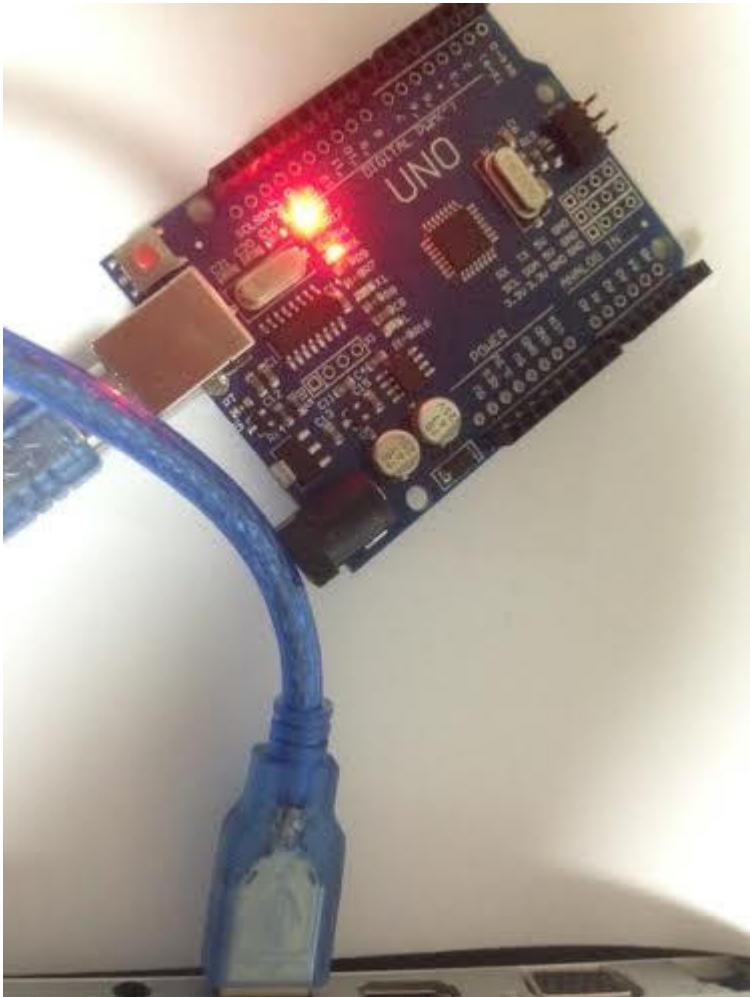
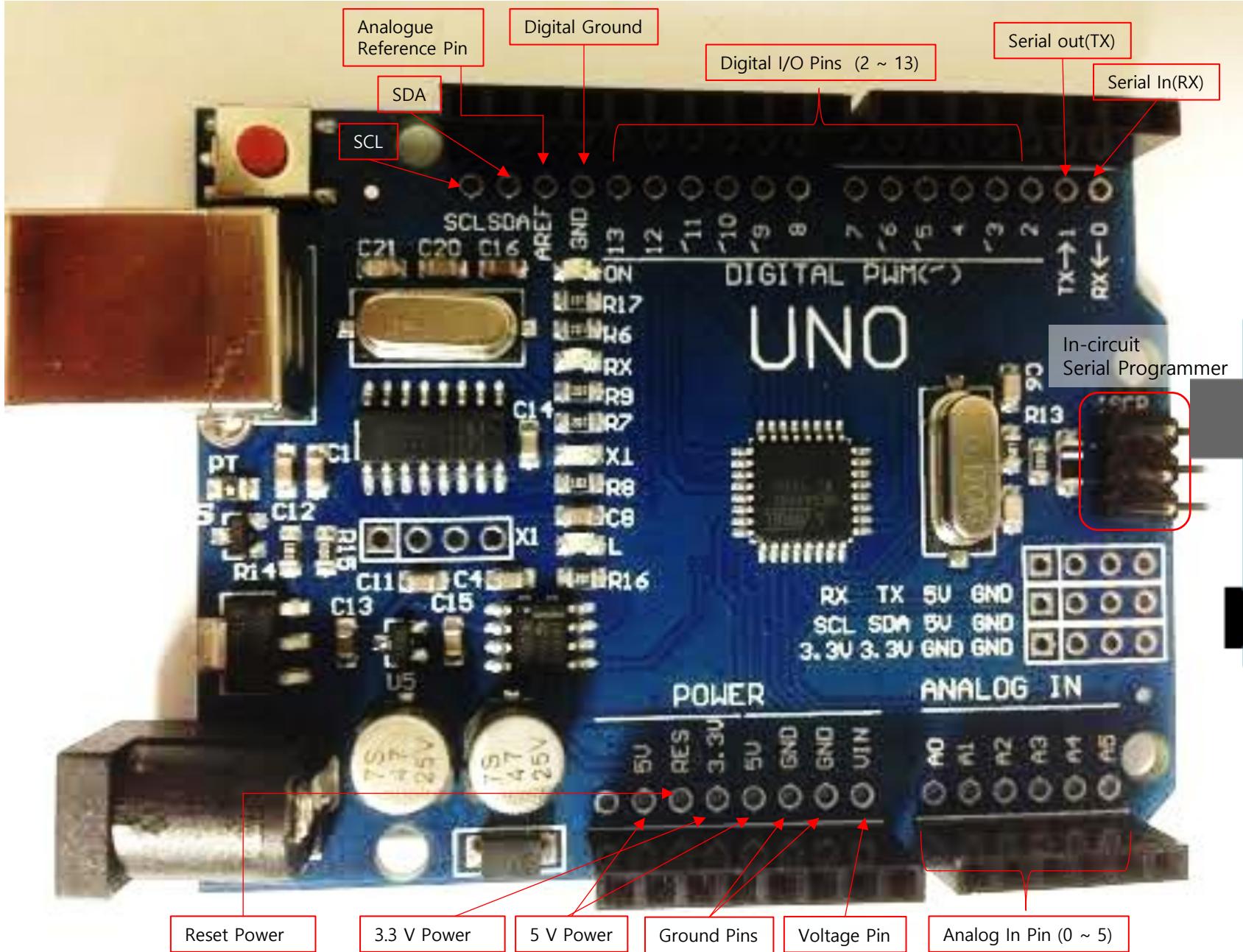


아빠와 아이가 함께 즐기는 아duino 놀이터



아두이노 보드 살펴보기

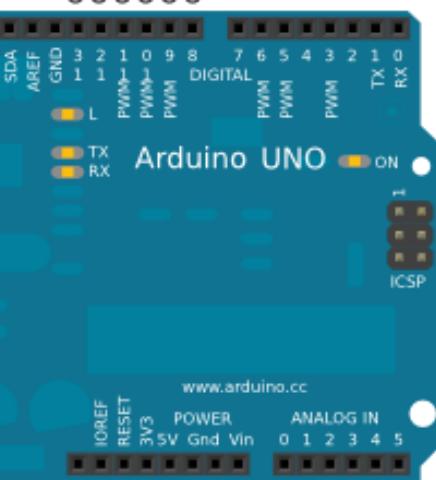




Analog Pin #	Function
0	Input (Digital or Analog) Bit 0
1	Input (Digital or Analog) Bit 1
2	Input (Digital or Analog) Bit 2
3	Input (Digital or Analog) Bit 3
4	Input (Digital or Analog) Bit 4
5	Input (Digital or Analog) Bit 5

Digital Pin #	Function
13	Output (Digital) Bit 5
12	Output (Digital) Bit 4
11	Output (Digital) Bit 3
10	Output (Digital) Bit 2
9	Output (Digital) Bit 1
8	Output (Digital) Bit 0

Not usually used **	5
Not usually used **	4
Not usually used **	3
Input (Digital) Trigger	2
Do not use! *	1
Do not use! *	0



Made with Fritzing.org

* Reserved for µManager Serial Communication

** Reserved for TLV5618 ACOTF Peripheral

아duino 프로그램 설치



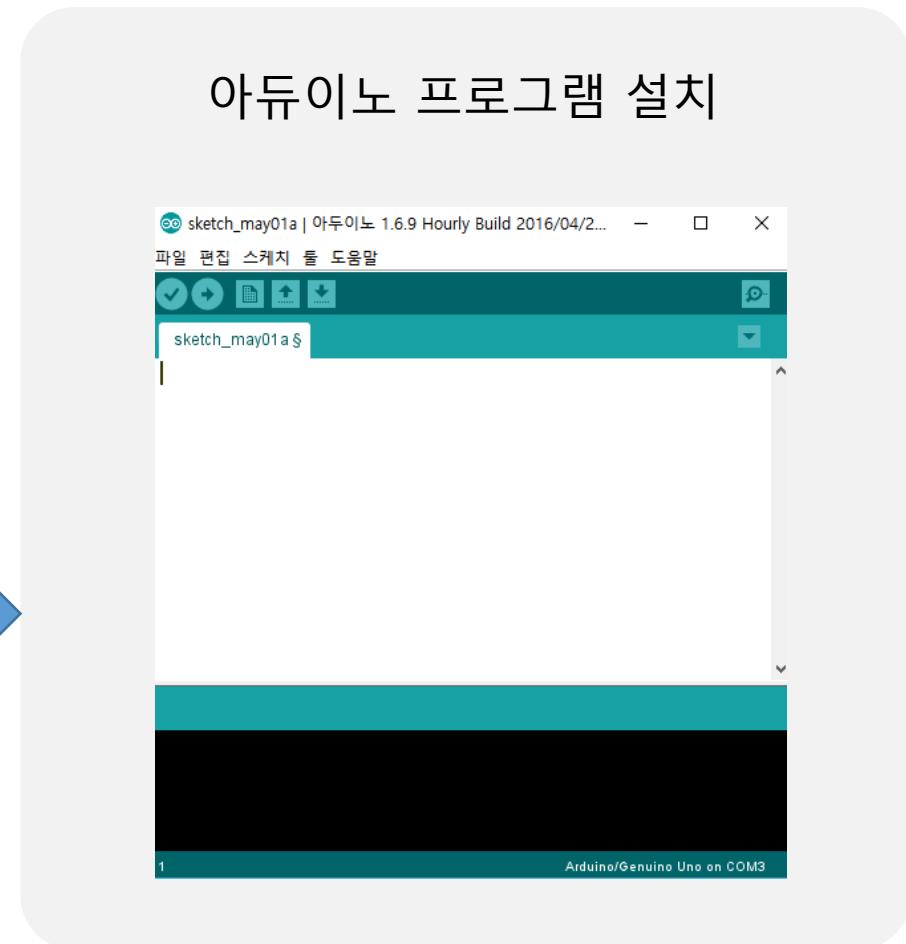
Download the Arduino Software



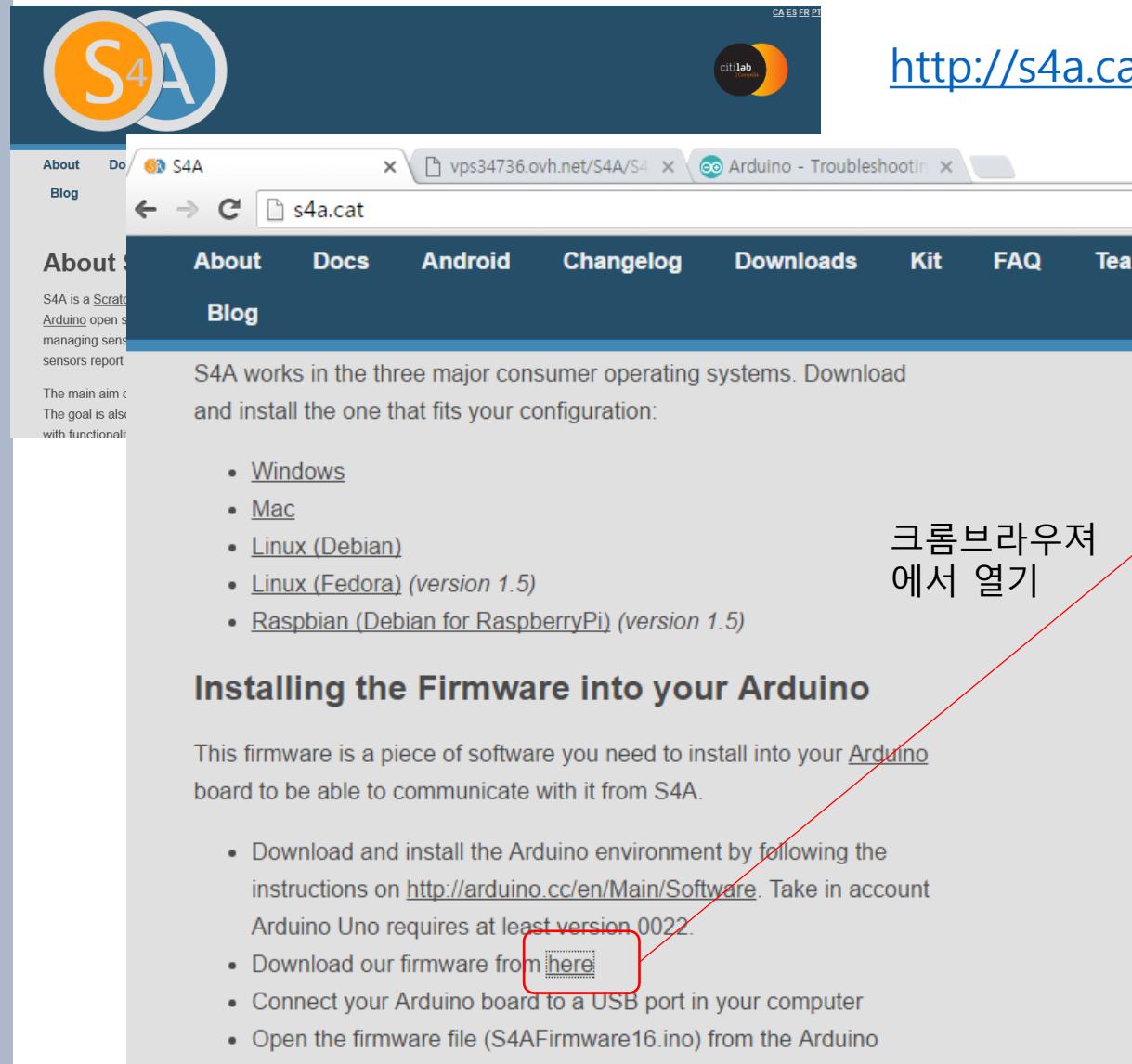
1. 아duino 페이지(www.arduino.cc) 접속
2. 최신 아duino 빌더 프로그램 설치

웹사이트 : <https://www.arduino.cc/en/Main/Software>

아duino 프로그램 설치



업데이트 소스 구하기

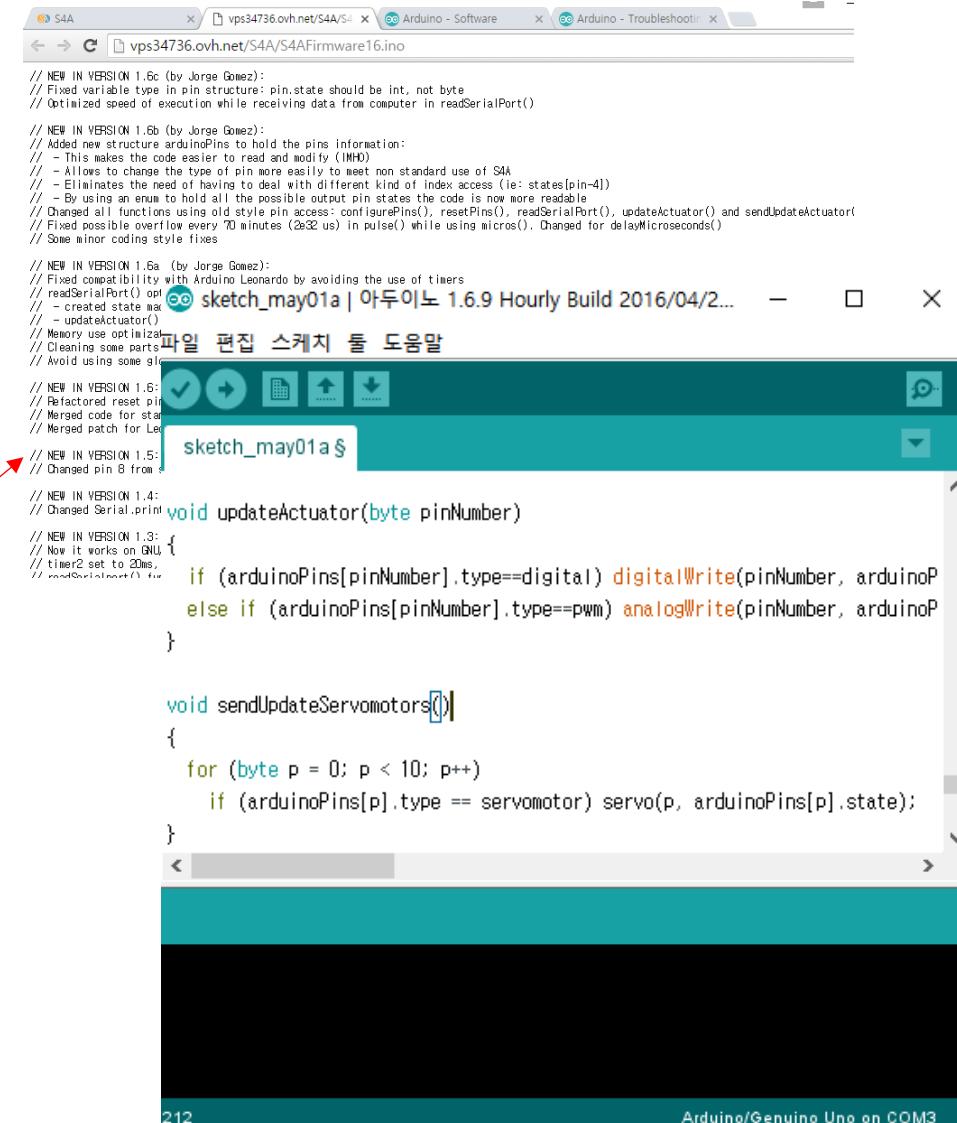


The screenshot shows the S4A website's 'About' page. At the top, there are links for 'About', 'Docs', 'Android', 'Changelog', 'Downloads', 'Kit', 'FAQ', and 'Team'. Below this, there's a 'Blog' section with a recent post about S4A working on three major consumer operating systems. A list of download links for different operating systems follows. The main content area is titled 'Installing the Firmware into your Arduino'. It explains the need to install the Arduino environment and provides instructions for connecting the board and opening the firmware file. A red box highlights the 'here' link in the final step.

크롬브라우저
에서 열기

1. 크롬 브라우저 설치
2. 소스를 복사하여 아두이노 빌더 창에 붙여넣기

<http://s4a.cat/>

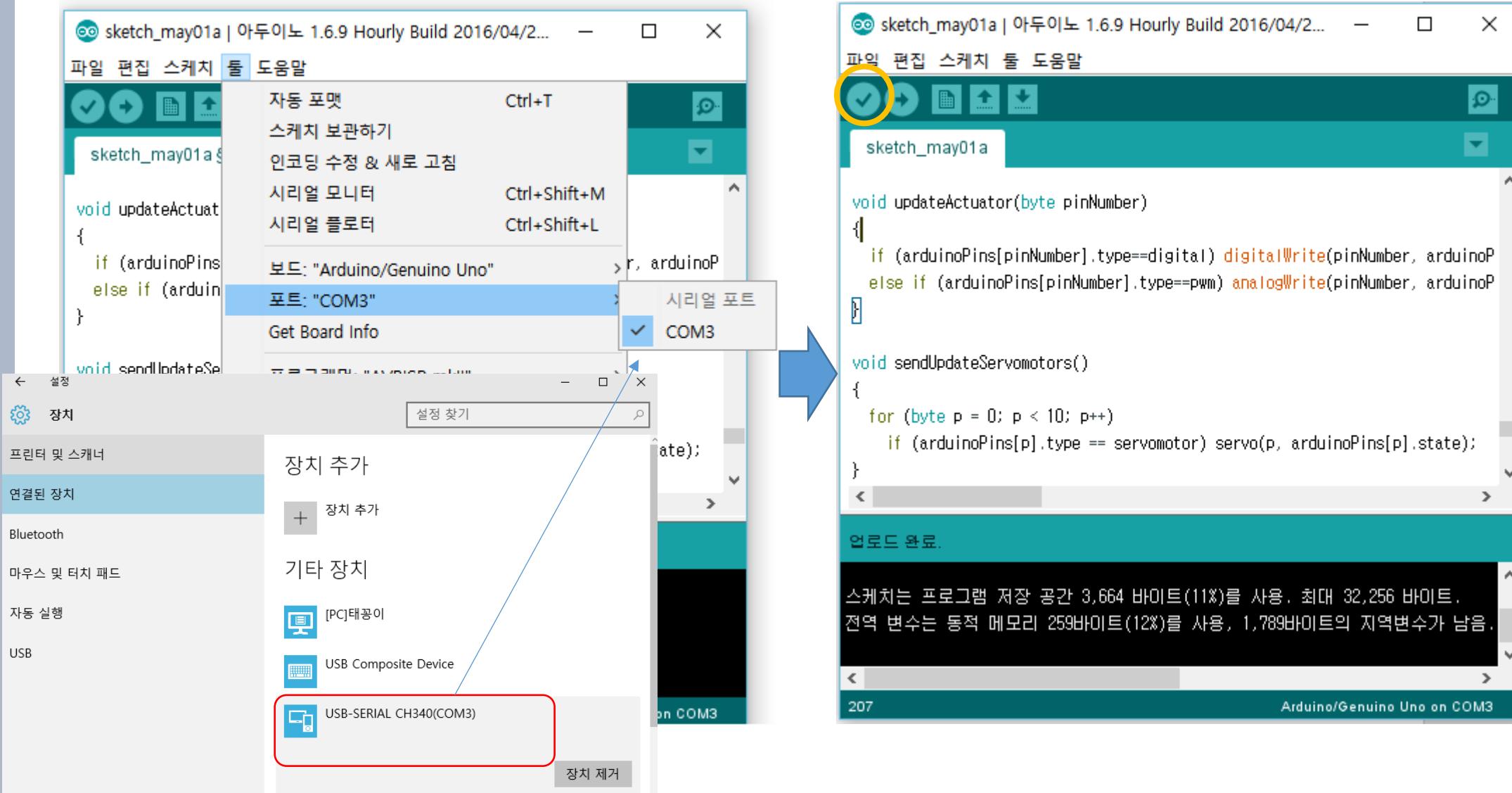


The screenshot shows a browser window displaying the Arduino IDE sketch 'sketch_may01a.ino'. The code contains several Korean comments explaining changes in different versions of the firmware. A red arrow points from the Korean text '크롬브라우저에서 열기' in the previous screenshot to the Korean comment '// NEW IN VERSION 1.5: // Changed pin 8 from 9' in the code.

```
// NEW IN VERSION 1.6 (by Jorge Gomez):  
// Fixed variable type in pin structure: pin.state should be int, not byte  
// Optimized speed of execution while receiving data from computer in readSerialPort()  
  
// NEW IN VERSION 1.6:  
// Added new structure arduinoPins to hold the pins information:  
// - This makes the code easier to read and modify (IMHO)  
// - Allows to change the type of pin more easily to meet the standard use of S4A  
// Eliminates the need of having to deal with different kind of index access (ie: states[pin-4])  
// By using an array to hold all the possible output pin states the code is now more readable  
// Changed all functions using old style pin access: configurePin(), resetPins(), readSerialPort(), updateActuator() and sendUpdateActuator()  
// Fixed possible overflow every 70 minutes (2e32 us) in pulse() while using micros(). Changed for delayMicroseconds()  
// Some minor coding style fixes  
  
// NEW IN VERSION 1.6a (by Jorge Gomez):  
// Fixed compatibility with Arduino Leonardo by avoiding the use of timers  
// readSerialPort() opt  
// - created state map  
// - updateActuator()  
// Memory use optimization  
// Cleaning some parts  
// Avoid using some global variables  
  
// NEW IN VERSION 1.6:  
// Refactored reset pin  
// Merged code for standard and RPI  
// Merged patch for Leonardo  
  
// NEW IN VERSION 1.5:  
// Changed pin 8 from 9  
  
// NEW IN VERSION 1.4:  
// Changed serial.print  
  
// NEW IN VERSION 1.3:  
// Now it works on GNU:  
// timer2 set to 20ms:  
// microcontroller:  
  
void updateActuator(byte pinNumber)  
{  
    if (arduinoPins[pinNumber].type==digital) digitalWrite(pinNumber, arduinoPins[pinNumber].state);  
    else if (arduinoPins[pinNumber].type==pwm) analogWrite(pinNumber, arduinoPins[pinNumber].state);  
}  
  
void sendUpdateServomotors()  
{  
    for (byte p = 0; p < 10; p++)  
        if (arduinoPins[p].type == servomotor) servo(p, arduinoPins[p].state);  
}
```

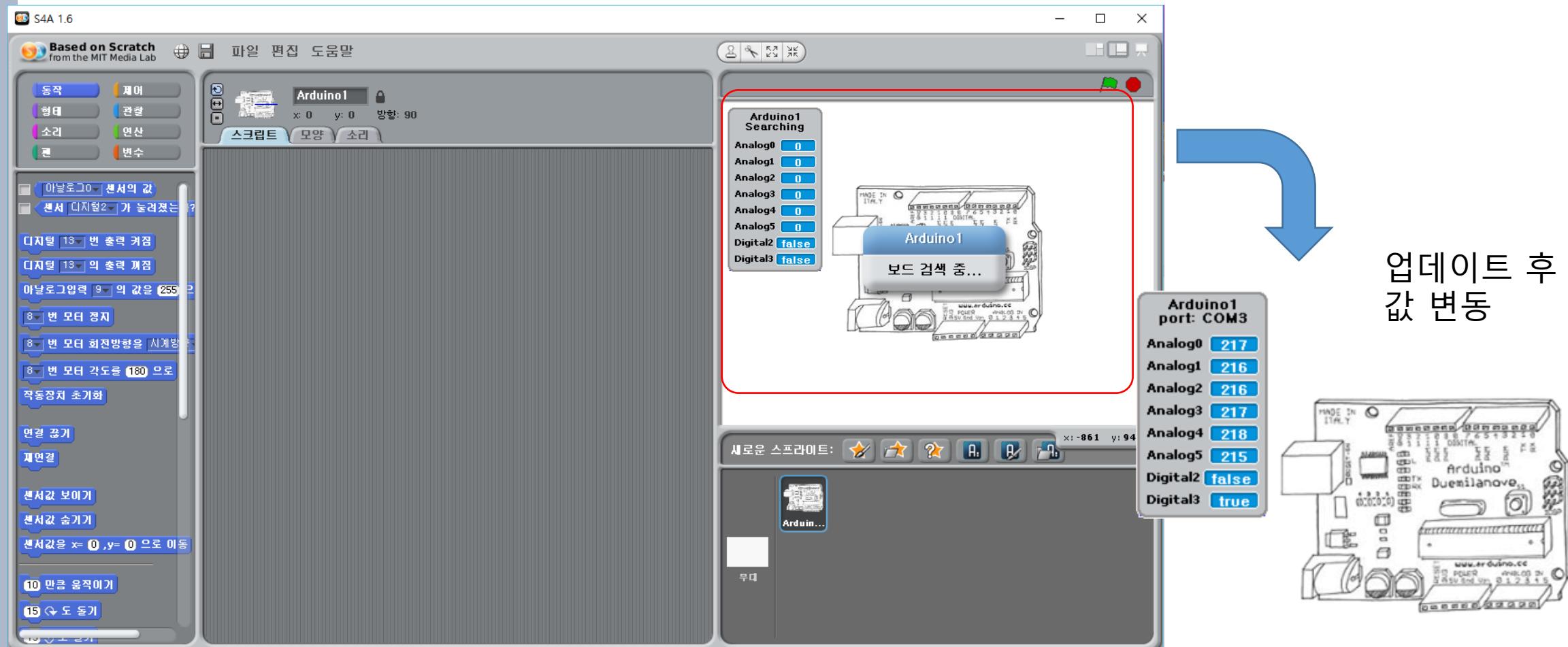
포트 설정 및 업로드

1. 시리얼 포트 설정
2. 보드를 PC에 연결하고 업데이트 실행



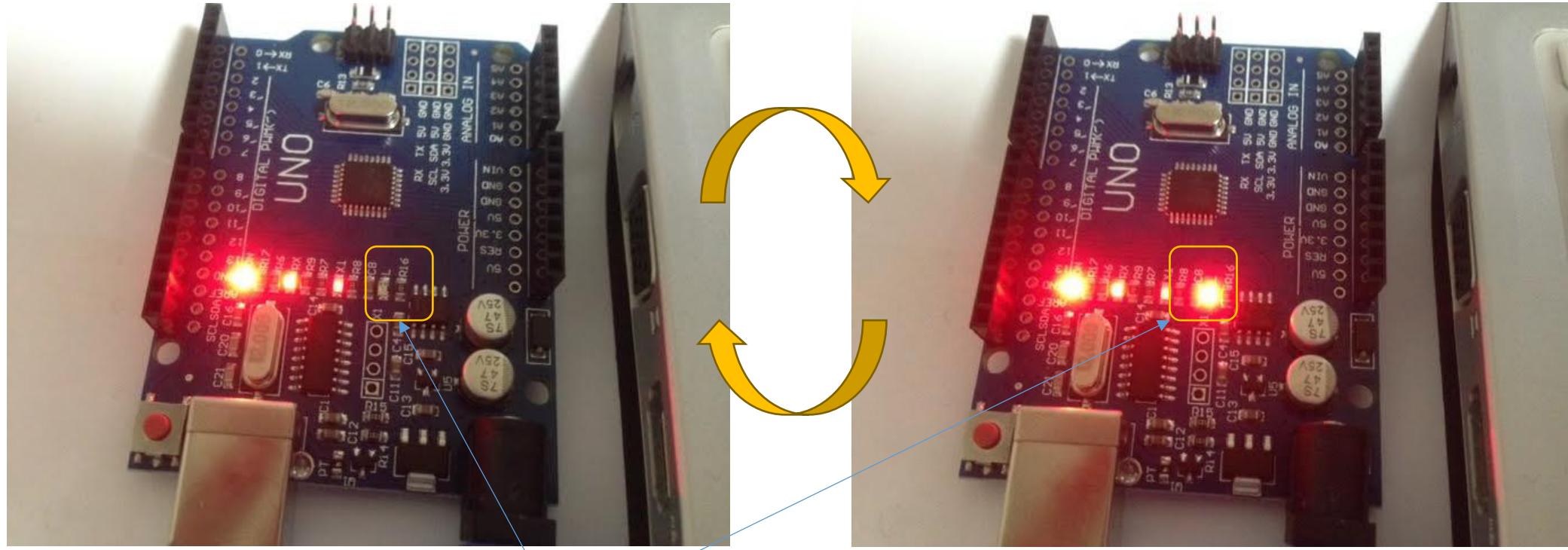
S4A상태

1. 업데이트를 하지 않으면 계속 보드 검색중 되풀이 됨
2. 아duino 보드 펌웨어 업데이트하면 값 변동 모니터링



상태 점검하기

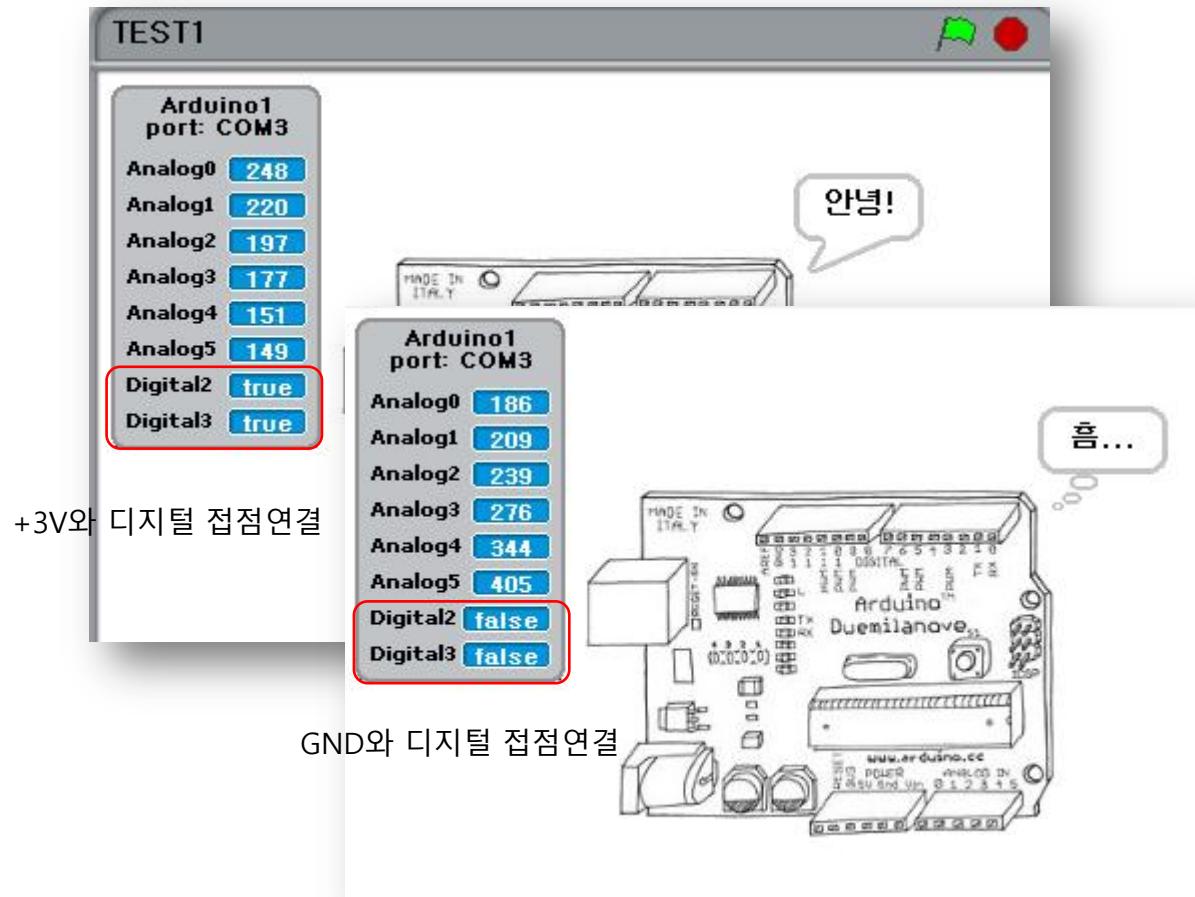
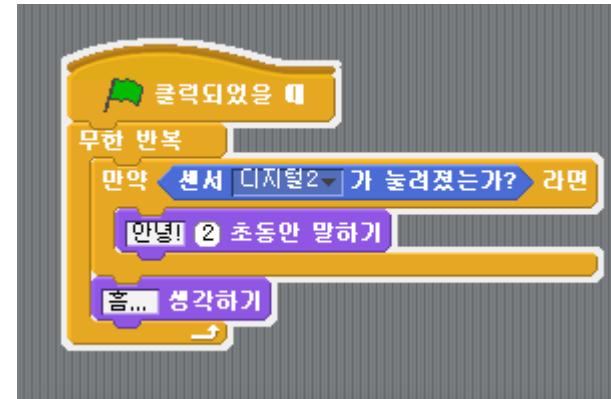
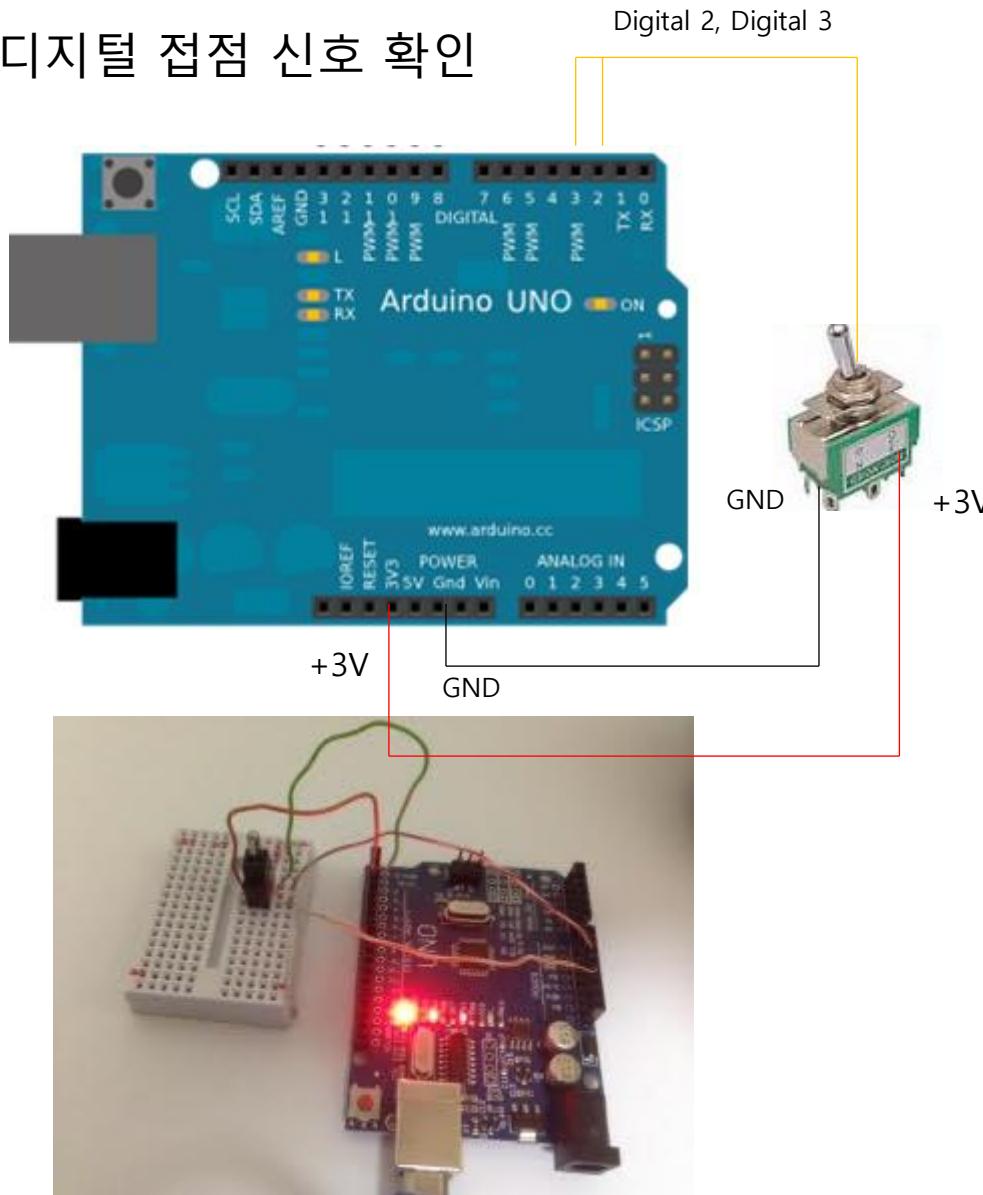
1. 아두이노 보드만 연결 후 디지털 13번을 출력을 주어 불빛이 변경되는 것을 확인 한다.



디지털 13번 출력

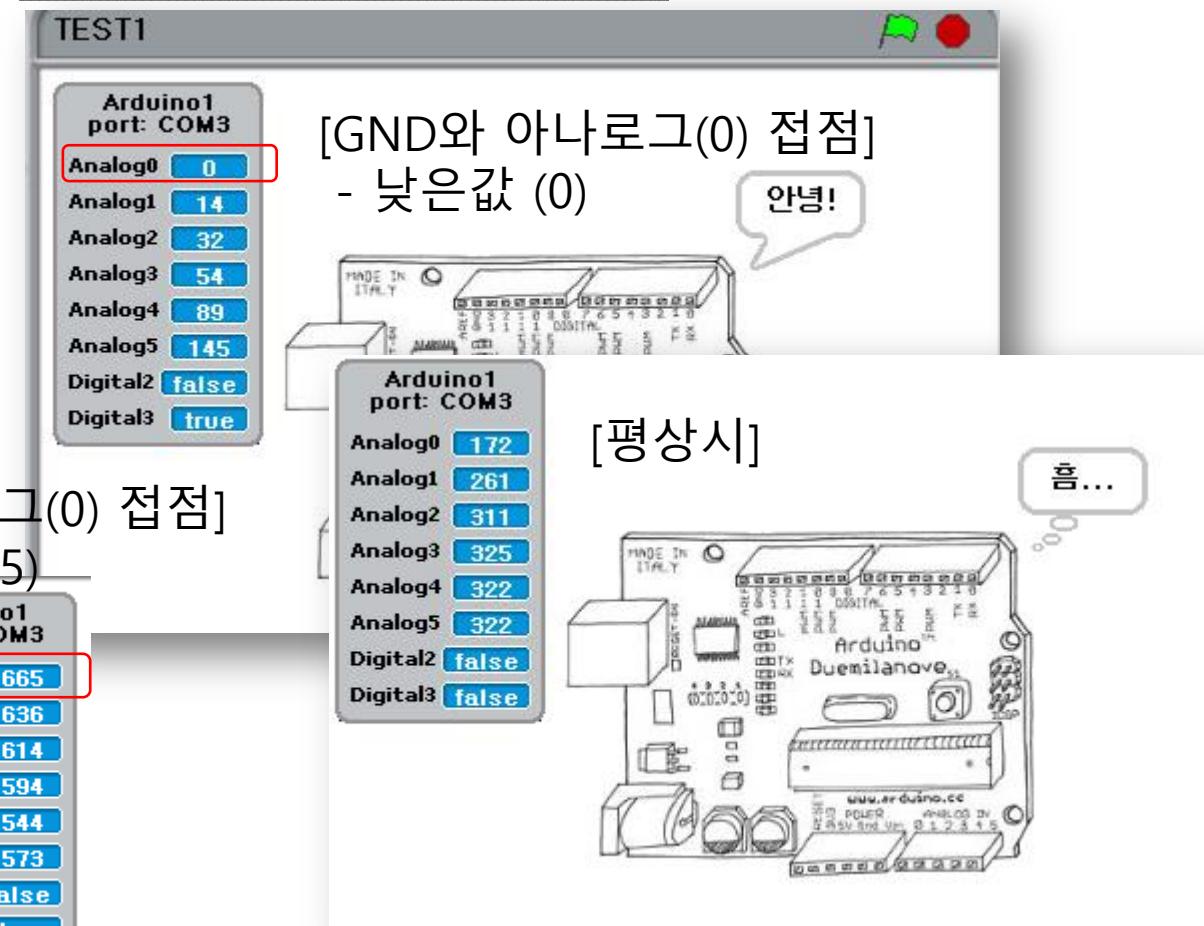
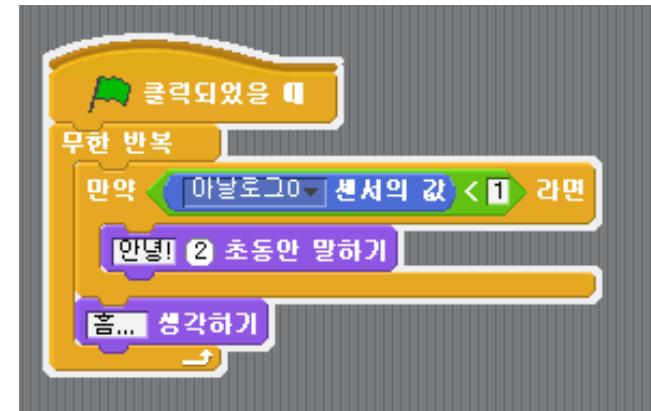
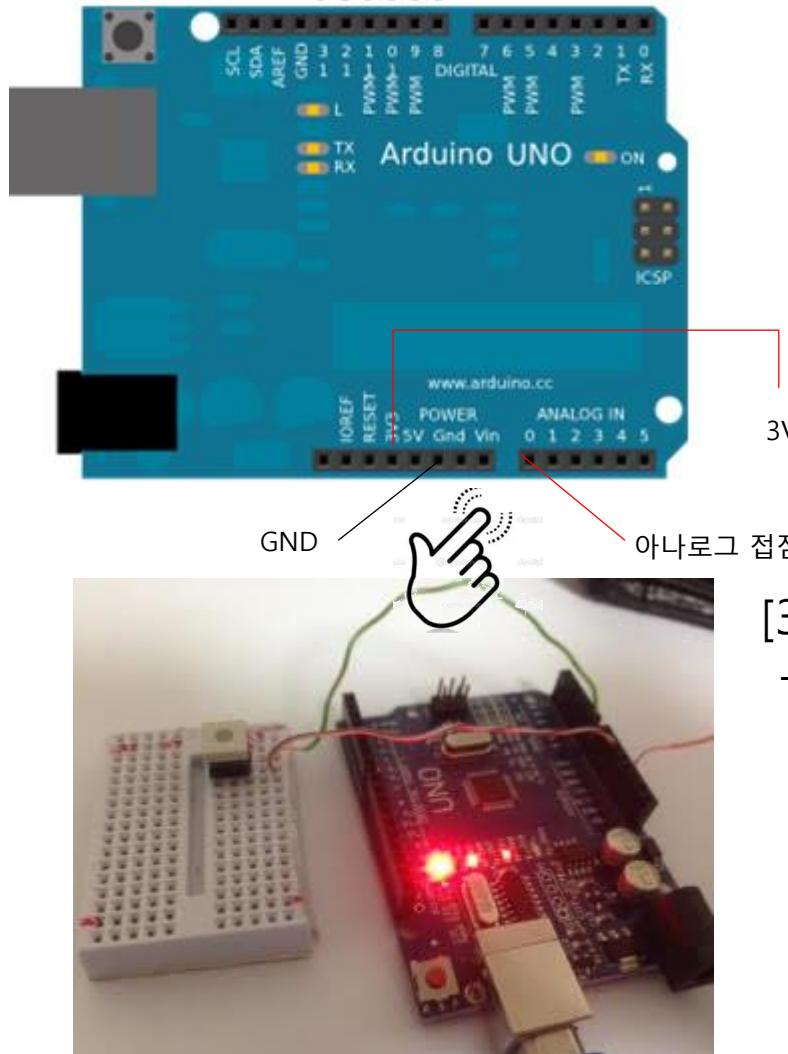
디지털 신호 받기

디지털 접점 신호 확인



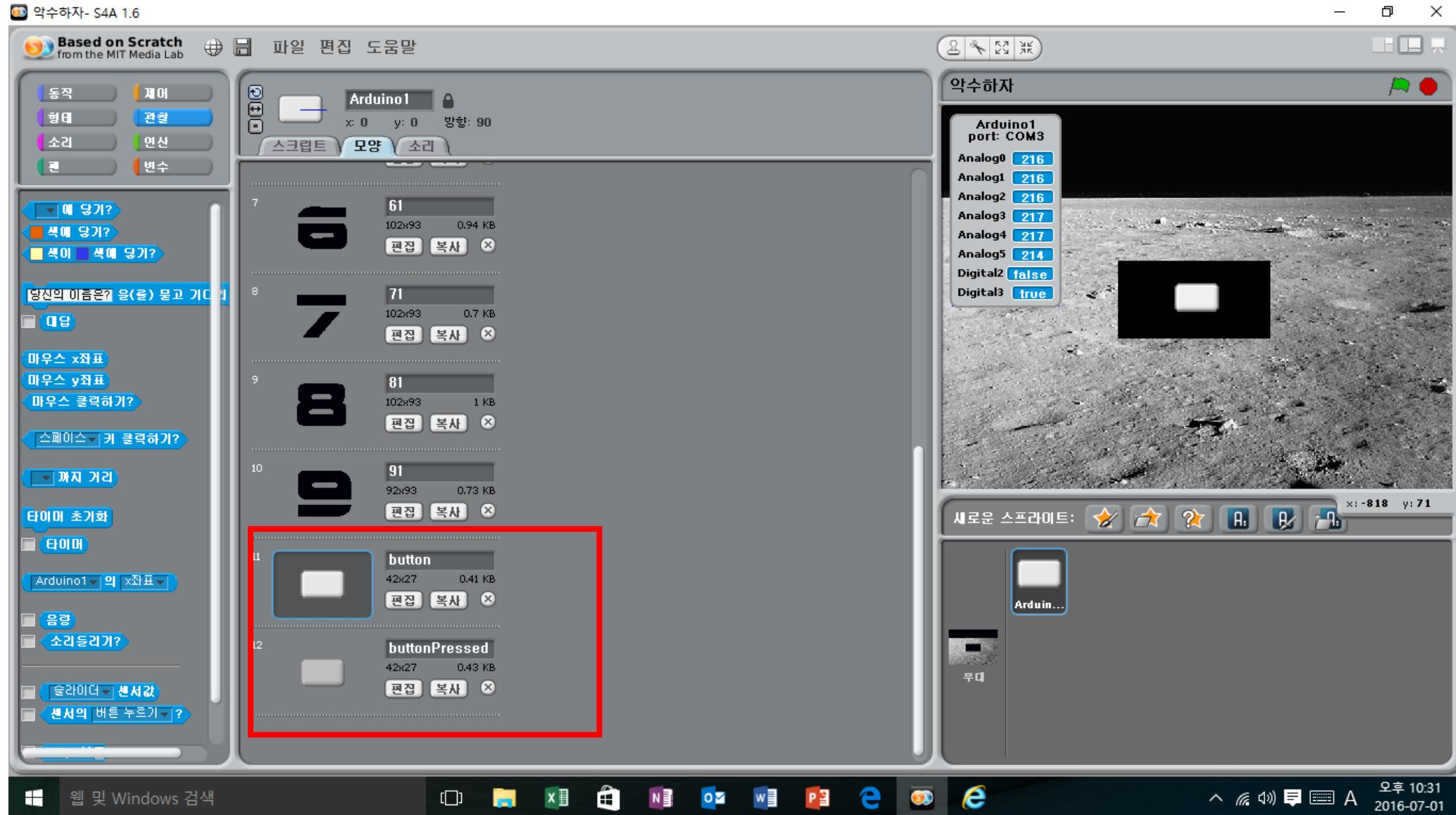
아나로그 신호 받기

아나로그 접점 신호 확인

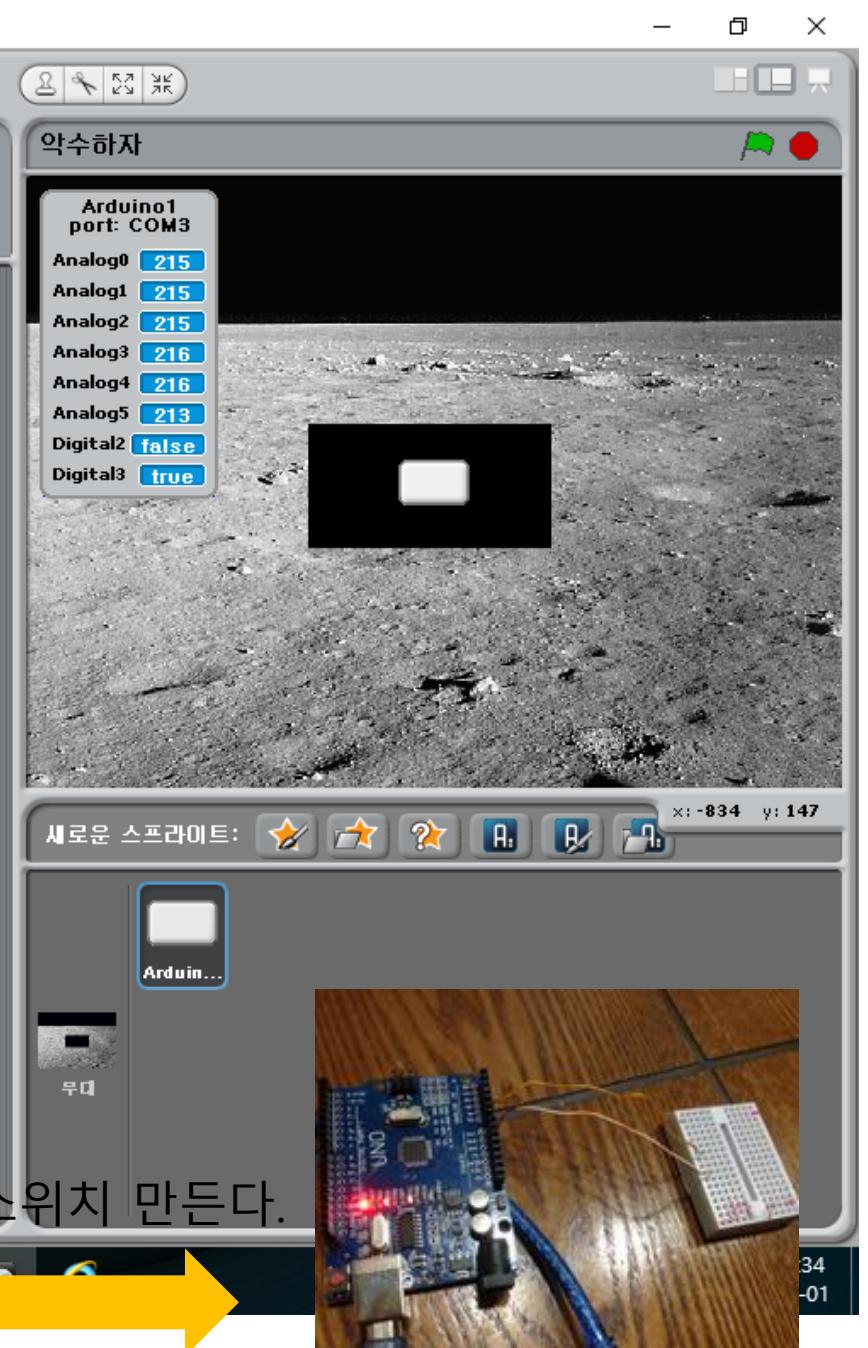
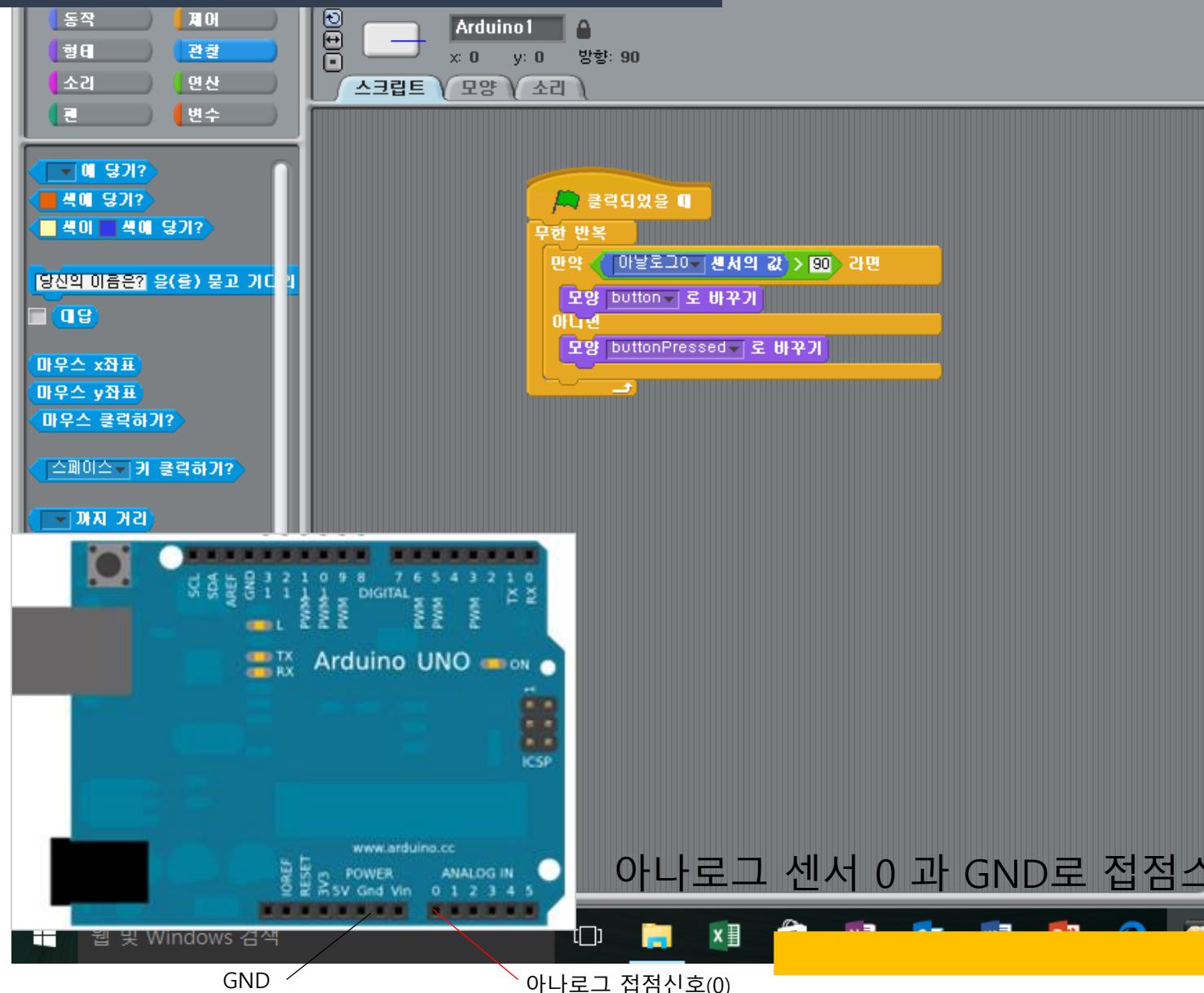


달나라 스위치(1/3)

접점 스위치를 눌러 스위치 그림 변화



달나라 스위치 (2/3)

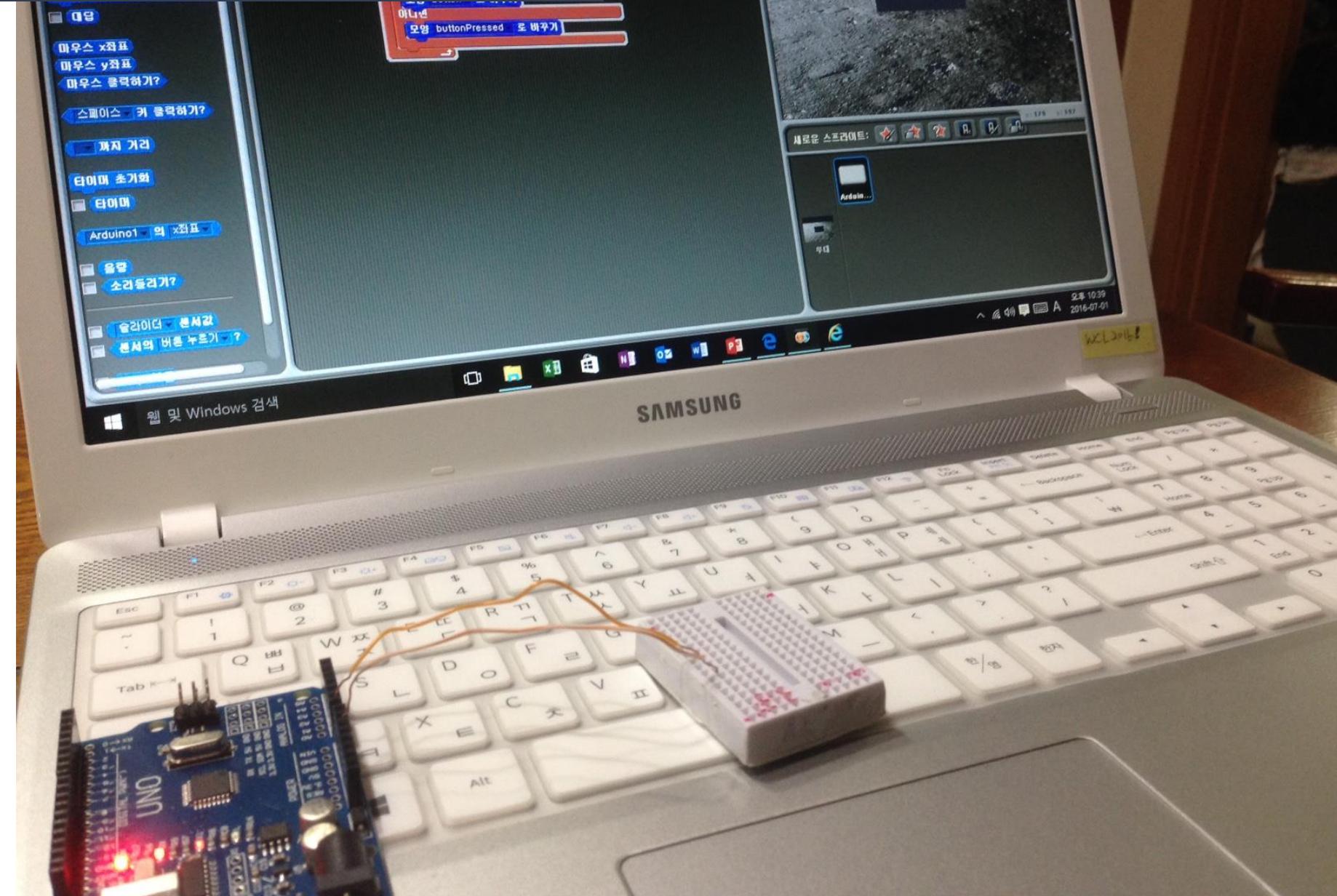


아나로그 센서 0 과 GND로 접점스위치 만든다.

GND

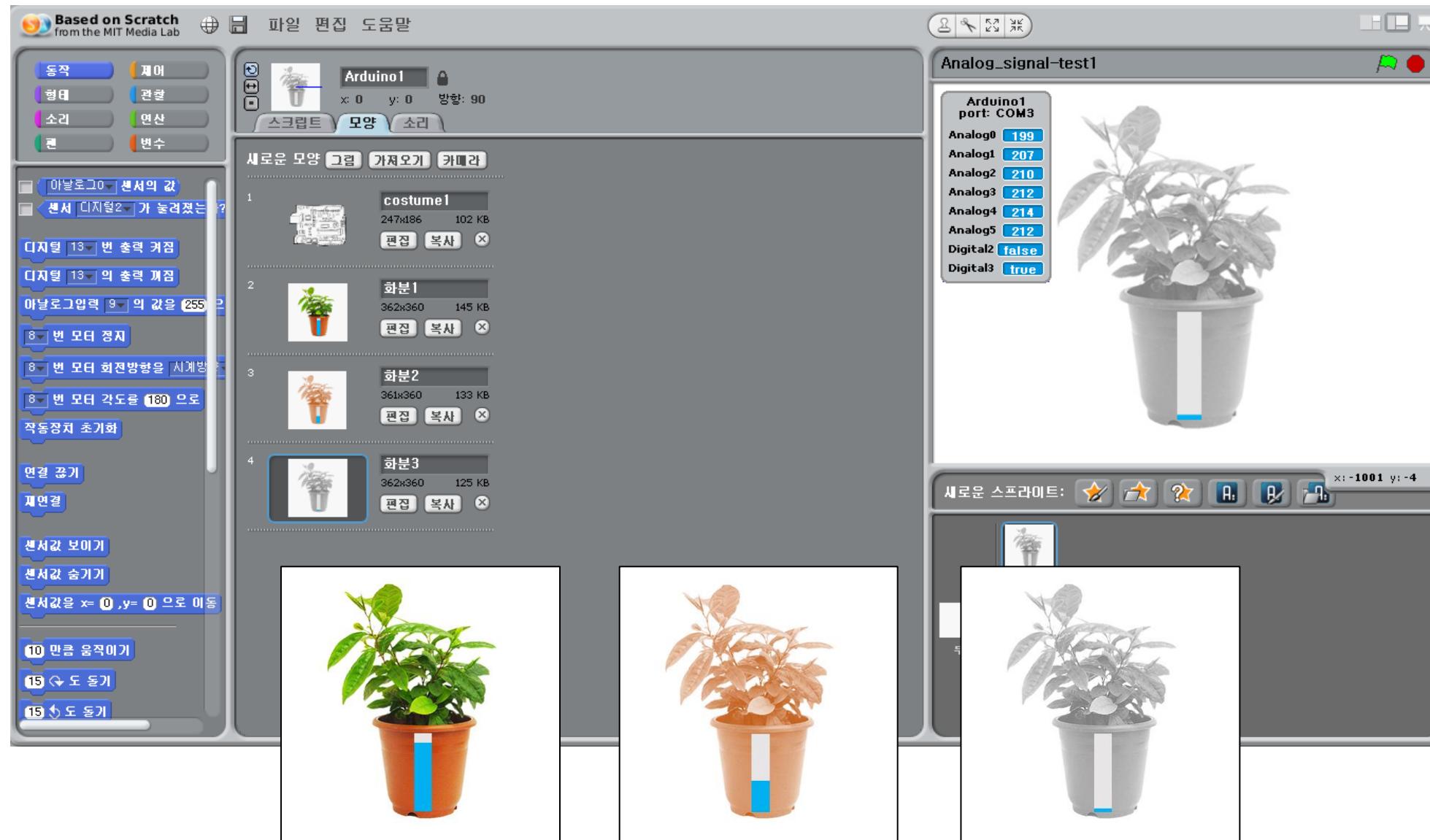
아나로그 접점신호(0)

달나라 스위치 (3/3)

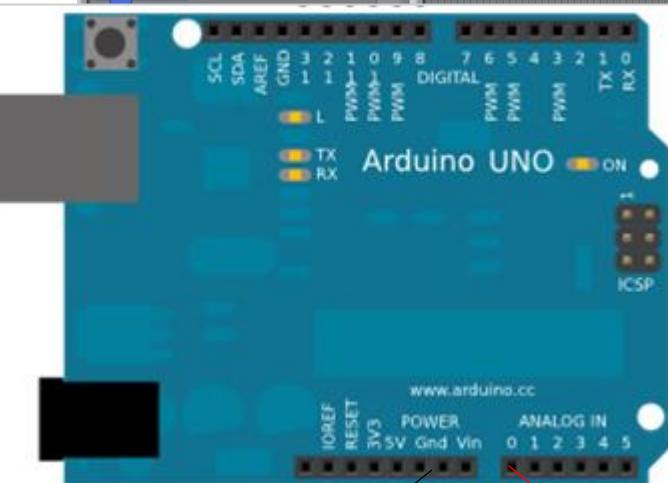
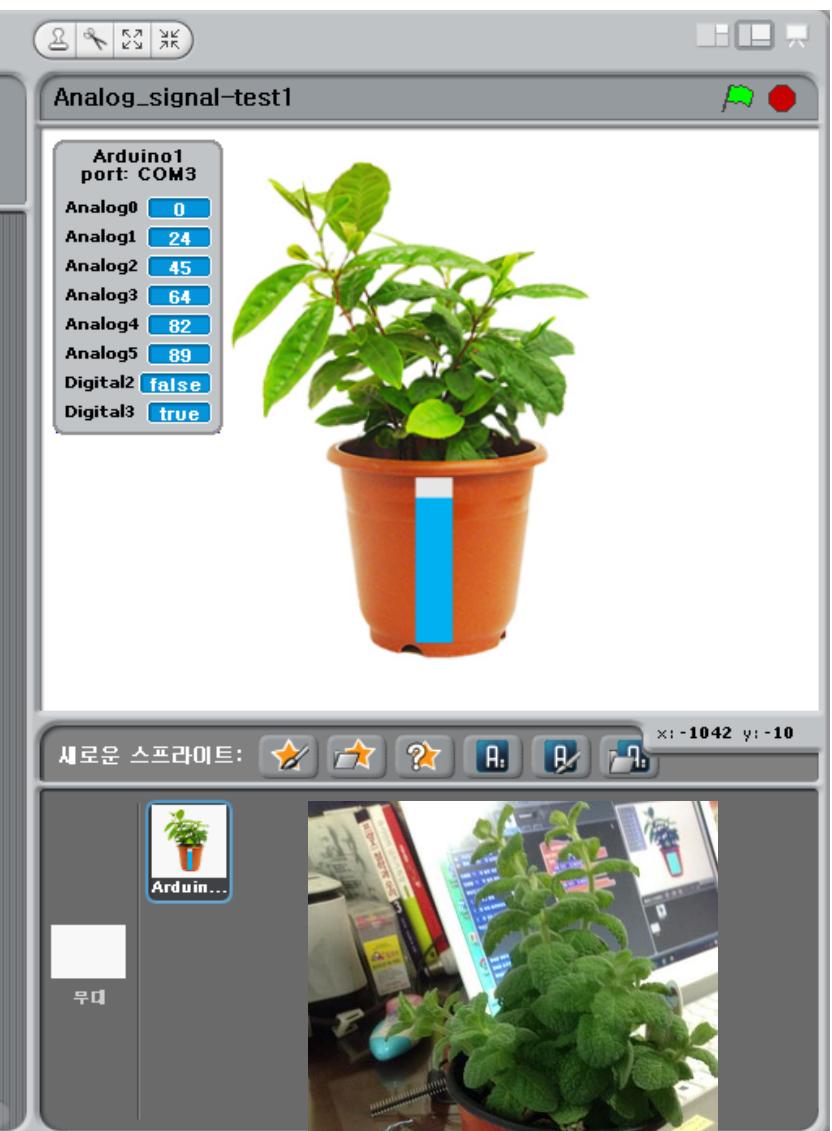


디지털 화분(1/3)

물의 양에 따른 3가지 형태 변화 그림 등록



디지털 화분(2/3)



GND

아나로그 접점신호(0)



디지털 화분(3/3)



센서 값이 90 이상인 경우



센서 값이 20보다 크고
90 이하인 경우



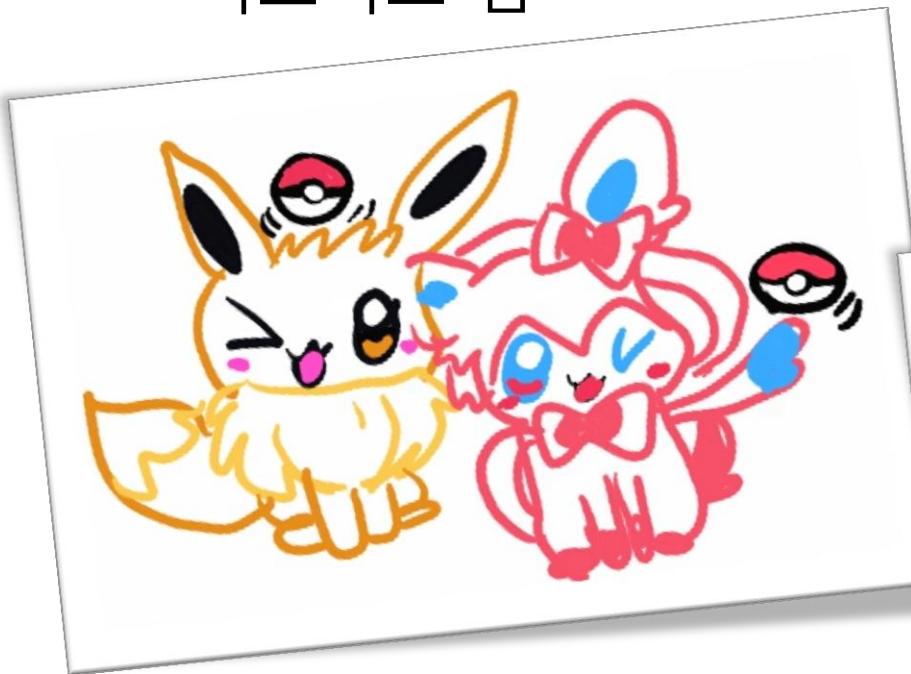
센서 값이 20 미만인 경우



누가이기나 (1/4)

아이와 함께 팀 디자인 하기
스토리 구상 및 그림 그리기

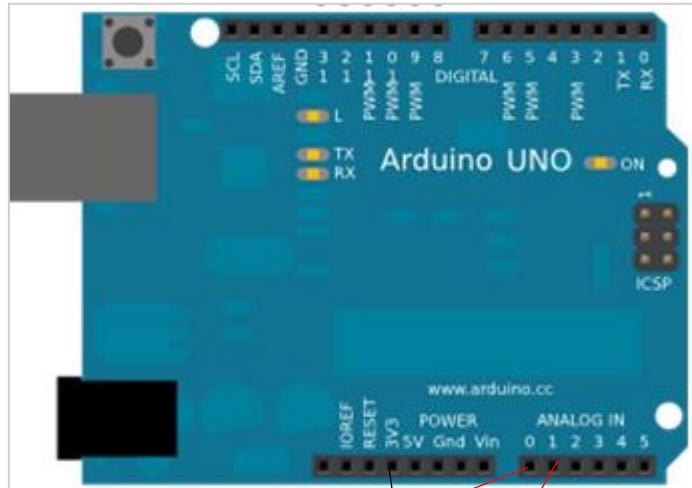
이브이즈 팀



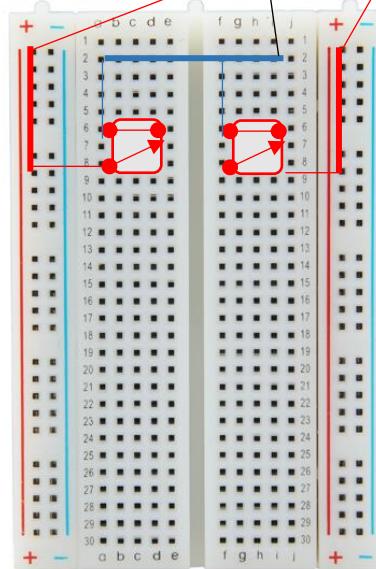
피카츄와 XY 팀



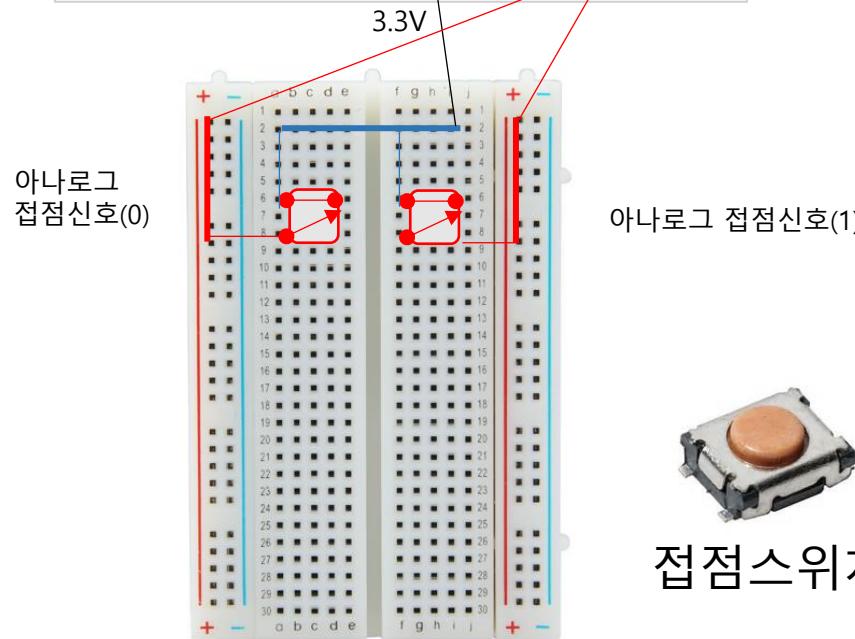
누가이기나 (2/4)



아나로그
접점신호(0)



접점스위치



변수함수를 통해
센서의 특정 값을
카운트하도록 설정

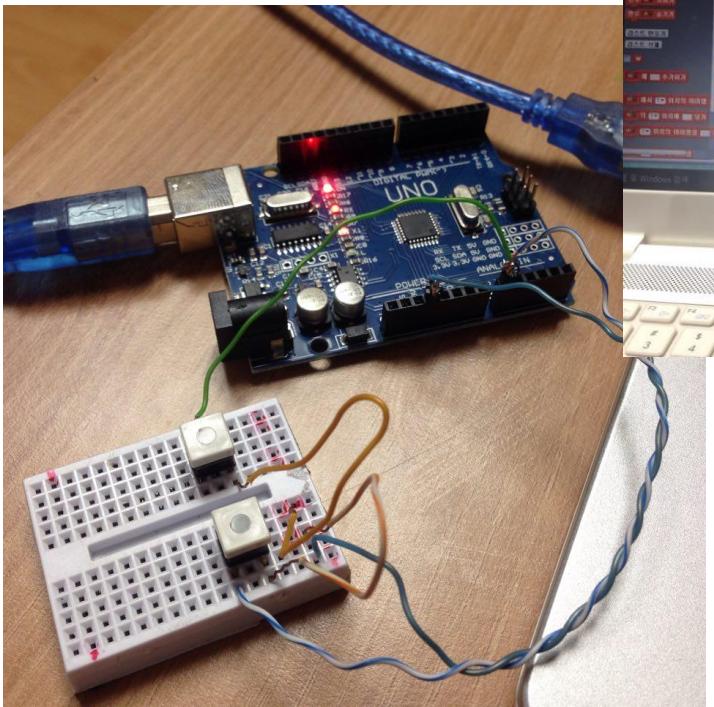
누적 값을 비교하여
결과를 그림으로 표시

누가이기나 (3/4)

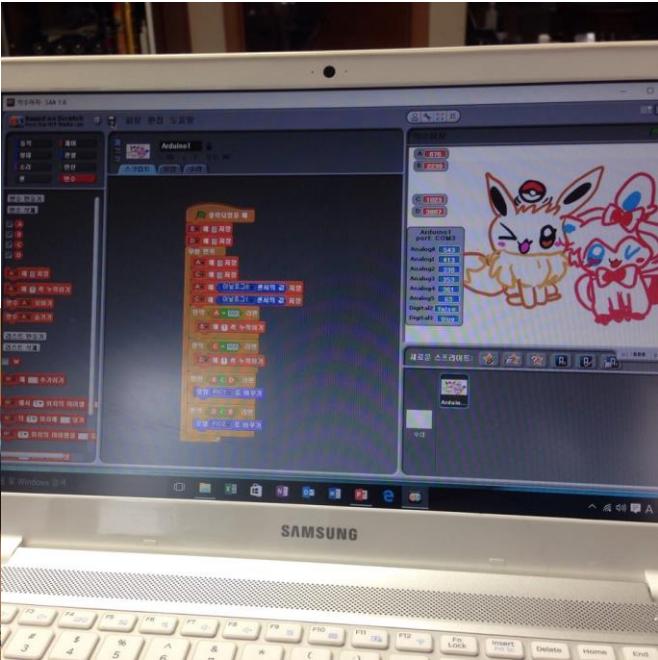


누가이기나 (4/4)

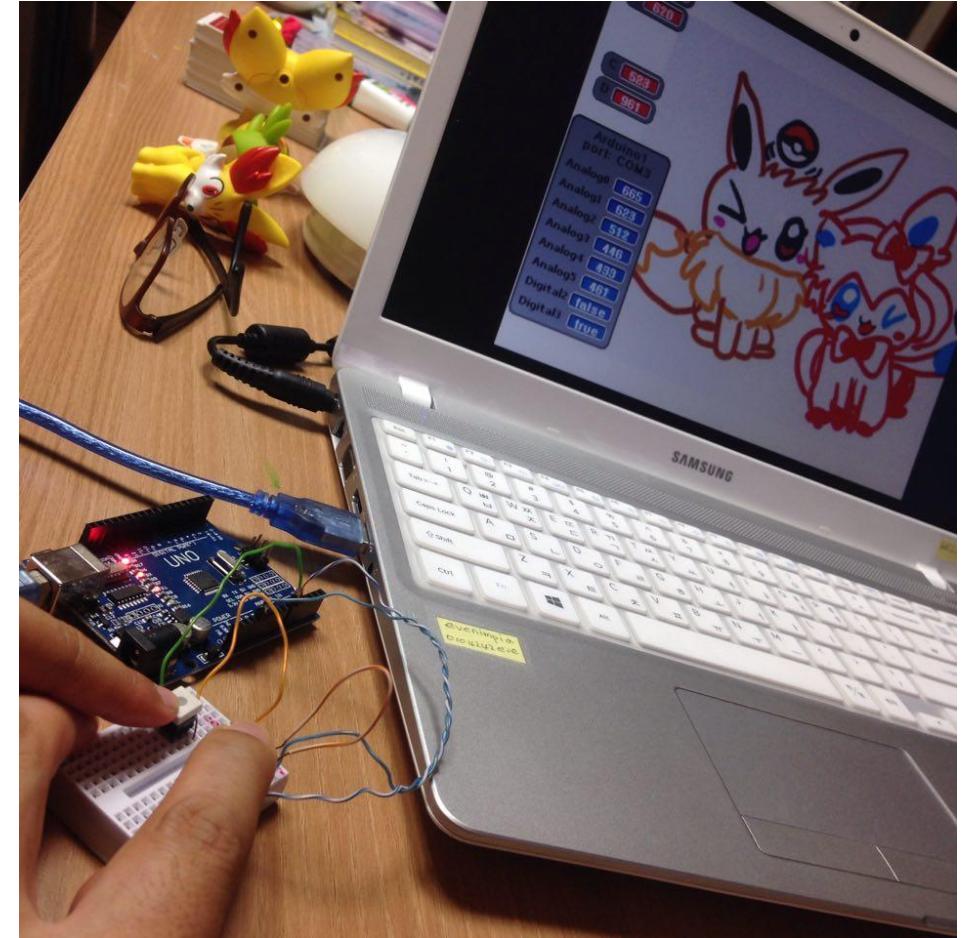
하드웨어 구성



프로그램 개발



아이와 게임



누가이기나2 (1/4)

아이와 함께 경기할 캐릭터 디자인 하기
스토리 구상 및 배경그림 만들기(파워포인트 활용)
출발시 새소리로 시작

이브이



피카츄

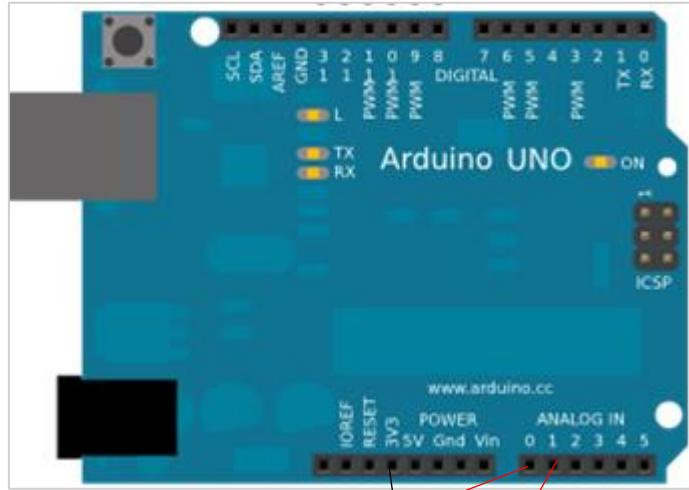


배경 없애고 아이콘 저장

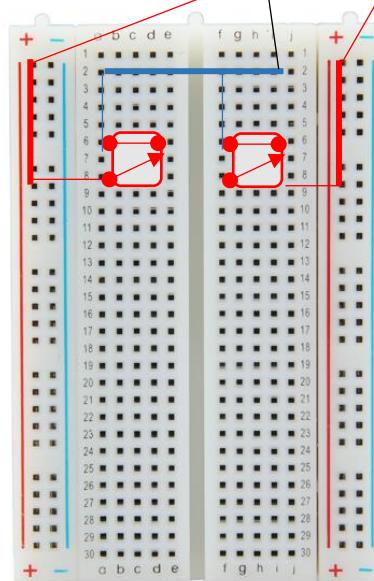
시나리오에 맞는 배경 만들기



누가 이기나2 (2/4)



아나로그
접점신호(0)



아나로그 접점신호(1)



접점스위치

우측상단에 결과 표시



우측하단에 표시



좌측하단에 표시

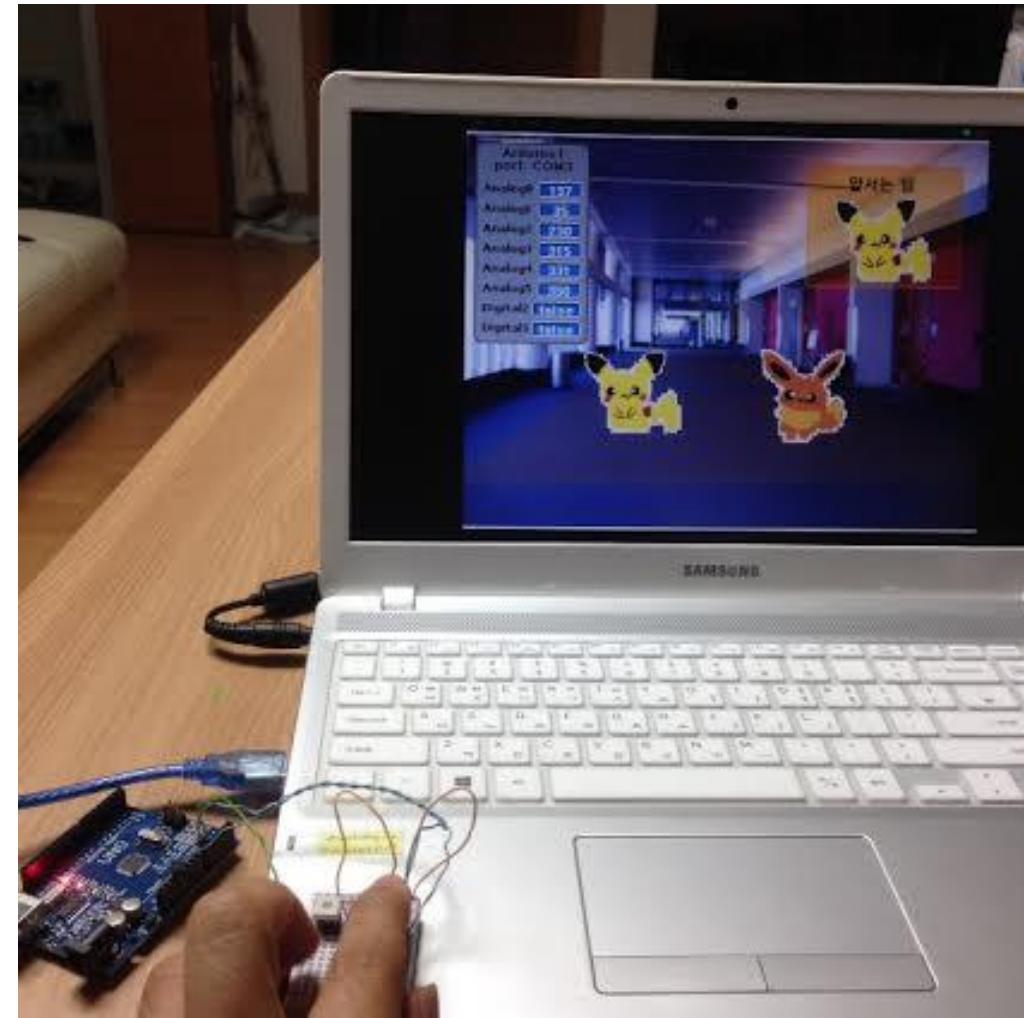
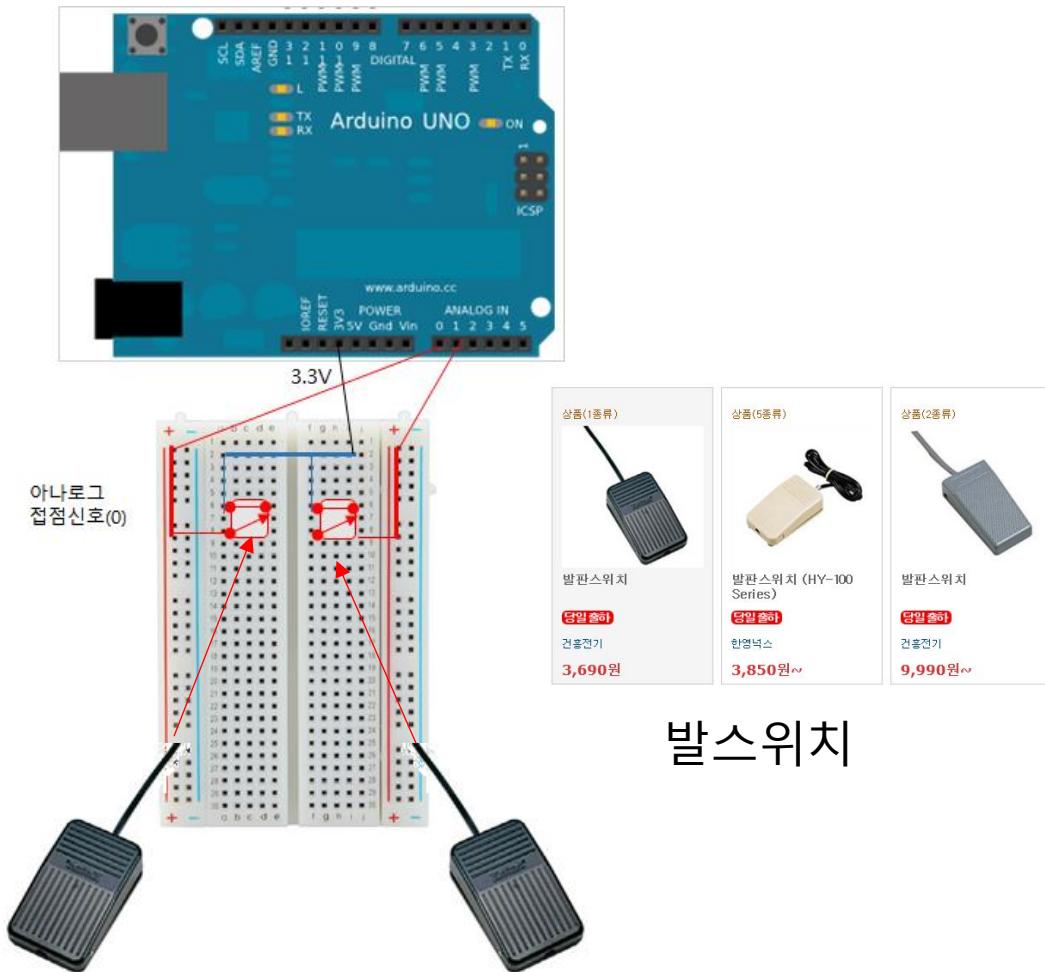


누가이기나2 (3/4)



누가이기나2 (4/4)

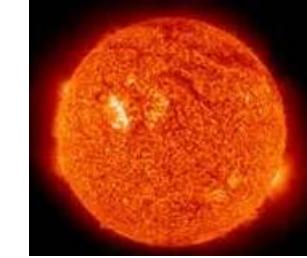
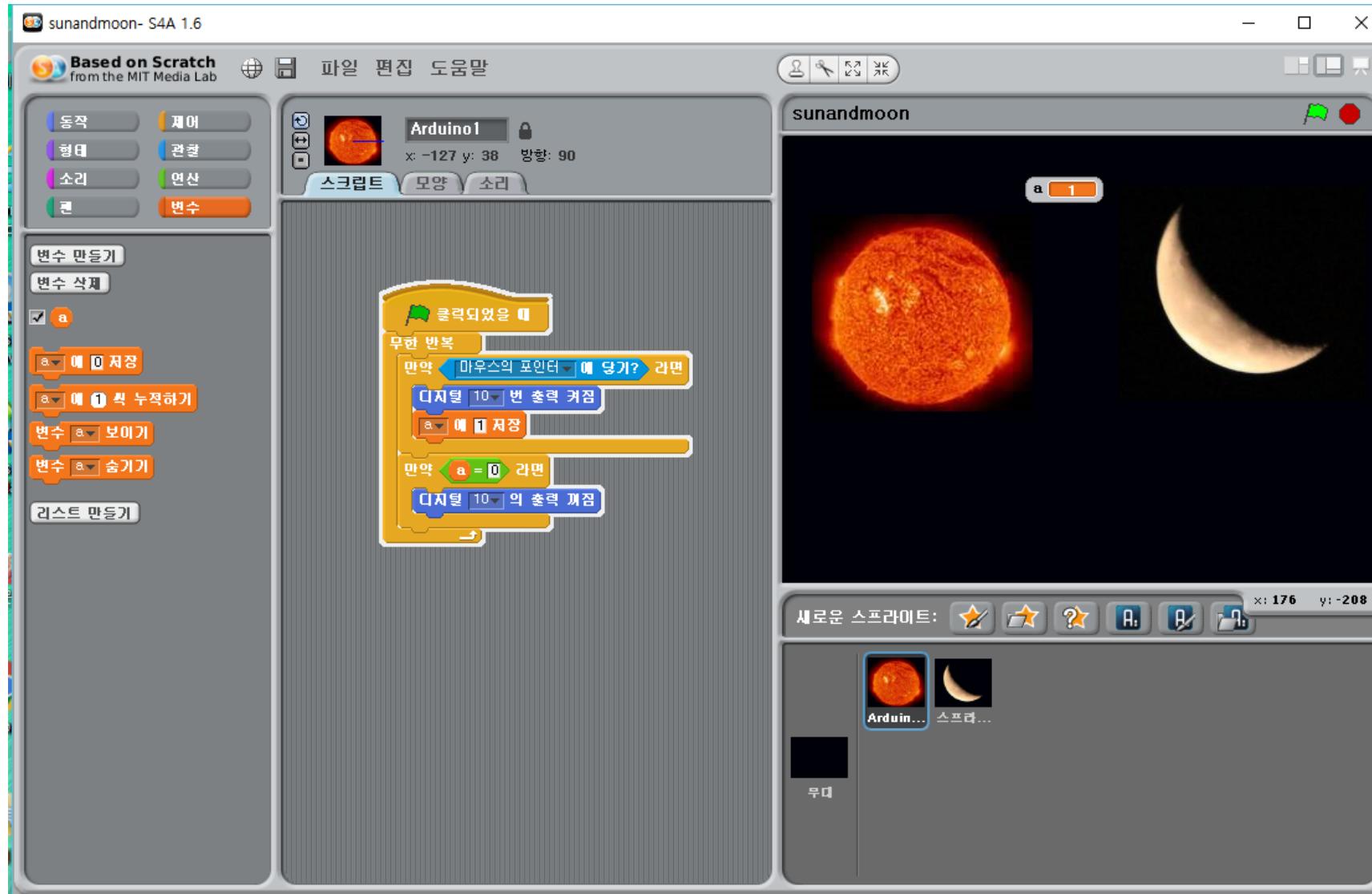
접점 스위치 대신 발스위치를 연결하여 만들어도 재미있는 게임이 됨



아이와 게임

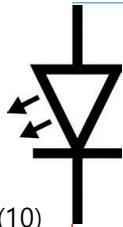
해와 달 조명(1/3)

검은 배경에 해와 달 그림 등록

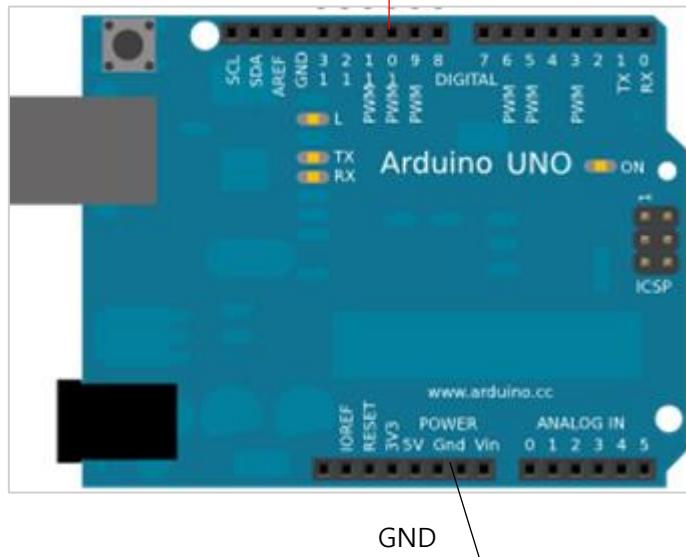


해와 달 조명(2/3)

LED 조명

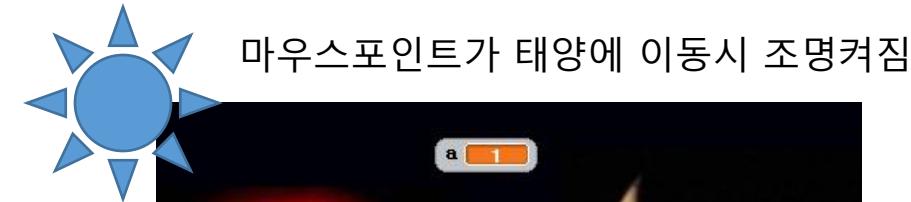
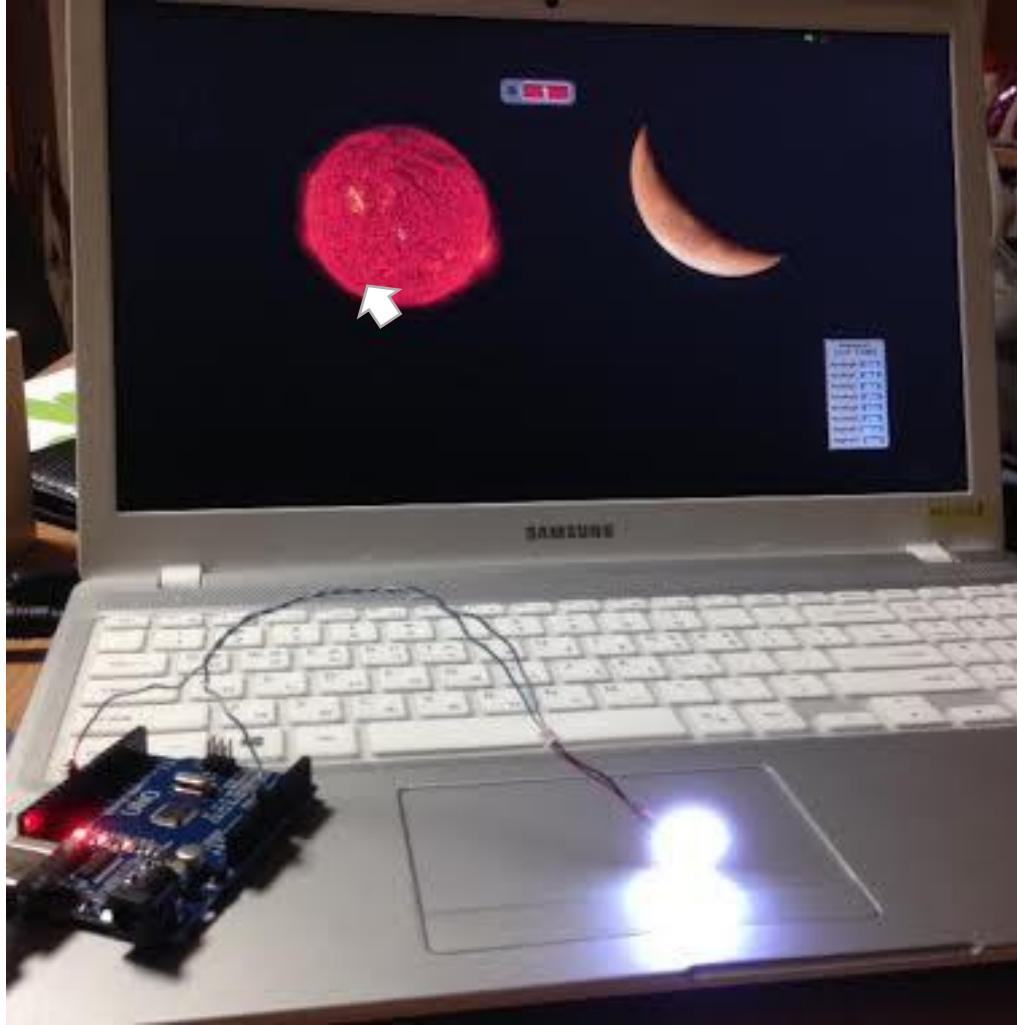


디지털 접점신호(10)



변수 "a"를 정하여
마우스가 해에 닿으면 변수 "a"는 1이 되며
켜지고 마우스가 달에 닿으면 "a"가 0이되
면서 LED조명이 꺼지게 소프트웨어 구성

해와 달 조명(3/3)

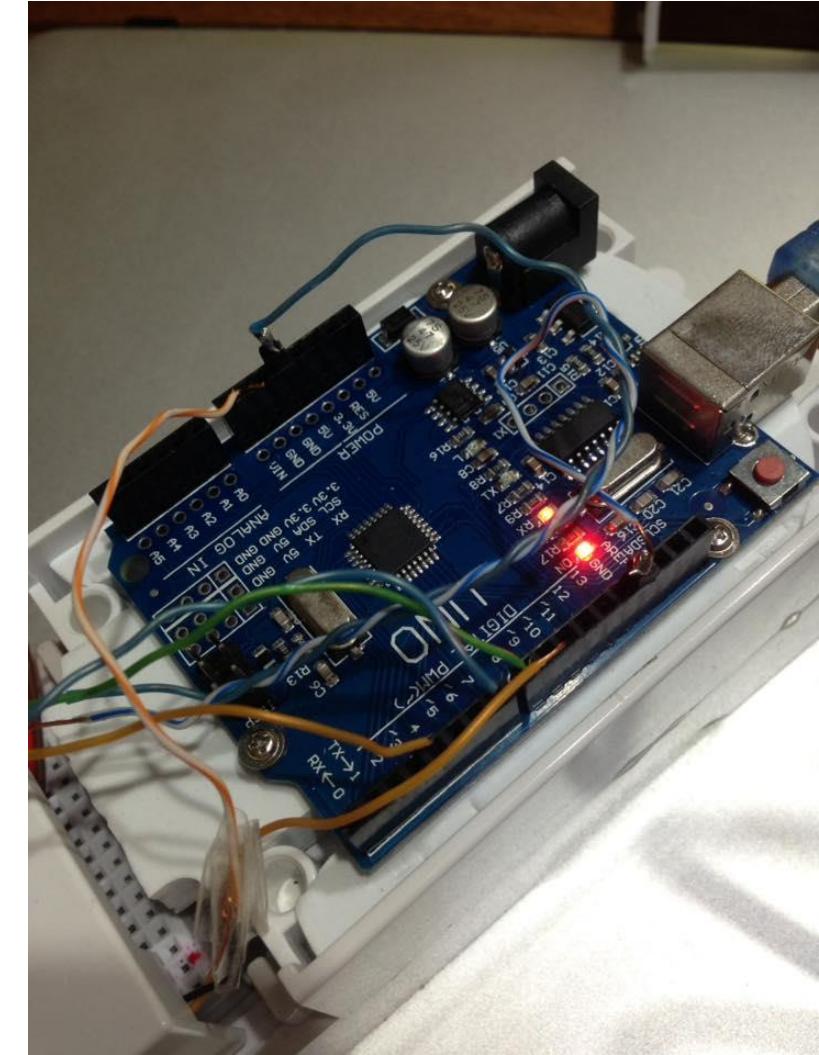
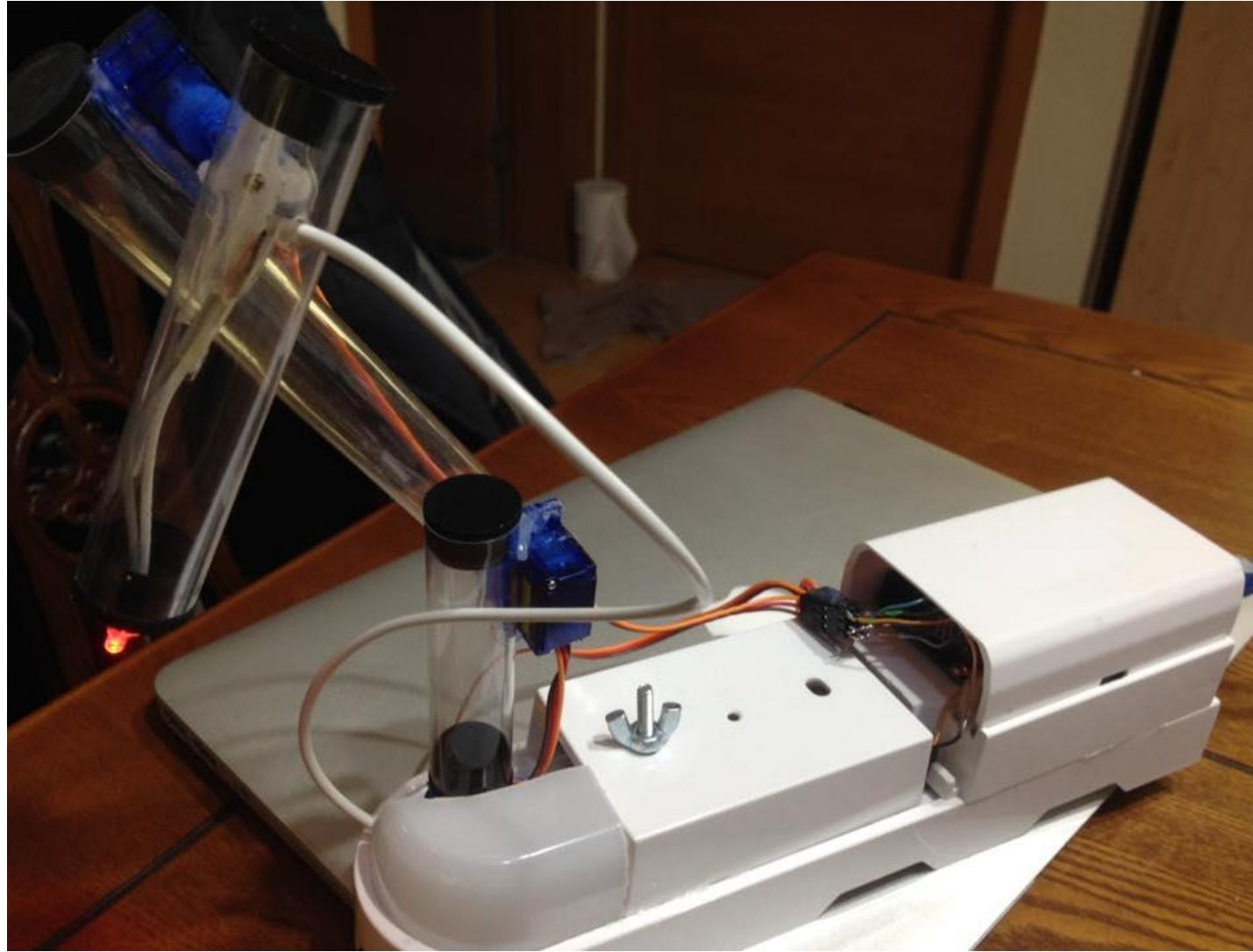


마우스포인트가 태양에 이동시 조명켜짐

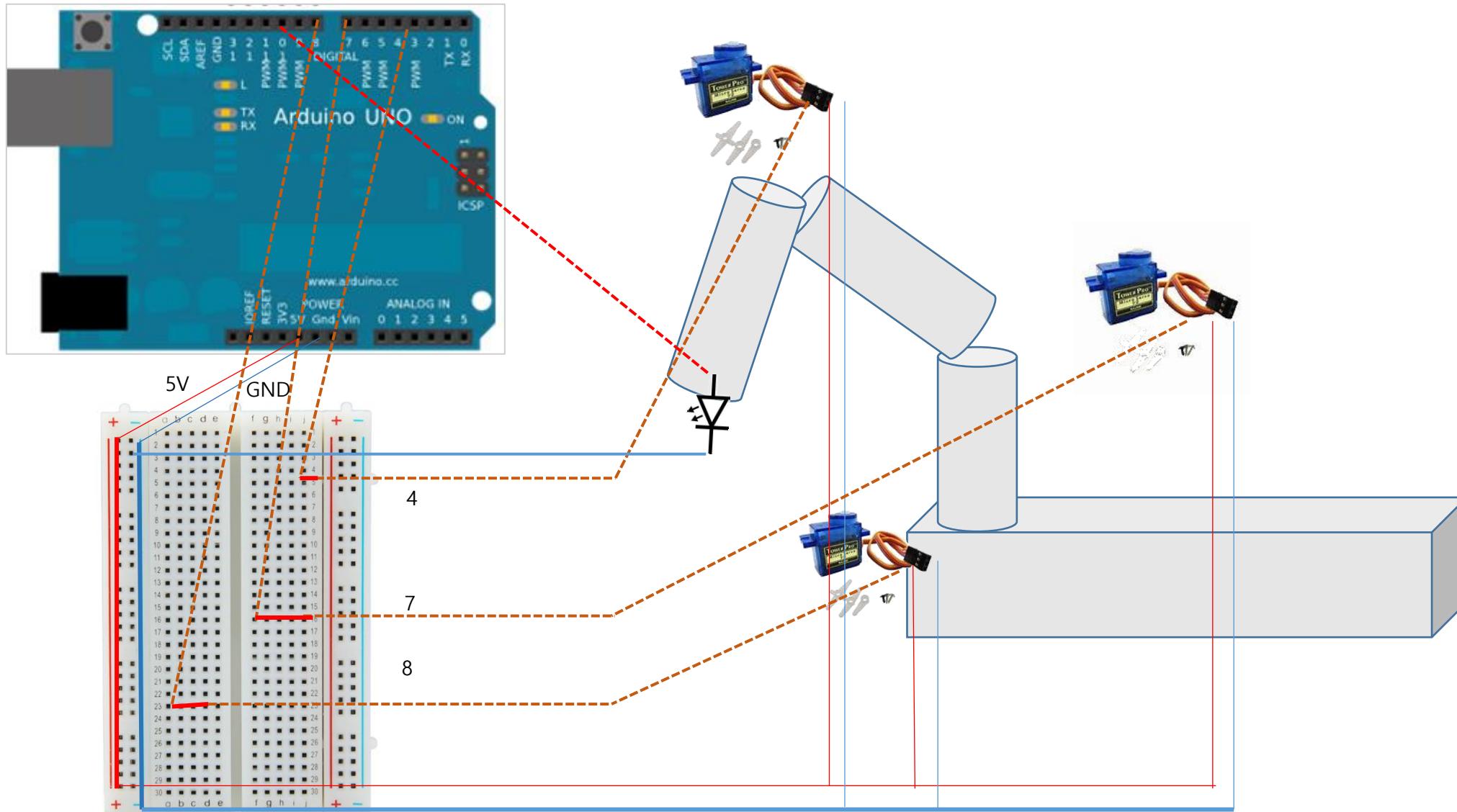


마우스포인트가 달로 이동시 조명켜짐

3축 관절 로봇 (1/3)



3축 관절 로봇 (2/3)



3축 관절 로봇 (3/3)

함수

소리

센서

마날로그0 센서의 값

센서 디지털2 가 눌렸을 때

디지털 13 번 출력 켜짐

디지털 13 의 출력 깨짐

마날로그입력 9 의 값을 255로

8 번 모터 경지

8 번 모터 회전방향을 시계방향으로

8 번 모터 각도를 180으로

작동장치 초기화

연결 끊기

제연결

센서값 보기

센서값 숨기기

센서값을 x= 0 ,y= 0 으로 이동

10 만큼 움직이기

15 ↵ 도 돌기

10 ↵ 도 돌기

Arduino1

스크립트 모양 소리

키 눌렸을 때 4 번 모터 각도를 180 으로

아래방향 화살표 키 눌렸을 때 4 번 모터 각도를 90 으로

위쪽 화살표 키 눌렸을 때 4 번 모터 각도를 270 으로

왼쪽 화살표 키 눌렸을 때 4 번 모터 각도를 Middle_P 으로

오른쪽 화살표 키 눌렸을 때 4 번 모터 각도를 SPIN 으로

z 키 눌렸을 때 만약 z 키 클릭하기? 라면 TOP 10 칙 누적하기 4 번 모터 각도를 TOP 으로

Robot_Arm_Motor

Arduino1 port: COM3

Analog0 193
Analog1 196
Analog2 201
Analog3 210
Analog4 225
Analog5 245
Digital2 false
Digital3 true

F-MODE

Key Key

A : Light On
S : Light Off

Q ↓
W ←
E ↑
R : Reset Position
Space Bar : Stop Motor

팔 : ↑ ↓
방향 : ← →

새로운 스프라이트:

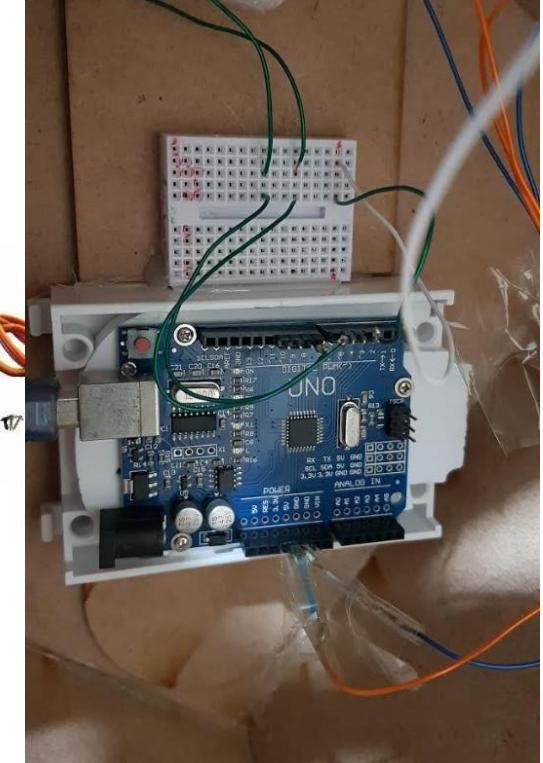
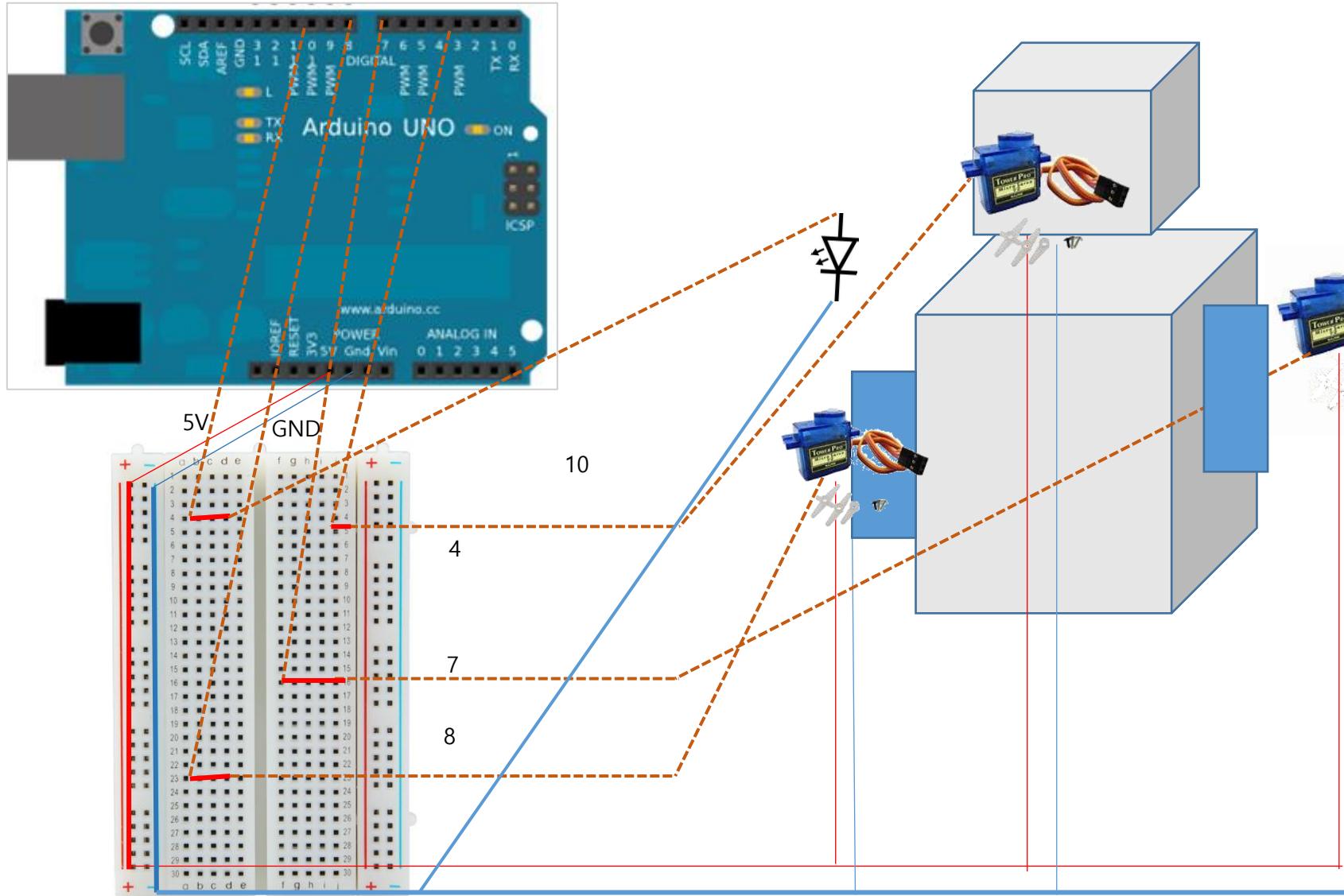
Arduin...

무대

3축 로봇 (1/4)



3축 로봇 (2/4)



3축 로봇 (3/4)

형태

소리

센서

마날로그0 센서의 값

센서 디지털2 가 눌려졌는가?

디지털 13 번 출력 거점

디지털 13 의 출력 깨짐

마날로그입력 9 의 값을 255로

8 번 모터 정지

8 번 모터 회전방향을 시계방향으로

8 번 모터 각도를 180으로

작동장치 초기화

연결 끊기

제연결

센서값 보기

센서값 숨기기

센서값을 x= 0 ,y= 0 으로 이동

10 만큼 움직이기

15 ↵ 도 돌기

15 ↲ 도 돌기

관찰

연산

변수

스크립트

x: 2 y: 1 방향: 90

e 키 눌렸을 때 4 번 모터 각도를 180 으로

w 키 눌렸을 때 4 번 모터 각도를 90 으로

d 키 눌렸을 때 4 번 모터 각도를 0 으로

아래방향 화살표 키 눌렸을 때 만약 아래방향 화살표 키 클릭하기? 라면 Middle_P 에 Middle_P - 10 저장 7 번 모터 각도를 Middle_P 으로

위쪽 화살표 키 눌렸을 때 만약 위쪽 화살표 키 클릭하기? 라면 Middle_P 에 Middle_P + 10 저장 7 번 모터 각도를 Middle_P 으로

왼쪽 화살표 키 눌렸을 때 만약 왼쪽 화살표 키 클릭하기? 라면 SPIN 에 SPIN + 10 저장 8 번 모터 각도를 SPIN 으로

오른쪽 화살표 키 눌렸을 때 만약 오른쪽 화살표 키 클릭하기? 라면 SPIN 에 SPIN - 10 저장 8 번 모터 각도를 SPIN 으로

x 키 눌렸을 때 만약 x 키 클릭하기? 라면 TOP 에 10 쪽 누적하기 4 번 모터 각도를 TOP 으로

z 키 눌렸을 때 만약 z 키 클릭하기? 라면 TOP 에 -10 쪽 누적하기 4 번 모터 각도를 TOP 으로

키 눌려요

Robot_Wall-E

LINK BOX

F Mode : M-F Key
G Mode : M-G Key

F-MODE

R : Reset Position
Space Bar : Stop Motor

맞춤조정 PRESET

W ↴

E ↵

미세조정:
Z(←) X(→)

원팔 :
↑ ↓ ↕ ↖ ↗ ↙ ↚ ↛

새로운 스프라이트: ★ ? A: B: C: D:

무대

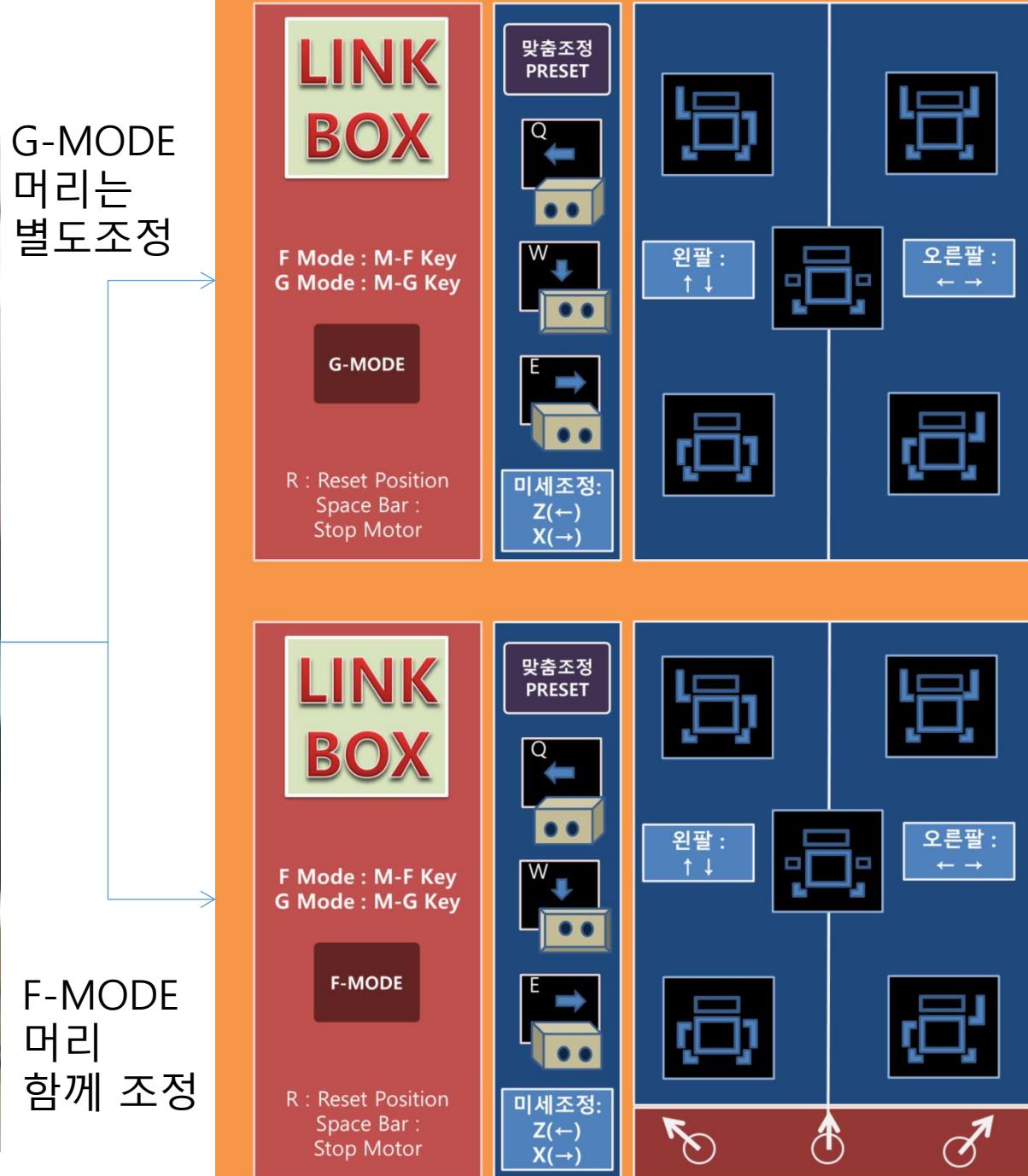
Arduino...

3축 로봇 (4/4)

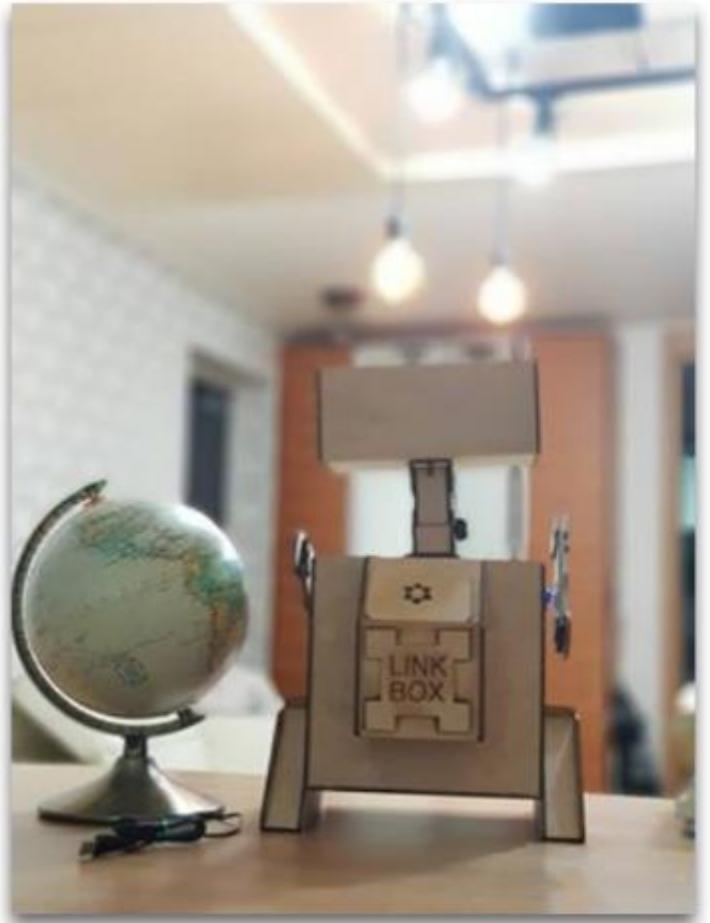


G-MODE
머리는
별도조정

F-MODE
머리
함께 조정



Link Box Project



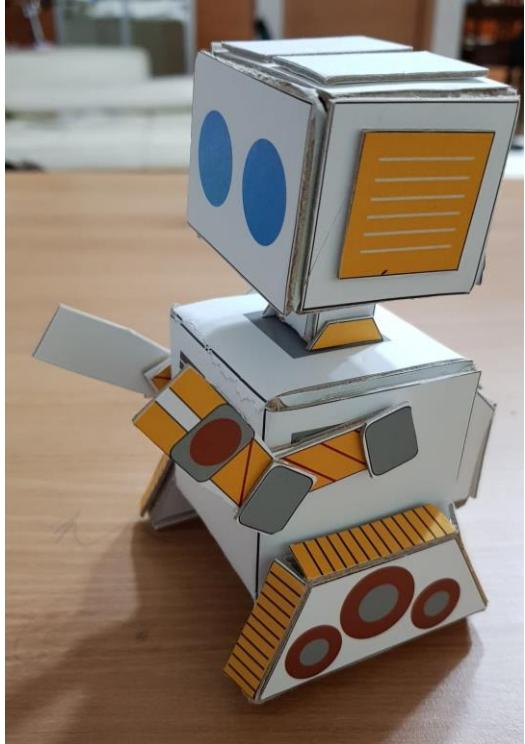
기술협력	감성분야
기술 커뮤니티와 협력생태계 구성	인간과 상호작용을 위한 감성디자인연구
기존, 가전 스마트폰 인공지능스피커 등 연결성 높인다.	사회적 이슈 사회적문제 해결

프로젝트가 추구하는 것

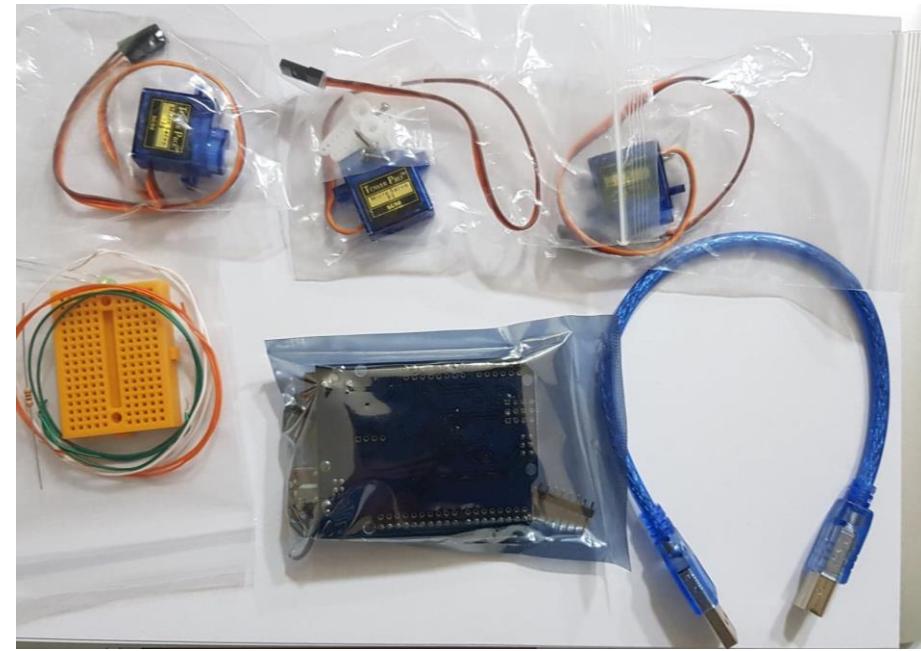
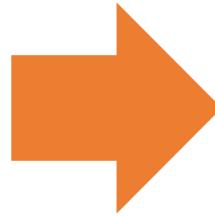
- 재활용**
 - 책상서랍에서 낭비되고 버려지는 중고 스마트폰의 재활용으로 환경문제 해결을 위한 업사이클 활동.
 - 버려지는 스마트폰 및 박스 등을 재활용한다.
- 편의성/활용성**
 - 음악 듣는 기능, 반려동물 관리, 화상회의 등
 - 기구물 디자인을 통해 카페 및 가정에서 인테리어용 소품 역할
 - 단순한 기능으로 활용성 높인다.

제작 준비물

1. 노트북
2. 아duino노킷 1 SET
3. 페이퍼크래프트 1 SET
4. 기타 : 양면테이프, 자, 가위 등



Starter Kit Sample



ADUINO KIT 1 SET



The Link-Box Starter Kit includes:

- 1 Link Box Projects paper craft (4 Pages)
- 2 Arduino UNO 1 EA
- 3 USB cable 1EA
4. Breadboard 1EA
5. Stranded jumper wires 1 SET
6. LED 1EA
7. Small servo motor 3EA
8. Resistors 50 Ohms 1EA



Link Box Projects paper craft

화면1

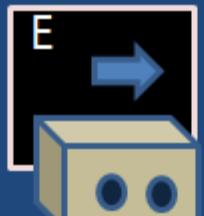
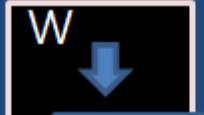
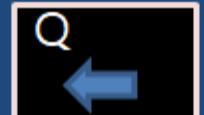
LINK BOX

F Mode : M-F Key
G Mode : M-G Key

G-MODE

R : Reset Position
Space Bar :
Stop Motor

맞춤조정
PRESET



미세조정:
Z(←)
X(→)

왼팔 :
↑ ↓



오른팔 :
← →



화면2

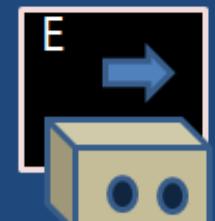
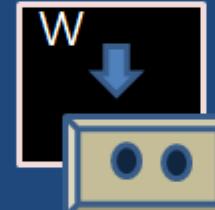
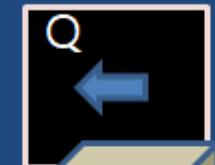
LINK BOX

F Mode : M-F Key
G Mode : M-G Key

G-MODE

R : Reset Position
Space Bar :
Stop Motor

맞춤조정
PRESET



미세조정:
Z(←)
X(→)

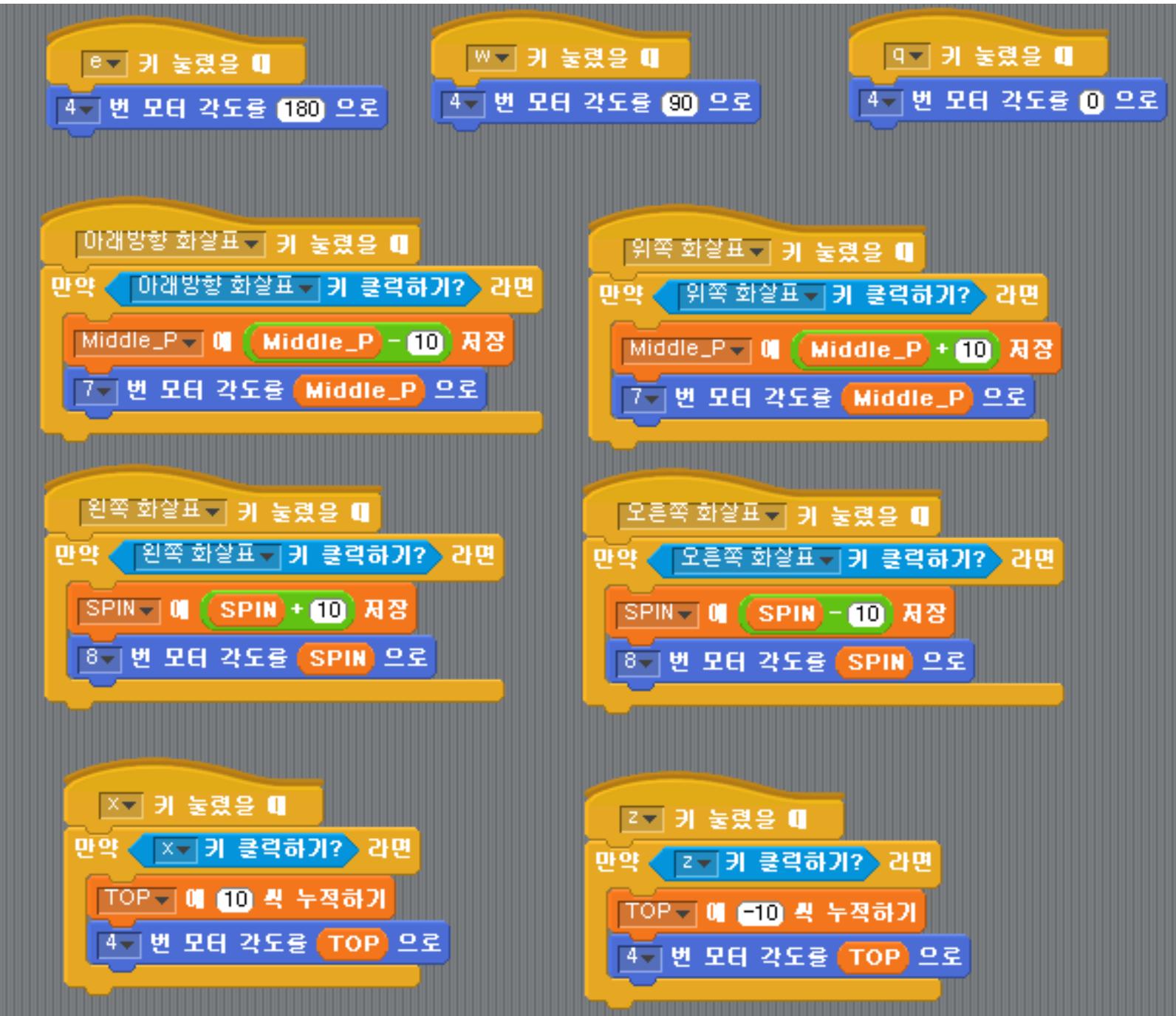
왼팔 :
↑ ↓



오른팔 :
← →



코드1



코드2



구성도

