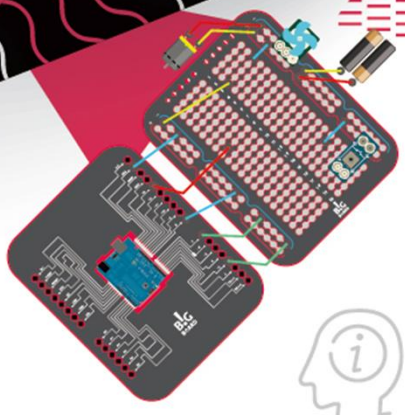
ARDUINO

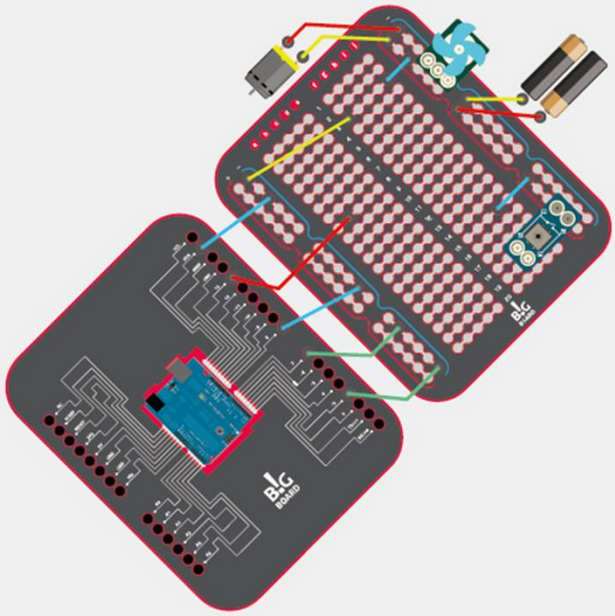
빅보드와 함께 하는
코딩놀이

"스케치용"



open source hardware

B!G BOARD



Bigboard starter Kit

빅보드와 함께 하는
코딩놀이

ARDUINO

open source hardware

Contents

BIG BOARD

01 코딩이란?

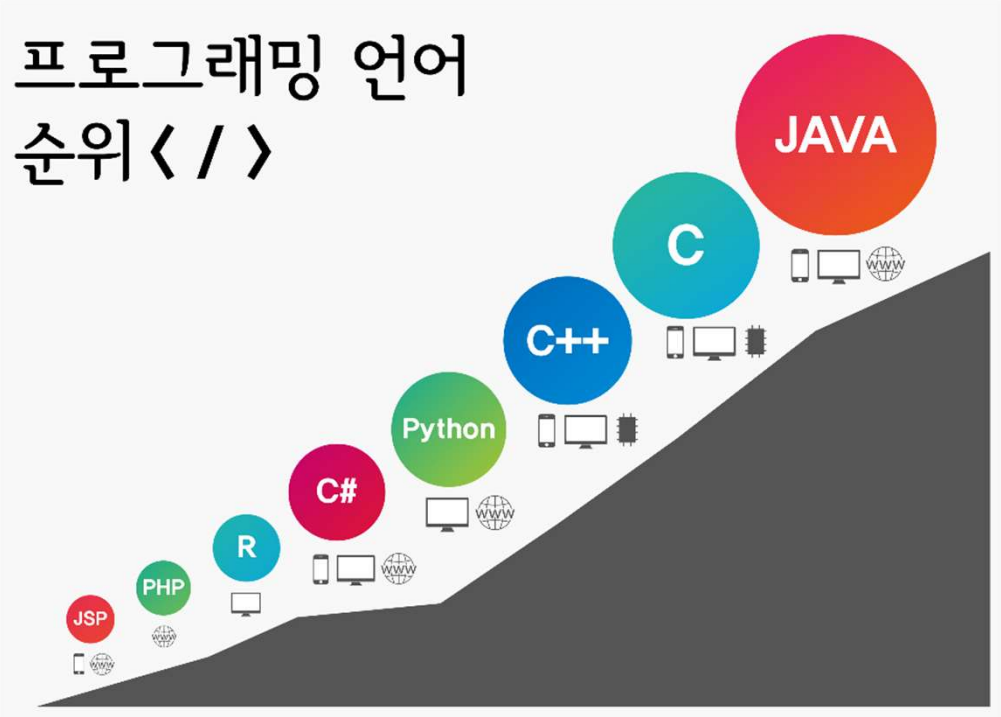
1) 코딩(프로그래밍)이란?

프로그램 언어를 이용해 컴퓨터에서 작동되는 프로그램을 만드는 과정을 코딩(Coding)이라 합니다. 프로그램 코드를 만드는 것을 이야기합니다.

2) 프로그램 언어

세상에는 참으로 다양한 종류의 컴퓨터 언어가 있어요. 프로그래밍 언어란 일반적으로 컴퓨터 언어가 명령을 만들기 위해 사용하는 언어를 말합니다. 프로그램은 컴퓨터를 실행시키기 위해 지시하는 명령어들의 한 묶음인 샘이에요.

스크래치(Scratch), 파이썬(Python), C, C++, 자바(Java), 자바스크립트 등이 있습니다.



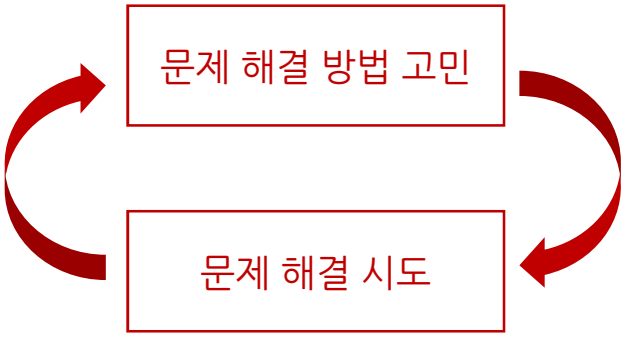
[그림 설명 : 프로그래밍 언어별 사용 순위를 표시합니다.]

3) 코딩 하는 법

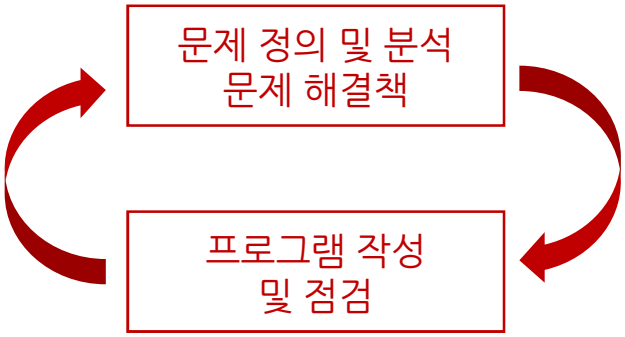
코딩하는 과정, 즉 프로그램을 만드는 과정은 사람들이 문제를 해결하는 과정과 비슷합니다.

우선 문제가 무엇인지 생각하고 이 문제를 해결할 수 있는 방법을 생각합니다. 방법이 생각이 나면 그 방법대로 문제를 해결해보고 문제가 해결되지 않으면 다시 해결 방법을 고민합니다. 이런 과정을 문제가 해결될 때까지 반복합니다.

코딩하는 법도 문제를 정의하고 이 문제의 해결책을 생각하고 해결책이 찾아지면 코드로 만들어 시도하고 해결될 때까지 이 과정을 반복합니다.



[문제 해결 과정]



[코딩 과정]

4) 코딩 교육이란?

코딩 교육은 코딩하는 방법을 교육하는 과정입니다. 이 과정의 주요 목표는 문제를 해결하는 방법을 습득하는 것과 코딩 기술을 배우는 것입니다. 이 과정에서 학습자의 창의력이나 사고력이 같이 발달될 수 있습니다.

5) 코딩교육의 필요성

문제를 해결하는 방법을 배울 수 있은 코딩 교육은 이전의 암기식 교육과는 다른 여러 가지 장점을 제공합니다.
여러 문제가 발행하더라도 이 문제를 창의적인 사고를 이용해 다양한 방법으로 문제를 해결을 시도할 수 있게 합니다.

[코딩 교육의 효과]

[선진국의 코딩교육시기]

[출처 : 2014년 12월05일 KBS 9시 뉴스보도]

▶ 한국 : 2018년 중학교, 2019년 초등, 고등 공교육 실시 예정

6) 코딩 교육의 핵심은?

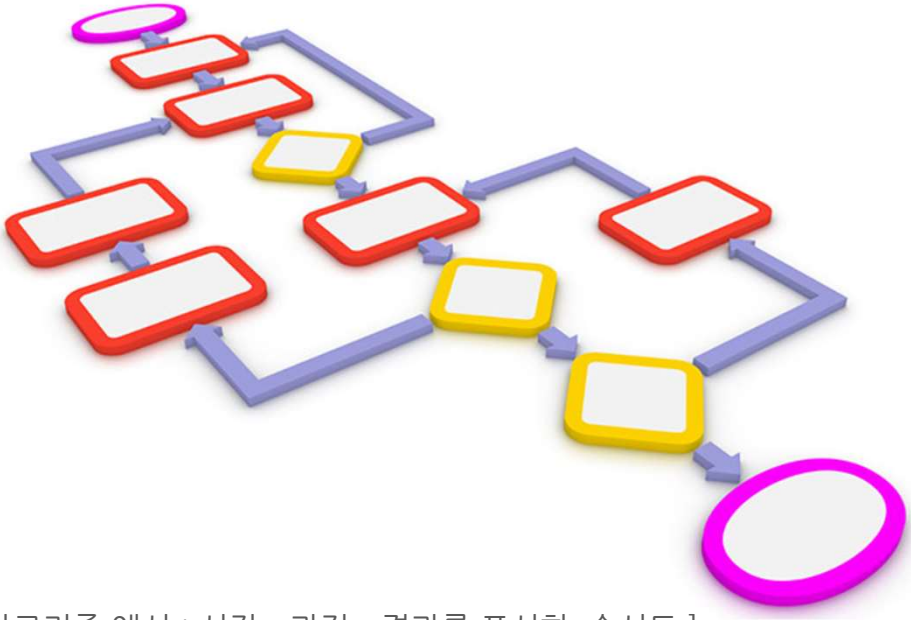
프로그래밍을 알아가는 과정에서 가장 중요한 것은 **문제를 해결하며 프로그램을 만드는 과정(*알고리즘) 을 이해하는 것입니다.**

프로그램에 사용하는 프로그램 언어를 아는 것은 그 과정일 뿐, 우선 순위가 아니며, 프로그래밍의 중심이 되어서는 안됩니다.

프로그래밍 언어와 개발 툴 사용 방법만 배워 다른 사람의 코드를 수정하는 반쪽 짜리 코더가 아닌, 스스로 문제를 해결해 나가는 '진정한 프로그래머'가 될 수 있도록 합시다!

*** 알고리즘 :**

어떤 문제의 해결을 위하여, 원하는 결과가 나오도록 하는 규칙(명령어)의 집합을 의미합니다. [메카솔루션 '아두이노를 활용한 기초 소프트웨어 교육' 교재 인용]



[알고리즘 예시 : 시작 - 과정 - 결과를 표시한 순서도]

7) 빅보드로 코딩 교육하기

피지컬 컴퓨팅이란?

디지털 기술 및 장치를 이용하여, 사용자로부터 물리적인 방식으로 정보를 입력 받아(Sensor) 처리한 결과를 물리적인 방식(Actuators)으로 출력하는 컴퓨팅입니다.

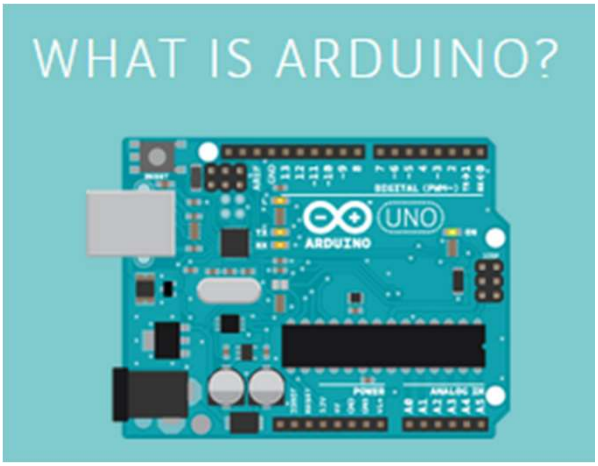
상호 작용 가능한 시스템이 현실세계의 모습을 각종 센서들을 통해 이해하고 모터나 LED, 그 밖의 각종 작동 가능한 기기들이 반응하는 시스템을 만드는 일련의 과정입니다.

본 교재는 창의적 사고와 융합 능력을 키우기 위해서,

코딩 프로그램 툴 (스케치) + 하드웨어 (아두이노=빅보드)를 활용하여 기초 프로젝트 예제들을 따라 하면서, 순차적으로 배우고 익히도록 설명을 할 예정입니다.

02 아두이노 소개

1) 아두이노 소개



“아두이노는 사용하기 쉬운 하드웨어와 소프트웨어를 기반으로 한 오픈 소스 전자 플랫폼입니다.

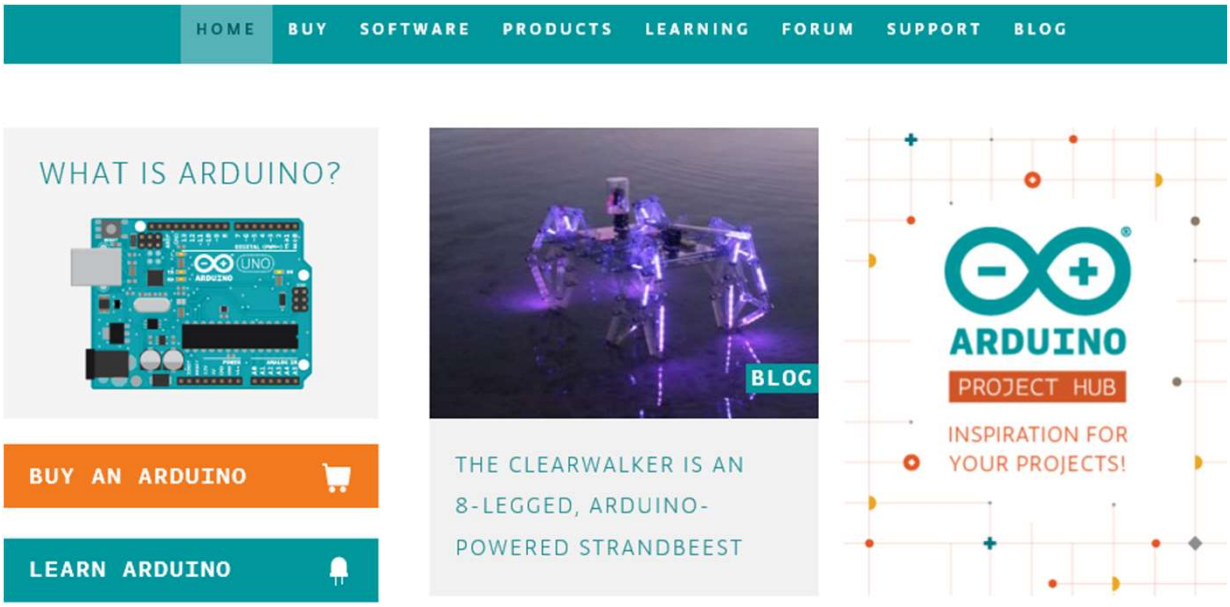
[그림1. 아두이노 회사 홈페이지]
(출처 : arduino.cc)

2) 오픈소스 하드웨어란?

open source hardware

오픈 소스는 말 그대로 아두이노를 이용하는 데에 관련된 모든 지적 재산물(source)이 열려있다(open)는 의미입니다. 아두이노의 소스라고 하면, 아두이노 하드웨어 장치의 회로도, 그리고 소프트웨어 원천 코드 등이 해당되겠습니다. 회사를 예로 들면 회사의 핵심 상품의 원천 기술을 모두 공개했다는 것과 같은 의미 입니다. 그래서 많은 사람들이 아두이노를 이용해서 프로젝트를 만들거나, 비즈니스에도 활용하는 기업들이 늘고 있습니다.

3) 아두이노의 구성



[그림2. 아두이노 회사 홈페이지] (출처 : arduino.cc)

”아두이노 보드”라는 것은 마이크로 컨트롤러 입니다. 아두이노 회사 사이트에서는 아두이노의 **소프트웨어+하드웨어+커뮤니티**를 모두 합쳐서 아두이노라고 부르고 있습니다.

따라서 아두이노는 플랫폼이라고 부를 수 있게 되는 겁니다. 그리고 아두이노는 다양한 것들을 만들 수 있는 기반이 되기 때문이기도 합니다.

03 아두이노 와 연동되는 소프트웨어

1) 아두이노 스케치(Sketch) - 본 교재에 적용

- 아두이노(https://www.arduino.cc/en/Main/Software) 그룹에서 제공하는 무료로 제공하는 통합프로그램
- 텍스트(타이핑)형태의 코딩 교육

2) 스크래치(Scratch)

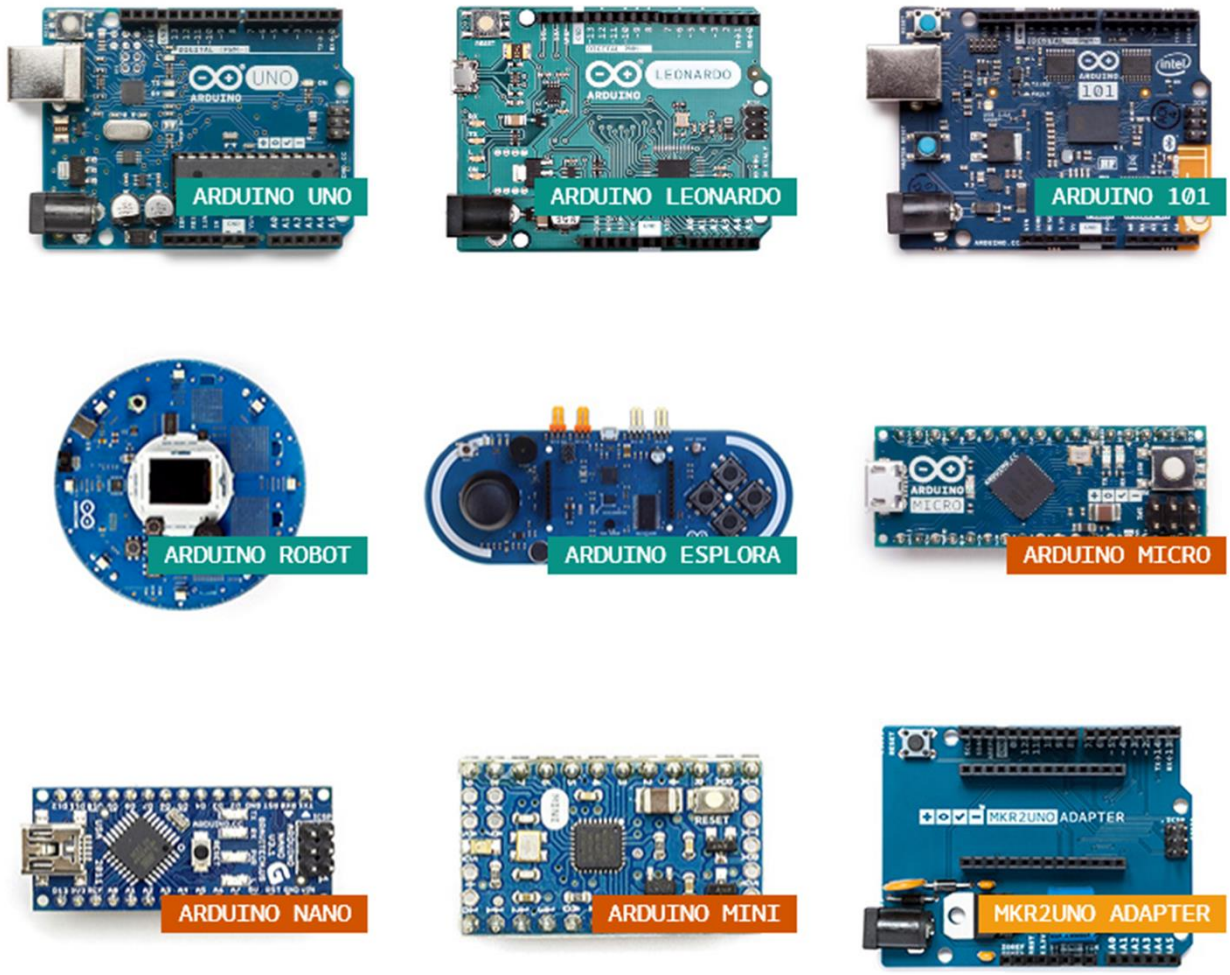
- 메사추세츠 공과대학 (MIT, https://www.scratchjr.org/) 에서 제공하는 무료로 제공하는 통합프로그램
- 그림/블록 형태의 코딩 교육

3) 엔트리(Entry)

- 한국의 네이버커넥터(www.play-entry.org) 재단에서 무료로 제공하는 프로그램
- 그림 / 블록 형태의 코딩 교육

04 아두이노의 하드웨어(보드)

1) 하드웨어(보드)- 기초 과정용 제품



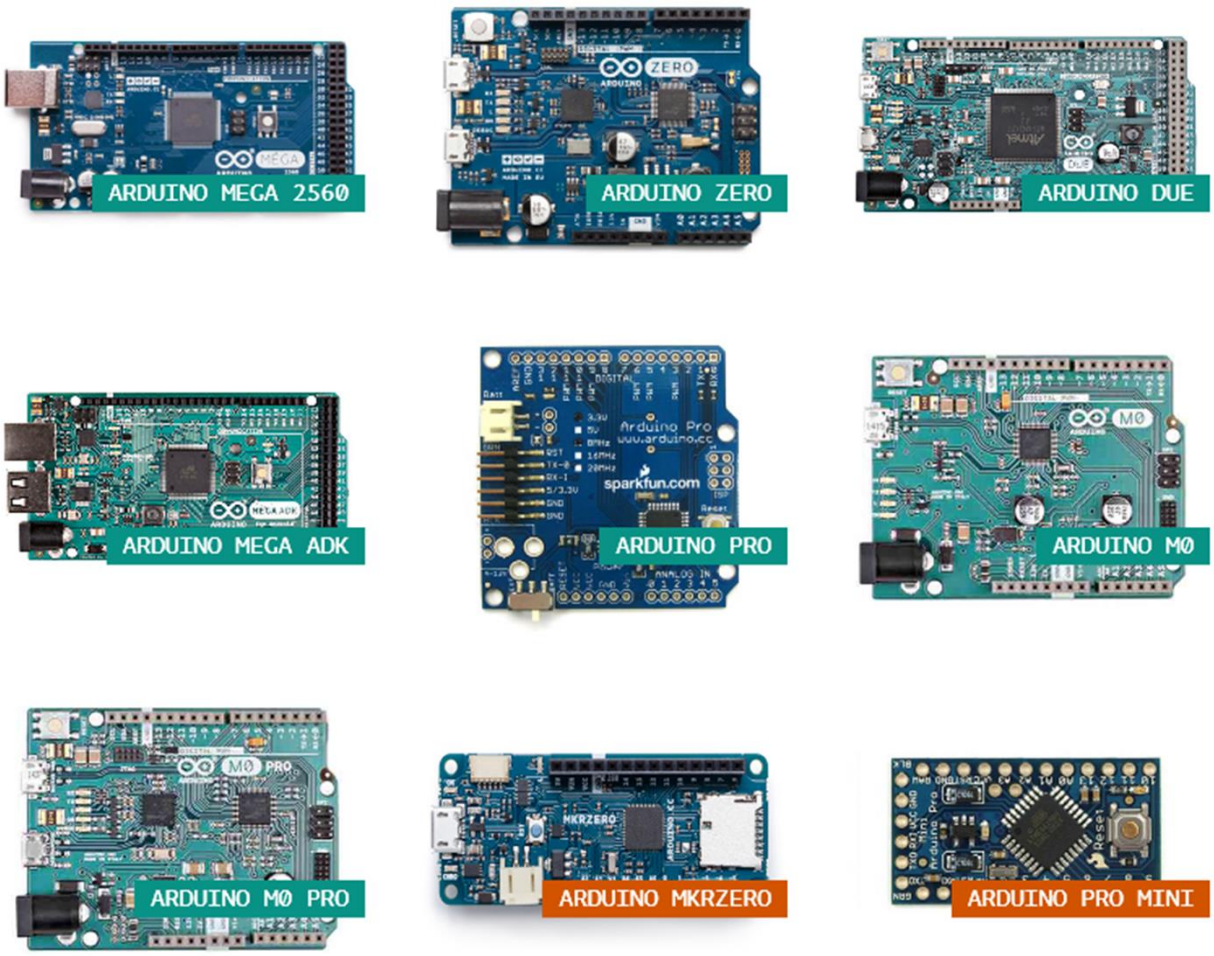
[그림3. 아두이노 회사 홈페이지] (출처 : arduino.cc)

기초 단계의 제품을 사용하여, 아두이노를 시작하세요!

사용하기 쉽고, 첫번째 창의적인 프로젝트를 수행할 준비가 되어 있어야 합니다. 이 보드와 모듈은 전자 장치 및 코딩을 배우고 조정할 때 가장 좋습니다.

04 아두이노의 하드웨어(보드)

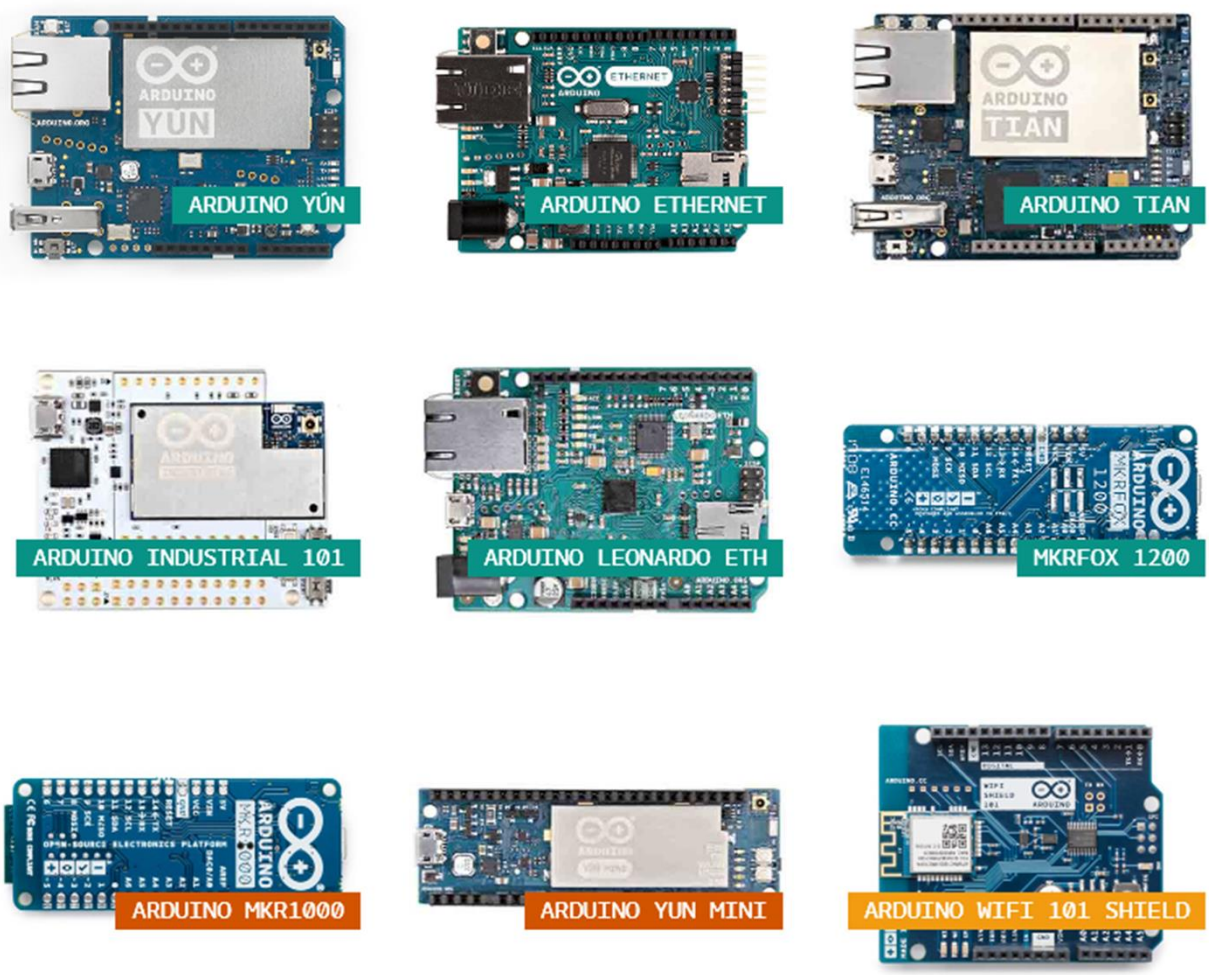
2) 하드웨어(보드) - 중급 과정용 제품



[그림4. 아두이노 회사 홈페이지] (출처 : arduino.cc)

고급 기능을 갖춘 보드 중 하나를 선택하거나 빠른 성능을 제공하는 보다 복잡한 프로젝트의 흥분을 경험하십시오.

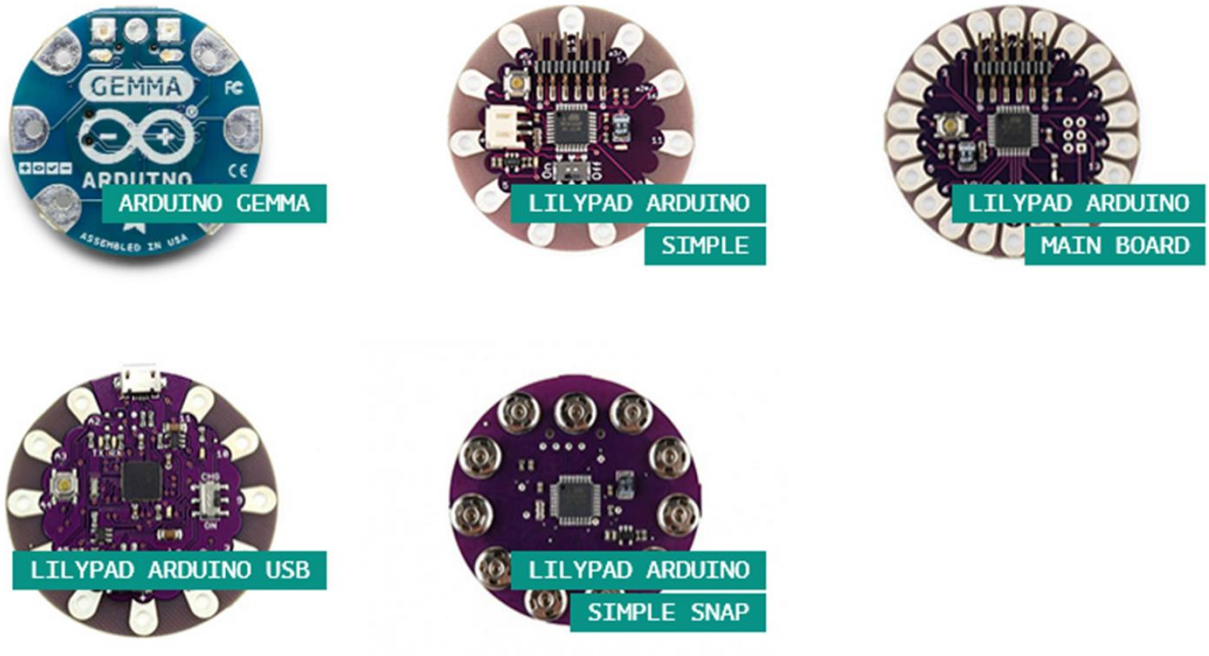
3) 하드웨어(보드) - 사물인터넷 과정용 제품



[그림5. 아두이노 회사 홈페이지] (출처 : arduino.cc)

이러한 사물인터넷 제품 중 하나를 사용하여 간편하게 장치를 만들고,
인터넷 연결을 통한 독창성을 열어보십시오.

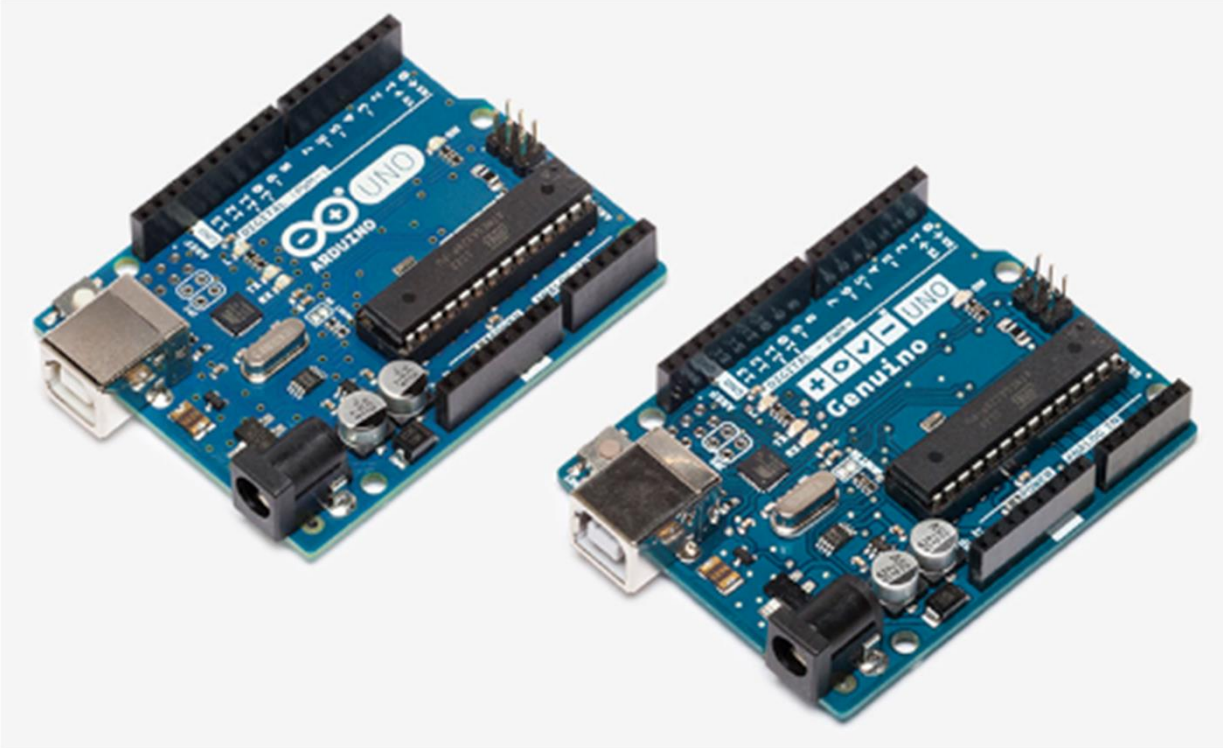
4) 하드웨어(보드) - 휴대 가능한 (웨어러블) 과정용 제품



[그림6. 아두이노 회사 홈페이지] (출처 : arduino.cc)

부드러운 프로젝트에 스마트 함을 더하고, 전자 제품의 힘을 섬유에 직접 바느질하는 마법을 발견하십시오.

5) 그 중에 아두이노 우노 보드 선택

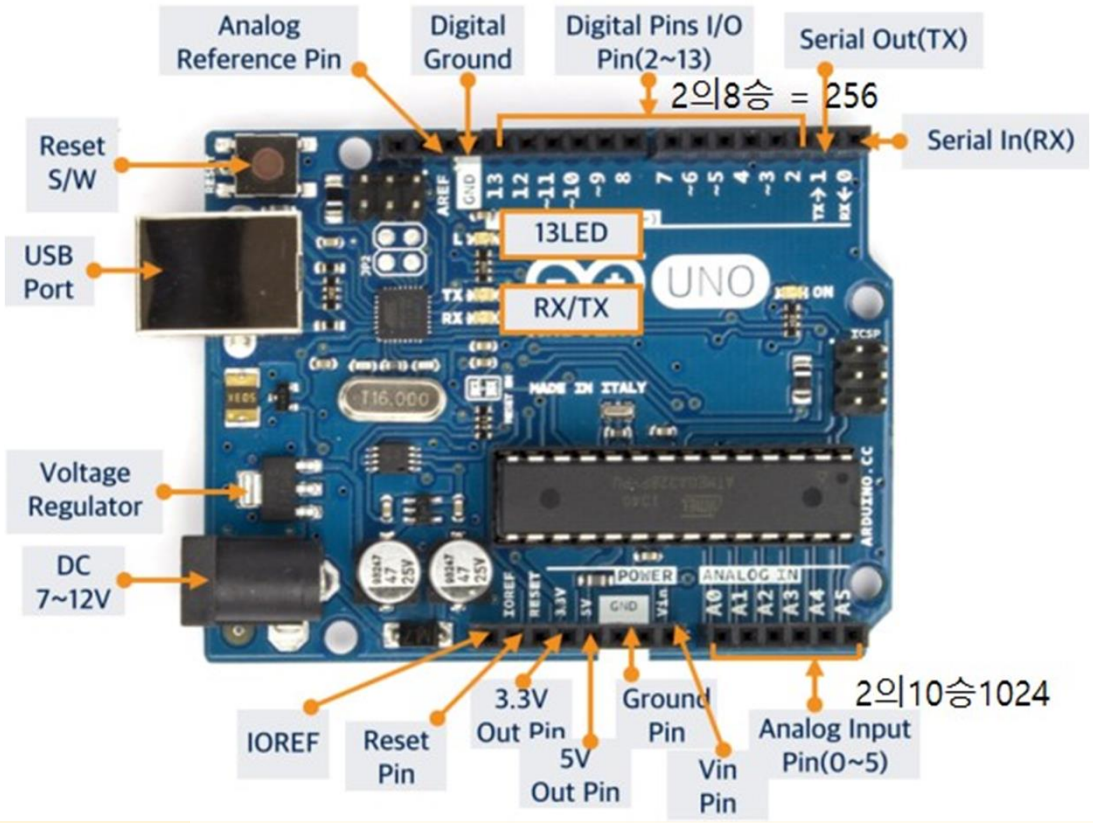


[그림7. 아두이노 회사 홈페이지] (출처 : arduino.cc)

우노(UNO)는 전자 제품 및 코딩을 시작하기에 가장 좋은 보드입니다. 아두이노 플랫폼에 처음 익숙해진 경험이라면, 우노가 가장 먼저 시작할 수 있는 보드입니다. 우노는 Arduino & Genuino 전체 제품군 중에 가장 많이 사용되고, 문서화된 보드입니다. 가장 먼저 출시된 제품이라 가장 많은 프로젝트와 교육용 자료가 잘 공유되어 있습니다.

최근 전자쇼핑몰에 아두이노우노 제품의 클론(100% 호환)제품이 6천원에 판매되고 있어, 가격의 부담이 전혀 없습니다.

6) 아두이노 우노 보드 살펴보기



Reset S/W	마이크로 컨트롤러를 리셋 시키기 위해 리셋 핀을 LOW로 만들었으며, 보통의 경우 쉴드에 리셋 버튼을 추가할 때 사용됩니다.
USB Port	USB Port를 통해서 컴퓨터와의 통신 및 전원을 공급합니다.
Voltage Regulator	LM1117칩으로 출력전압을 5V로 공급해주는 역할을 하며 1A정도의 전류를 처리할 수 있습니다.
DC 7~12V	USB 포트를 통해서 전원을 공급받지 않고 아두이노 단독으로 동작을 할 때, DC파워 잭에 7~12V DC Adapter를 연결해서 전원을 공급하면 됩니다.
IOREF	보드 상의 MCU(Micro Controller Unit)가 동작하는 전압에 대한 레퍼런스를 쉴드에 제공해서 읽힌 전압에 따라 쉴드가 적절한 전원 소스를 선택하거나 Voltage translator를 활성화 시키도록 합니다.
Reset Pin :	마이크로 컨트롤러를 리셋 시키기 위해 리셋 핀을 LOW로 만들었으며, 보통의 경우 쉴드에 리셋 버튼을 추가할 때 사용됩니다.

3.3V Out	보드 상의 레귤레이터를 통해 3.3V를 출력합니다, 최대 전류는 50mA
5V Out	보드 상의 레귤레이터에서 생성된 5V를 출력합니다.
Ground Pin	그라운드 핀, 기준이 되는 전압, 0V입니다.
Vin Pin	외부 전원 소스를 사용할 경우 사용하는 입력전압 핀입니다.
Analog Input	A0~A5핀은 센서 등에서 들어오는 아날로그 값을 읽기 위한 핀입니다.
Serial Rx/Tx	시리얼 통신에 사용되는 핀 입니다.
Digital I/O	Digital값의 입출력을 위한 핀이며 (~)는 PWM핀으로 0~255까지 256단계의 아날로그 출력이 가능합니다. 2번과 3번핀은 이벤트 처리를 위한 인터럽트(Interrupt)기능을 갖고 있습니다.
Digital Ground	그라운드 핀, 기준이 되는 전압, 0V입니다.
Analog Reference Pin	아날로그 입력의 레퍼런스 전압. analogReference()함수와 같이 사용됩니다.

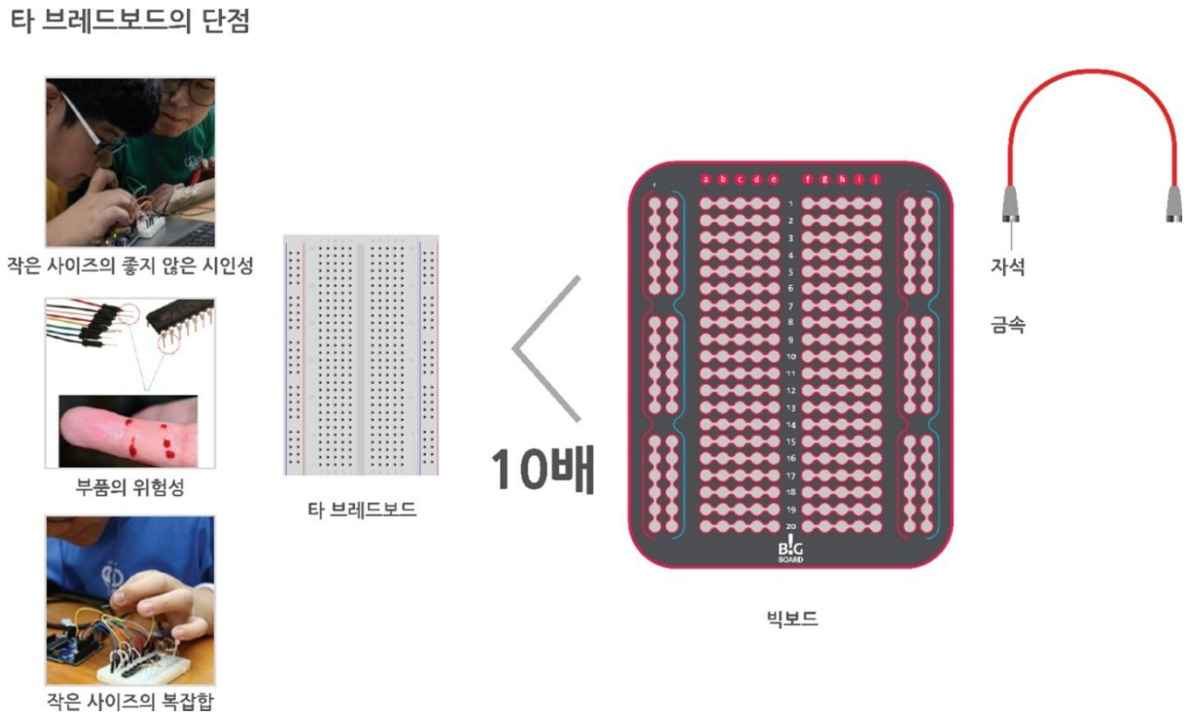
7) 아두이노 전기적 특성

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

05 빅보드 소개하기

1) 빅보드의 탄생 이야기

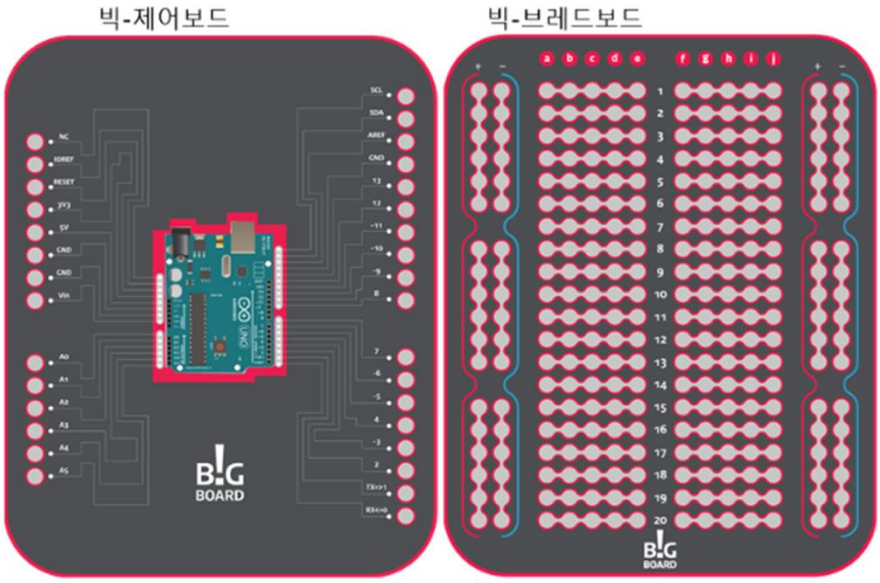
”아두이노 우노”는 초기 2005년 개발 당시는 해당 연령이 대학생들이었습니다. 또한, 전공자들의 기초 교육용으로 활용되었습니다.
현재는 어린이부터 성인에 이르기까지 창의력과 문제해결 능력을 키우는 코딩용 교구로 선진국부터 늘리 활용되고 있습니다.
이에 빅보드 연구원들은 불편한 점들을 고민하기 시작했습니다.

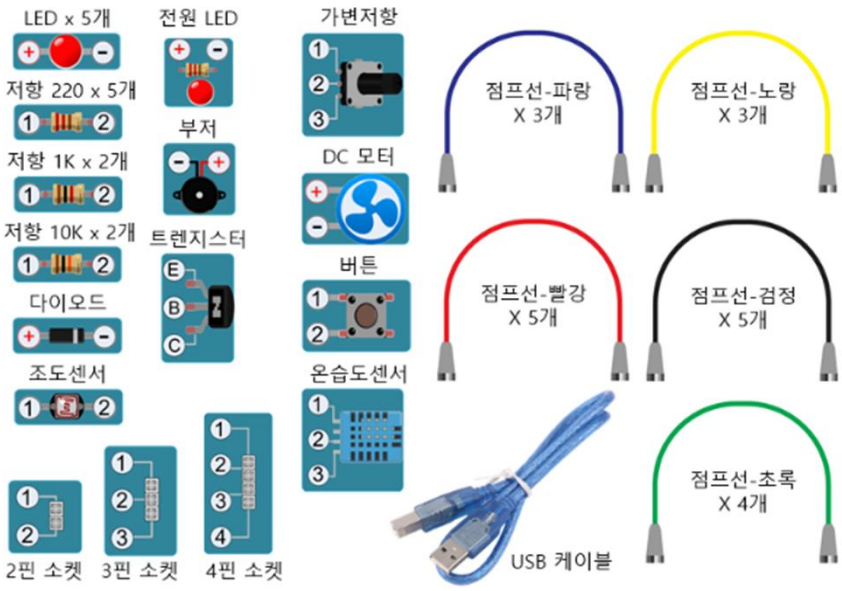


”빅보드” 는 아두이노 우노와 브레드보드와 똑같은 모양과 동작으로 10배 크기로 만들어 졌습니다.
또한, 자석을 이용한 연결로 다칠 우려가 없이 안전하며, 연결이 쉬워서 누구나 쉽게 보고, 배우며, 빠른 성취감을 느낄 수 있습니다.

2) 빅보드 기본형 - 전체 구성

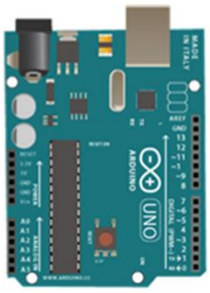
❖ 빅보드 기본형에는 빅-제어보드, 빅-브레드보드, 점프선, 여러 가지 부품들로 구성됩니다. 상세한 구성은 아래 그림과 같습니다.

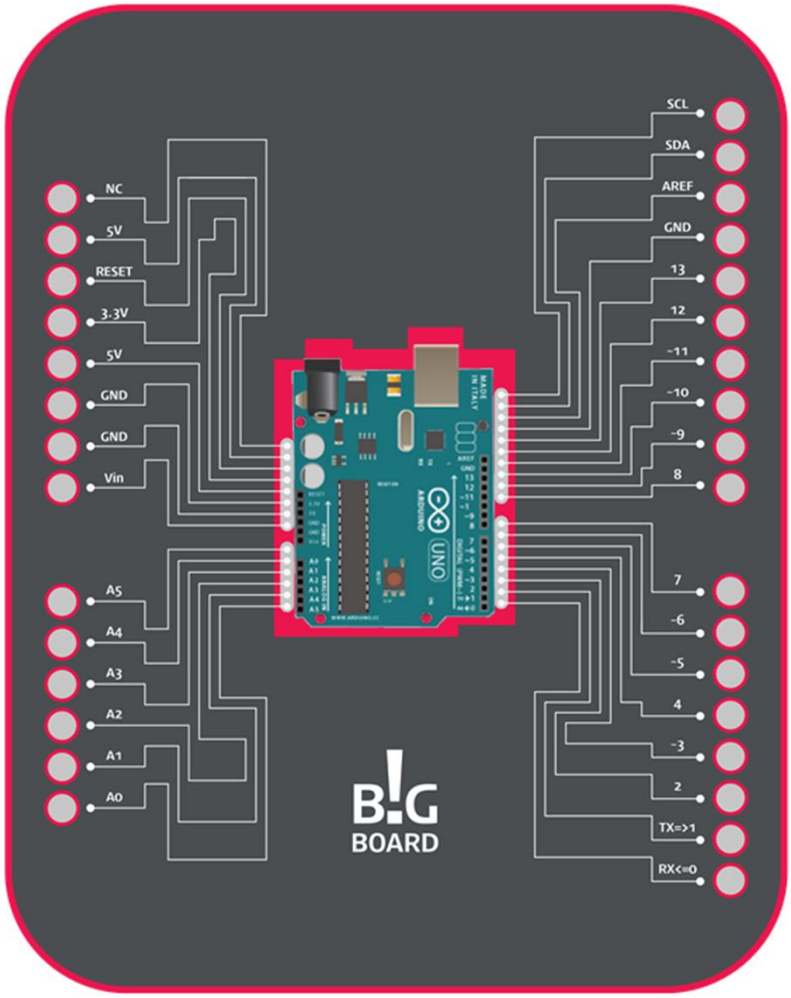




*위의 구성은 품질 향상을 위해, 변경될 수 있습니다.

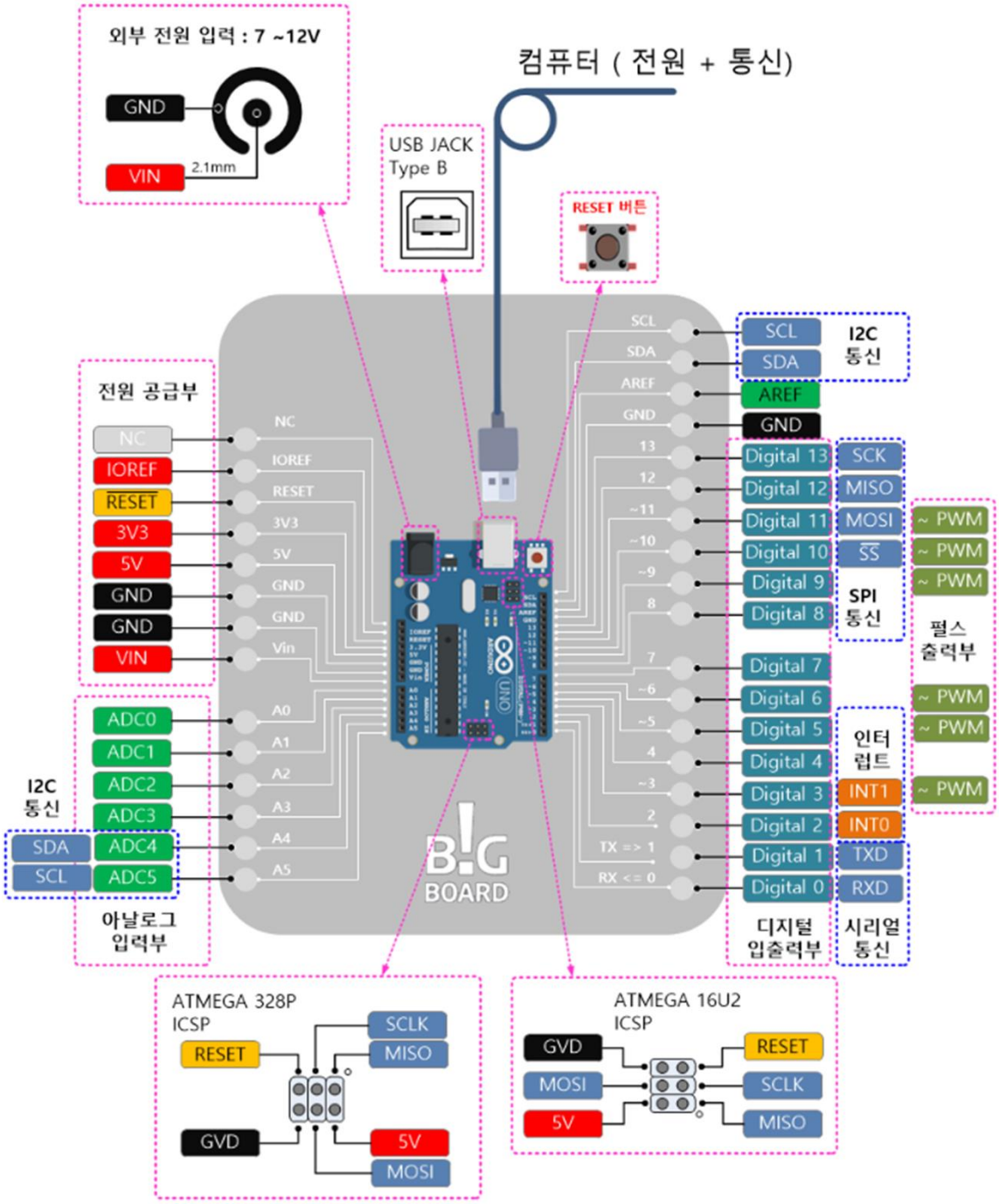
3) 빅 - 제어 보드 설명



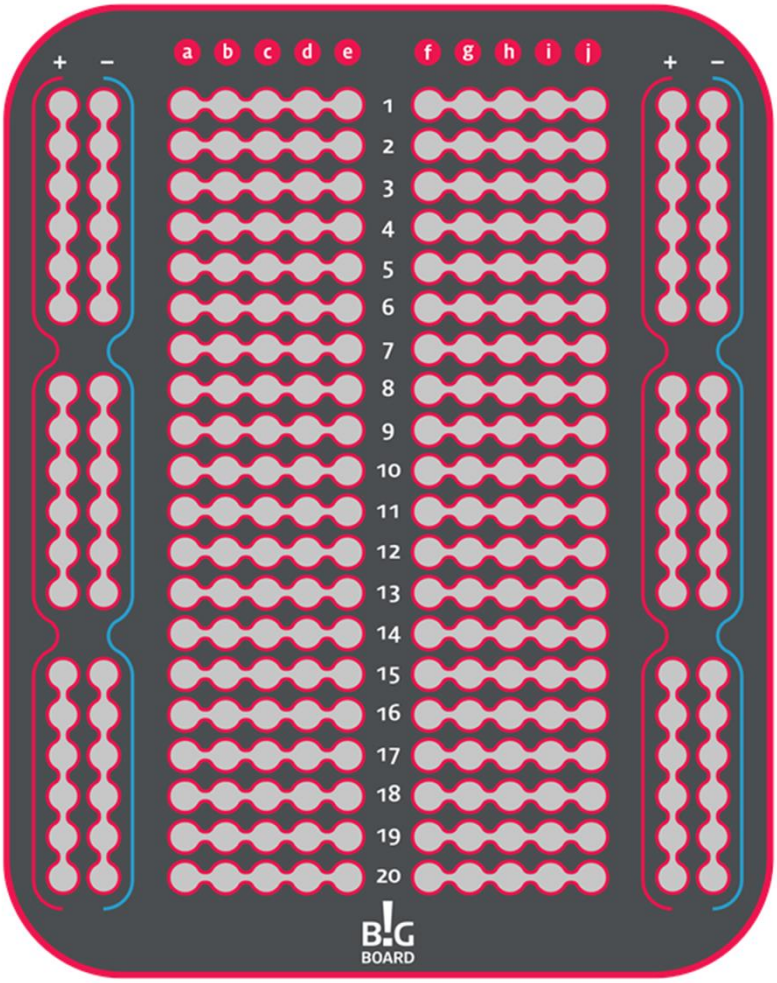


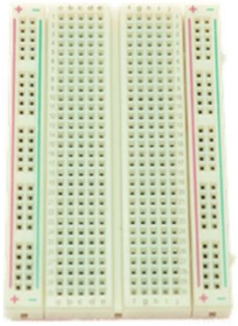
- 빅-제어보드 = 아두이노우노 x 10배
- 물리적 크기만 10배 하였으므로, 동작이나 사용 프로그램은 모두 똑같습니다.
- 또한, 빅-브레드보드 내에 부품과 회로 연결용 점프선(자석)이 연결될 수 있는 구조로 되어 있습니다.

4) 빅 - 제어 보드 설명


외부 전원 입력 : 7 ~12V
GND
VIN
2.1mm
컴퓨터 (전원 + 통신)
USB JACK
Type B
RESET 버튼
전원 공급부
NC
IOREF
RESET
3V3
5V
GND
GND
VIN
I2C
통신
ADC0
ADC1
ADC2
ADC3
SDA
SCL
아날로그
입력부
SCL
SDA
AREF
GND
Digital 13
Digital 12
Digital 11
Digital 10
Digital 9
Digital 8
Digital 7
Digital 6
Digital 5
Digital 4
Digital 3
Digital 2
Digital 1
Digital 0
I2C
통신
SCK
MISO
MOSI
SS
SPI
통신
~ PWM
펄스
출력부
인터
럽트
INT1
INT0
TXD
RXD
디지털
입출력부
시리얼
통신
ATMEGA 328P
ICSP
RESET
SCLK
MISO
GVD
5V
MOSI
ATMEGA 16U2
ICSP
GVD
MOSI
5V
RESET
SCLK
MISO

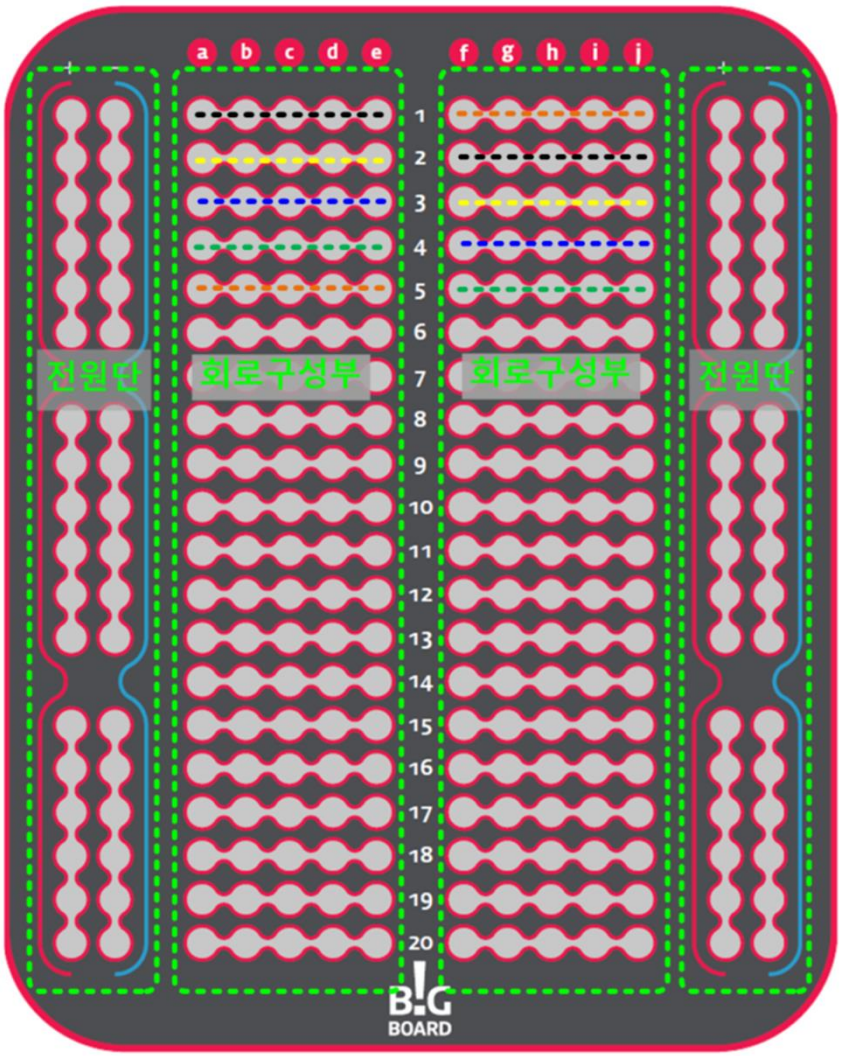
5) 빅 - 브레드 보드 설명





- 빅-브레드보드 = 일반 브레드보드 x 10배
- 물리적 크기만 10배 하였으므로, 동작이나 사용 모두 똑같습니다.
- 또한, 자석을 활용한 부품과 점프선이 연결될 수 있는 구조로 되어 있습니다.

6) 빅-브레드 보드 설명

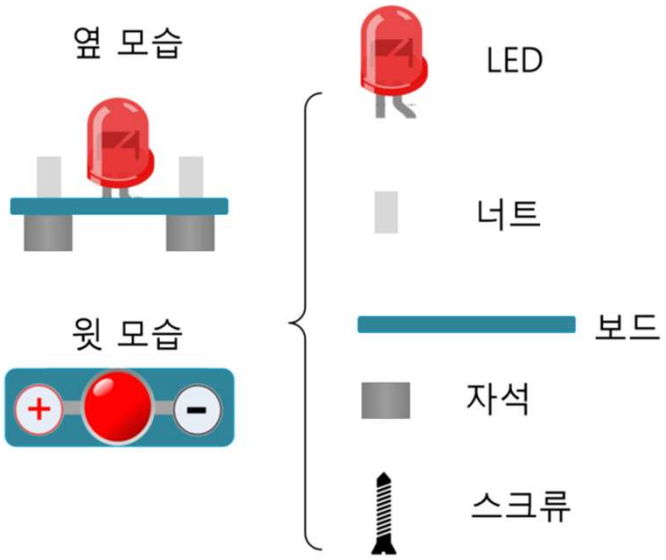


빅-브레드보드는 전원단과 회로구성으로 구별됩니다. 전원단은 5V, 3.3V, GND 에 해당하는 연결을 합니다. 또한, 세로 홀들은 모두 연결되어 있습니다.

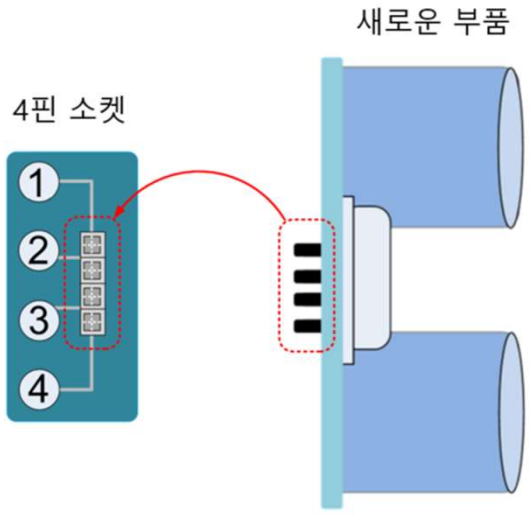
회로구성부는 자유롭게 회로 구성이 가능하며, 왼쪽 5개(a~b~c~d~e)와 오른쪽 5개(f~g~h~i~j)는 각각 연결되어 있습니다.

7) 빅-부품 구성

부품통에는 다양한 종류의 부품들이 있습니다.
모든 부품은 전기가 통하는 자석과 스크류로 구성되어 있습니다.

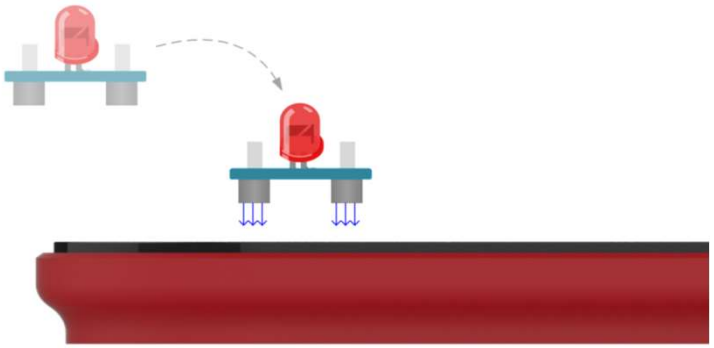


새로운 부품을 사용하고자 할 경우에는 아래와 같이 부품의 핀 숫자에 맞는 소켓을 이용해서, 꽂으시면 빅보드에서도 사용할 수 있습니다.



8) 빅-부품 사용 방법

빅-부품은 손으로 너트부분을 잡아서, 아래의 그림과 같이 빅-브레드 보드 위에 살짝 다가가면 자석의 힘에 의해서, 부드럽게 붙습니다. 분해, 역시 가볍게 손으로 분리가 가능합니다.



취급주의

인공심장이나 체내형 심실보조장치, 체내형 의료기를 장착하고 계신 분은 절대 사용하지 마시길 바랍니다.

자석을 삼킬 경우 생명을 위협하는 사고로 이어질 수 있습니다. 만일, 삼켰을 경우 즉시 의사의 진단을 받고 지시를 따르세요.

영유아(6세이하)는 반드시 보호자의 감독이 필요하며, 손이 닿지 않는 곳에 보관하십시오

외부의 강한 충격에 의해서, 깨지거나 도금이 벗겨질 수 있습니다.

자석끼리나 금속 물체에 서로 흡착하는 힘이 강합니다. 안이하게 생각하거나 방심하면 다칠 위험이 있습니다.

휴대전화, 아날로그 시계, 자기 카드, 자기테이프 등에 가까이 하지 마십시오 내용이 손상될 수 있습니다.

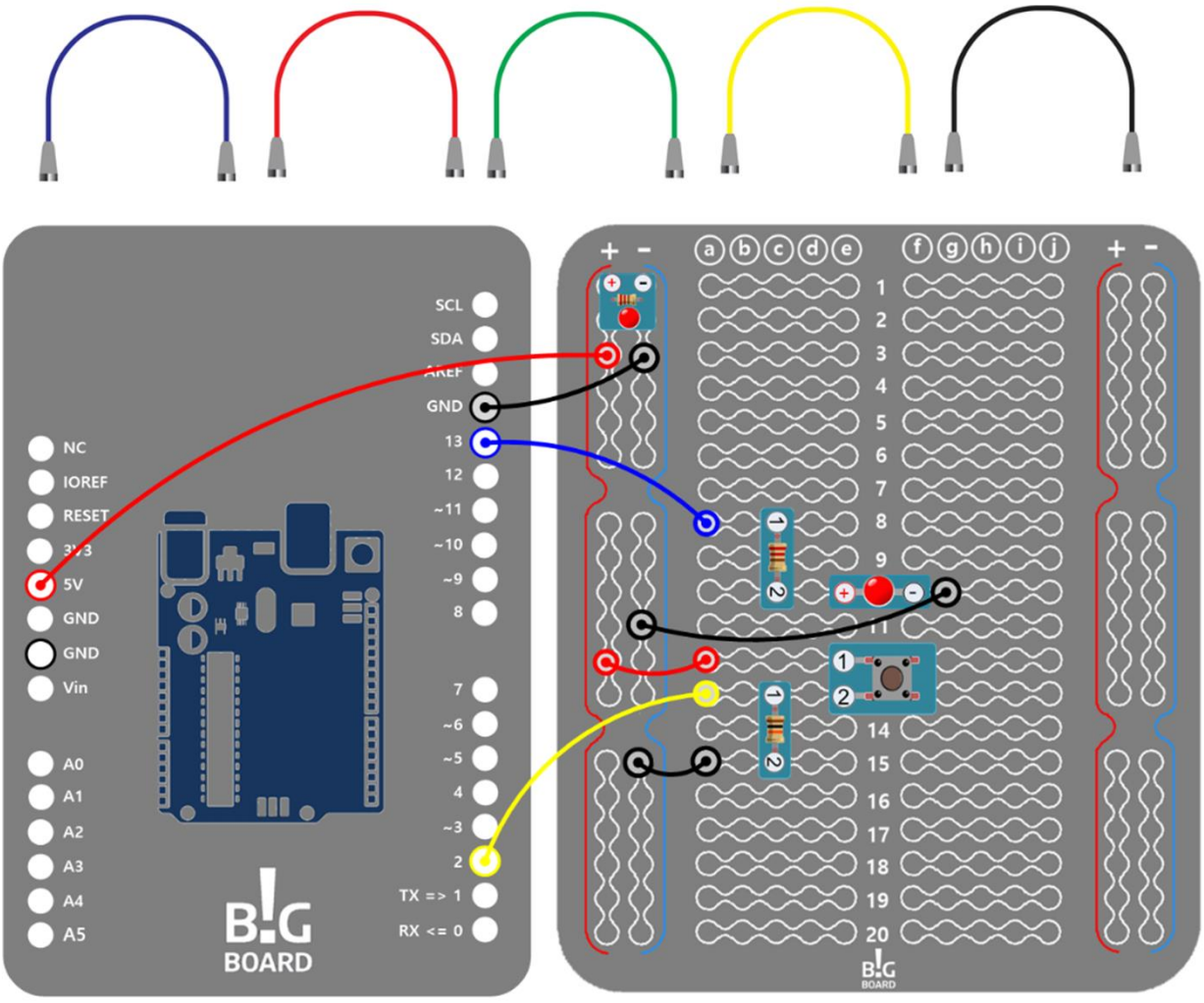
9) 점프선 설명

점프선은 아래의 그림과 같은 구조로 되어 있습니다.
몰드 금형을 이용하여, 손으로 잡기 편하며, 외부 충격에 강하게 만들어져, 오랜 동안 사용하여도, 고장이 발생하지 않습니다.
전기적 전원이나 신호, 전기가 통하는 네오디늄 자석과 연결되어 있습니다.

10) 회로 연결 방법

점프선은 아래의 그림과 같이 빅-제어보드에서 빅-브레드보드로 전원과 GND를 연결하거나, 빅-브레드보드 내에서 부품끼리의 신호를 연결 하는데 사용됩니다.

전원은 빨강색, GND는 검정색, 신호선은 파란색과 노란색을 사용하면, 복잡한 회로에서도 명확하게 구분이 가능합니다.
특히, 녹색은 길이가 길어서, 먼 곳에 연결할 때 편리합니다.

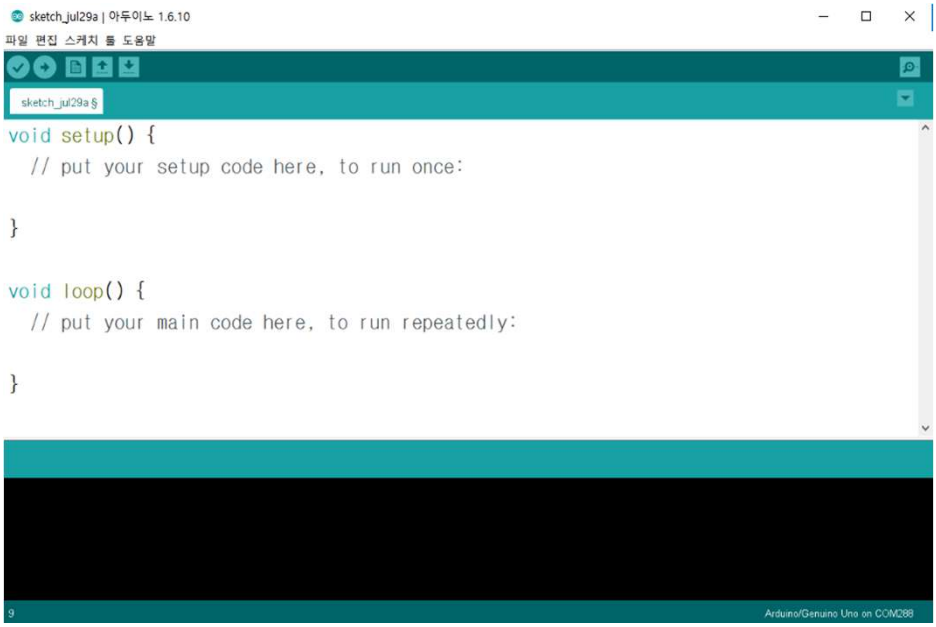


06 스케치(Sketch) 란?

1) 스케치 소개

- ✓ 아두이노(https://www.arduino.cc/)에서 제공하는 통합 개발 환경 (IDE)을 **스케치**라고 합니다.
- ✓ 누구나 무료로 소프트웨어 교육을 받을 수 있게 개발된 비 영리 소프트웨어 교육 플랫폼입니다.
- ✓ 자바 (Java) 와 C를 기반으로 개발되는 크로스 플랫폼 응용 소프트웨어 입니다.
- ✓ 타이핑(글자) 입력 형태로 코딩합니다.

2) 스케치 설명



[그림? 컴퓨터에 설치된 스케치 화면]

- 타이핑 방식의 코딩 방식으로 구문 강조, 괄호 찾기, 자동 들여쓰기 기능이 포함된 에디터와 한 번의 클릭으로 컴파일과 업로드가 가능한 컴파일러 기능을 포함하고 있습니다.
- 아두이노 동작을 위해서 C++ 언어 기반을 사용합니다.
- 장점으로 다양한 전자 부품들을 활용한 복잡한 프로젝트까지 수행이 가능합니다.
- 컴파일러는 avr-gcc을 사용합니다. 따라서 avr-gcc가 제공하는 많은 C언어의 표준라이브러리를 함수를 사용할 수 있습니다.
- 아두이노에서 제공하는 여러 하드웨어 보드와 호환 보드들을 연결해서 사용할 수 있습니다.

06 스케치(Sketch) 란?

3) 스케치 특징

스케치는 코드를 만들 수 있고, 만든 코드의 오류를 찾아주고, 완성된 코드를 컴퓨터가 읽을 수 있도록 기계어로 변환(컴파일)해 서 하드웨어 보드에 업로드를 할 수 있는 통합개발 프로그램을 말합니다.

스케치는 2가지 큰 특징을 갖고 있습니다.
1) 타이핑(글자) 입력 방식의 프로그래밍
스케치내의 에디터 장에 직접 사용자가 간단한 함수나 라이브러리를 글자형태로 입력해야 합니다. 처음은 어려울 수 있으나, 복잡한 프로그래밍이 가능하여, 다양한 프로젝트 수행이 가능한 장점이 있습니다.

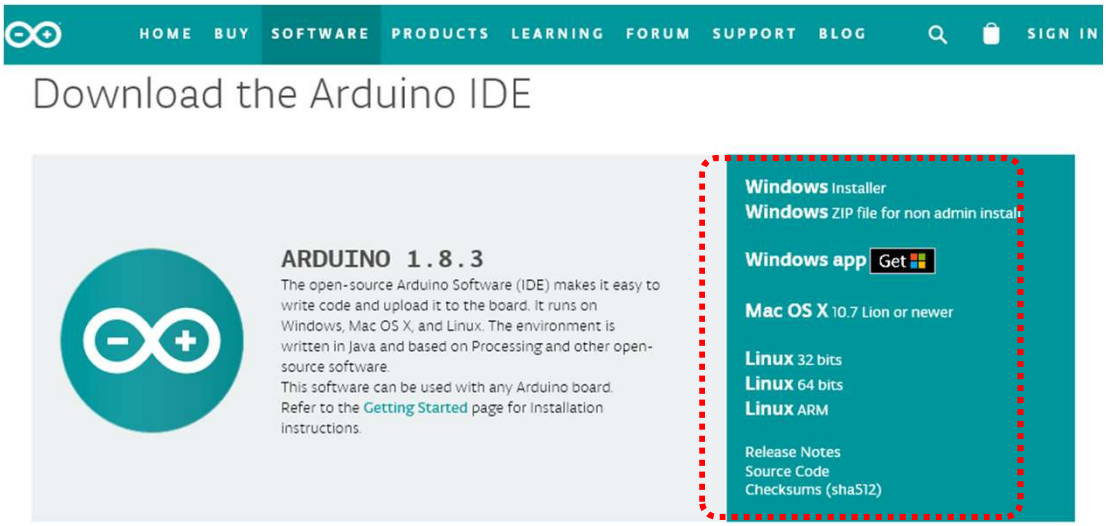
2) 하드웨어 연결 기능
외부에 하드웨어(전자 보드)를 연결하여, 통신(데이터 송수신)으로 다양한 프로그램을 작동시킬 수 있으며, 보다 흥미롭고 재미있게 프로젝트 수행이 가능합니다.
참고로, 한번 업로드 된 이후에는 내부에 독자적인 마이크로 프로세스가 내장되어 있어서, 컴퓨터 연결 없이도 동작이 가능 합니다.

06 스케치(Sketch) 란?

4) 스케치- 아두이노 IDE (통합 개발환경) - 프로그램 설치하기

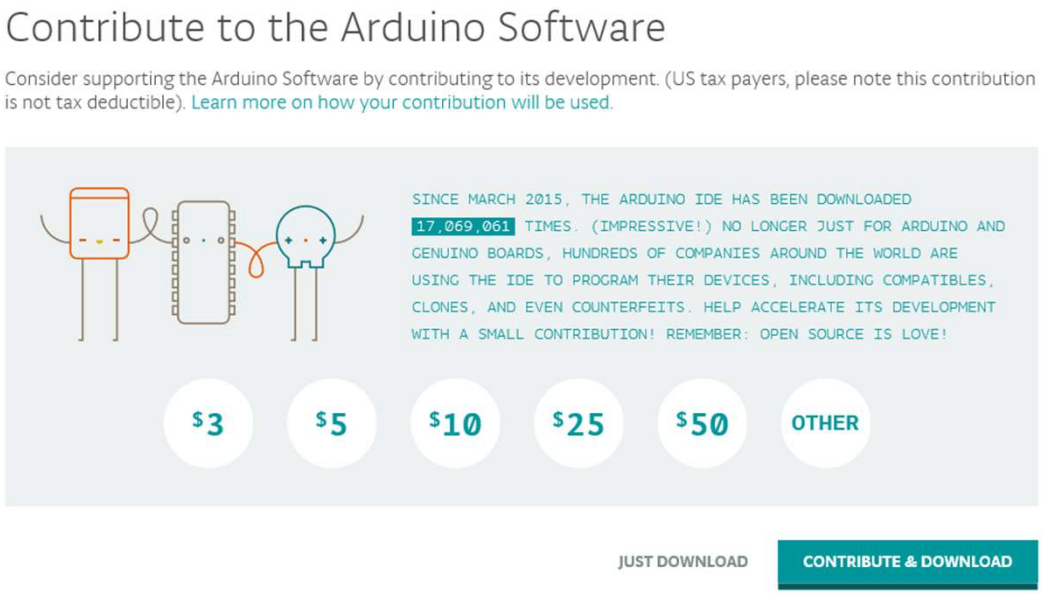
1) 먼저, 아두이노 사이트에 접속합니다.
https://www.arduino.cc/

2) 메뉴 창에서 “SOFTWARE” (다운로드) 창을 선택합니다.
https://www.arduino.cc/en/Main/Software



3) 우측 창에서 사용자의 운영시스템에 맞는 프로그램을 선택해서 설치를 합니다.
여기서는 **Windows** 운영체계를 선택해서 설치해 보겠습니다.

4) 윈도우용 설치 파일을 다운로드하기 위해 "Windows installer"를 클릭하면 아래와 같은 화면이 나옵니다.



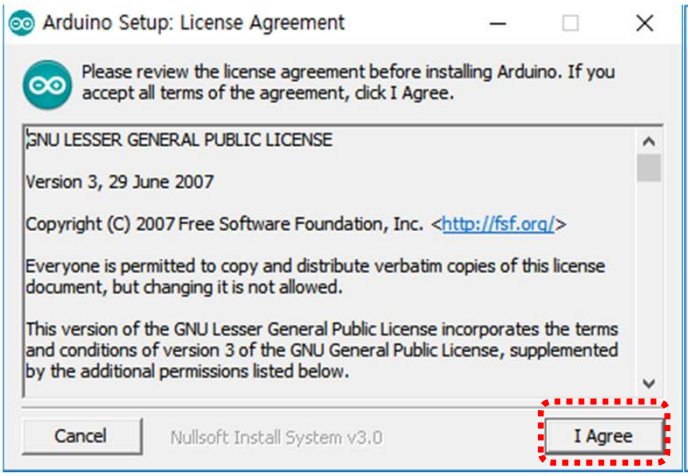
"CONTRIBUTE&DOWNLOAD"를 클릭하면 원하는 금액을 기부(?)하고 다운로드 할 수 있습니다.
기부를 원하지 않으려면, "JUST DOWNLOAD"를 클릭하면 스케치를 설치할 수 있습니다.

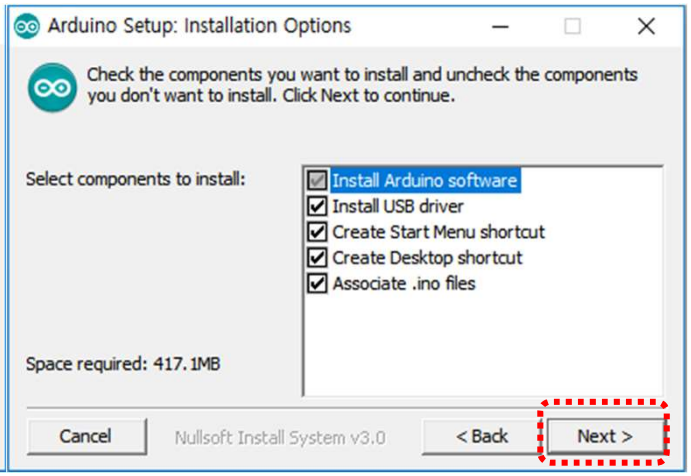
5) 그러면, 아래의 창처럼 설치 프로그램의 실행 창이 표시됩니다.

여기서, 실행 아이콘을 선택합니다.
 그러면, 실행 프로그램을 사용자 컴퓨터로 다운로드 합니다.

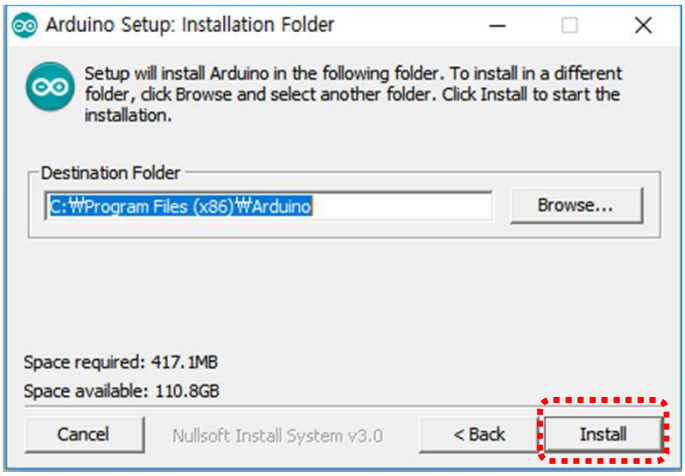
06 스케치(Sketch) 란?

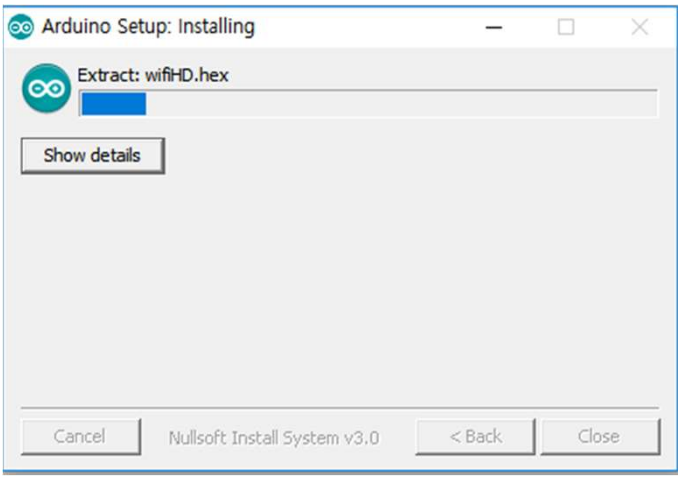
6) 다운로드가 완료되면, 사용자 컴퓨터에 설치 과정을 진행합니다.





"I Agree" 선택 => "Next" 선택을 합니다.

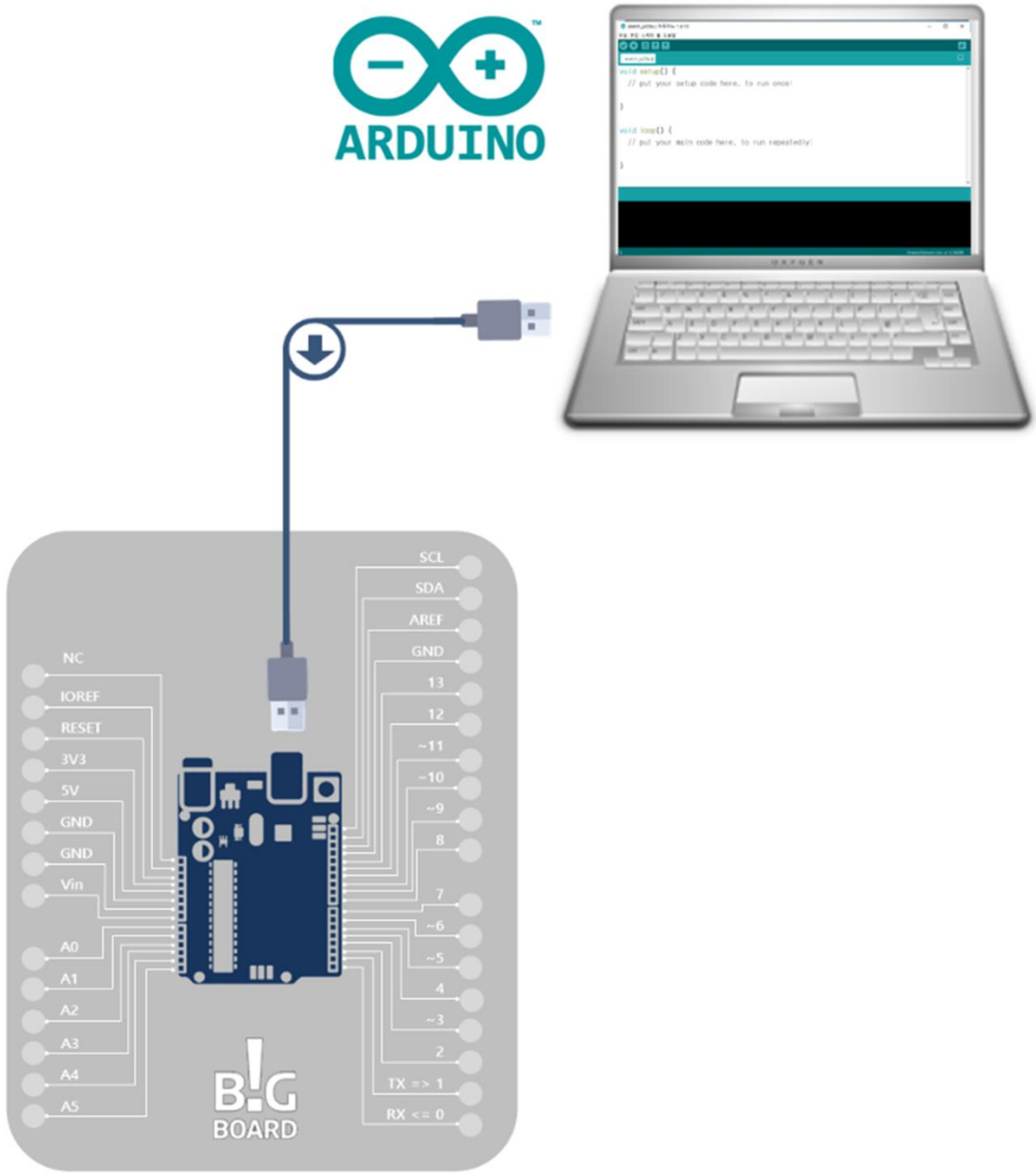




"Install" 선택 => 설치를 시작합니다.
"Completed" 문구나 나타나면, 설치가 완료된 것입니다.
"Close"를 선택하세요!

07 스케치와 빅보드 연결하기

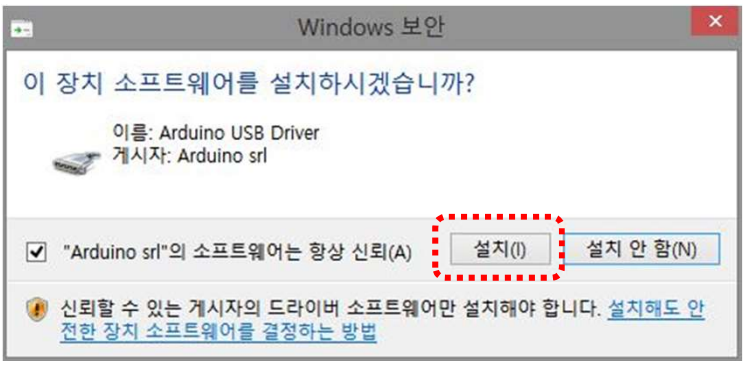
1) 빅보드와 컴퓨터 간에 USB 케이블을 연결합니다.



2) 다음으로, 자동으로 USB Driver 설치를 진행합니다.

USB 케이블을 연결 시, 아래의 화면처럼 드라이블 설치를 진행합니다.

"설치"를 선택합니다.

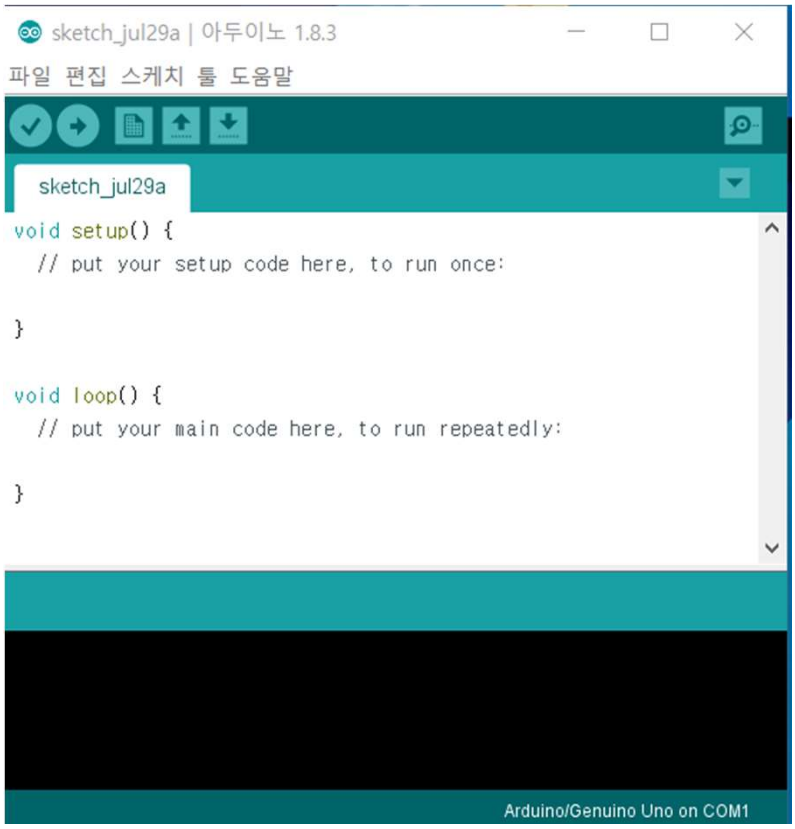


"설치"를 선택합니다.

3) 스케치 실행을 합니다.

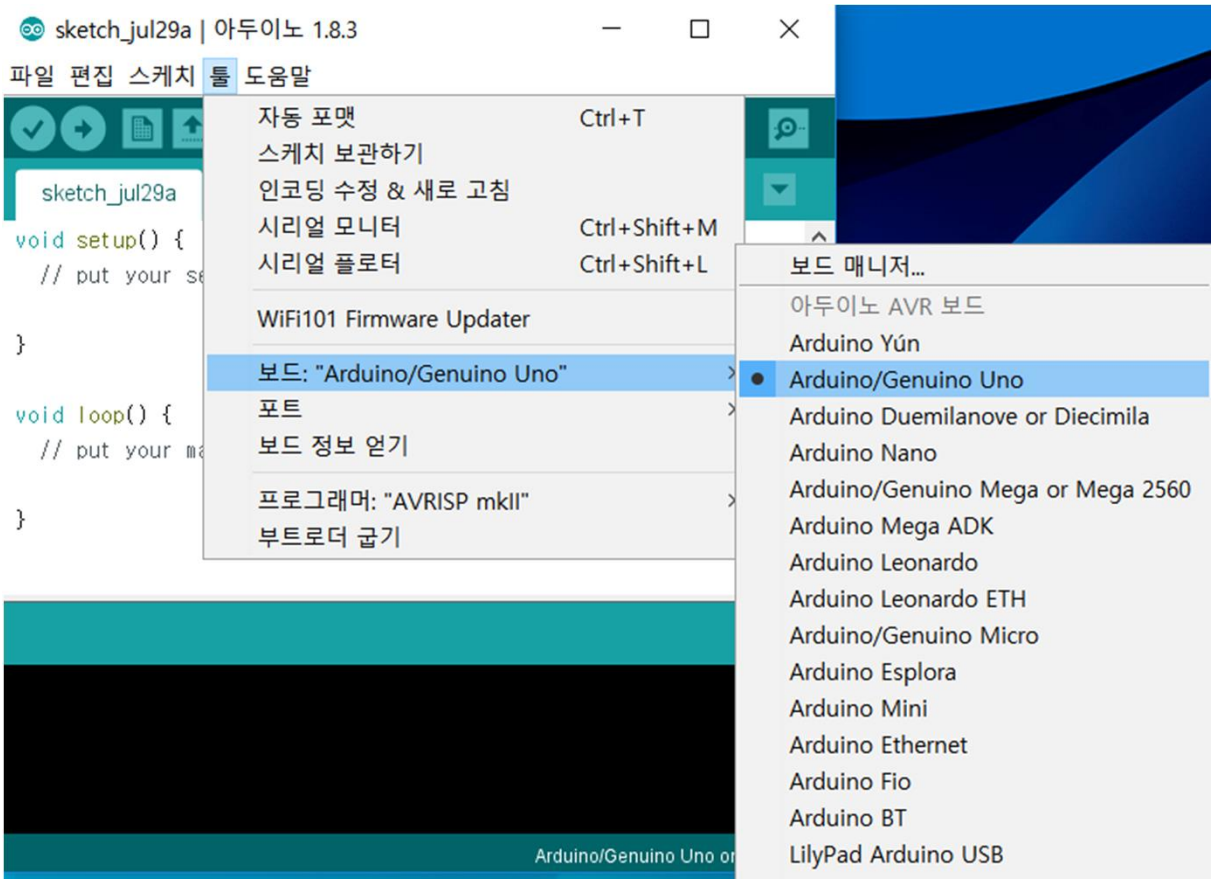
바탕화면에서 아두이노 스케치 아이콘을 실행합니다.

아래의 사진처럼 스케치가 실행됩니다.



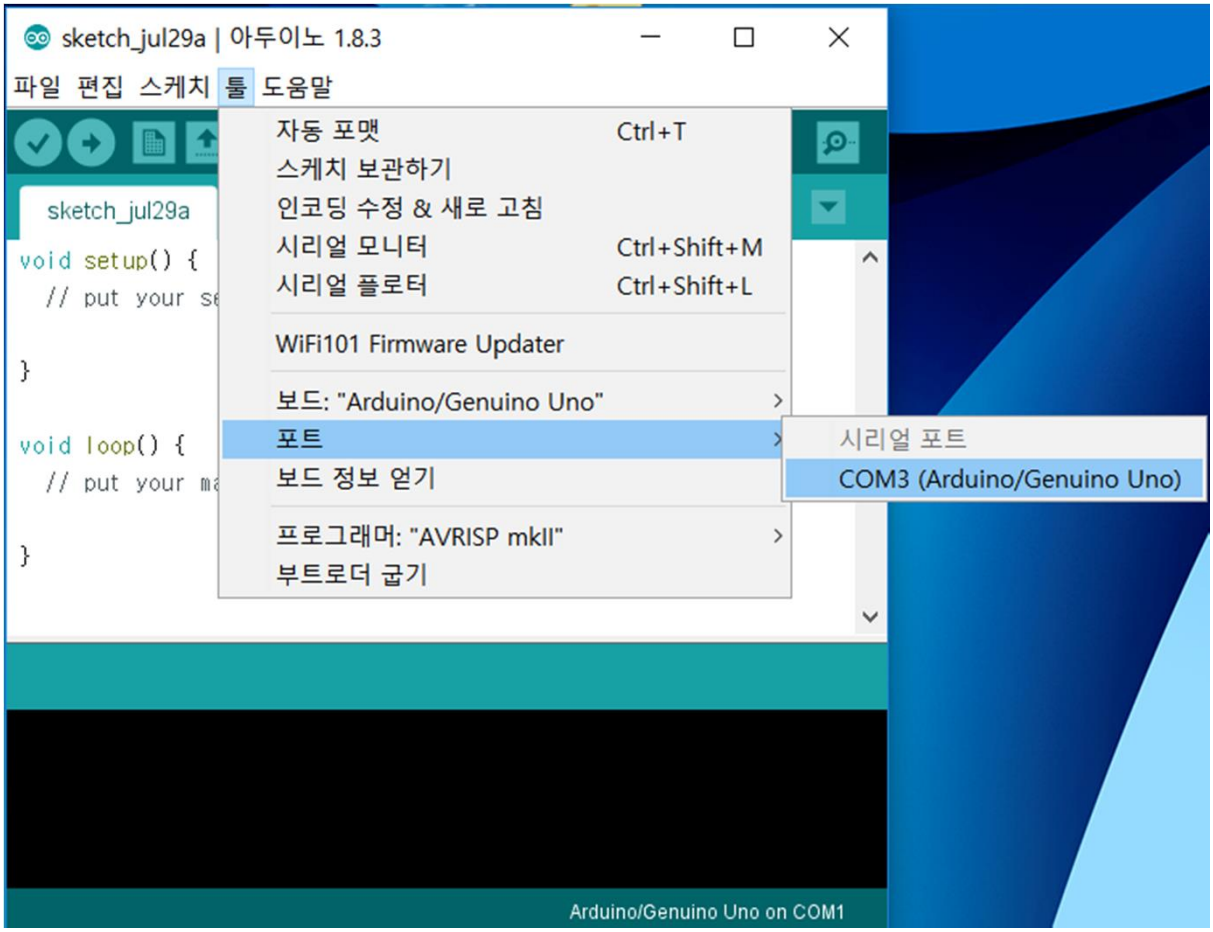
4) 하드웨어 보드 선택

아래의 화면처럼, 툴 => 보드 => Arduino/Genuino Uno 를 선택합니다.
빅보드는 Arduino Uno 보드를 적용했습니다.



5) USB 통신 포트 설정

아래의 화면처럼, 툴 => 포트 => COM(Arduino/Genuino Uno) 를 선택합니다.
설명자의 컴퓨터는 COM3입니다.

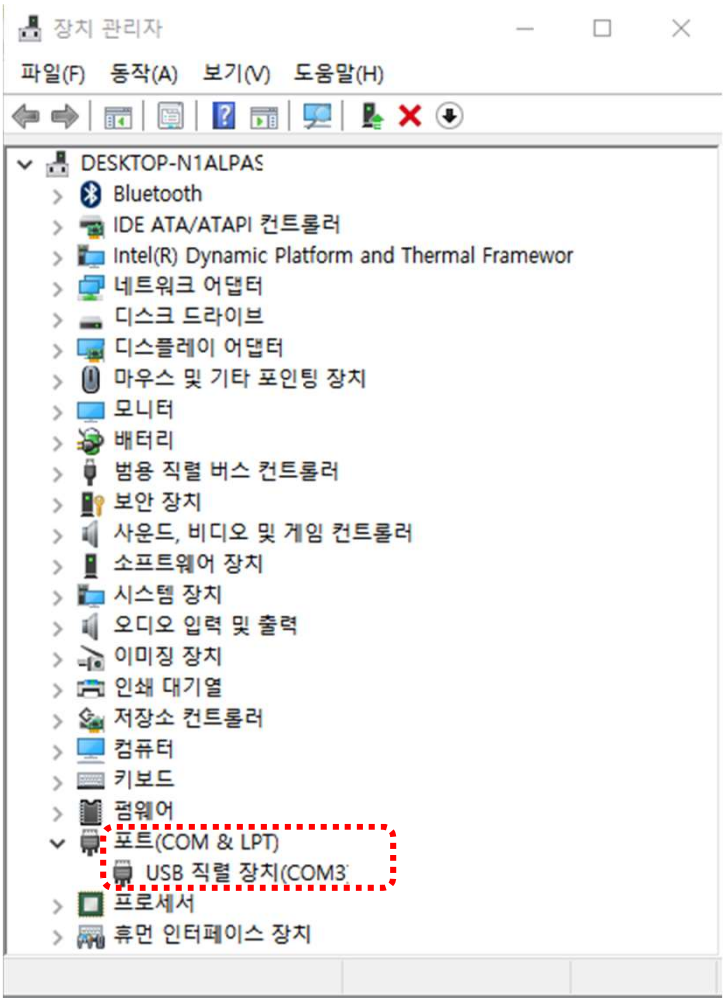


* 주의사항

사용자의 컴퓨터에 따라서, COM 번호가 상이할 수 있습니다.

Tip) USB 통신 포트 번호 확인 방법

컴퓨터의 설정 => 장치관리자 를 실행합니다.
그리고, 포트((COM & LPT)에 설치된 COM 번호를 확인하여,
선택해야 합니다.



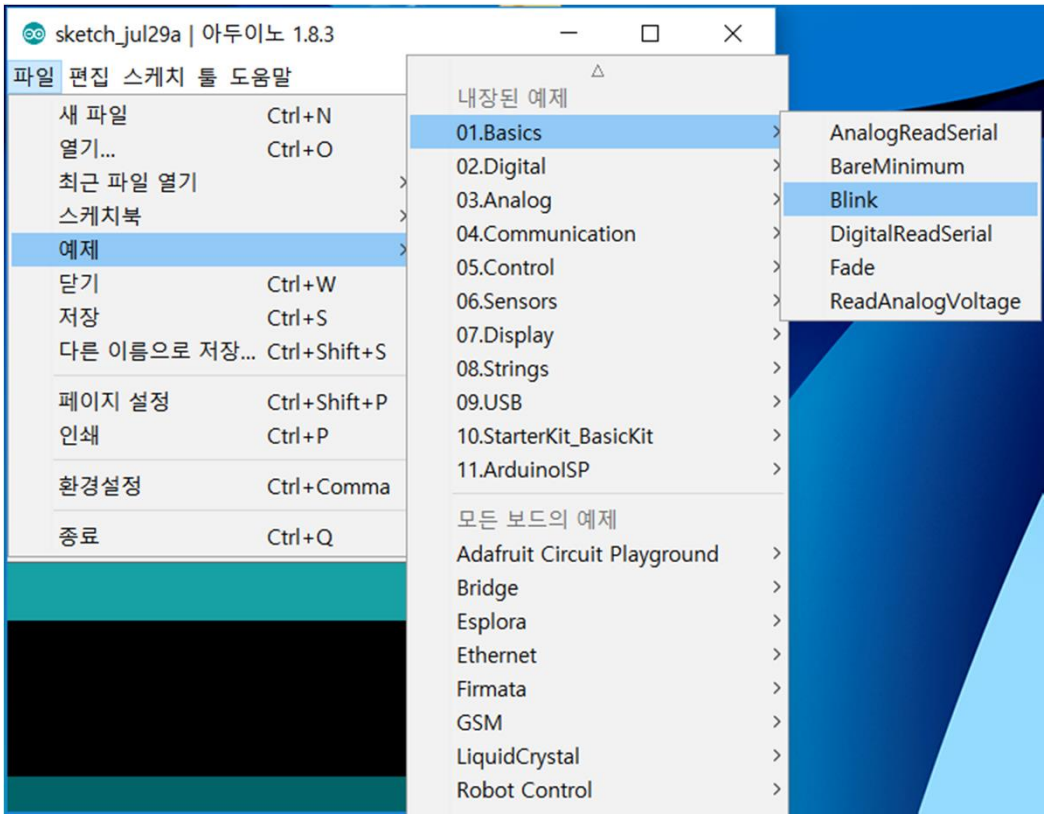
* 주의사항

사용자의 컴퓨터에 따라서, COM 번호가 상이할 수 있습니다.

정상동작 확인하기 - 간단 예제

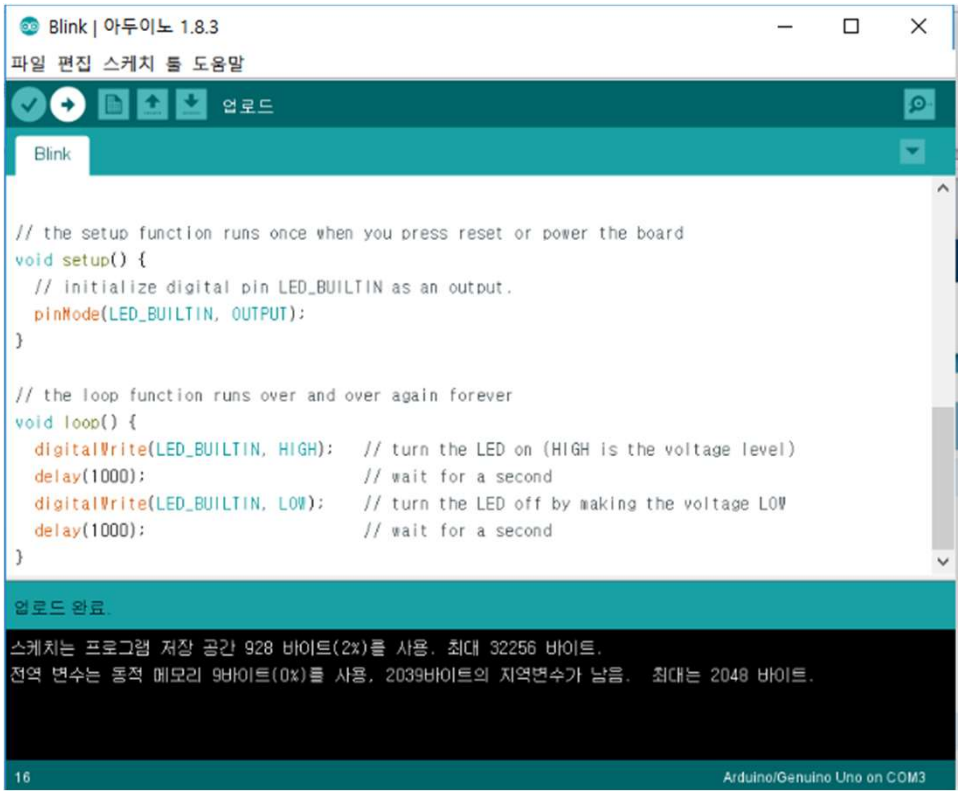
(도전) 아두이노 보드(빅보드) 내의 13핀에 연결된 LED(램프)가 1초 동안 켜지고 꺼지기를 반복하기입니다.

1) 스케치 화면에서 미리 설치된 예제를 불러옵니다.
2) 파일 => 예제 => 01.Basics => Blink 를 선택합니다.



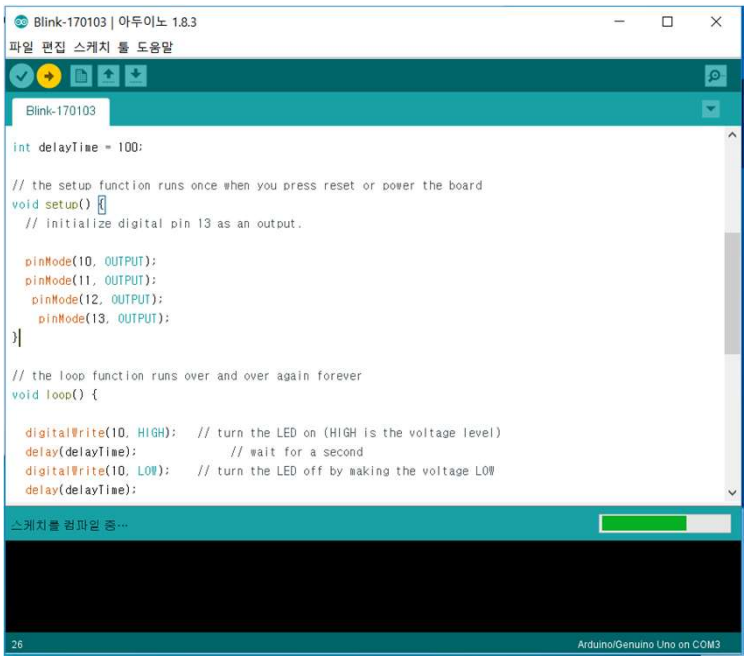
정상동작 확인하기 - 간단 예제

3) 불러온 예제 파일이 아래의 사진처럼 나타납니다.

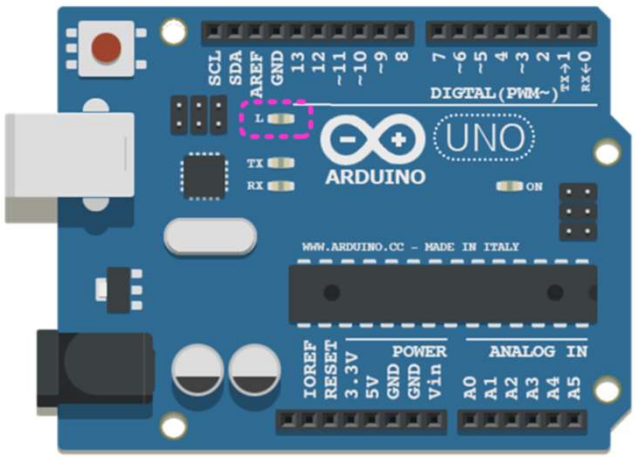


4) 예제 파일을 아두이노우노 (빅보드)로 프로그램을 업로드 합니다.
아래의 사진처럼 화살표를 선택합니다.

5) 기계어로 변환(컴파일) 후 아두이노우노(빅보드)로 업로드 됩니다.



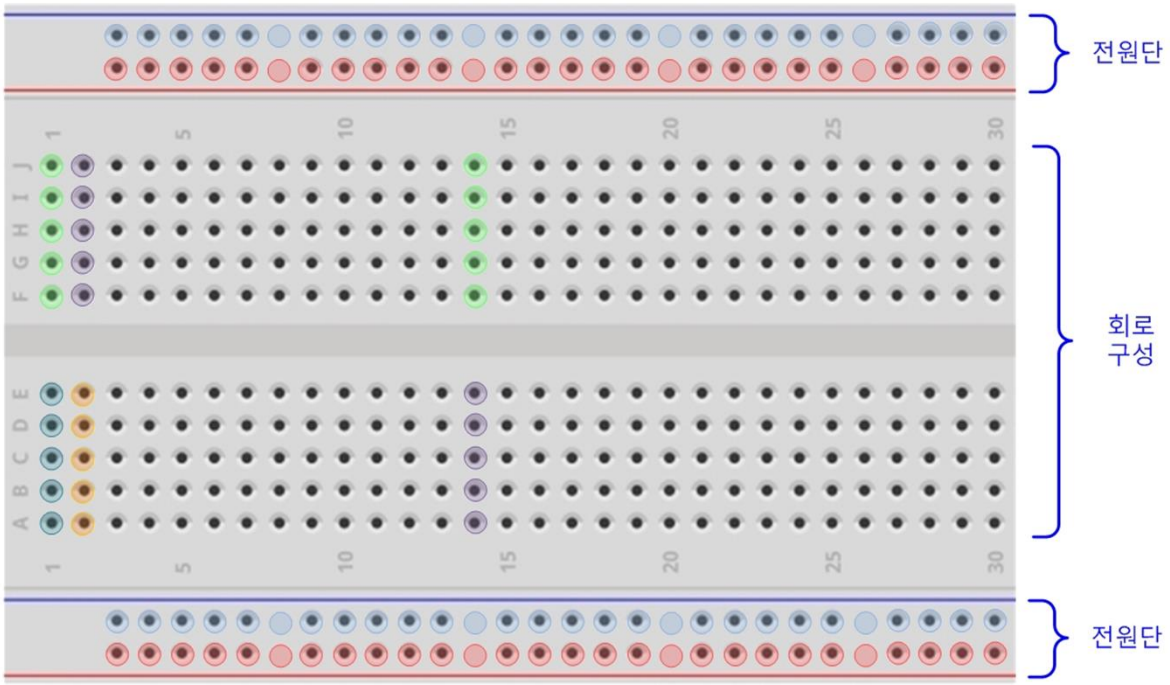
6) 아두이노우노 보드에서 정상 동작을 확인합니다. 아래의 그림처럼 L 자 옆의 LED가 1초 간격으로 깜빡이면, 정상 동작하는 것입니다.



이제부터 본격적으로 코딩 공부를 시작해 볼까요!

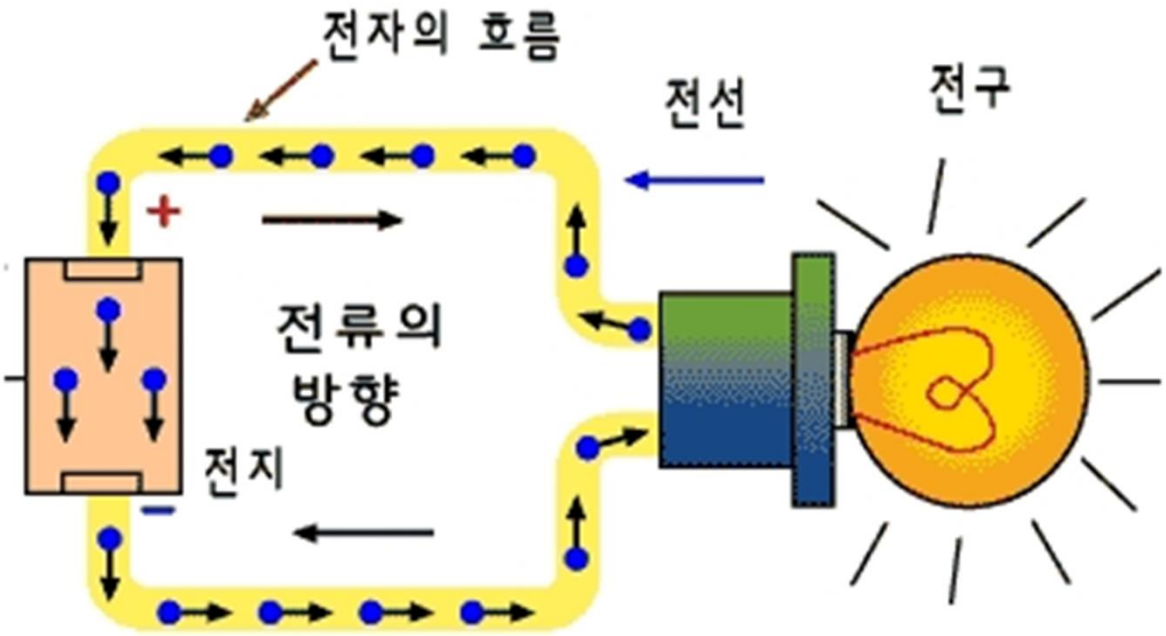
08 전자회로 기초 설명

1) 브레드 보드 설명



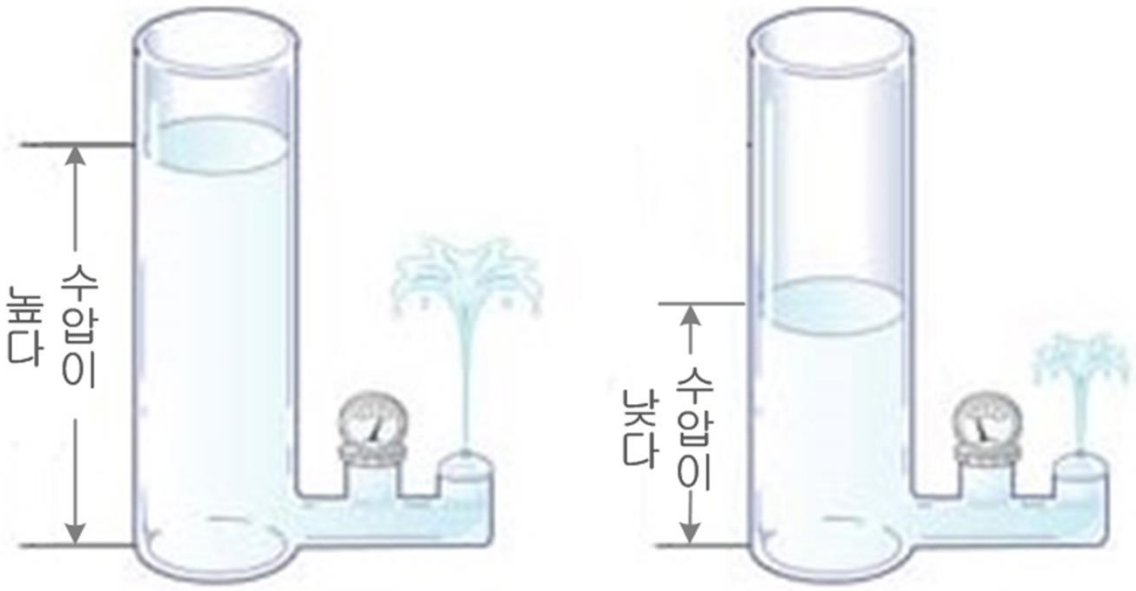
- ✓ 납땜 없이 자유롭게 회로 구성이 가능한 만능 보드입니다.
- ✓ 윗면과 아래면은 전원 연결용으로 가로로 연결되어 있습니다.
- ✓ 중간 면은 회로구성용으로 세로로 연결되어 있습니다.

2) 기본 전자회로 설명



- ✓ 전기는 전자의 흐름입니다. 전기는 (+) => (-)로 흐릅니다.
- ✓ 전기를 가지고 있는 것은 전지이며, 이를 흐르게 하는 것은 전선입니다.
- ✓ 전기는 조명을 켜거나 소리를 내거나 모터를 움직이게 합니다.

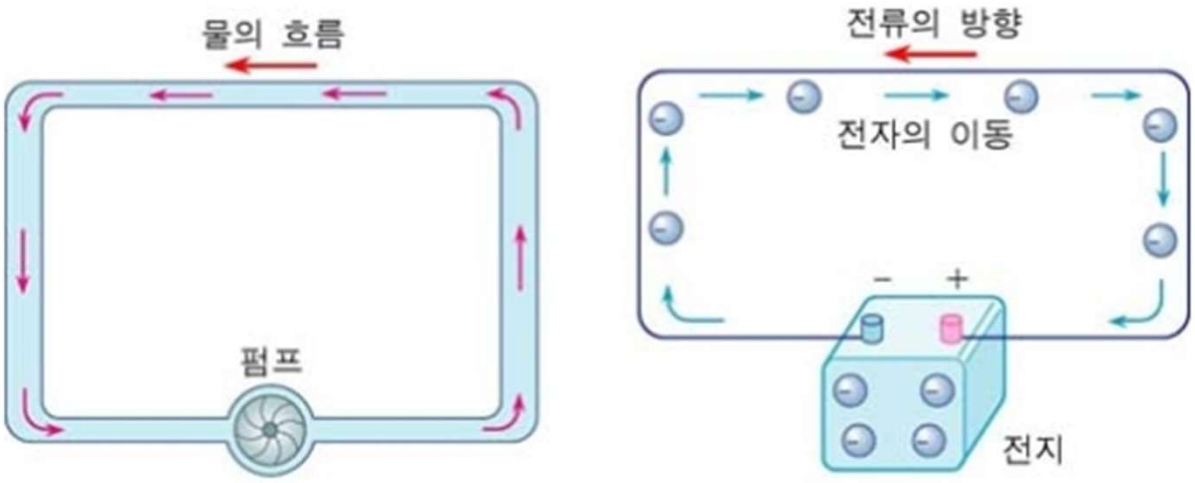
3) 전압 설명



[수압과 물줄기의 세기]

- ✓ 전기적인 위치에너지(전위, 높이)의 차이입니다.
- ✓ 물에 비유하면, 수위(물의 높이), 즉 수압으로 생각할 수 있습니다.
- ✓ 전압은 일반 건전지나 전압 생성장치(충전기, 파워공급장치) 를 통해서 얻어집니다

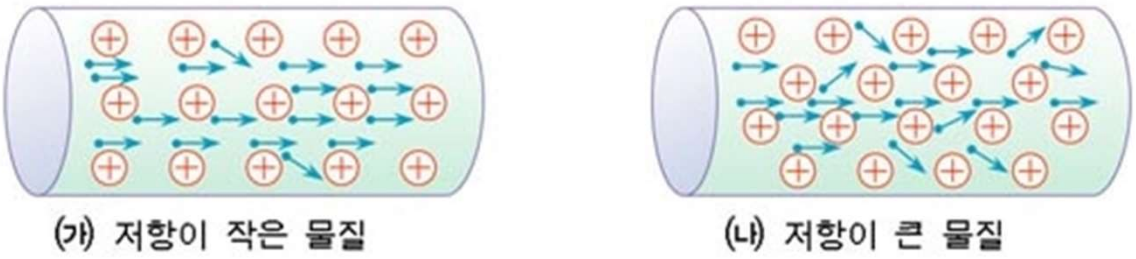
4) 전류 설명



[물의 흐름과 도선 속의 전자의 이동 및 전류의 방향 비교]

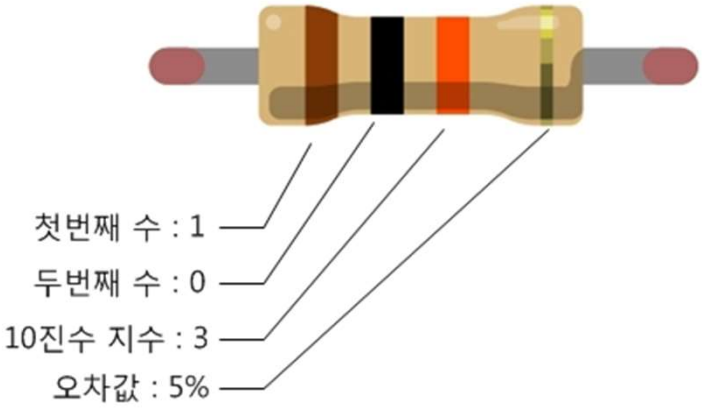
- ✓ 전자가 가지고 있는 전하의 흐름으로, 전선과 같은 도체에서 일어납니다.
- ✓ 전류의 세기는 전압의 세기와 저항에 의해서 영향을 받습니다.

5) 저항 설명



[금속 내에서 전자의 운동]

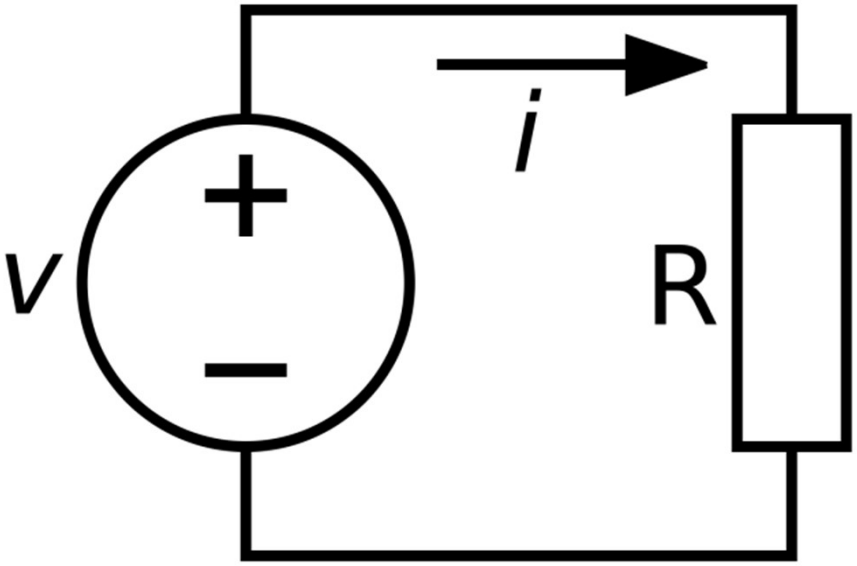
- ✓ 전류의 흐름을 방해하는 정도를 나타내는 수치를 저항이라고 합니다.
- ✓ 저항 값이 낮을 수록 전류의 흐름을 적게 방해 합니다.



10 x 10^3 = 10KΩ 편차 5%

첫번째, 두번째, 세번째 색띠의 숫자										네 번째 색띠		
흑	갈	적	황	노	녹	청	보	회	백	금	은	없음
0	1	2	3	4	5	6	7	8	9	5%	10%	20%

6) 옴의 법칙



✓ 도체(저항)의 두 지점 사이에 나타나는 전위차에 의해 흐르는 전류가 일정한 법칙에 따르는 것을 말합니다.

✓ 전압(V) = 전류(I) X 저항(R)

1) 전압의 단위는 볼트(V, volt)

2) 전류의 단위는 암페어(A, ampere)

3) 저항의 단위는 옴(Ω, ohm)

✓ 도체 양단에 전위차가 존재할 때, 도체의 저항의 크기와 전류는 반비례 관계를 갖습니다.

✓ 저항은 전류를 제한하는 요소로 작용합니다.

09 코딩의 기초 설명

1) 용어 설명

➢ 아두이노 = 하드웨어 보드 + 통합개발 환경(IDE) + 오픈소스 그룹

✓ 하드웨어보드 = (디지털 + 아날로그) x (입력 + 출력)

✓ 통합개발 환경 = 스케치작업(코딩) + 컴파일 + 업로드

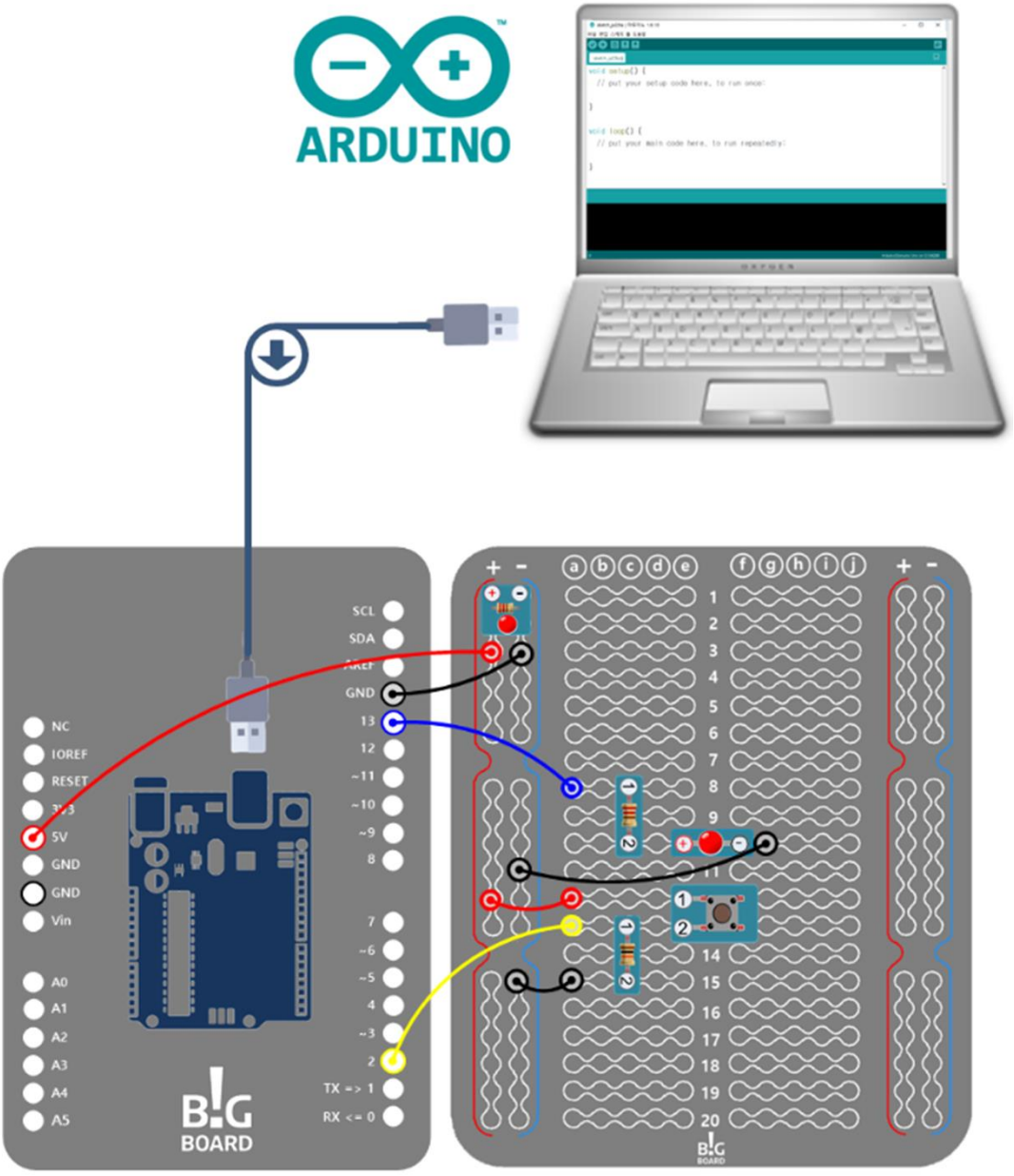
- 소스코드 = 스케치 (아두이노는 소스 프로그램을 ' 스케치'라 합니다.)
- 컴파일 = 스케치를 마이크로 컨트롤러가 알아듣게 바꾸는 작업
- 업로드 = 컴파일 된 것을 USB 케이블로 아두이노 보드에 옮기는 작업

✓ 오픈소스 그룹 = 블로그 + 카페 + 커뮤니티 + 행사를 통한 정보 교류

✓ 소스 저장하는 곳 = 스케치 북

2) 전체 진행 순서

3) 코딩 교육 전체 시스템 구성



코딩 교육을 위한 전체 구성입니다.
컴퓨터(스케치 설치된)와 빅보드는 USB 케이블을 연결하여, 전원 공급과 프로그램을 할 수 있습니다.

4-1) 스케치 프로그램의 기초

아두이노 언어는 C언어와 유사하지만, 더 간단합니다.
아두이노 라이브러리는 C언어로 작성되고, 통합개발환경은 Java로 만들어 졌습니다.

통합개발환경에서 스케치 코딩 영역(적색 테두리 안)에서
코딩(프로그램)을 합니다.

4-2) 스케치 프로그램의 기초

코딩은 3가지 요소 = 변수 + 구조 + 함수

- 문법
  - ✓ **// 두줄 슬래시** 뒤는 한 라인의 설명문, 코드 실행 안됨
  - ✓ **/*** 사이는 여러 라인의 설명문, 코드 실행 안됨 ***/**
  - ✓ **{ 중괄호** 사이는 코드 블록 **}**
  - ✓ **;** 한 줄의 코드 끝에는 **세미콜론**을 붙임

이름	열기	닫기
소괄호, 괄호, 손톱괄호, 손톱묶음	(	)
겹소괄호, 이중소괄호	⟪	⟫
중괄호	{	}
대괄호, 꺾쇠	[	]

- 변수 ; 상수의 차이점
  - ✓ int : 2Byte (16bit)정수 (-32,768 ~ +32,768)
  - ✓ long : 4Byte(32bit) 정수 (-2,147483,648 ~ +2,147,483,647)
  - ✓ boolean : 1bit 참(True) or 거짓(False)
  - ✓ float : 4Byte 소수 (-3.4E+38 ~ +3.4E38)
  - ✓ char : 1바이트, ASCII 코드 'A'=65, 문자열은 "ABC"로 표현

- 구조 = void setup (소괄호) {중괄호 } + void loop ( ) { }
  - ✓ void setup () { // 이 곳에 있는 코드는 한번만 실행됨 }
  - ✓ void loop () { // 이 곳에 있는 코드는 반복해서 실행됨 }
  - ✓ void 는 리턴해 주는 값이 없음. 그 함수로만 종료

4-3) 스케치 프로그램의 기초

➢ 함수

✓ 산술 연산자 (수학적 계산)

1) = : 대입
2) + : 덧셈
3) - : 뺄셈
4) * : 곱셈
5) / : 나눗셈
6) % : 나누고 남은 값 (예 : 7%4 =3)

✓ 비교 연산자 (논리적 비교)

1) == : 같은가?
2) != : 같지 않은가?
3) < : 작은가?
4) > : 큰가?

✓ 제어구조 ; 프로그램은 코드를 차례대로 실행하나, 제어문에 의해서 순서를 바꿉니다.

```
1)If (조건) { }  // 조건은 반드시 참, 진리값이 1 이어야만 함.
                 // 양수일 때는 0외는 모두 참.
  else if (조건) { }
  else { }
```

2) `for (int i=0; i<10; i++){ }` // i=0에서 9까지, { } 을 10회 반복 실행합니다.

4-4) 스케치 프로그램의 기초

➢ 디지털 함수

1) **pinMode(pin, mode)** : 해당 핀과 입/출력으로 정합니다.

2) **digitalWrite(pin, value)** : 디지털 출력 (해당 핀, 출력 값)을 정합니다. 출력 값은 High/Low 2가지입니다.

➢ 아날로그 함수

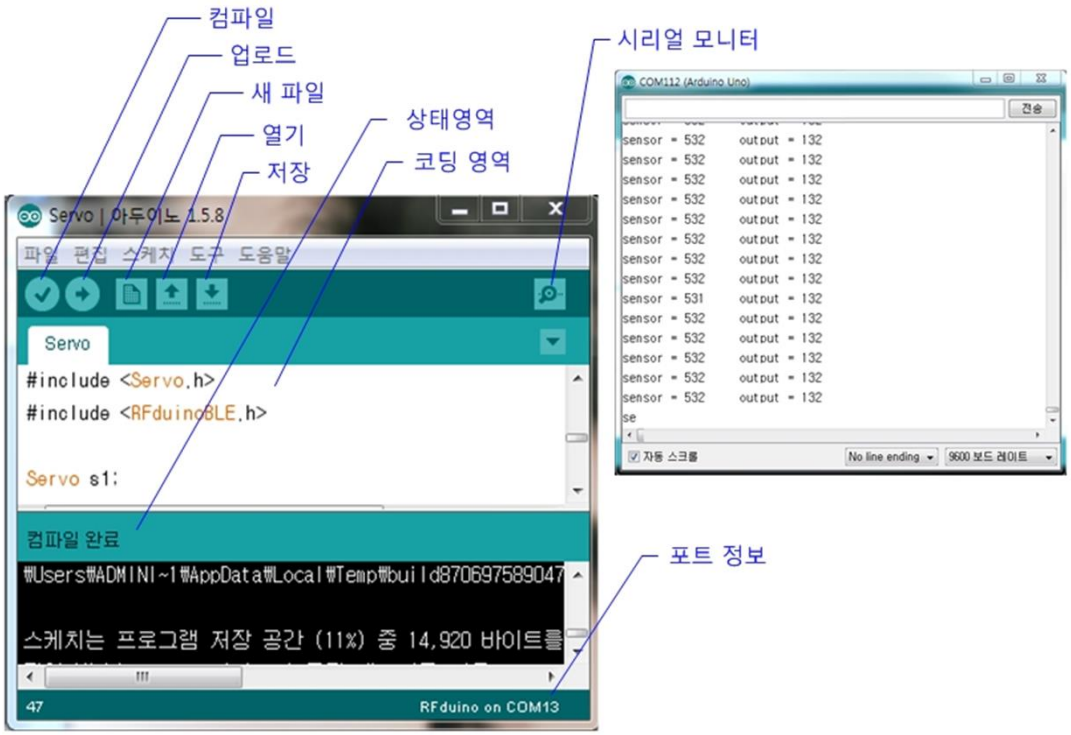
1) **pinMode(pin, mode)** : 해당 핀과 입/출력으로 정합니다.

2) **analogWrite(pin, value)** : 아날로그 출력 (해당 핀, 출력 값)을 정합니다. 출력값의 영역은 0 ~ 255 (8bit) 입니다.

3) **int A = analogRead(pin)** : 아날로그 입력 값을 해당 핀에서 2 Byte 정수로 읽습니다.

변수 A의 자료형은 정수형으로, analogRead (pin)의 값을 읽어서, 변수 A에 저장을 한다. analogRead 값의 영역은 0~1023(10bit) 입니다.

5) 스케치 메뉴 설명

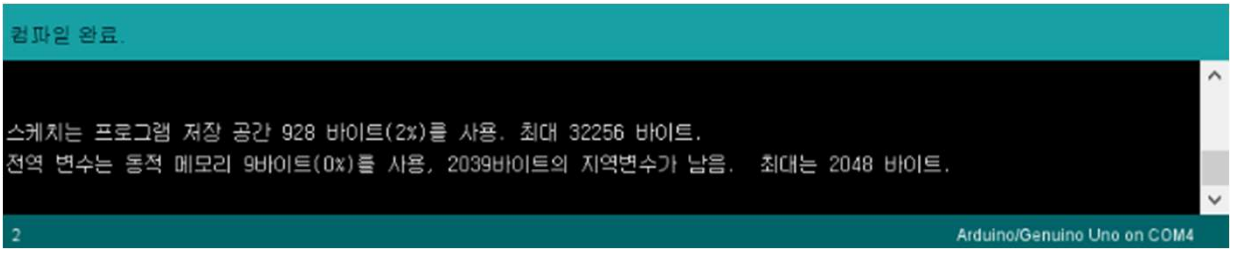


- ✓ 컴파일 : 코드의 오류를 확인
- ✓ 업로드 : 보드로 코드를 보드로 업로드 함
- ✓ 새 파일 : 새로운 스케치를 만듦
- ✓ 열기 : 스케치북에서 스케치를 가져옴
- ✓ 저장 : 스케치를 저장함
- ✓ 상태영역 : 상태 정보를 표시 영역
- ✓ 코딩영역 : 스케치 직접 작성 및 편집하는 영역
- ✓ 시리얼 모니터 : 시리얼 모니터를 열어봄
- ✓ 포트 정보 : PC USB 연결된 포트 정보

6) 컴파일 하기

✓ 화면 왼쪽 상단의 ✓버튼을 눌러 불러온 예제 프로그램 소스를 컴파일 합니다.

✓ 완료되면, 왼쪽 하단에 '컴파일 완료' 라고 나타납니다.
창에는 프로그램 저장 공간과 사용프로그램의 용량을
표시해 줍니다.



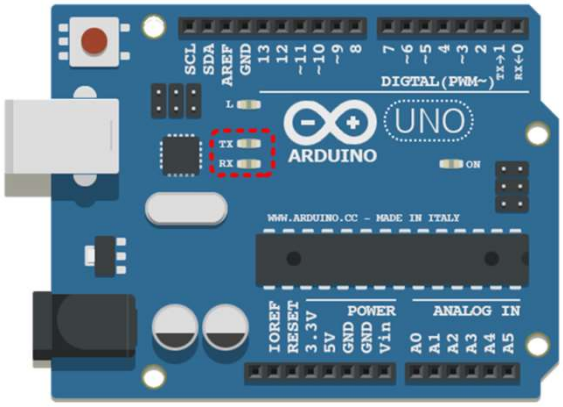
만약, 컴파일 에러 발생시에는 불러온 프로그램 소스가 제대로인지 아니면 소스 수정을 했을 경우, 수정된 구문의 문법이 정확한지를 확인하고 다시 컴파일을 해야 합니다.

7) 업로드 하기

화면 왼쪽 상단의 ⇨버튼을 눌러, 컴파일 완료된 예제 프로그램 소스를 업로드(다운로드)합니다.

실행 후 업로드 완료 라고 나타납니다.

업로드 될 때, 보드는 TX RX LED가 깜빡입니다.



그러면, 보드는 스케치의 프로그램으로 동작을 합니다.

10 프로젝트 목차

10 프로젝트1_LED깜빡이기

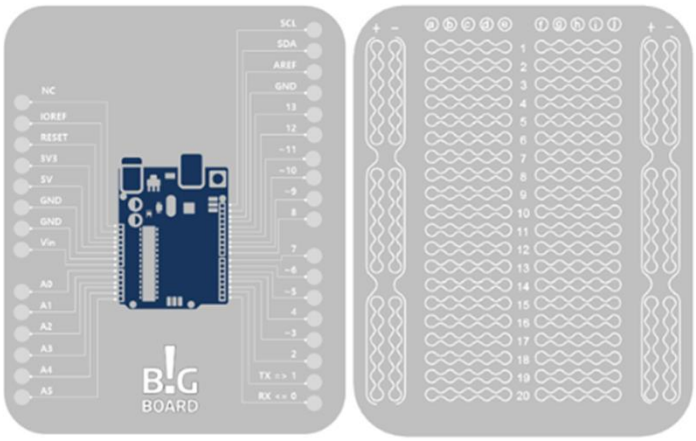
1) 학습 목표

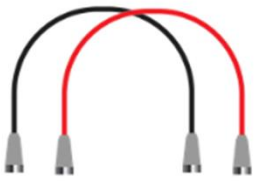
- ✓ LED에 대한 기초 개념과 특성을 알아 봅니다.
- ✓ 스케치와 빅보드 연결을 이해합니다.

2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	LED 부품	1	빛을 발생하는 부품
3	저항 220Ω	1	전류량을 조절하는 부품

빅보드

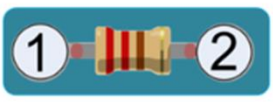




점프선

USB 케이블

LED



저항 220Ω

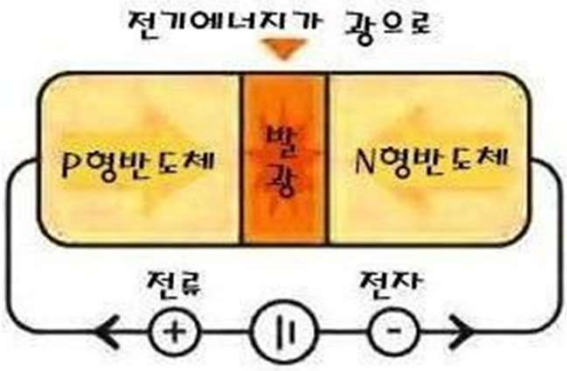
3) 핵심이론

➢ LED에 대한 기초 개념과 특성

✓ 한쪽 방향으로 전류가 흐르도록 제어하는 반도체 소자를 다이오드라고 합니다. 다이오드 중에 전기 에너지를 빛 에너지로 변환하는 것을 발광 다이오드(LED)라 합니다.

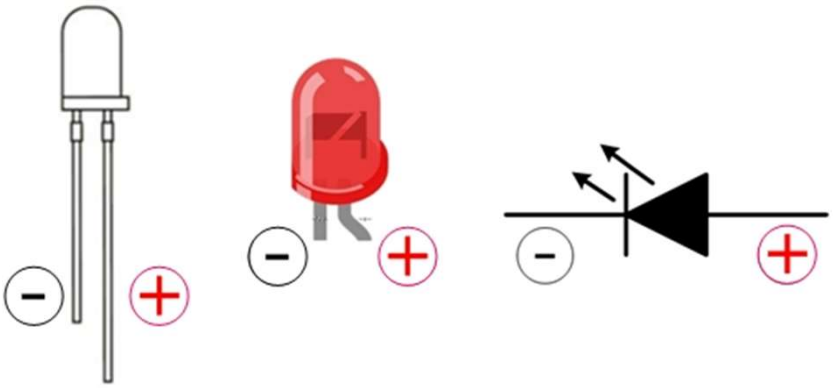
✓ LED는 Light Emitting Diode의 약자입니다.

✓ P형, N형 반도체를 접합한 다이오드에 순방향 전압을 인가하면 전자와 정공이 결합하면서 빛이 발광되는 원리로 빛이 발생합니다.

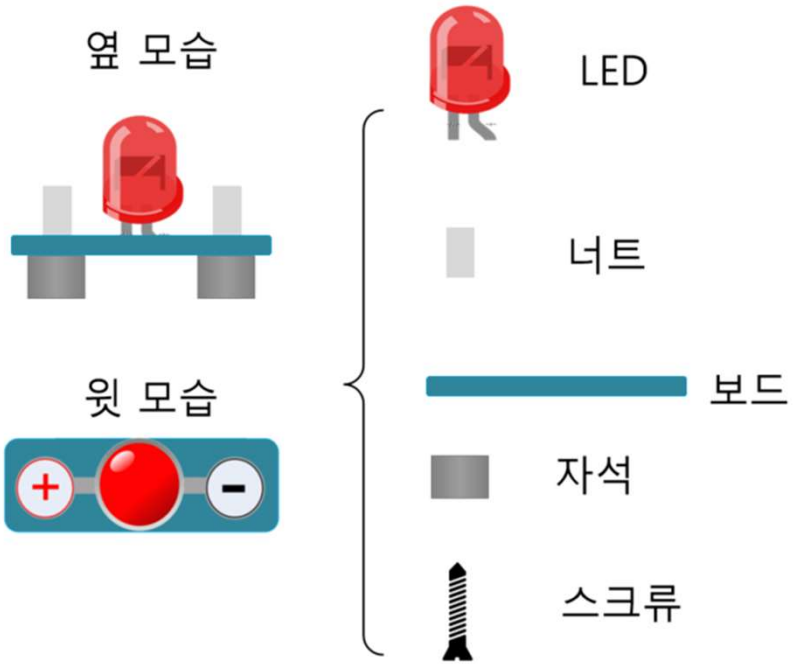


10 프로젝트1_LED깜빡이기

✓ LED는 극성(방향)이 있으므로, 연결 시 극성을 잘 확인해야 합니다.

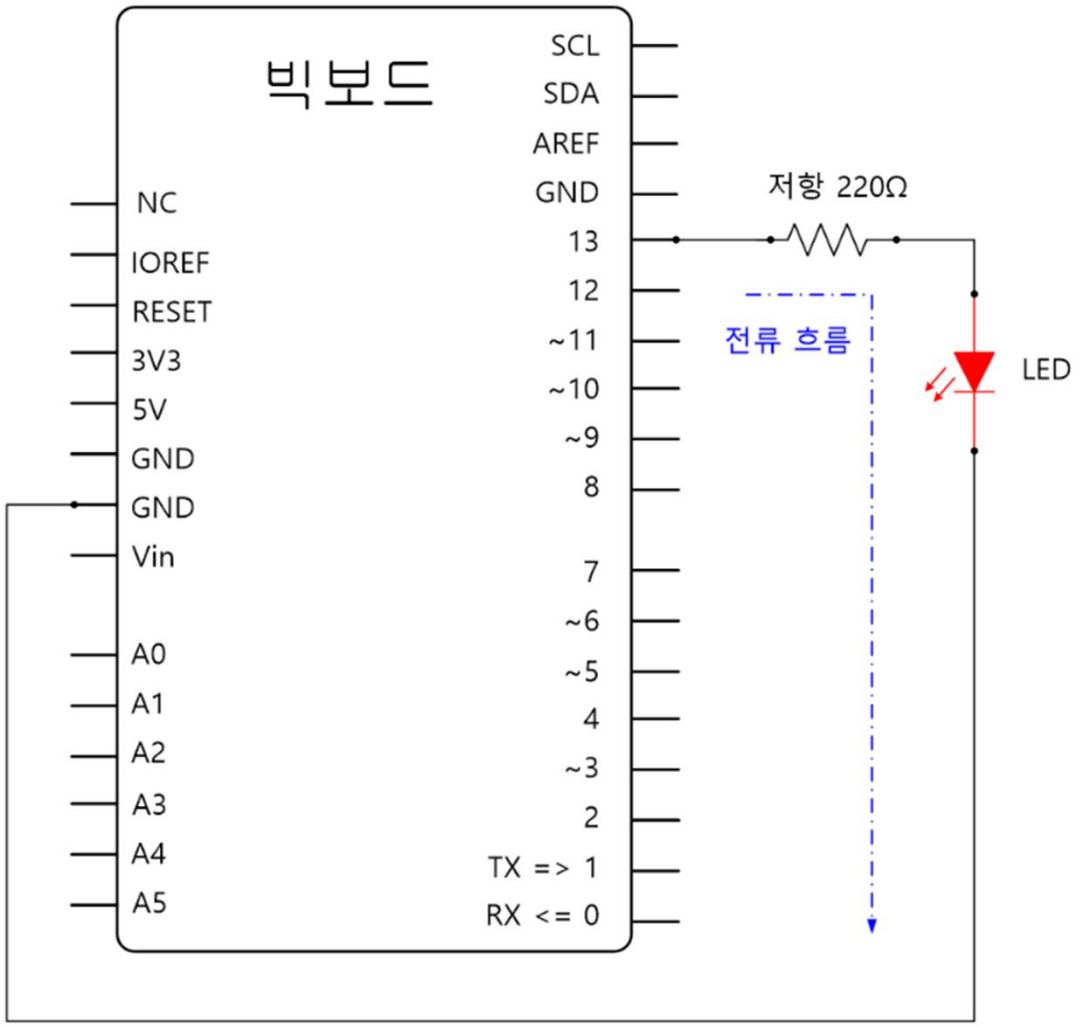


✓ 빅보드의 LED 부품은 아래와 같이, (LED + 자석) 연결 형태로 구성되어 있으며, 극성의 혼돈을 피하기 위하여, 극성 표시가 되어 있습니다.

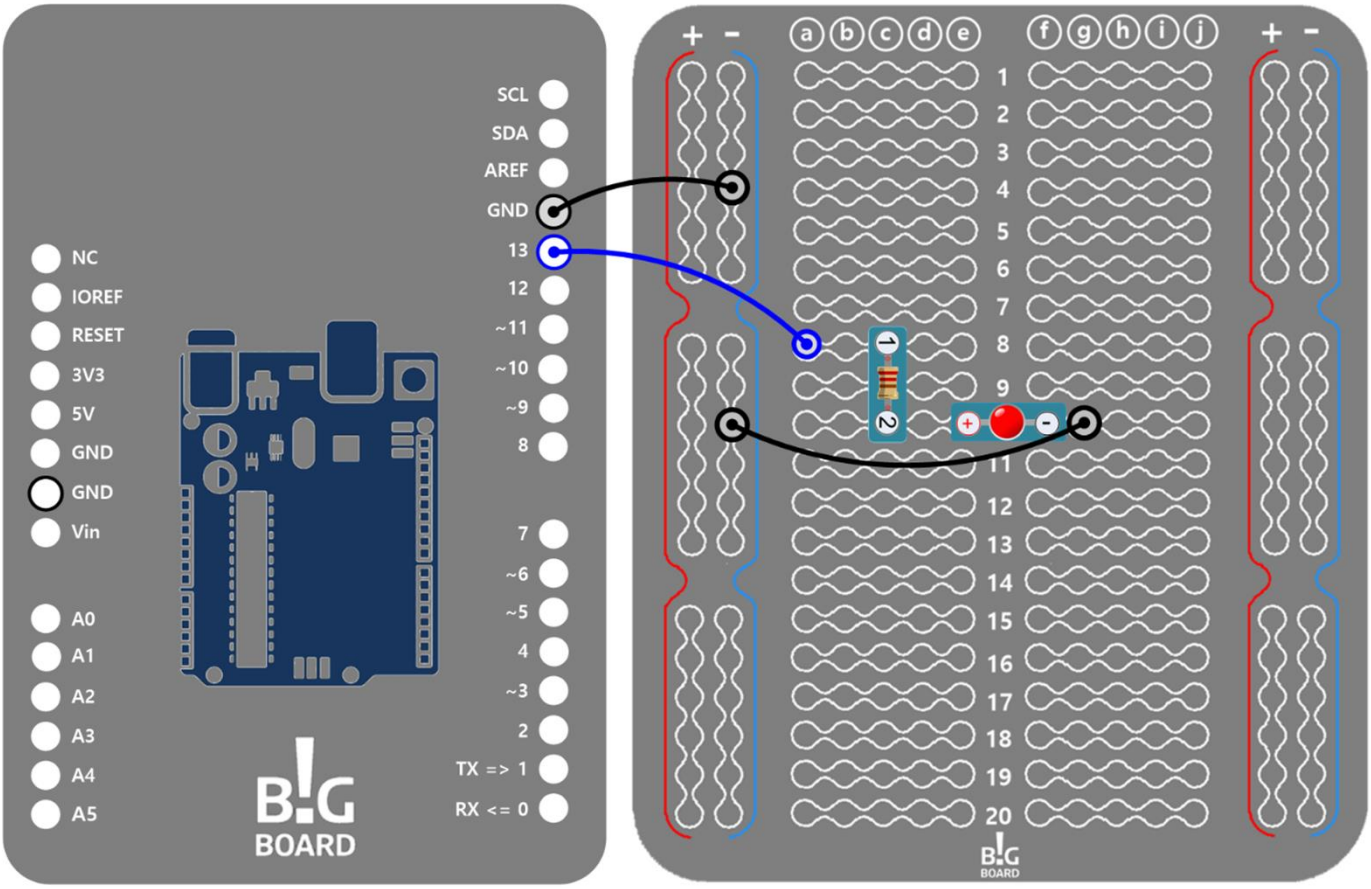


4) 회로 설명

미션) 13핀에 연결된 LED(램프)가 1초(1000ms) 동안 켜지고 꺼지기를 반복 하기입니다.



5) 하드웨어 연결 하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18
19
20

6) 코딩하기 -1

```
/*
  Bigboard-Starter-kit
  Project 1 - LED  깜빡이기 ; 설명문
*/
```

/* 주석 처리 */

여러 줄 주석 처리시 사용합니다.

보통은 프로그램을 관리하거나 메모하는 기능으로

기계어 변환 시 적용되지 않습니다.

즉, 프로그램 영역에 있지만, 전혀 동작과 상관없는 내용입니다.

```
// 초기 한번 설정
void setup() {
  pinMode(13, OUTPUT);   // 13번 핀을 출력으로 설정 (설명문)
}
```

// 주석 처리 부분, 한 줄 주석 처리시 사용합니다.

Void setup() { 코드 } // 초기 한번 설정하는 영역으로

보통은 이 영역에 환경 설정을 합니다.

pinMode(13, OUTPUT) ; // 13번 핀(pin)을 출력으로 설정합니다.

6) 코딩하기-2

```
// 반복 실행
void loop() {
 digitalWrite(13, HIGH);  // 디지털출력으로 13번 핀을 High로 출력
 delay(1000);   // 1000ms 동안 상태 유지

 digitalWrite(13, LOW);   // 디지털출력으로 13번 핀을 Low로 출력
 delay(1000);          // 1000ms 동안 상태 유지
 }
```

// 주석 처리 부분, 한 줄 주석 처리시 사용합니다.

void loop() { 코드 } // 계속 반복 실행하는 영역으로

보통은 이 영역에서 프로그램 동작 관련 코딩을 합니다.

digitalWrite(13, HIGH) ; // 13번 핀을 디지털 HIGH로 출력을 합니다.

delay(1000); // 설정 시간 동안 대기를 합니다. 시간의 단위는 ms입니다.

digitalWrite(13, LOW) ; // 13번 핀을 디지털 LOW로 출력을 합니다.

[동작]

13번 HIGH => 1초 대기 => LOW => 1초 대기 => 처음으로 다시 반복

11 프로젝트2_순차적으로 LED깜빡이기

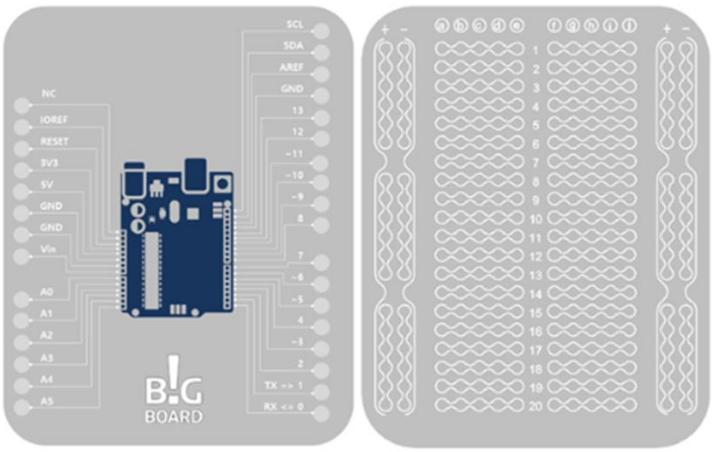
1) 학습 목표

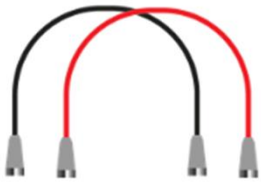
- ✓ 여러 개 LED를 이용한 순차적 켜기/끄기를 알아 봅니다.
- ✓ 디지털 출력을 이해합니다.

2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	LED 부품	3	빛을 발생하는 부품
3	저항 220Ω	3	전류량을 조절하는 부품

빅보드





점프선

USB 케이블

LED x 3개

저항 220Ω x 3개

3) 핵심 이론

➢ 디지털 출력 (Digital Out)의 기초 개념과 특성

✓ 디지털 신호는 0과 1로 표현됩니다.

✓ 신호 0은 GND를 의미하며, 모든 신호는 GND를 기준으로 그 전압차를 가지고, 판단합니다.

✓ 신호 1은 HIGH, 출력전압 만큼의 전압 값을 의미합니다.

✓ 빅보드의 1(HIGH)의 출력전압은 5.0V입니다.

디지털 신호

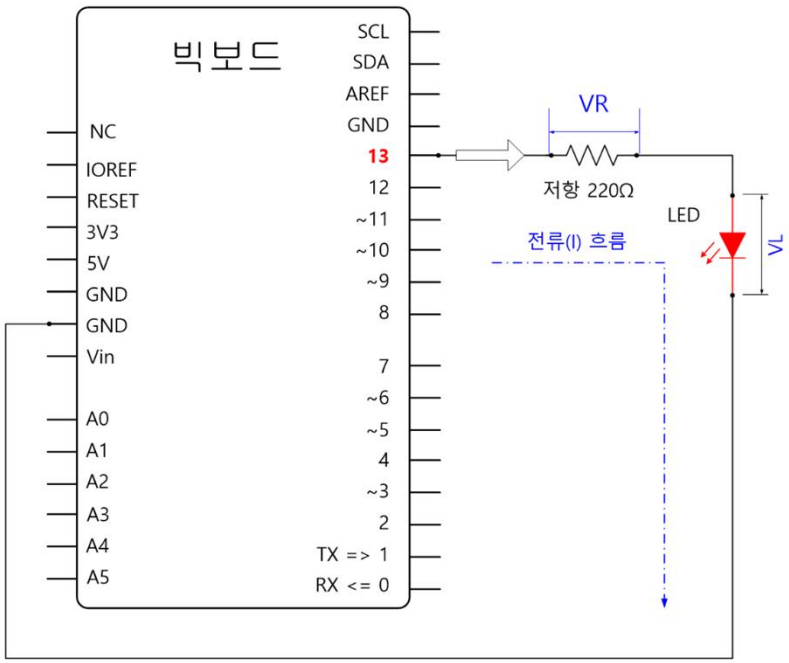
✓ 디지털 출력을 13번 핀으로 선택하고, 그 출력값을 1(HIGH) 값으로 하면, LED에 전류가 공급됩니다. 그러면 LED가 켜집니다.

✓ 또한, 출력값이 0(LOW)일 때는 전류 공급이 없어져서, LED가 꺼집니다.

➢ 디지털 출력과 LED 동작

✓ 디지털 출력값이 HIGH일 때, 5V가 출력되며, 전류의 량은 5V에서 LED에 동작에 필요한 전압(VL, 2.0V)를 뺀 나머지 전압 VR에 저항값에 의해서 결정됩니다.

✓ 저항이 220Ω일 때, 약 14mA정도의 전류가 흘려서, LED의 필요 전류(5~20mA)정도로 밝기가 적당히 유지됩니다.



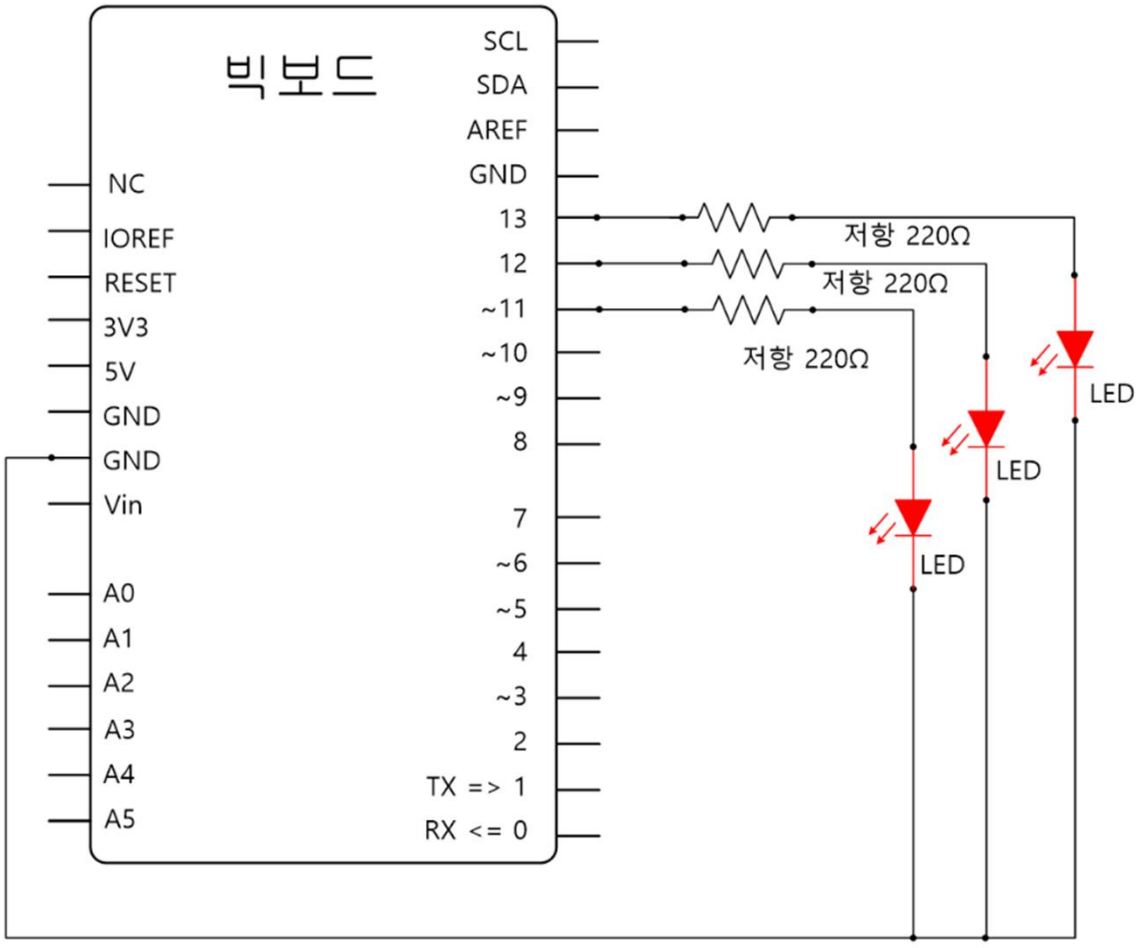
5V = VR + VL = VR -2.0

VR = 5.0 - 2.0 = 3.0

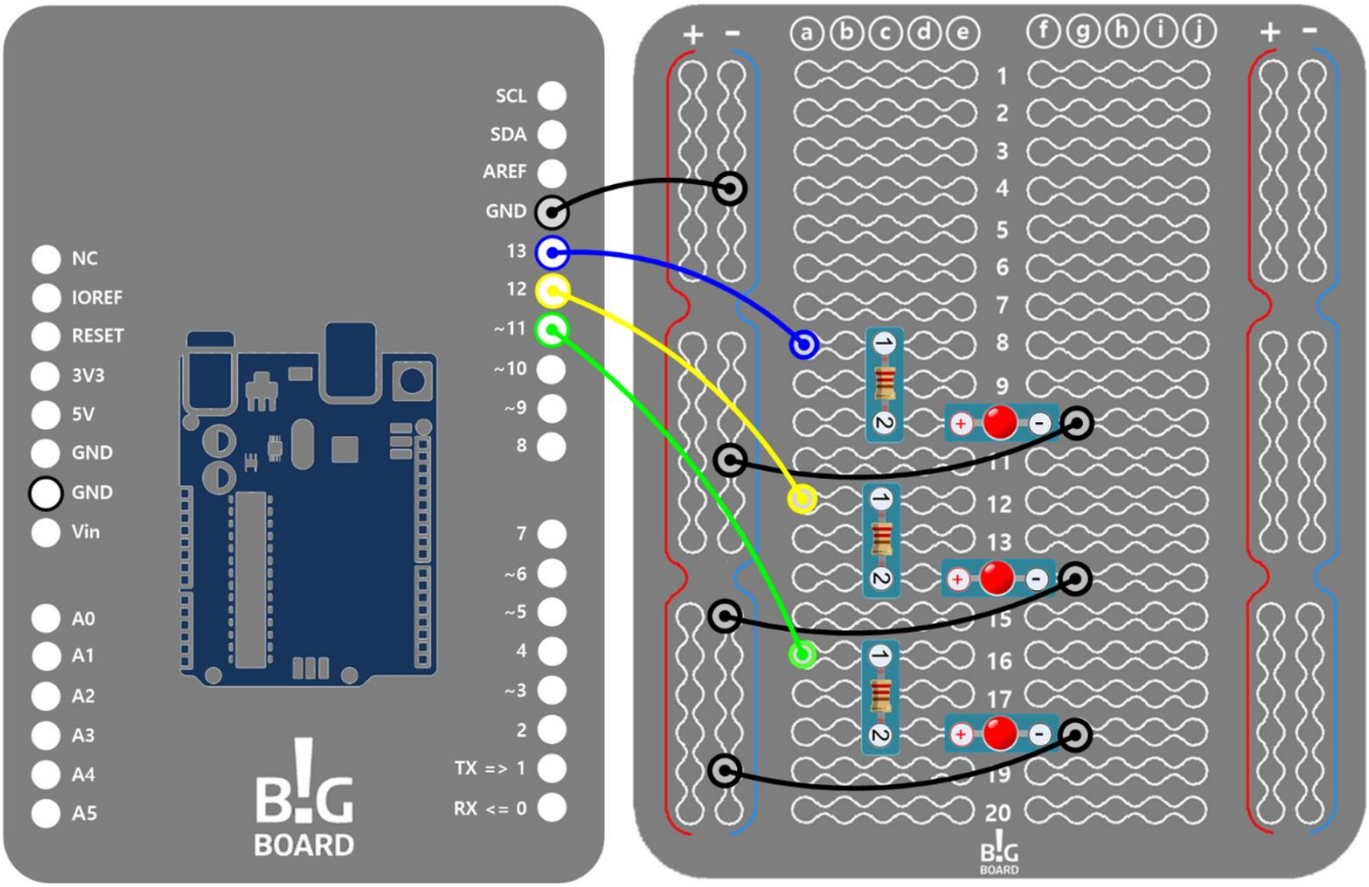
I = 3.0 / 220 =0.014A = 14mA

4) 회로 설명

미션) 11번 / 12번 / 13번 핀에 연결된 LED(램프)가 순차적으로 켜지고,
1초 후 꺼지는 동작을 반복하기입니다.



5) 하드웨어 연결하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
12
13
16
17
20

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 2 - LED  순차적으로 깜빡이기 ; 설명문
*/

// 초기 한번 설정
void setup() {
  pinMode(11, OUTPUT);   // 11번 핀을 출력으로 설정 (설명문)
  pinMode(12, OUTPUT);   // 12번 핀을 출력으로 설정 (설명문)
  pinMode(13, OUTPUT);   // 13번 핀을 출력으로 설정 (설명문)
}
```

// 주석 처리 부분, 한 줄 주석 처리시 사용합니다.

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(11, OUTPUT) ; // 11번 핀을 출력으로 설정합니다.

pinMode(12, OUTPUT) ; // 12번 핀을 출력으로 설정합니다.

pinMode(13, OUTPUT) ; // 13번 핀을 출력으로 설정합니다.

6) 코딩하기-2

```
void loop() {
 digitalWrite(11, HIGH);  // 디지털출력으로 11번 핀을 High로 출력
 delay(1000);   // 1000ms 동안 상태 유지
 digitalWrite(11, LOW);   // 디지털출력으로 11번 핀을 Low로 출력
 delay(1000);          // 1000ms 동안 상태 유지

 digitalWrite(12, HIGH);  // 디지털출력으로 12번 핀을 High로 출력
 delay(1000);   // 1000ms 동안 상태 유지
 digitalWrite(12, LOW);   // 디지털출력으로 12번 핀을 Low로 출력
 delay(1000);          // 1000ms 동안 상태 유지

 digitalWrite(13, HIGH);  // 디지털출력으로 13번 핀을 High로 출력
 delay(1000);   // 1000ms 동안 상태 유지
 digitalWrite(13, LOW);   // 디지털출력으로 13번 핀을 Low로 출력
 delay(1000);          // 1000ms 동안 상태 유지
 }
```

[동작]

11번 HIGH => 1초 대기 => LOW => 1초 대기 => 12번 HIGH => 1초 대기 => LOW => 1초 대기 => 13번 HIGH => 1초 대기 => LOW => 1초 대기 => 처음으로 다시 반복

12 프로젝트3_스위치로 LED제어하기

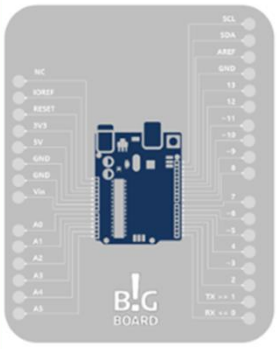
1) 학습 목표

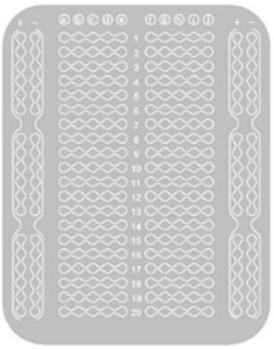
- ✓ 스위치로 LED를 켜기/끄기를 제어합니다.
- ✓ 디지털 입력을 이해합니다.

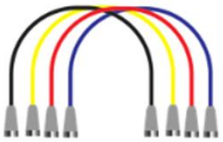
2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	LED 부품	1	빛을 발생하는 부품
3	저항 220Ω	1	전류량을 조절하는 부품
4	텍트 스위치	1	스위치 입력하는 부품
5	저항 10KΩ	1	스위치 입력 안정화하는 부품

빅보드





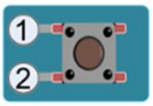


점프선

USB 케이블

LED x 1개

저항 220Ω x 1개



스위치 x 1개

저항 10KΩ x 1개

3) 핵심 이론

- ✓ 스위치는 전자 부품의 세계에서는 전류의 흐름을 막거나 계속 흐르게 하는 용도로 사용되는 도구입니다.
- ✓ 우리 일상 생활에서도 스위치는 많이 사용됩니다.
- ✓ 방안의 전구를 켤 때의 스위치, 비밀번호를 입력하는 스위치, 컴퓨터 키보드, 게임기 조이스틱, 음료수 자판기의 스위치 등 그 종류는 아주 많습니다.

푸시버튼 스위치(PB SW)

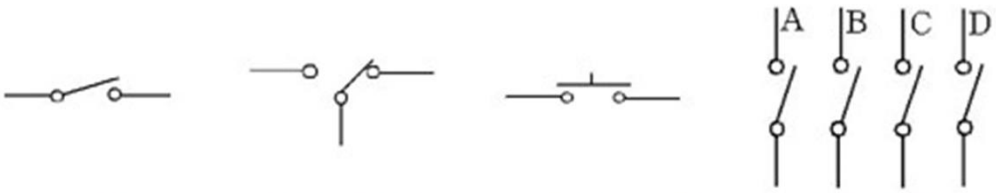
슬라이딩 스위치

토글 스위치

딥 스위치(DIP SW)

디지털 스위치(Digital SW)

일반적으로 전자회로에서 사용하는 스위치 기호는 다음과 같다.



일반적인 스위치 기호 | 토글 스위치 (toggle SW) | 푸시버튼 스위치 (PB SW) | 딥 스위치 (DIP SW)

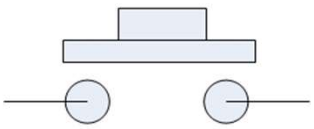
12 프로젝트3_스위치로 LED제어하기

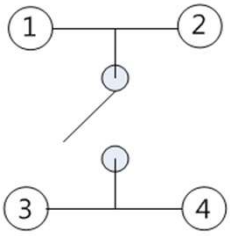
➢ 스위치(키, Key)의 동작원리

- ✓ 푸시 버튼 스위치나 텍트 스위치는 두 개의 지점을 연결시키는 전자 부품입니다.
- ✓ 텍트 스위치는 누를 때만 연결합니다. 푸시 버튼 스위치는 누를 때만 연결되는 부품과 계속 유지되는 되는 부품 형태도 있습니다.
- ✓ 여기서 스위치, 키(Key)는 누를 때만 연결되는 텍트 버튼 스위치를 사용합니다.

PUSH Swich

TAC Swich



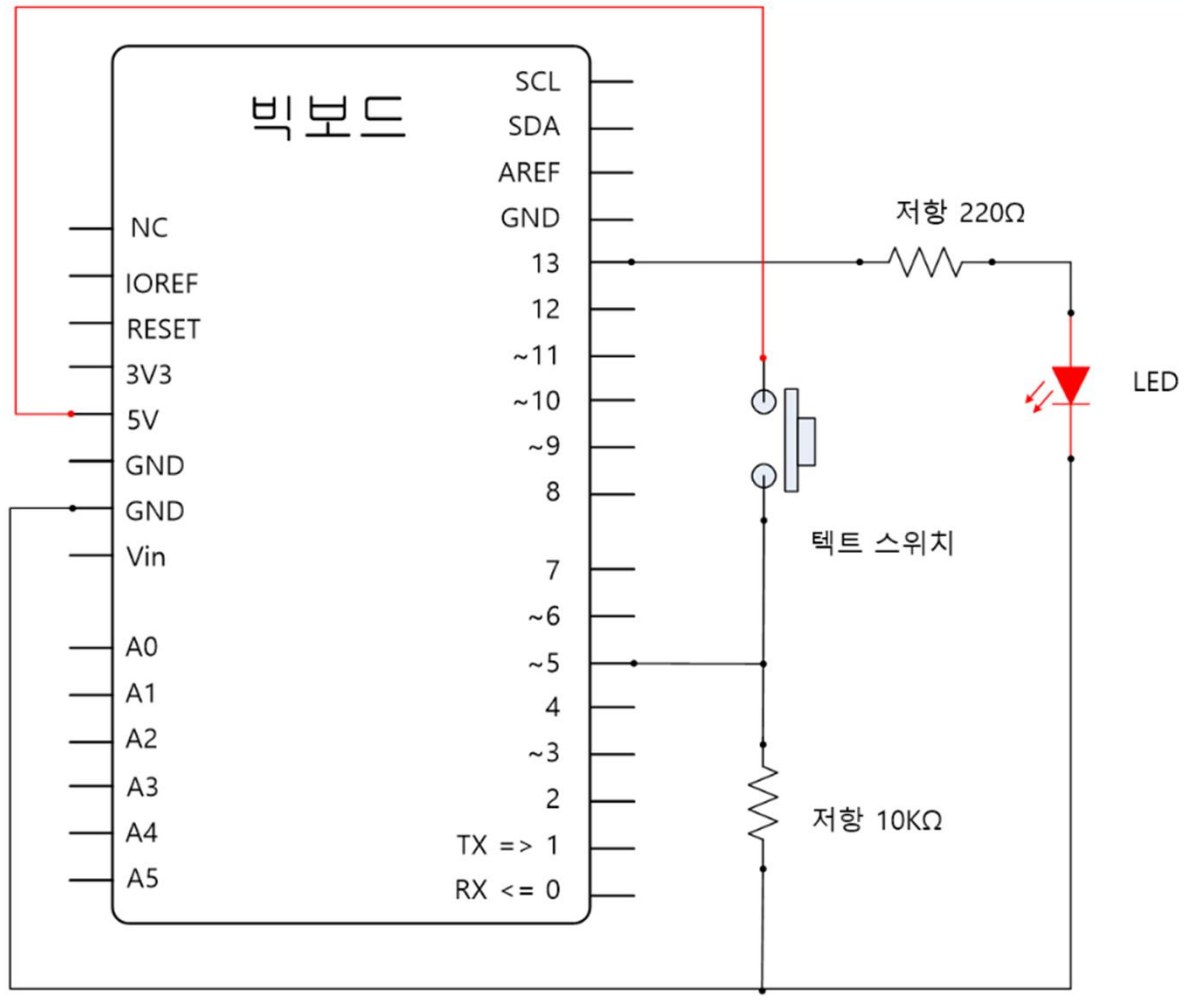


➢ 스위치의 응용 - 디지털 입력

- ✓ 스위치를 디지털 입력으로 사용할 수가 있습니다.
- ✓ 스위치 입력을 High (+5.0V)에서 인식하거나, Low (0V)에서 인식하게 디지털 입력단을 설정할 수가 있습니다.
- ✓ 디지털 신호는 보드의 마이크로 컨터롤러에 인가되는 전압(VCC)에 따라서, +5V, +3.3V 처럼 달라질 수 있습니다.

4) 회로 설명

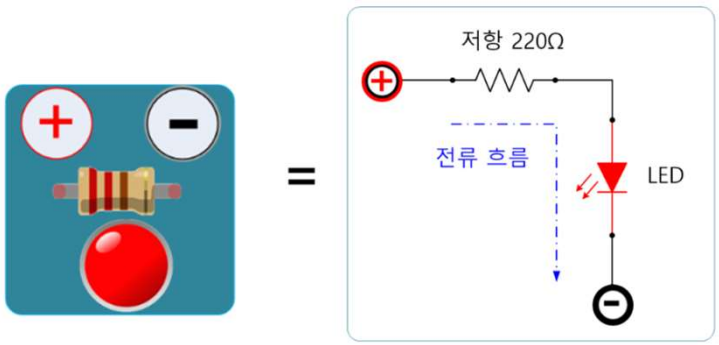
미션) 텍트 스위치를 누를 때만, LED가 켜지고, 누르지 않으면, LED가 꺼지도록 하세요!



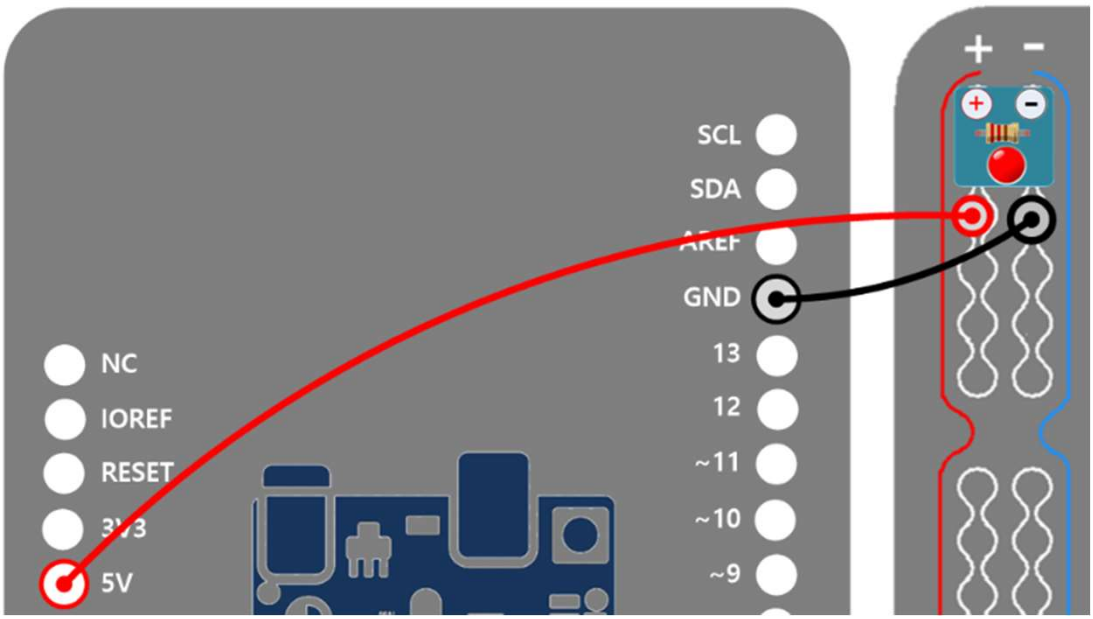
✓ 디지털 5번에 연결된 스위치의 저항은 풀 다운(pull-down)으로 연결되었기 때문에 버튼이 안 눌러졌을 때 0(false), 눌러졌을 때 1(true)의 값을 갖습니다.

5) 하드웨어 연결하기 - 팁

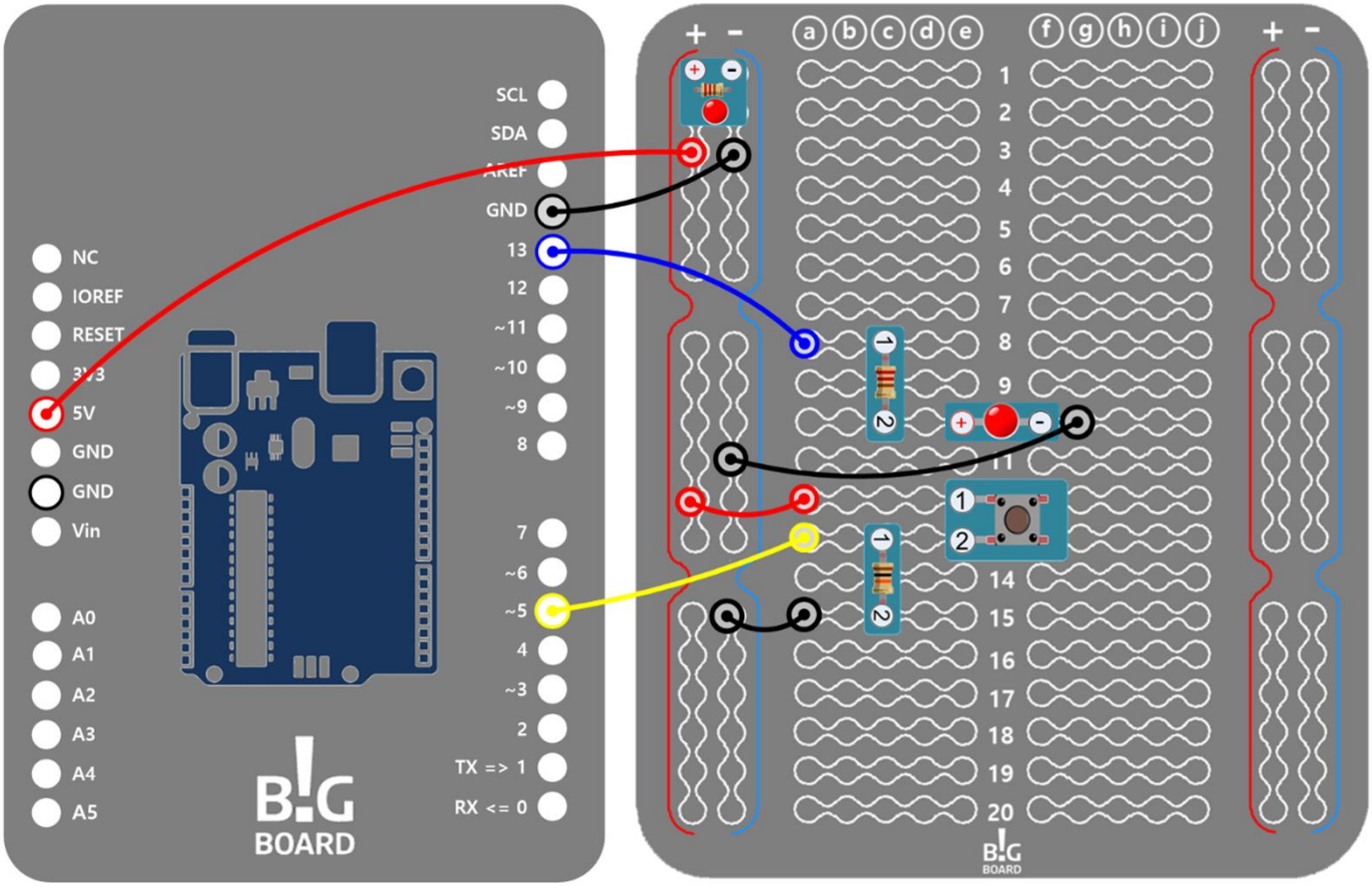
- ✓ 회로 동작을 위해서, 전원 공급은 필수입니다.
- ✓ 전원이 제대로 공급되고 있는지 확인하는 방법으로 전원 LED 부품을 활용해 보세요! .



- ✓ 항상 회로를 꾸밀 때, 먼저 전원 LED 부품이 켜져 있어야만 정상적으로 동작한다는 것입니다.
- ✓ 잘못 연결했을 때, 전원 LED는 꺼집니다.



5) 하드웨어 연결하기


SCL
SDA
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
14
15
16
17
18
19
20

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 3 - 버튼을 누르는 동안 LED켜기 ; 설명문
*/
int button = 5;  // 버튼 핀을 5번으로 설정
int LED = 13;     // LED핀을 13번으로 설정
int buttonState = 0; // 입력 핀의 상태를 읽기 위한 변수
```

/* 주석 처리 * /

[초기 설정]

int button = 5; // int (정수 변수)의 이름을 button으로 명명하고, 그 값을 5로 설정

int LED = 13 // int (정수 변수)의 이름을 LED 로 명명하고, 그 값을 13로 설정

int buttonState = 0; // int (정수 변수)의 이름을 buttonState로 명명하고, 그 값을 0 으로 설정

6) 코딩하기-2

```
void setup() {
pinMode(LED , OUTPUT);  // LED핀을 출력으로 설정
pinMode(button, INPUT);
}
```

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(LED , OUTPUT); // LED핀을 출력으로 설정
pinMode(button, INPUT); // button 핀을 입력으로 설정.

6) 코딩하기-3

```
void loop() {
 buttonState = digitalRead(button);

 if(buttonState  == HIGH) {  // 버턴의 상태가 High 이면
    digitalWrite(LED , HIGH);   // LED 핀을 High로 출력
  }
  else {    // 버턴의 상태가 High가 아니면
    digitalWrite(LED , LOW);  // LED 핀을 Low로 출력
}}
```

void loop() { } // 계속 반복 하기

buttonState = digitalRead(button);

// digitalRead () 디지털 값을 읽어서, buttonState 값에 저장을 합니다.

if(조건) // 조건문입니다.

{ 참일 때, 실행 } // 조건이 만족, 참(1, HIGH)일 때, 실행

else { 거짓일 때 실행 } // 만족 못할 때, 거짓(0, LOW)일 때, 실행

[동작]

버튼을 누르면, 5번 핀에 LOW => HIGH 값으로 변경이 됩니다. 그러면, 그 상태 값이 참이 되므로, LED를 켜줍니다.

13 프로젝트4_피에조 부저로 소리내기

1) 학습 목표

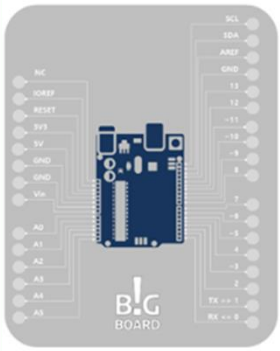
✓ 스위치로 부저에게 소리내기를 합니다.

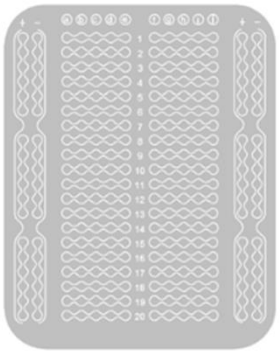
✓ 디지털 출력을 이해합니다.

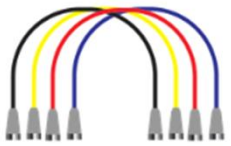
2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	텍트 스위치	1	스위치 입력하는 부품
3	저항 10KΩ	1	스위치 입력 안정화하는 부품
4	부저	1	소리를 내는 부품

빅보드

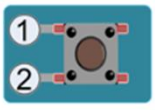






점프선

USB 케이블



스위치 x 1개

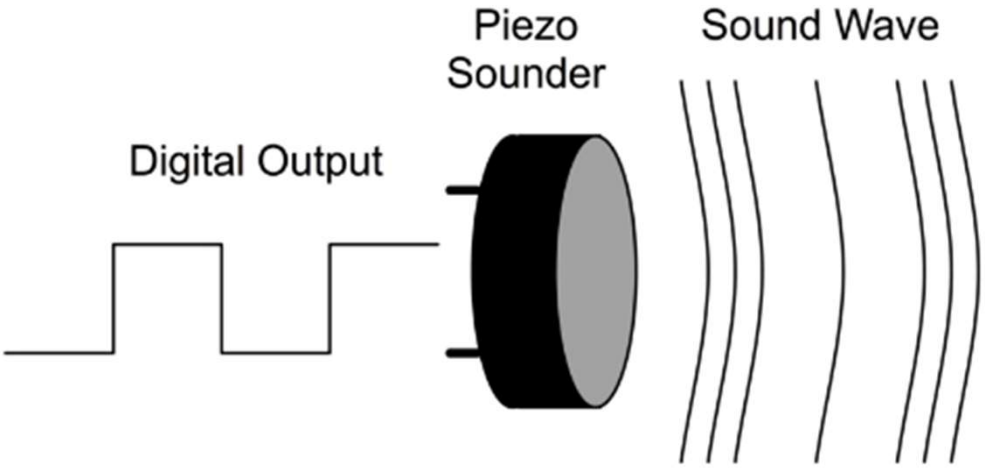
저항 10KΩ x 1개

부저 x 1개

3) 핵심 이론

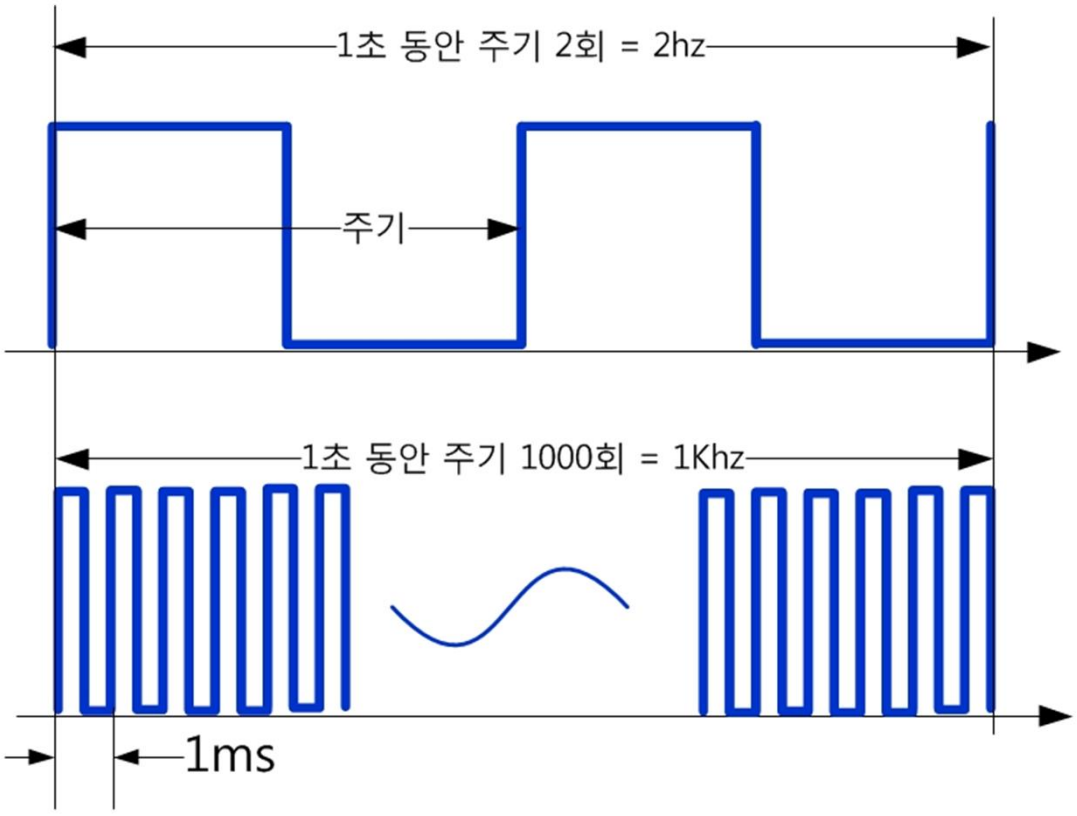
➢ 피에조 부저, 스피커 (Piezoelectric Speaker)

- ✓ 피에조 부저는 크기가 작고, 저렴하지만 간단하게 전기 신호를 주어서 소리를 내게 하는 전자 부품입니다.
- ✓ 피에조 필름에 진동이나 힘을 가하면, 힘에 비례하여, 전압이 발생하여, 압전센서 역할을 하고, 반대로 필름에 전압을 넣을 경우 기계에서 소리가 납니다.



➢ 주파수 (Frequency)

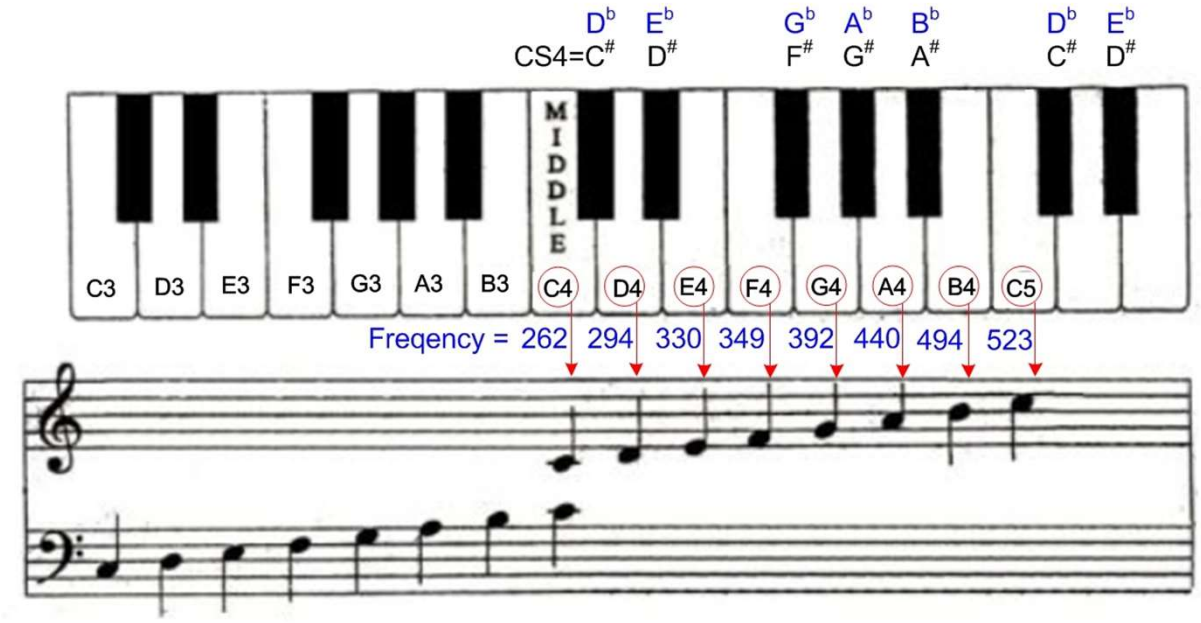
- ✓ 디지털 출력의 High / Low 신호가 일정한 간격으로 반복된다면, 1초 동안 반복된 회수를 주파수(Frequency)라 하고, 단위는 헤르츠(Hz)로 표기합니다.
- ✓ 희망하는 주파수에 맞는 High/Low 반복 비율로 제어해야만 소리를 만들 수가 있습니다.
- ✓ 만약에 1Khz 소리를 발생하려면, 0.5ms 단위로 High/Low를 반복하여야 합니다.



➢ tone() 함수 이해하기

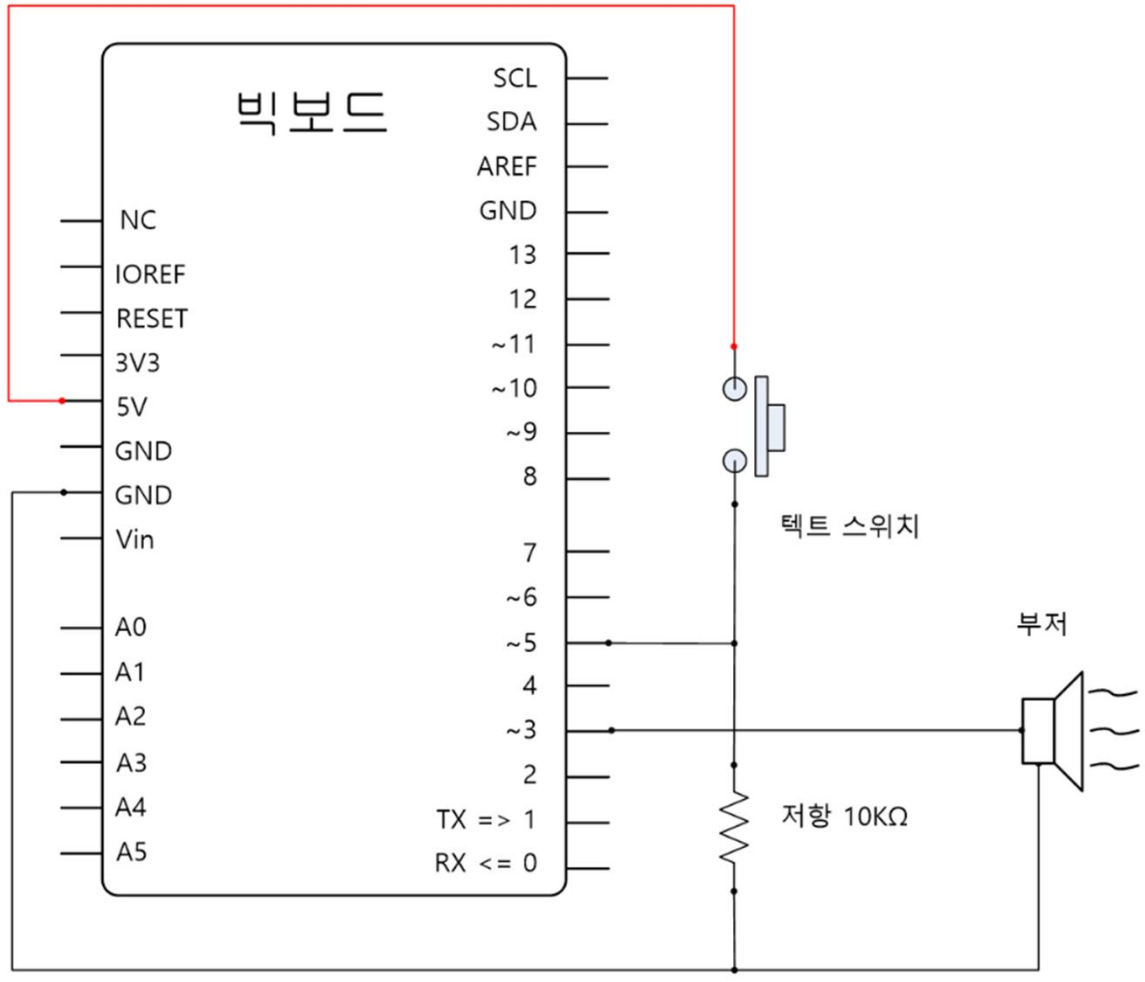
- ✓ tone 함수를 사용하면, 간단히 원하는 주파수를 만들 수가 있습니다. 함수는 tone(출력 핀 번호, 주파수, 시간) 입니다.
- ✓ 예로, tone(3, 1000, 1000) ; 3번 핀에 1000hz 주파수로 1000ms 동안 출력합니다.
- ✓ 사용 가능한 핀은 ~ 무늬 표시된 3, 5, 6, 9, 10, 11번입니다.

- ✓ 음표는 [C, D, E, F, G, A, B, C]로 표시하는데, 우리말로 [도, 레, 미, 파, 솔, 라, 시, 도] 이며, 소리의 높낮이에 따라서, 주파수가 달라집니다. C 음은 보통 262Hz로 입니다.
- ✓ 좋은 멜로디의 소리가 출력되려면, 음(소리)의 높낮이와 박자(길이)가 조화롭게 되어야 합니다.

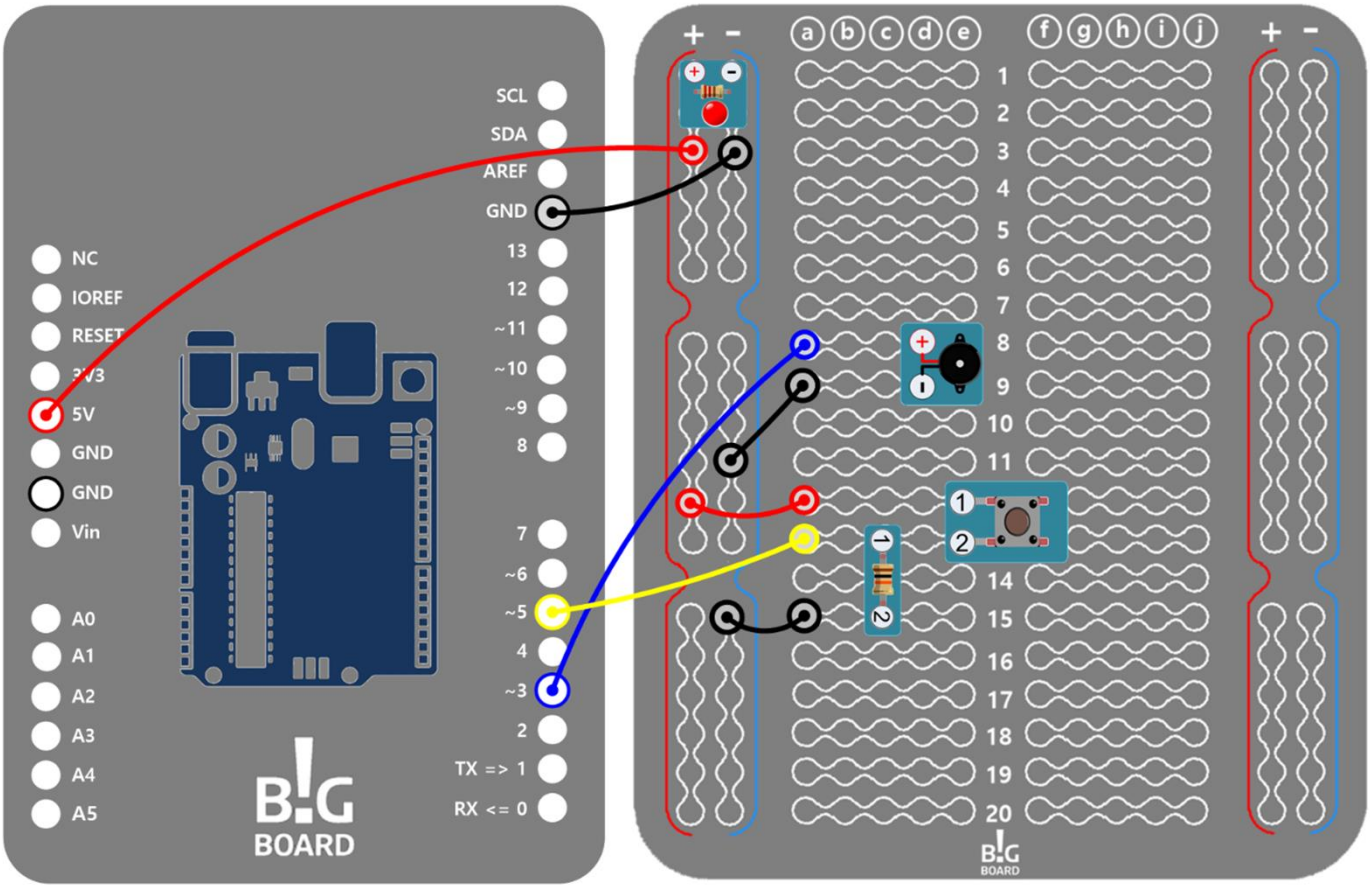


4) 회로 설명

미션) 텍트 스위치를 누를 때만, 부저 소리가 나오고, 누르지 않으면, 부저 소리가 꺼지도록 하세요!



5) 하드웨어 연결 하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
10
11
14
15
16
17
18
19
20

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 4 - 피에조 부저로 소리내기 ; 설명문
*/
int speakerPin = 3;
int button = 5;
int buttonState = 0;
```

/* 주석 처리 * /

[초기 설정]

int speakerPin = 3;

// int(정수 변수)의 이름을 speakerPin 으로 명명하고,

// 그 값을 3으로 저장

int button = 5;

// int(정수 변수)의 이름을 button 으로 명명하고, 그 값을 5로 저장

int buttonState = 0;

// int(정수 변수)의 이름을 buttonState 로 명명하고, 그 값을 0으로 저장

6) 코딩하기-2

```
void setup() {
pinMode(speakerPin, OUTPUT);  // 스피커 핀을 디지털 출력으로 설정
pinMode(button, INPUT);  // 버튼 핀을 디지털 입력으로 설정
}
```

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(speakerPin, OUTPUT);
// speakerPin 핀을 출력으로 설정

pinMode(button, INPUT);
// button 핀을 입력으로 설정

6) 코딩하기-3

```
void loop() {
buttonState = digitalRead(button); // 버튼이 눌려졌는지를 읽음
 if(buttonState == HIGH) {  // 조건; 버튼의 상태가 High 이면
 tone(speakerPin, 2000, 1000); // 스피커에 2000hz의 음을 1초 동안 내기
 delay(1000);
 }
  else {   // 버튼의 상태가 High가 아니면
  noTone(speakerPin); // 스피커 핀의 출력을 하지 않습니다.
}}
```

void loop() { } // 계속 반복 하기

tone(출력 핀 번호, 주파수, 시간)

// 톤을 발생시키는 함수입니다.

// 출력 핀 중에 ~ 표시가 있는 3, 5, 6, 9, 10, 11번에 연결을 합니다.

noTone(출력 핀 번호)

// 톤을 발생을 중단 시키는 함수입니다.

[동작]

버튼을 누르면, 5번 핀에 LOW => HIGH 값으로 변경이 됩니다. 그러면, 그 상태 값이 참이 되므로, 3번 핀으로 톤(tone) 출력을 합니다.

14 프로젝트5_조도 센서로 LED 제어하기

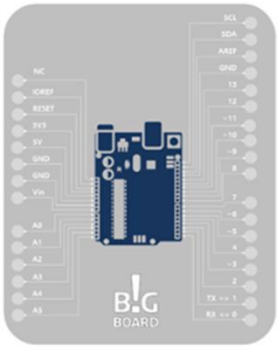
1) 학습 목표

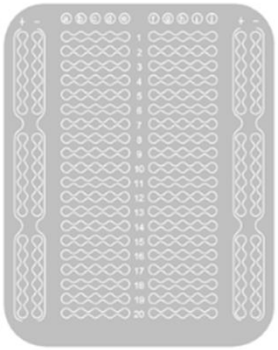
✓ 조도 센서를 이해합니다.
✓ 아날로그 입력을 이해합니다.

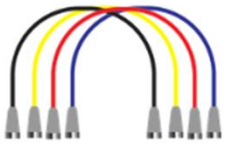
2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	LED 부품	1	빛을 발생하는 부품
3	저항 220Ω	1	전류량을 조절하는 부품
4	조도 센서	1	빛의 밝기를 측정하는 부품
5	저항 10KΩ	1	밝기 측정 회로의 구성 저항

빅보드







점프선

USB 케이블

LED x 1개

저항 220Ω x 1개

조도 센서 x 1개

저항 10KΩ x 1개

14 프로젝트5_조도 센서로 LED 제어하기

3) 핵심 이론

➢ 조도센서 (CDS)

✓ 조도센서 는 주변의 밝기를 측정하는 센서입니다.

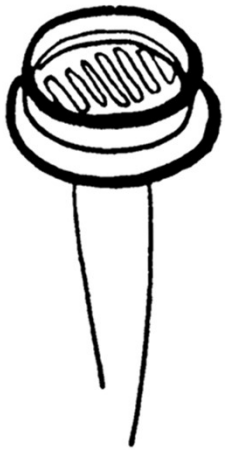
✓ 조도센서는 저렴한 가격과 활용도 때문에 많이 사용하는 광도전 효과를 이용한 반도체 포토 센서입니다.

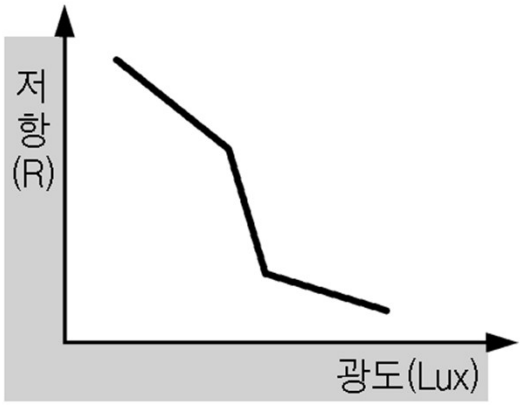
✓ 어두워지면 자동으로 켜지는 가로등, 자동차의 헤드라이트, 밝기 에 따라 변하는 핸드폰 화면 액정, 실내 스탠드 등 실생활에서도 쉽게 찾아 볼 수 있습니다.

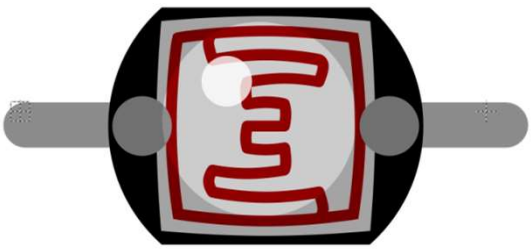
➢ 포토 센서의 특성 및 구조

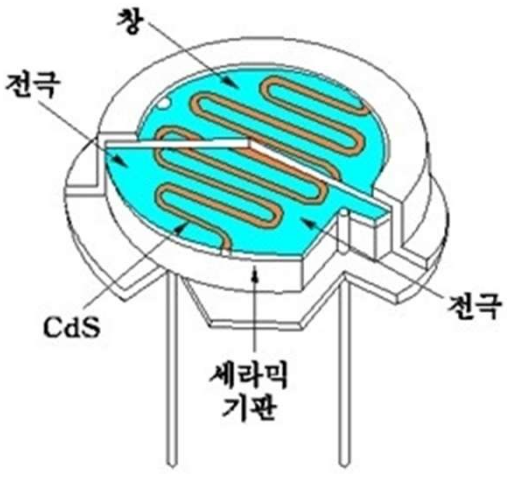
✓ 주위가 밝으면 저항이 줄어들고, 주위가 어두우면 저항이 커지는 특성입니다.

✓ CDS라고 불리는 이유는 CDS Photoresistor를 만드는 주 재료가 카드륨(CD), 황(S)의 화합물인 황화카드뮴(CDS)이기 때문입니다.









14 프로젝트5_조도 센서로 LED 제어하기

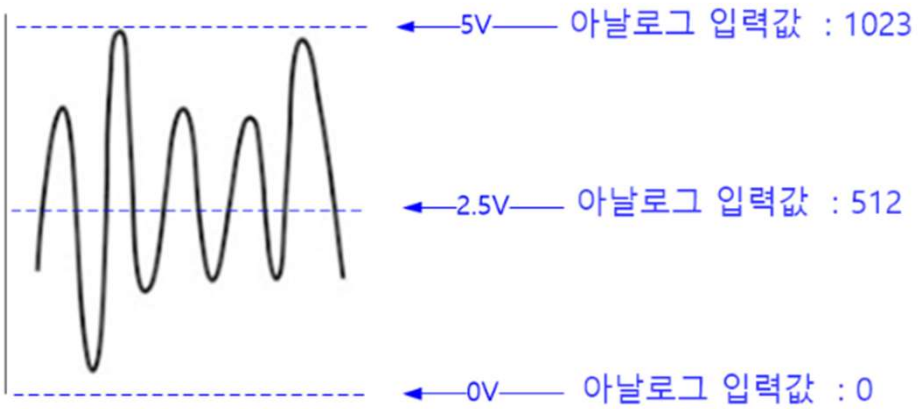
➢ 아날로그

✓ 아날로그(analogue)는 어떤 자료를 '길이', '각도' 또는 '전류'와 같이 외부적인 원인에 의해 연속적으로 변하는 것들을 물리량으로 나타내는 일. 자동차의 속도를 바늘의 각도로 표시해주는 속도 측정계, 수은주의 길이로 온도를 나타내는 온도계, 상대적으로 얕게 패이거나 깊게 패인 여러 홈들과 바늘의 마찰로 인해 녹음된 소리가 나오는 음반(LP)이 아날로그의 예이다. 디지털에 대비되어 쓰인다.

✓ 우리가 거시적인 자연에서 얻는 신호는 대개 아날로그이다. 이를테면, 빛의 밝기, 소리의 높낮이나 크기, 바람의 세기 등이 있다.

[위키피디아 참조]

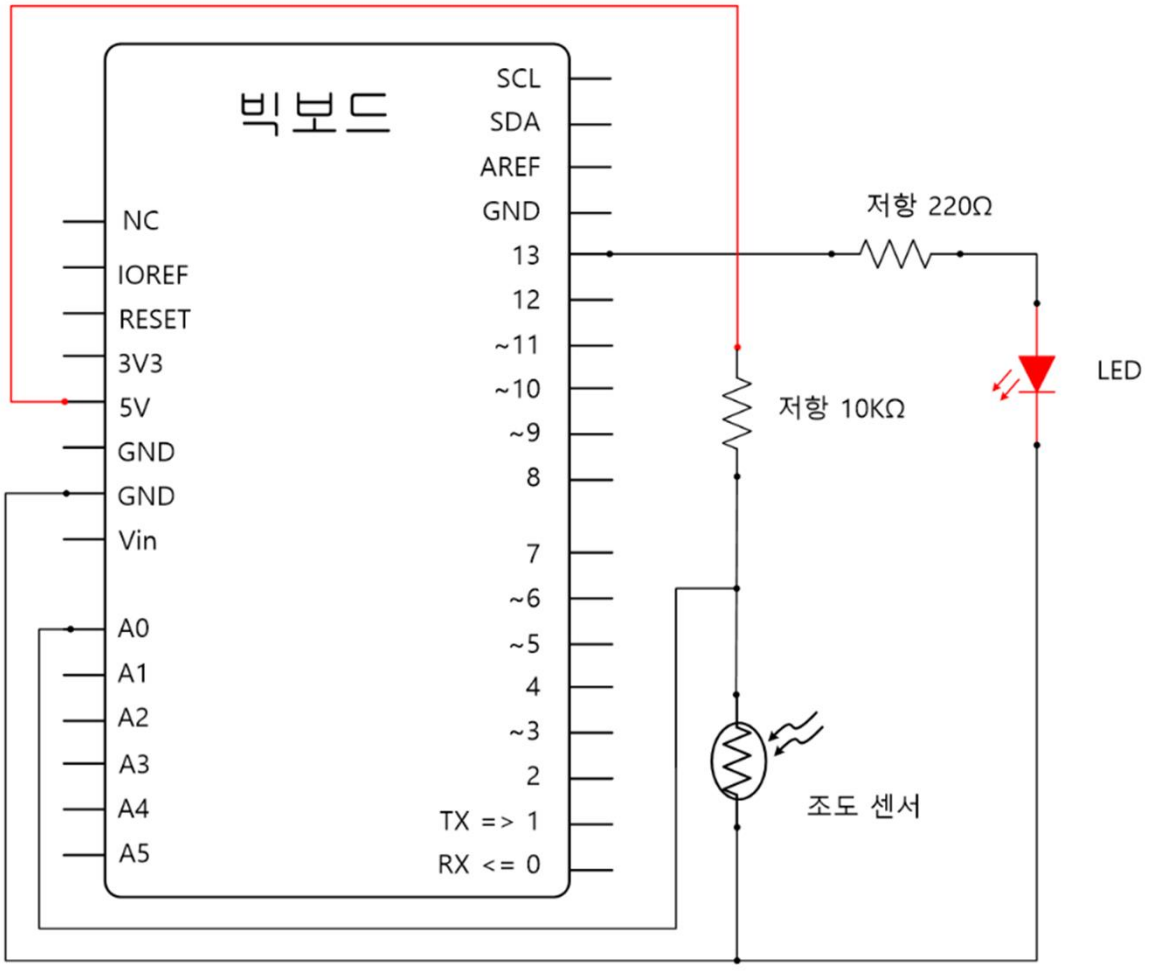
✓ 빅보드에서는 아날로그 발생 신호를 0 ~ 5V 전압 범위 내에서 읽을 수 있으며, 그 신호 값은 0 ~ 1023 값으로 표현합니다.



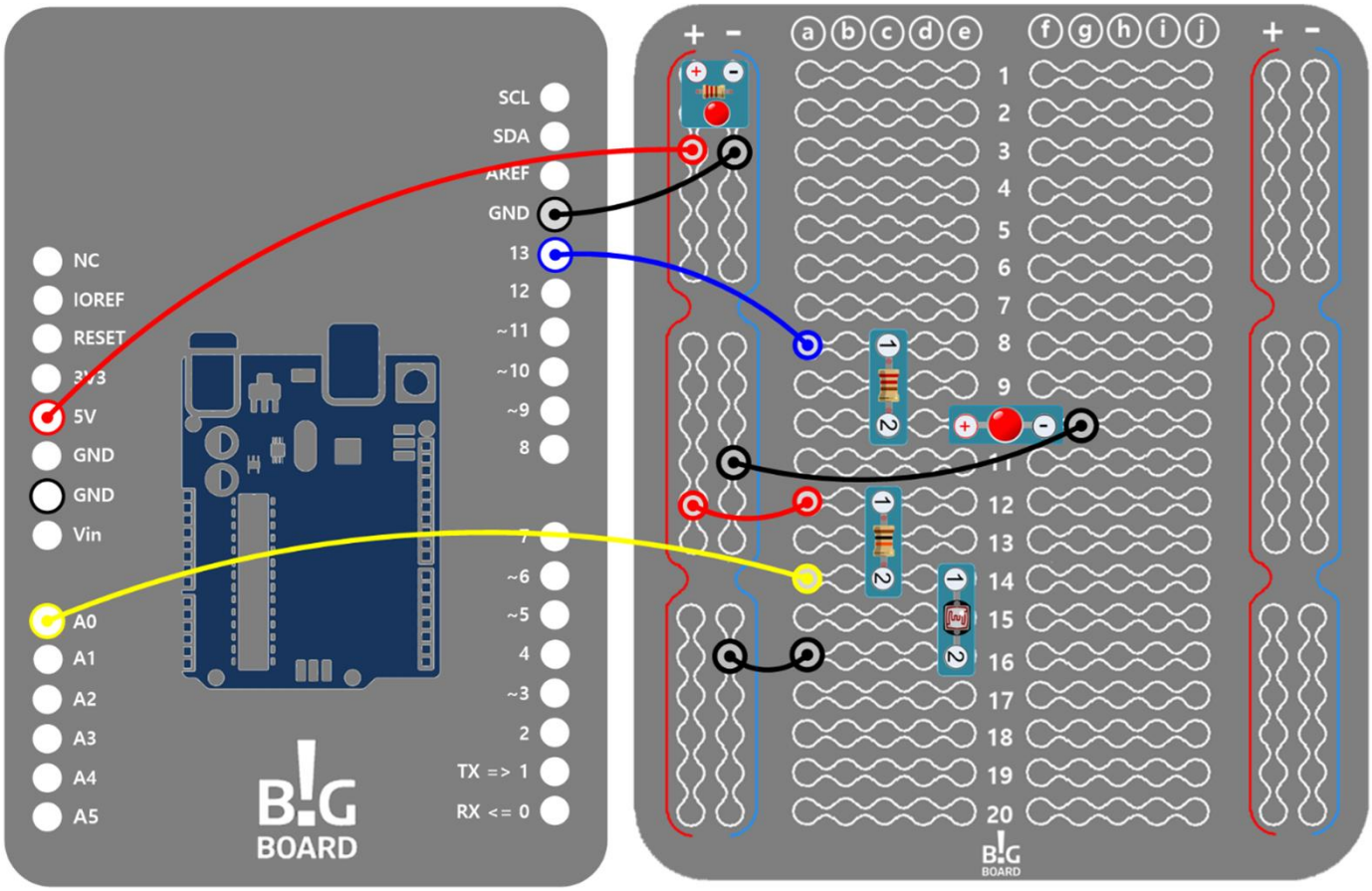
아날로그 신호 (입력)

4) 회로 설명

미션) 조도 센서를 이용해서, 어두우면, LED를 켜도록 하세요!



5) 하드웨어 연결 하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 5 - 조도센서로 LED 제어하기 ; 설명문
*/
int  CDS = A0;
Int  LED = 13;
Int  SensorValue = 0;
```

/* 주석 처리 */

int CDS = A0; // int(정수 변수)의 이름을 CDS 로 명명하고,
그 값을 A0 로 저장

int LED = 13; // int(정수 변수)의 이름을 LED 로 명명하고,
그 값을 13으로 저장

int SensorValue = 0; // int(정수 변수)의 이름을 SensorValue 로 명명하고,
그 값을 0으로 저장

6) 코딩하기-2

```
void setup() {
 pinMode(CDS,  INPUT);  // CDS 핀을 입력으로 설정
 pinMode(LED,  OUTPUT);  // LED 핀을 출력으로 설정
  }
```

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(CDS, INPUT); // CDS 핀을 입력으로 설정

pinMode(LED, OUTPUT); // LED 핀을 출력으로 설정

```
void loop() {
  SensorValue = analogRead(CDS);
if ( SensorValue >=600 ) {
    digitalWrite(LED, HIGH);
 }
    else {
digitalWrite(LED, LOW);
 }}
```

void loop() { } // 계속 반복 하기

SensorValue = analogRead(CDS); // CDS 값을 아날로그 입력으로 읽음

if (SensorValue >=600) { // 센서 값이 600보다 같거나 크면
digitalWrite(LED, HIGH); // LED 를 디지털 High 로 출력

else { // 센서 값이 600보다 작으면
digitalWrite(LED, LOW); // LED 를 디지털 LOW 로 출력

[동작]
조도 센서의 값을 읽어서, 그 값이 600보다 크거나 같으면, 참이 되므로,
LED를 켜주고, 값이 600보다 작으면, 거짓으로 LED를 꺼줍니다.

15 프로젝트6_가변저항으로 밝기 조절하기

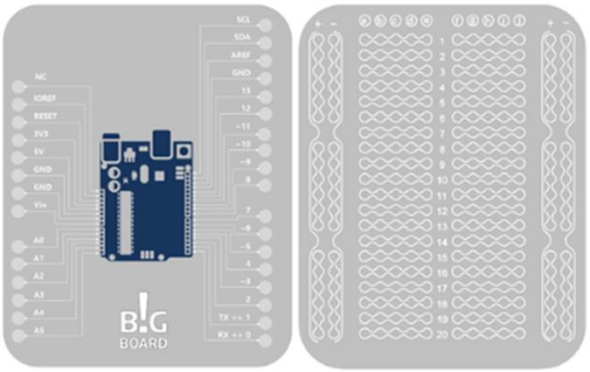
1) 학습 목표

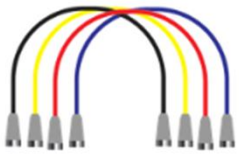
- ✓ 가변 저항을 이해합니다.
- ✓ 아날로그 입력을 이해합니다.

2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	LED 부품	1	빛을 발생하는 부품
3	저항 220Ω	1	전류량을 조절하는 부품
4	가변저항	1	입력 전압을 가변하는 부품

빅보드



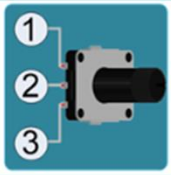


점프선

USB 케이블

LED x 1개

저항 220Ω x 1개



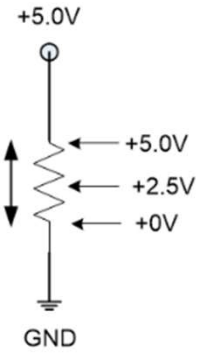
가변저항 x 1개

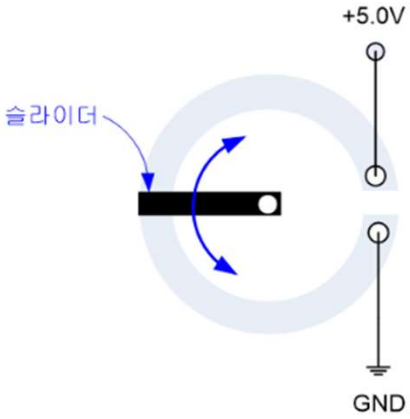
3) 핵심 이론

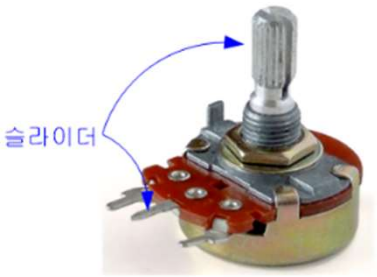
➢ 가변 저항기

✓ 가변저항기는 둥글게 생긴 트랙이 있는데, 저항과 같은 역할을 합니다. 가변저항기가 일반저항과 다른 점은 슬라이더라 불리는 중간연결 입니다. 이 슬라이더는 가변저항기를 돌리면 같이 돌아 가는데, 0~ 최대 저항 값까지 가변 합니다.

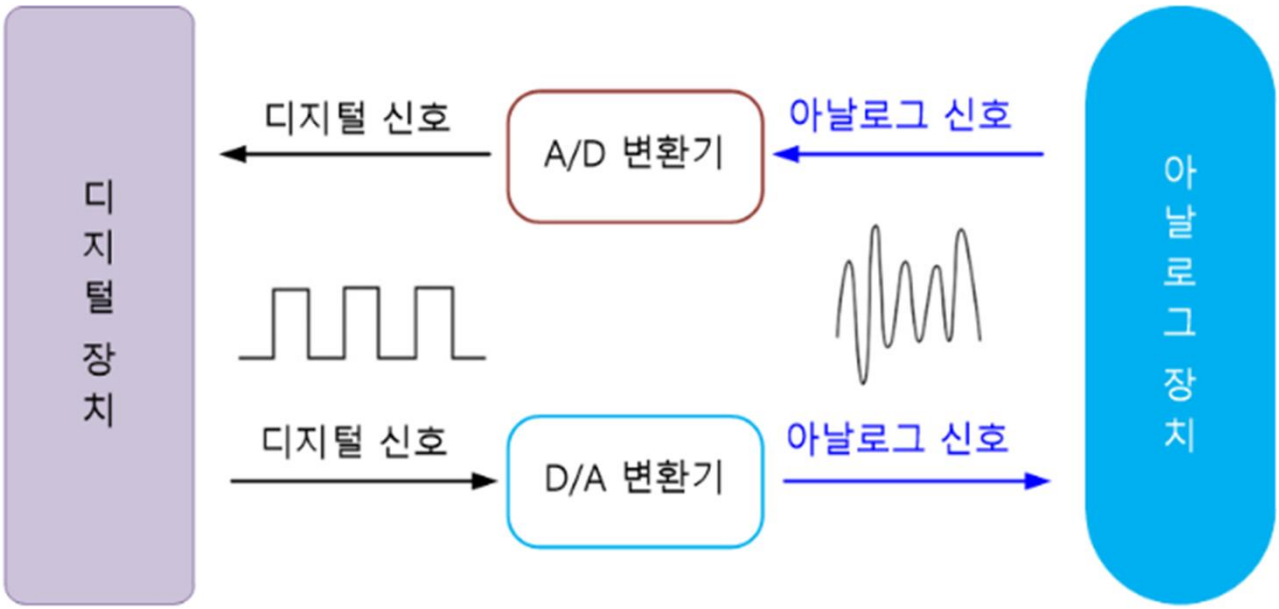
✓ 가변 저항기 끝에 5.0V를 입력하면, 슬라이더의 전압은 5.0V 쪽으로 돌리면 점차 5.0V가 되고 GND 쪽으로 돌리면 0V가 됩니다.







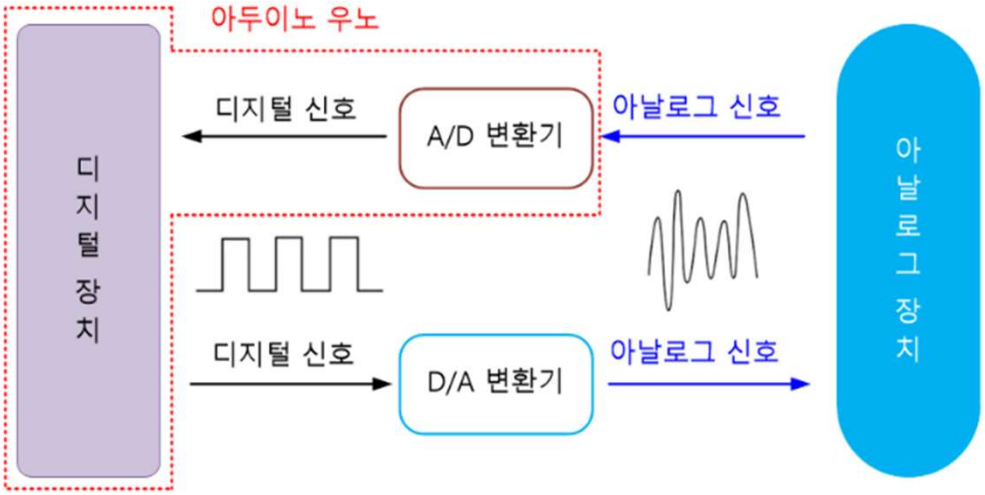
➢ 아날로그-디지털 변환회로(ADC)
자연계의 소리나 밝기를 디지털 장치 가 알아 들을 수 있도록 변환하는 회로입니다.



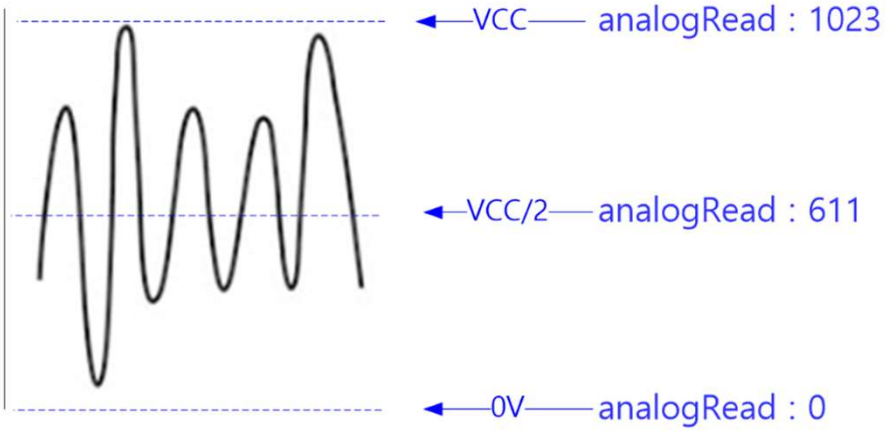
➢ 디지털-아날로그 변환회로(DAC)
디지털 장치 의 출력 값을 소리나 빛으로 전환하는 회로입니다.

➢ 아두이노 우노의 ADC

✓ ADC가 내장되어 있어서, 아날로그 신호를 입력 받을 수 있습니다.



✓ 물리적 전압 0 ~ 5V(Vcc) 를 디지털 장치(마이크로 프로세스)에서는 0 ~1023 영역으로 읽습니다.



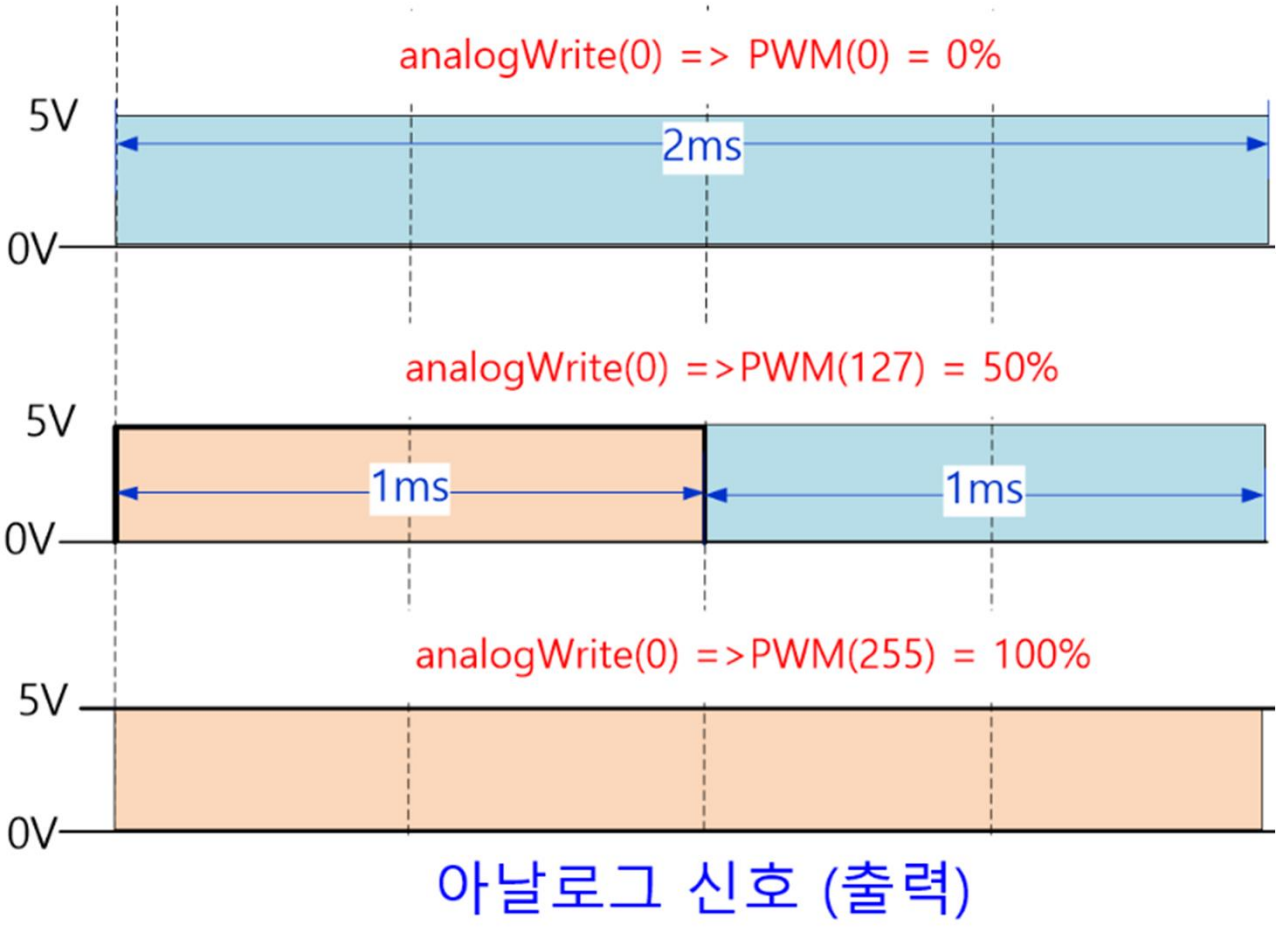
아날로그 신호 (입력)

➢ **아두이노 우노**의 DAC

✓ DAC 는 내장되어 있지 않아서, 펄스(PWM) 발생으로 아날로그 출력을 대신합니다.

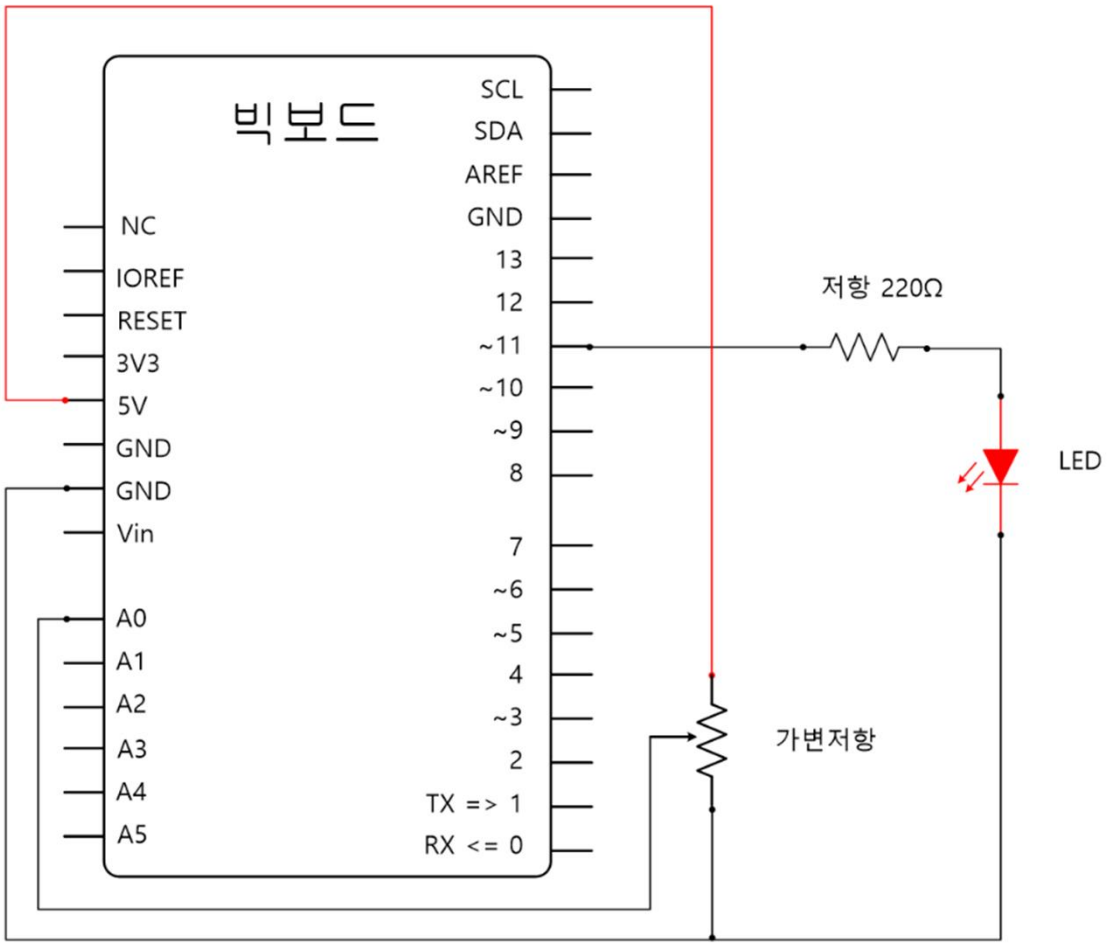
✓ 펄스 발생 값의 영역은 0 ~ 255 입니다.

✓ 펄스 발생 값이 255에 가까울 수록 LED는 밝기가 밝아지며, DC Motor는 속도가 빨라집니다.

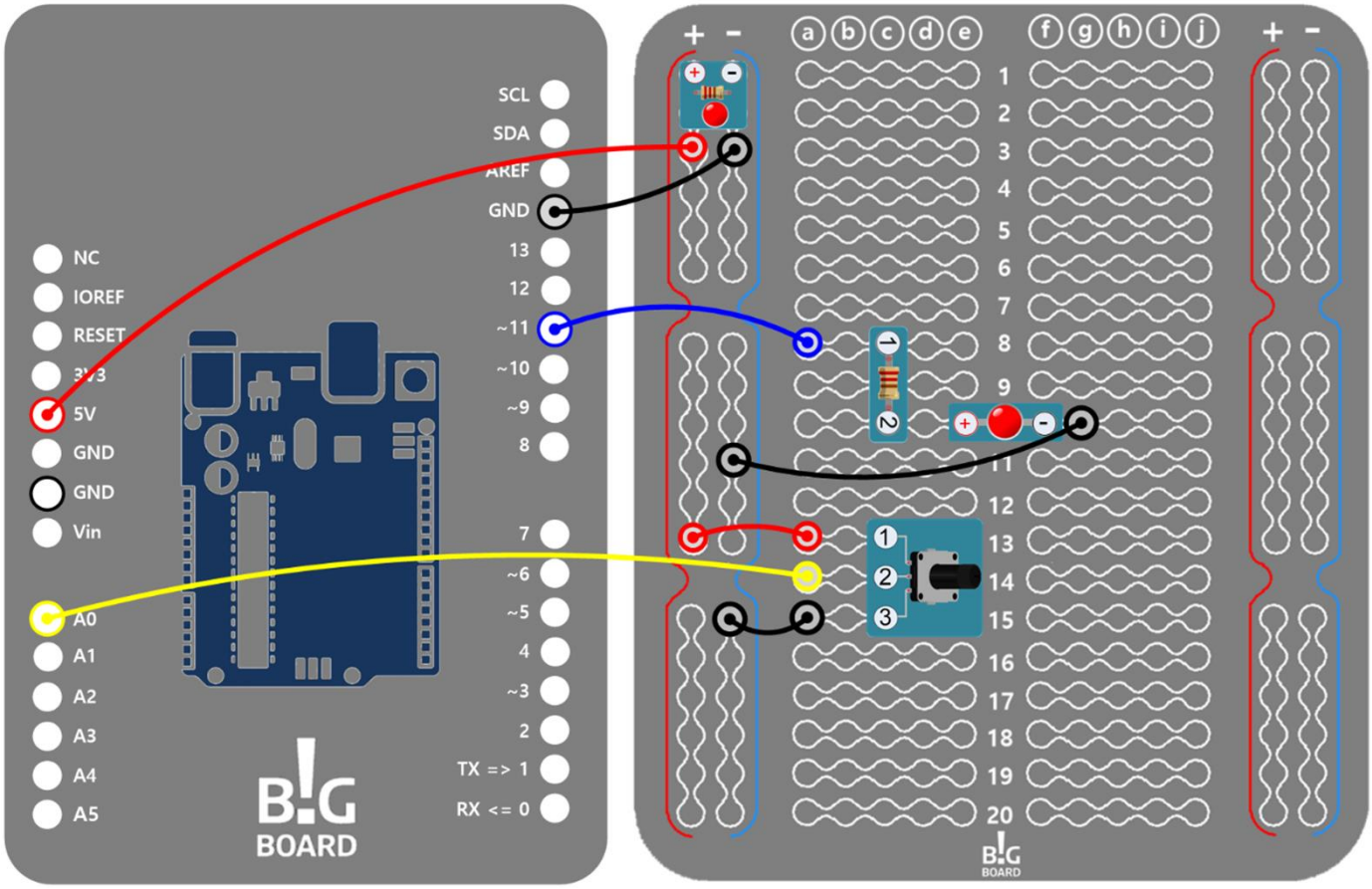


4) 회로 설명

미션) 가변 저항을 이용해서, LED 밝기를 제어하세요!



5) 하드웨어 연결 하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
+ -
a b c d e
f g h i j
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 6 - 가변저항으로 LED 밝기 제어하기 ; 설명문
*/
int VR = A0;
int LED = 11;
int SensorValue = 0;
```

/* 주석 처리 */

int VR = A0;

// int(정수 변수)의 이름을 VR 로 명명하고, 그 값을 A0 로 저장

int LED = 11;

// int(정수 변수)의 이름을 LED 로 명명하고, 그 값을 11으로 저장

int SensorValue = 0;

// int(정수 변수)의 이름을 SensorValue 로 명명하고, 그 값을 0으로 저장

6) 코딩하기-2

```
void setup() {
 pinMode(VR, INPUT);  // VR 핀을 입력으로 설정
 pinMode(LED, OUTPUT);  // LED 핀을 출력으로 설정
 }
```

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(VR, INPUT); // VR 핀을 입력으로 설정

pinMode(LED, OUTPUT); // LED 핀을 출력으로 설정

```
void loop() {
  SensorValue = analogRead(VR);
  int Speed = map(SensorValue,0,1023,0,255);
  analogWrite(LED,Speed);
}
```

void loop() { } // 계속 반복 하기

SensorValue = analogRead(VR);
// VR의 전압값을 아날로그 입력으로 읽음
int Speed = map(SensorValue,0,1023,0,255);
// int(정수 변수)의 이름을 Speed 로 명명하고,
// 그 값을 map함수를 이용해서, SensorValue 값 0 ~ 1023 을
// 비율적으로 0 ~ 255로 저장
// map(입력변수, 시작값, 종료값, 출력시작값, 출력종료값)
analogWrite(LED,Speed);
// 아날로그 출력으로 LED핀에 Speed 값을 출력

[동작]
가변저항에 의한 전압 변동 값을 아날로그 입력으로 읽고, 그 값을 map함수를 이용해서, 아날로그 출력으로 전환하여 LED 밝기를 제어합니다.

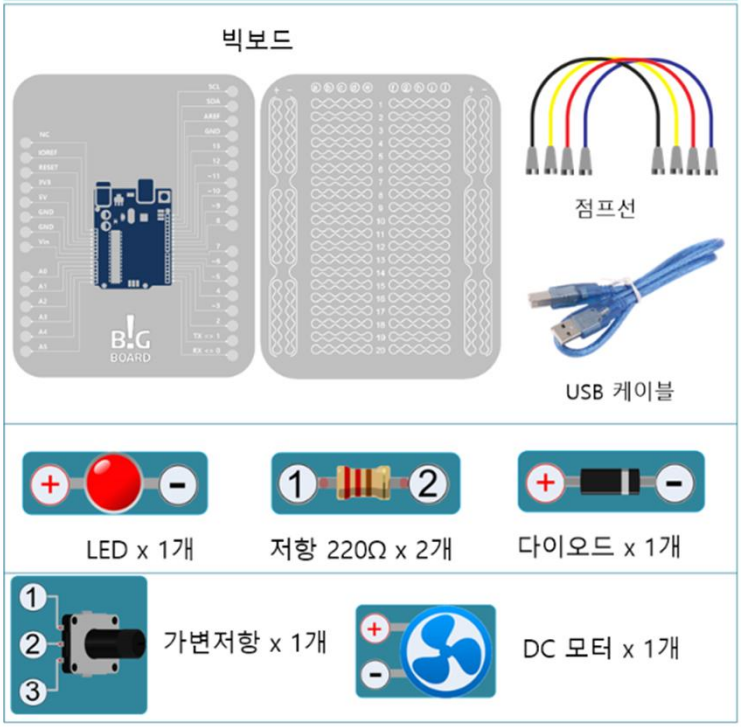
16 프로젝트7_가변저항으로 모터 속도 조절하기

1) 학습 목표

- ✓ DC 모터를 이해합니다.
- ✓ 모터 속도를 제어 합니다.

2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	LED 부품	1	빛을 발생하는 부품
3	저항 220Ω	1	전류량을 조절하는 부품
4	가변저항	1	입력 전압을 가변하는 부품
5	트렌지스터	1	전류량을 증폭하는 부품
6	다이오드	1	역전류 보호 회로
7	DC 모터	1	회전을 하는 부품



3) 핵심 이론

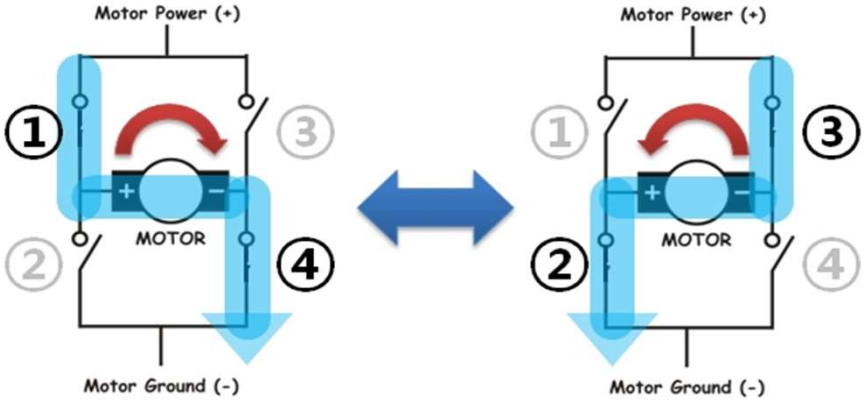
➢ DC 모터

✓ DC 모터는 2라인(전원, GND)만 연결하면 됩니다.

✓ DC 모터는 자동차의 뒷바퀴 회전체나 헬기의 프로펠러 회전체 로 많이 활용됩니다.

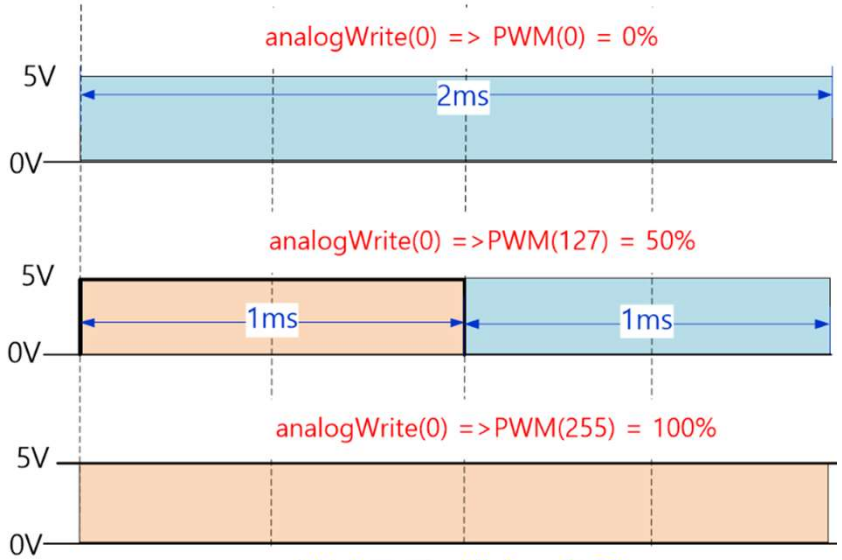
16 프로젝트7_가변저항으로 모터 속도 조절하기

✓ 모터는 입력해 주는 전원의 방향에 따라서, 순 / 역방향으로 변경됩니다.



✓ 모터의 속도는 코딩에서는 analogWrite()의 제어값 0 ~ 255 (가장 빠름)로 제어가 가능합니다.

✓ 회로 구현에서는 트랜지스터를 이용하여, 소모 전류가 높은 모터의 전류 량을 조절합니다.

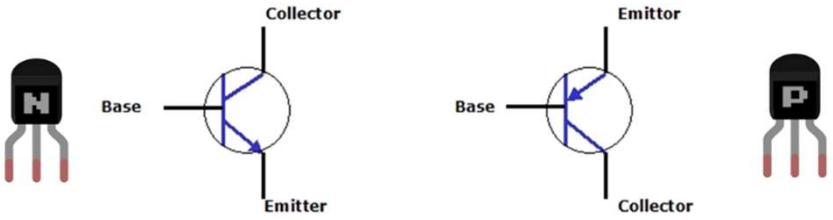


아날로그 신호 (출력)

➢ 트랜지스터

✓ 디지털 출력 핀은 DC 모터를 회전하기에는 공급 전류가 부족합니다. 이때 외부에 트랜지스터를 활용해서, 많은 전류 공급이 가능합니다.

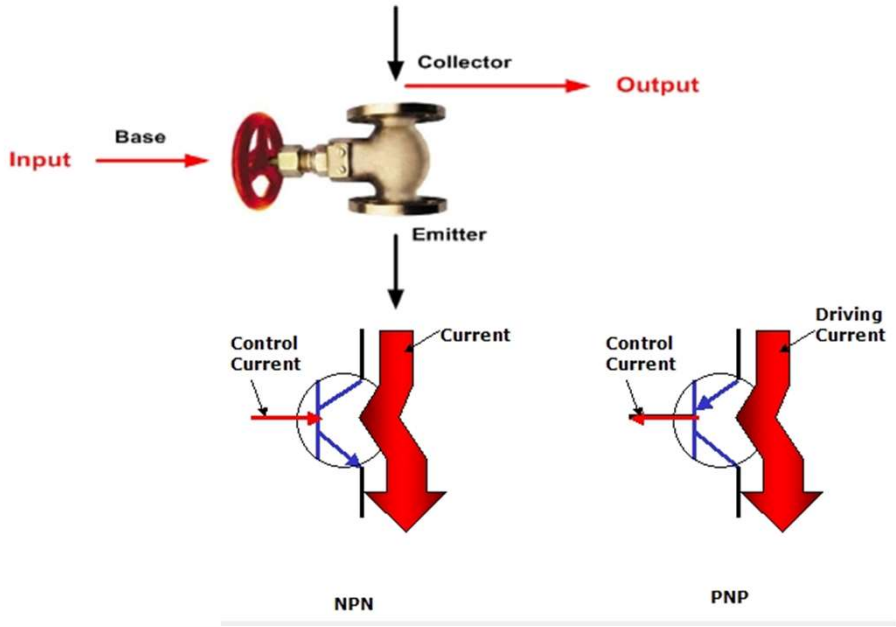
✓ 트랜지스터는 여러 종류가 있습니다.



NPN 트랜지스터 PNP 트랜지스터

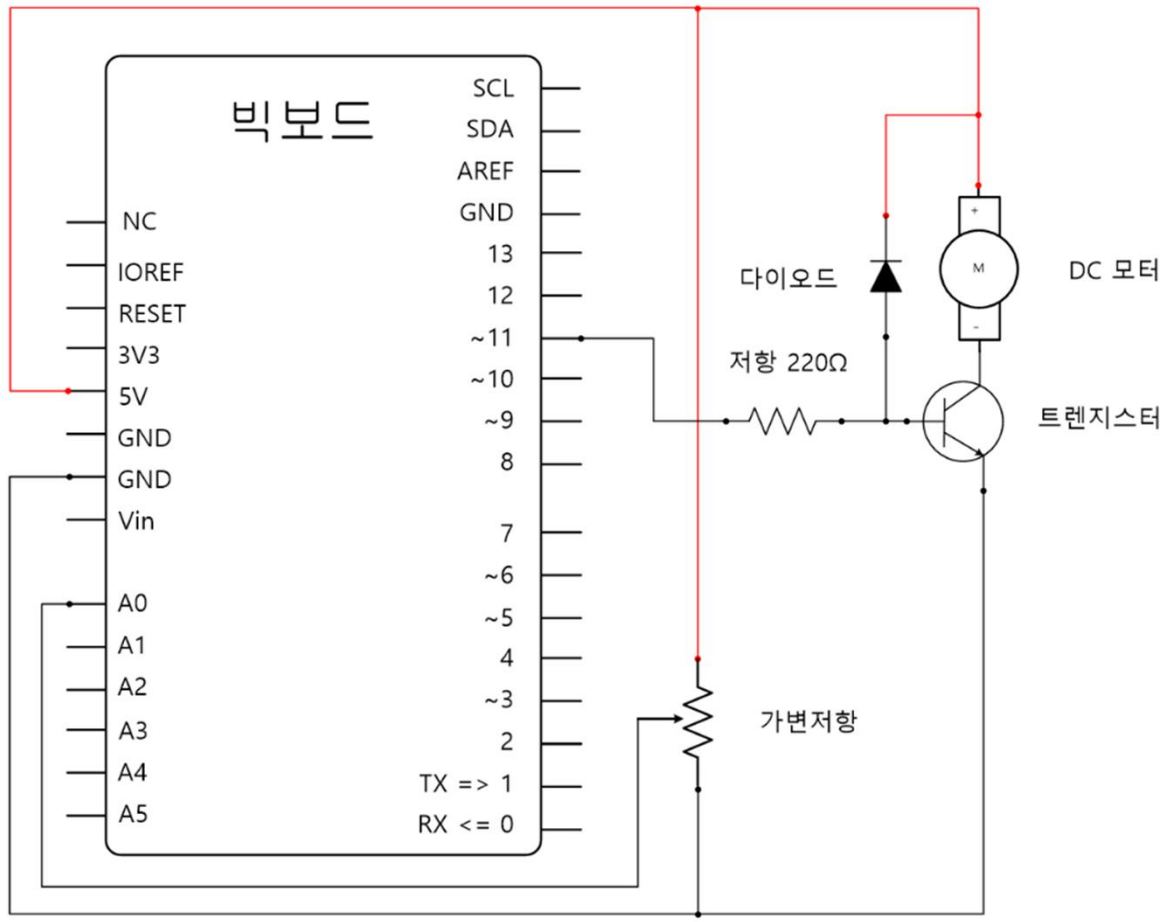
✓ 트랜지스터에는 이미터(emitter), 콜렉터(collector), 베이스(base)로 세 개의 단자가 있습니다.

✓ NPN 트랜지스터의 기본적인 동작 원리는 작은 전류가 베이스로 흘러 들어가면, 콜렉터에서 이미터로 큰 전류가 흐르는 것입니다. 그래서, 이를 증폭기라고 표현하기도 합니다.

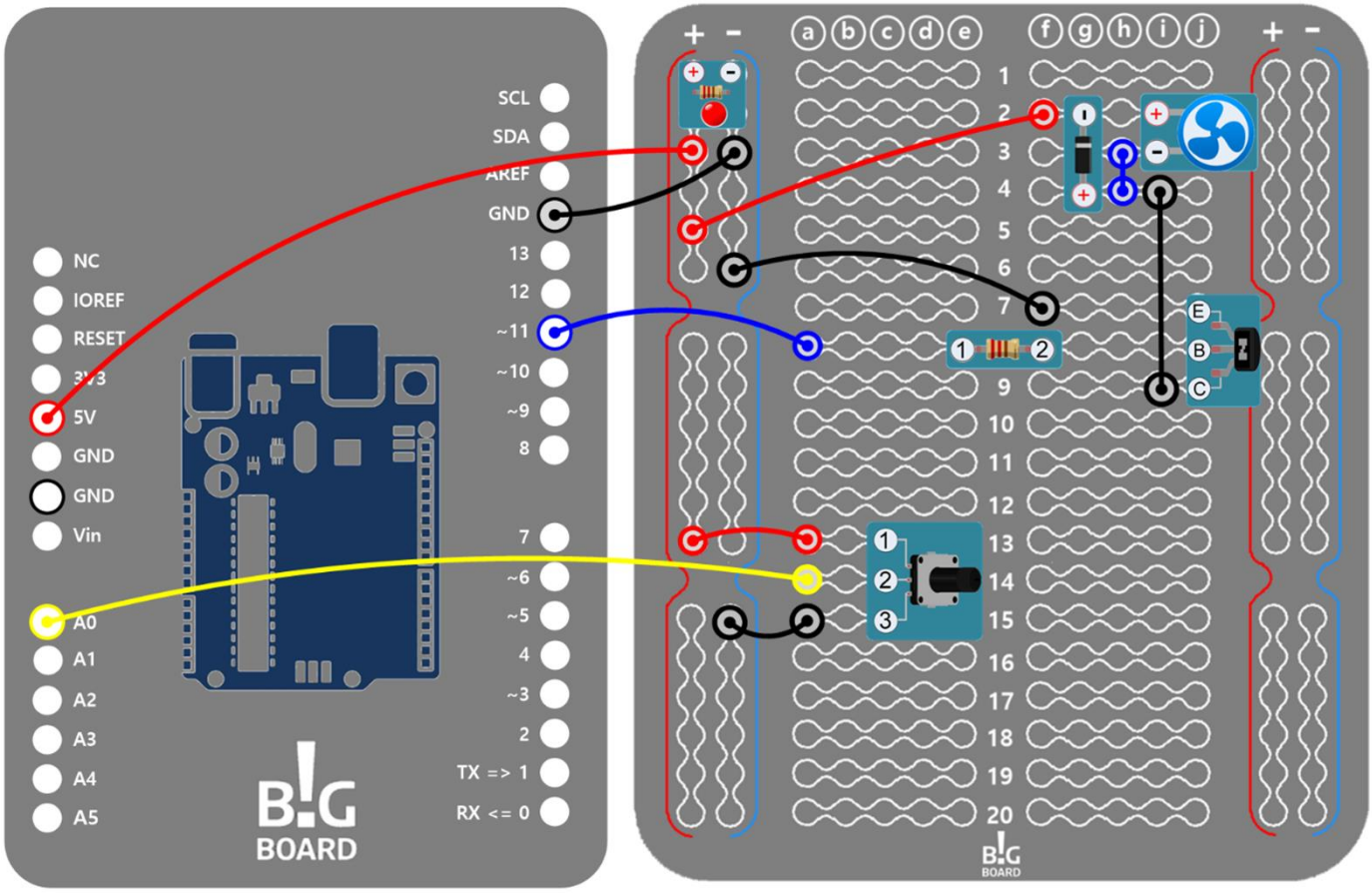


4) 회로 설명

미션) 가변 저항을 이용해서, 모터 속도를 제어하세요!



5) 하드웨어 연결 하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
9
10
11
12
13
14
15
16
17
18
19
20
E
B
C

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 7 - 가변저항으로 모터 속도 조절하기 ; 설명문
*/
int Enable = 11;
int Potent = A0;
int SensorValue = 0;
```

/* 주석 처리 */

int Enable = 11;

// int(정수 변수)의 이름을 Enable 로 명명하고, 그 값을 11 로 저장

int Potent = A0;

// int(정수 변수)의 이름을 Potent 로 명명하고, 그 값을 A0 로 저장

int SensorValue = 0;

// int(정수 변수)의 이름을 SensorValue 로 명명하고, 그 값을 0 으로 저장

6) 코딩하기-2

```
void setup() {
 pinMode(Enable, OUTPUT);
pinMode(Potent, INPUT);
}
```

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(Enable, OUTPUT);
// Enable (동작(속도) 제어) 핀을 출력으로 설정

pinMode(Potent, INPUT);
// Potent (가변 저항의 전압값) 핀을 입력을 설정

6) 코딩하기-3

```
void loop() {
 SensorValue = analogRead(Potent);
int Speed = map(SensorValue, 0, 1023, 0, 255);
analogWrite(Enable, Speed);
}
```

void loop() { } // 계속 반복 하기

SensorValue = analogRead(Potent);

// 가변저항 전압값을 아날로그 입력으로 읽어서, SensorValue에 저장

int Speed = map(SensorValue, 0, 1023, 0, 255);

// SensorValue 핀의 값 (0 ~ 1023)범위 내에서 읽고,

Speed에 (0 ~ 255)로 비례적으로 저장.

analogWrite(Enable, Speed);

// 아날로그 출력으로 Enable 핀에 Speed 값으로 출력

[동작]

가변저항에 의한 전압 변동 값을 아날로그 입력으로 읽고, 그 값을 map함수를 이용해서, 아날로그 출력으로 전환하여 , DC Motor의 속도를 제어합니다.

17 프로젝트8_ 컴퓨터에 출력하기

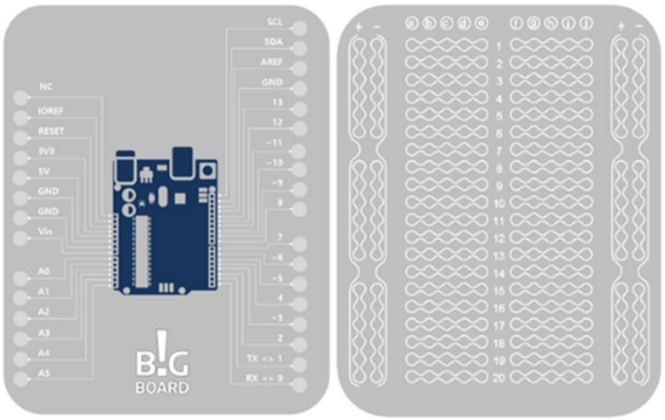
1) 학습 목표

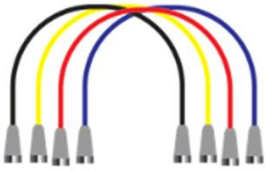
- ✓ 조도 센서를 이해합니다.
- ✓ 시리얼 모니터 출력을 이해합니다.

2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	조도 센서	1	빛의 밝기를 측정하는 부품
3	저항 10KΩ	1	밝기 측정 회로의 구성 저항

빅보드





점프선

USB 케이블

조도 센서 x 1개

저항 10KΩ x 1개

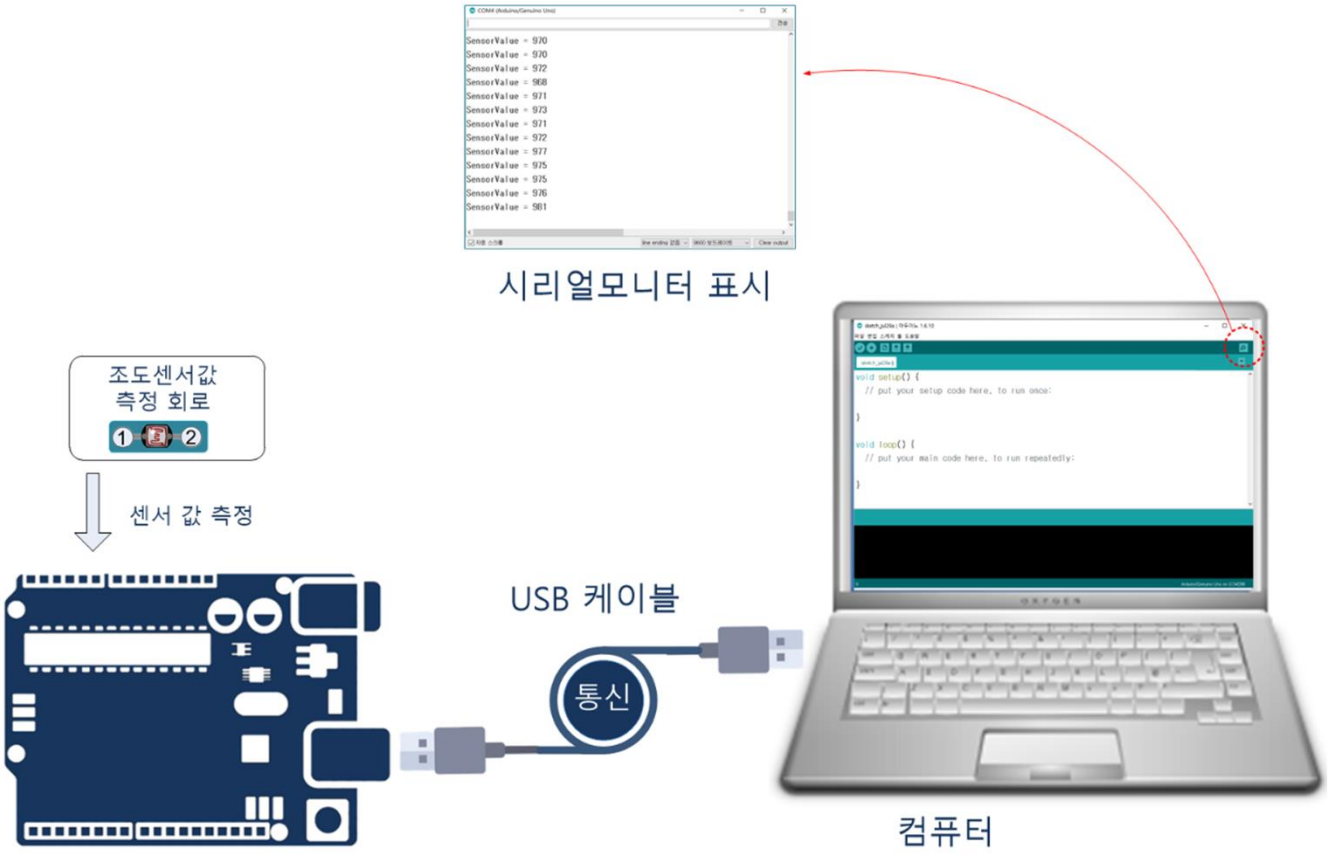
3) 핵심 이론

➢ 시리얼 통신 이해하기

✓ 컴퓨터(PC)와 보드간의 정보 교환을 위해 물리적으로 연결된 것을 시리얼 통신이라고 합니다.

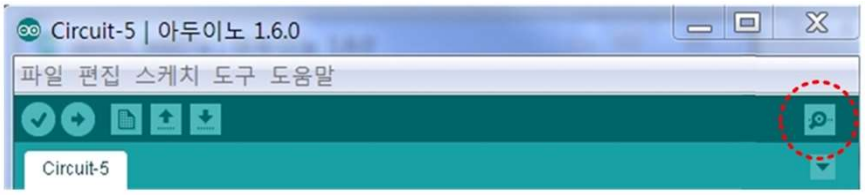
✓ 보드에 입력된 센서의 값을 PC와 연결된 USB 케이블을 통해 메시지를 받거나 명령을 통해, 보드를 제어를 할 수 있습니다.

✓ 그러므로, 보드의 상태를 디버깅 (오류나 상태 값을 파악) 하는 데 활용합니다.



17 프로젝트8_ 컴퓨터에 출력하기

✓ 스케치 프로그램 상단에  아이콘을 클릭하면, 시리얼 모니터 창이 열립니다.



✓ 동작은 아두이노와 PC간의 전송 속도를 하단 창에서 선택합니다. 9,600 ~ 115,200 bps까지 지원합니다.

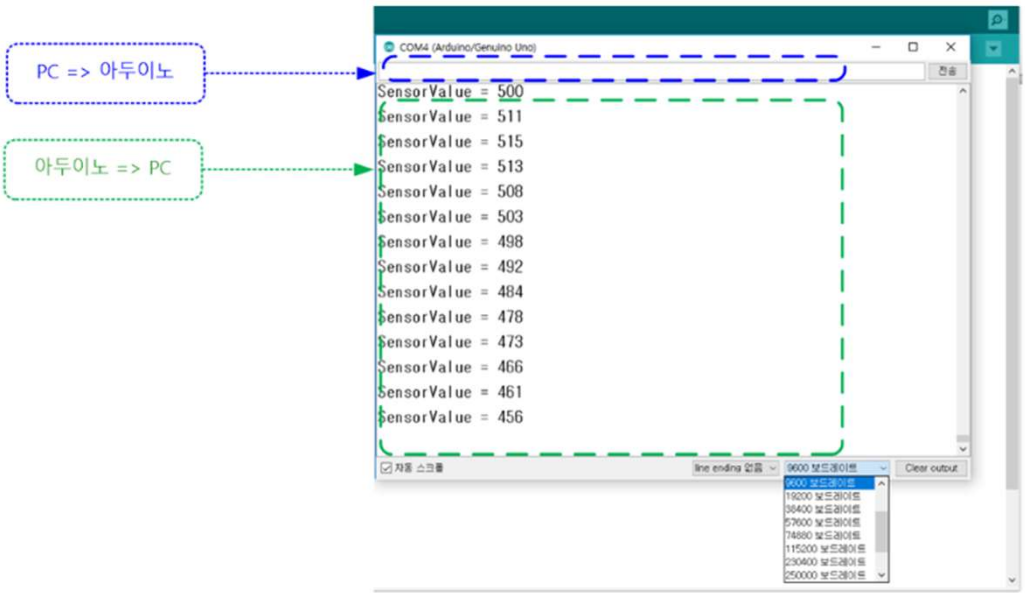
❖**주의 사항 : 시리얼 모니터 시작 할 때, 단말기는 자동 리셋 증상이 일어납니다. 그러므로, 사용 초기에 시리얼 모니터를 실행을 하세요!**

✓ PC에서 아두이노로 문자 보내기

시리얼 모니터의 상단 부분에 전송할 문자를 입력하고 '전송'이란 아이콘을 클릭하면, 문자가 전송됩니다.

✓ 아두이노에서 PC로 응답하기

시리얼 모니터의 중간 부분에 전달받은 문자를 표시합니다.



➢ 시리얼 통신 함수 이해하기

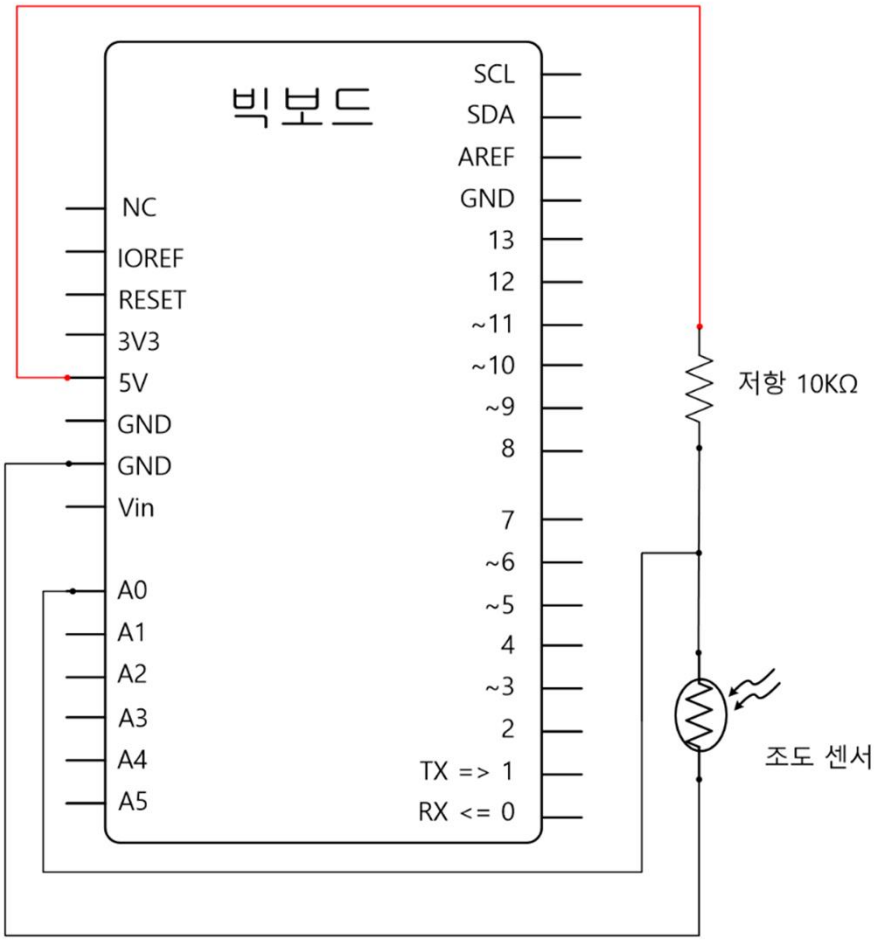
- 시리얼 통신 프로그램 설명
  - Serial.begin() : 시리얼 통신을 시작합니다.
    Serial.begin(baud) : 전송 속도를 설정할 수 있습니다.
    (9600) ~ (115200) ; 9,600 ~ 115,200bps
  - Serial.flush() : 시리얼 포트 안에 존재하는 데이터를 비웁니다.
  - Serial.end() : 시리얼 통신을 종료합니다.

- 아두이노 => PC : 아두이노에서 PC로 데이터 전송
  - Serial.print(val) : 데이터를 아스키 코드 형식에 맞추어서 출력
  - Serial.print("문자") : ""안의 문자를 출력
  - Serial.print(val,DEC) : DEC (DEC/HEX/OCT) 로 출력
  - Serial.println(val) : 데이터를 아스키 코드 형식에 맞추어서 출력하며, 줄 바꿈을 합니다.
  - Serial.available() : 아두이노로 부터 받은 정보가 있는지를 확인하여, 읽은 정보의 Byte 숫자를 리턴합니다.

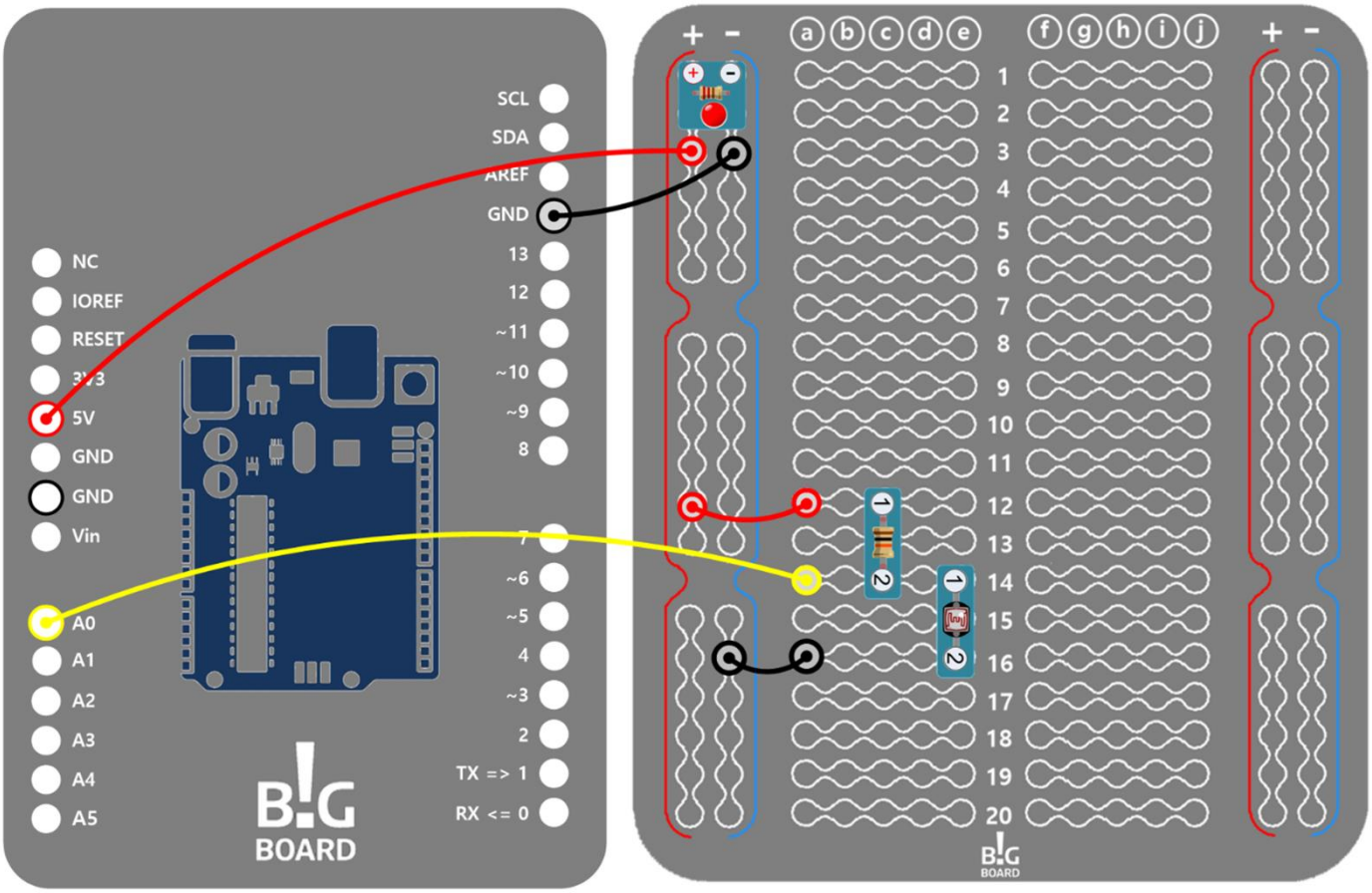
- PC => 아두이노 : PC에서 아두이노로 데이터 전송
  - Serial.read() : PC에서 입력한 정보를 아두이노로 전송합니다.

4) 회로 설명

미션) 조도 센서 값을 측정해서, 컴퓨터 화면에 출력하세요!



5) 하드웨어 연결 하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

6) 코딩하기-1

```
/*
  Bigboard-Starter-kit
  Project 8 – 컴퓨터에  출력하기 ; 설명문
*/
int CDS = A0;
int SensorValue = 0;
```

/* 주석 처리 */

int CDS = A0;

// int(정수 변수)의 이름을 CDS 로 명명하고, 그 값을 A0 로 저장

int SensorValue = 0;

// int(정수 변수)의 이름을 SensorValue 로 명명하고, 그 값을 0으로 저장

6) 코딩하기-2

```
void setup() {
 pinMode(CDS,  INPUT);  // CDS 핀을 입력으로 설정
 Serial.begin(9600);
 }
```

void setup() { 코드 } // 초기 한번 설정하는 영역으로
보통은 이 영역에 환경 설정을 합니다.

pinMode(CDS, INPUT);
// CDS 핀을 입력으로 설정

Serial.begin(9600);
// 시리얼 통신을 속도 9600bps 으로 시작합니다.

6) 코딩하기-3

```
void loop() {
SensorValue = analogRead(CDS);
Serial.print("SensorValue = ");
Serial.println(SensorValue);
delay(100);
}
```

void loop() { } // 계속 반복 하기

SensorValue = analogRead(CDS);

// CDS의 센서 값을 아날로그 입력으로 읽음

Serial.print("SensorValue = ");

// 시리얼 모니터에 . " " 안의 글자를 출력합니다.

SensorValue = 는 계속 출력합니다.

Serial.println(SensorValue);

// 시리얼 모니터에 SensorVaule 값을 출력하고, 줄 바꾸기를 합니다.

[동작]

조도 센서의 값을 읽어서, 시리얼 모니터로 출력 합니다.

18 프로젝트9_온/습도 값을 출력하기

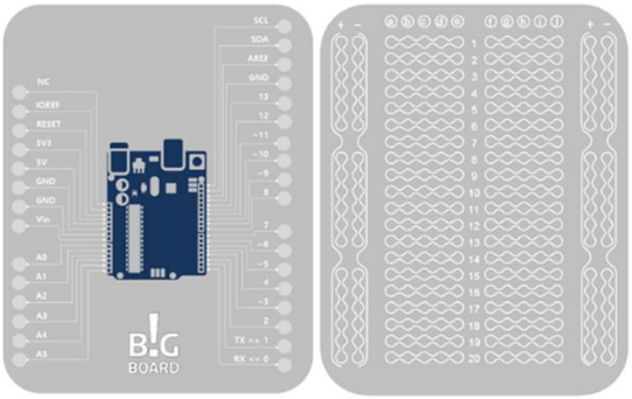
1) 학습 목표

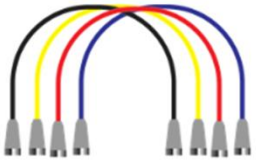
- ✓ 온/습도 센서를 이해합니다.
- ✓ 데이터 통신을 이해합니다.

2) 준비물

No	부품명	수량	자세한 설명
1	빅보드	1	아두이노 우노 호환
2	온습도센서	1	온도/습도 측정 센서

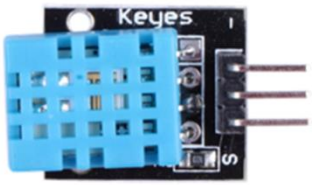
빅보드





점프선

USB 케이블



온습도센서 (DHT11)

18 프로젝트9_온/습도 값을 출력하기

3) 핵심 이론

➢ 온/습도 센서 이해하기

✓ DHT11은 온도와 상대 습도를 측정해 주는 센서로 장시간 사용해도 변화율이 낮은 제품입니다.

✓ 온도는 2℃, 습도는 ± 5%의 오차 범위를 갖습니다.

✓ 온도 센서는 에어컨, 보일러, 냉장고 등 많은 곳에서 사용합니다.

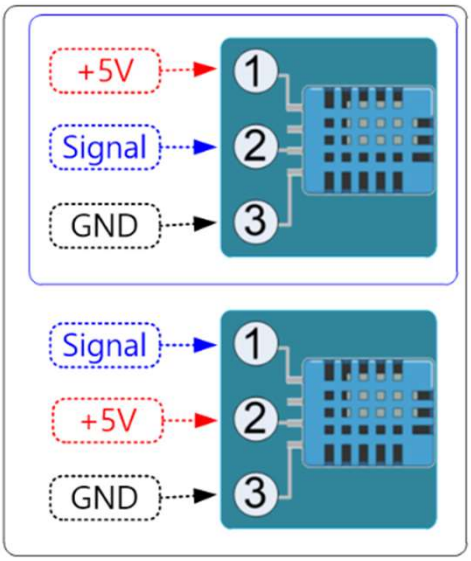
✓ 습도 센서는 하우스 작물 재배, 병원, 동물원 등 많은 곳에서 사용합니다.

18 프로젝트9_온/습도 값을 출력하기

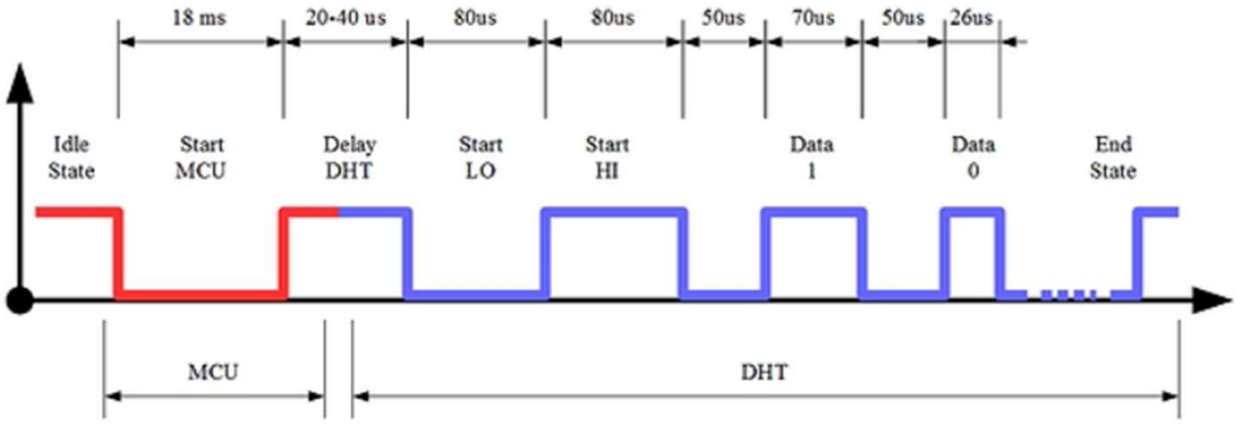
✓ DHT11 제품은 2가지 핀 맵이 있으므로 주의하세요!

✓ 여기서는 5V / Signal / Gnd 핀 맵으로 설명을 합니다.

[부품 방향 주의]

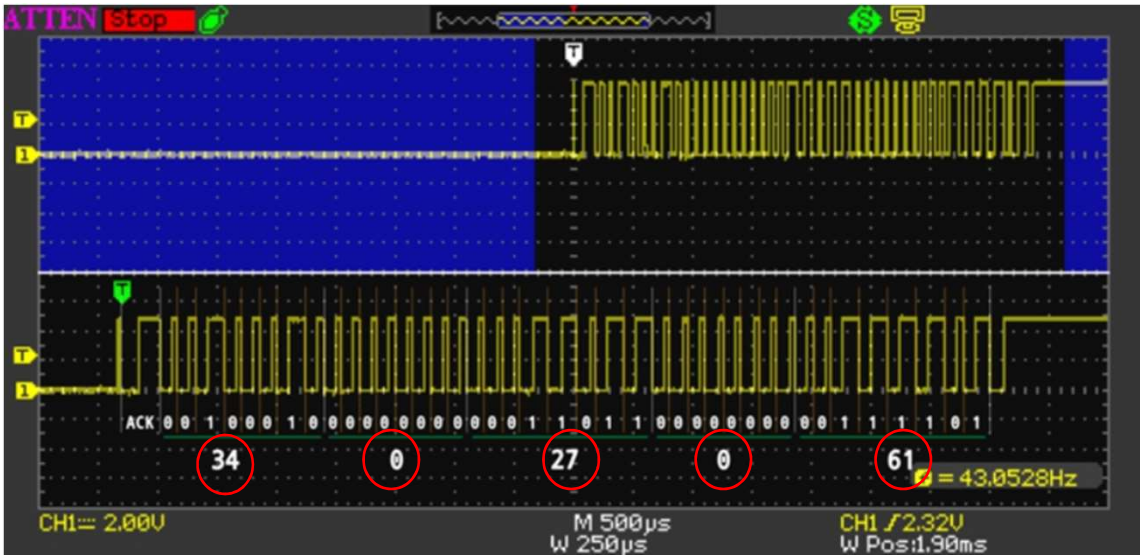


✓ Signal 핀의 출력 데이터 형태



18 프로젝트9_온/습도 값을 출력하기

➢ 실제 디지털 스코프 장비로 측정한 화면



➢ 온/습도 센서 (DHT11) 의 신호 데이터 분석

✓ 습도의 정수(Humidity Integer) : 8bit = 34

✓ 습도의 소수(Humidity Fraction) : 8bit = 0

✓ 온도의 정수(Temperature Integer) : 8bit = 27

✓ 온도의 소수(Temperature Fraction) : 8bit = 0

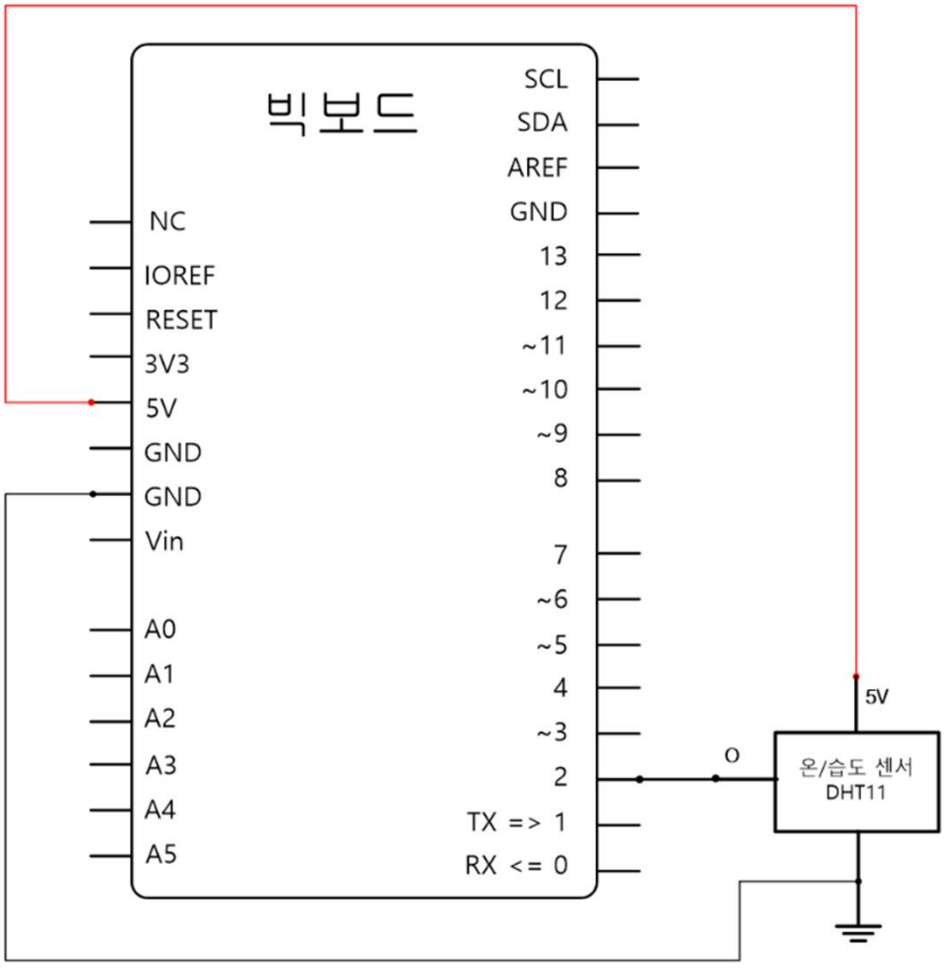
✓ Check sum : 8bit = 습도의 정수 + 소수 + 온도의 정수 + 소수

= 34 + 0 + 27 + 0 = 61 (값이 일치하므로, 유효값)

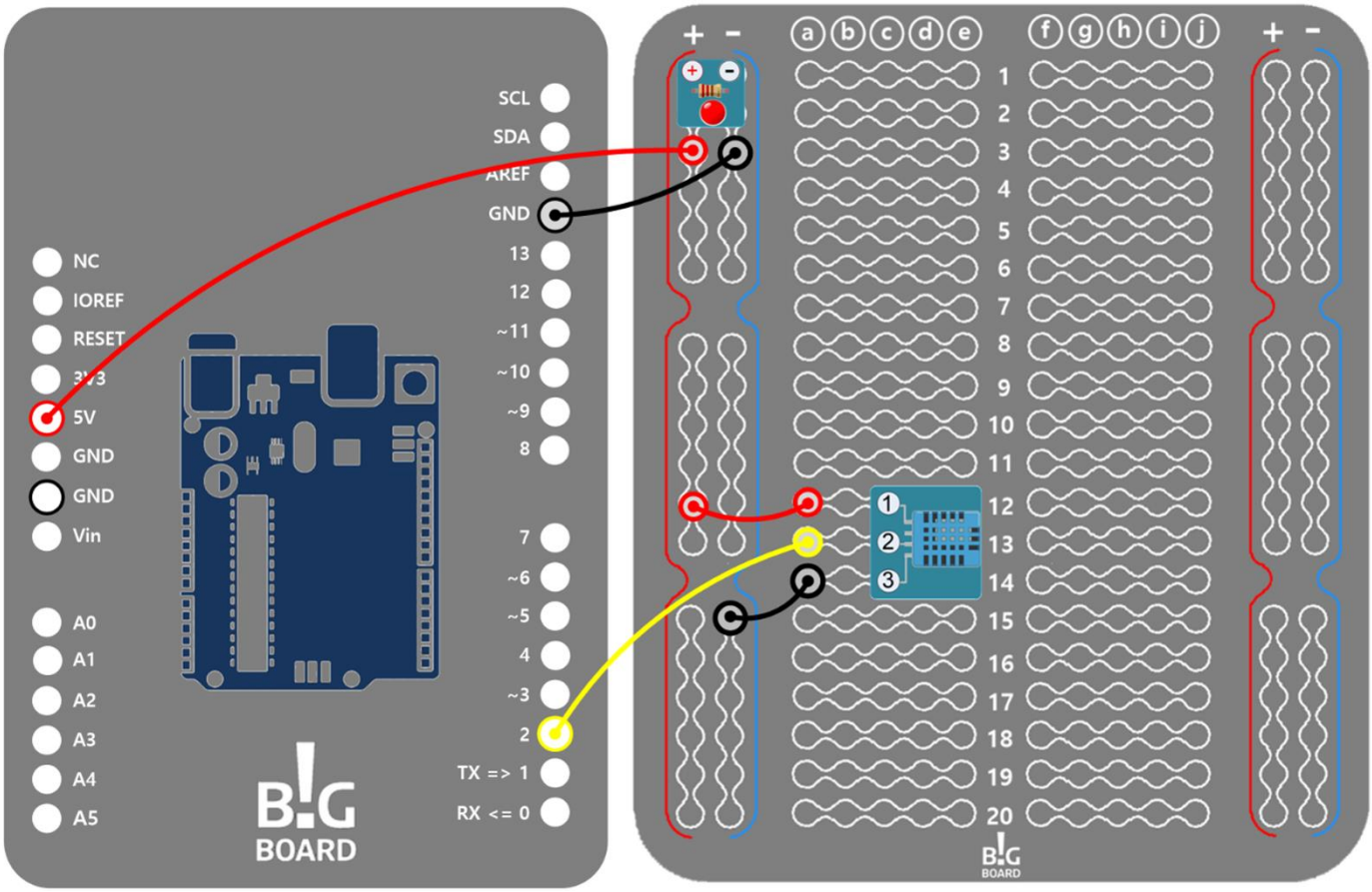
=> 습도 : 34.0%, 온도 : 27.0℃

4) 회로 설명

미션) 온/습도 센서의 값을 컴퓨터 모니터에 출력하세요!



5) 하드웨어 연결하기


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX => 1
RX <= 0
NC
IOREF
RESET
3V3
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
BOARD
a b c d e
f g h i j
1
2
3
BOARD

18 프로젝트9_온/습도 값을 출력하기

6) 코딩하기-1

```
/*  주석문 */
int DHpin = 2; //정수 변수 명을 DHpin으로 명명하고, 2 값을 저장
byte dat [5]; // 길이가 5byte인  배열의 이름을  dat로 명명함

byte read_data () {
byte data;
 for (int i = 0; i < 8; i ++) {
  if (digitalRead (DHpin) == LOW) {
   while (digitalRead (DHpin) == LOW); // wait for 50us
   delayMicroseconds (30); // determine the duration of the high level to
determine the data is '0 'or '1'
   if (digitalRead (DHpin) == HIGH)
    data |= (1 << (7-i)); // high front and low in the post
   while (digitalRead (DHpin) == HIGH); // data '1 ', wait for the next one
receiver
  } }
return data;
}
// 온/습도센서로 부터 받은 데이터를 분석하는 함수  코딩
```

6) 코딩하기-2

```
void start_test () {
  digitalWrite (DHpin, LOW); // bus down, send start signal
  delay (30);
digitalWrite (DHpin, HIGH);
  delayMicroseconds (40); // Wait for DHT11 response
  pinMode (DHpin, INPUT);
  while (digitalRead (DHpin) == HIGH);
  delayMicroseconds (80); // DHT11 response, pulled the bus 80us
  if (digitalRead (DHpin) == LOW);
  delayMicroseconds (80);
for (int i = 0; i < 4; i ++) // receive temperature and humidity data, the parity
bit is not considered
    dat[i] = read_data ();

  pinMode (DHpin, OUTPUT);
  digitalWrite (DHpin, HIGH);
}
```

// 아두이노 보드에서 온/습도 센서로 신호 시작을 제어하는 함수

6) 코딩하기-3

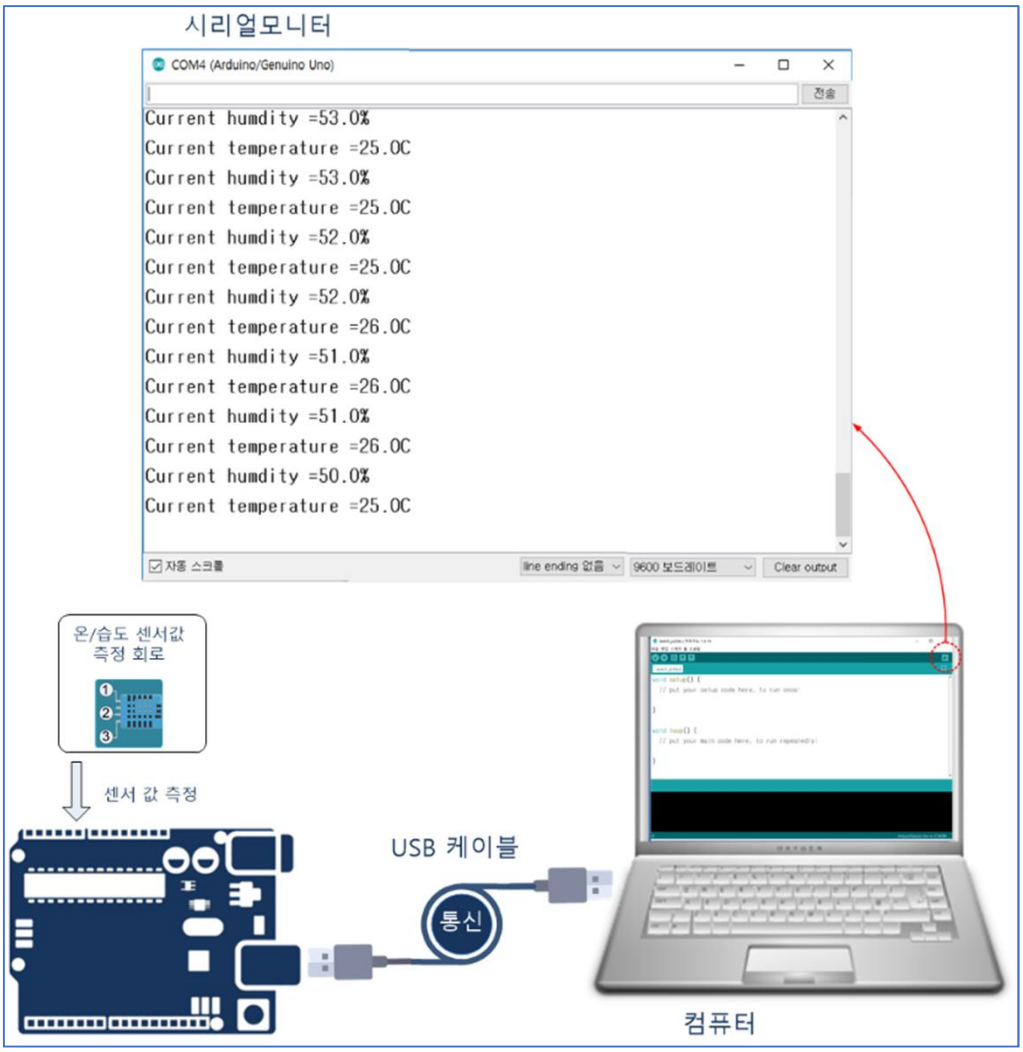
```
void setup () {
  Serial.begin (9600); // 시리얼 통신을  속도 9600bps 으로 시작
  pinMode (DHpin, OUTPUT); // Dhpin을 출력으로 설정
}
void loop () {
  start_test ();
  Serial.print ("Current humdity =");
  Serial.print (dat [0], DEC); // display the humidity-bit integer;
  Serial.print ('.');
  Serial.print (dat [1], DEC); // display the humidity decimal places;
  Serial.println ('%');
  Serial.print ("Current temperature =");
  Serial.print (dat [2], DEC); // display the temperature of integer bits;
  Serial.print ('.');
  Serial.print (dat [3], DEC); // display the temperature of decimal places;
  Serial.println ('C');
  delay (700);
}
```

[동작]

온/습도 센서의 값을 시리얼 모니터에 출력을 합니다.

18 프로젝트9_온/습도 값을 출력하기

7) 시리얼 모니터 출력하기

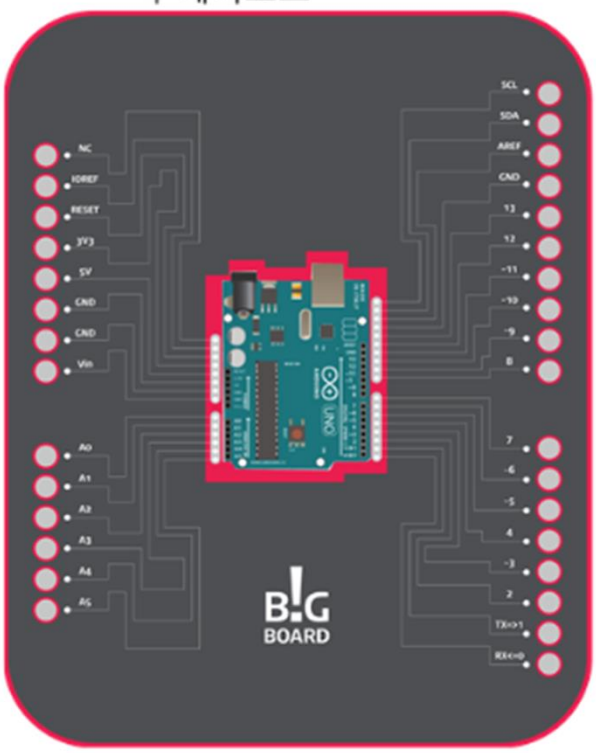


참고사항

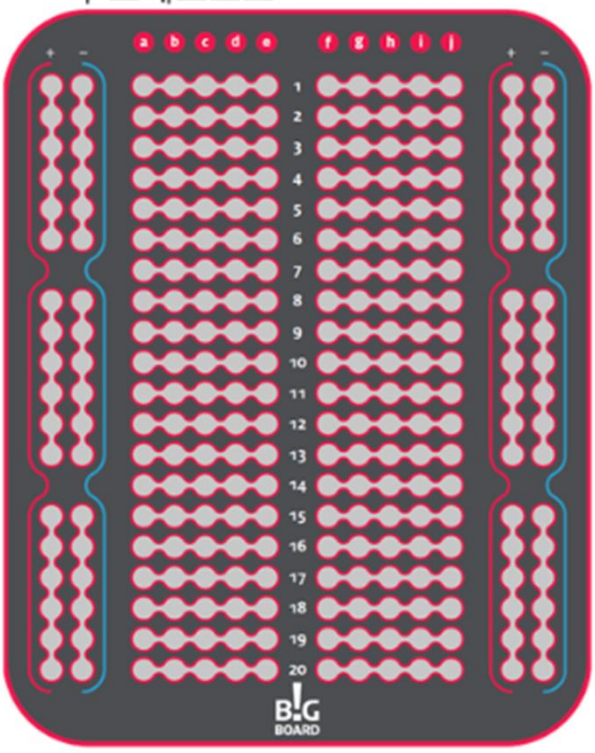
본 교재는 스케치와 빅보드의 연동과 관련된 내용을 소개하는 정도의 프로젝트 예제를 준비했습니다.

스케치의 재미있고, 다양한 콘텐츠는 아두이노 사이트를 참조하시길 바랍니다. https://www.arduino.cc/

빅-제어보드



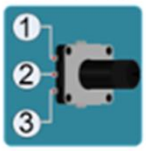
빅-브레드보드



LED x 5개

전원 LED

가변저항



저항 220 x 5개

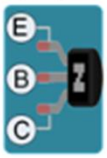
부저

저항 1K x 2개

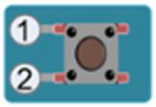
DC 모터

저항 10K x 2개

트렌지스터



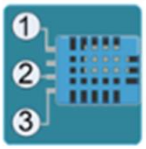
버튼



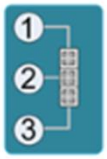
다이오드

조도센서

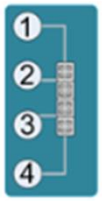
온습도센서



2핀 소켓



3핀 소켓



4핀 소켓

Bigboard starter Kit

빅보드와 함께 하는 코딩놀이

[네이버 카페]
http://cafe.naver.com/arduinobigboard
자세한 정보는 네이버카페(아두이노빅보드)에서 제공합니다.

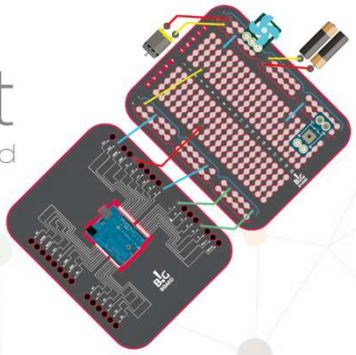
NAVER 카페 아두이노빅보드 검색

(주)비아이에스웍스, 블루이노 공동 지음

빅보드 고객센터

총판 : (주)비아이에스웍스
홈페이지 : www.bisworks.co.kr
전화 : 070-4369-6005
주소 : 경기도 성남시 분당구 성남대로171번길 17, 씨티밸리오피스텔 619호
(주) 비아이에스웍스
이메일 : help@bisworks.co.kr
제조사 : 블루이노

Bigboard starter Kit



BIG BOARD