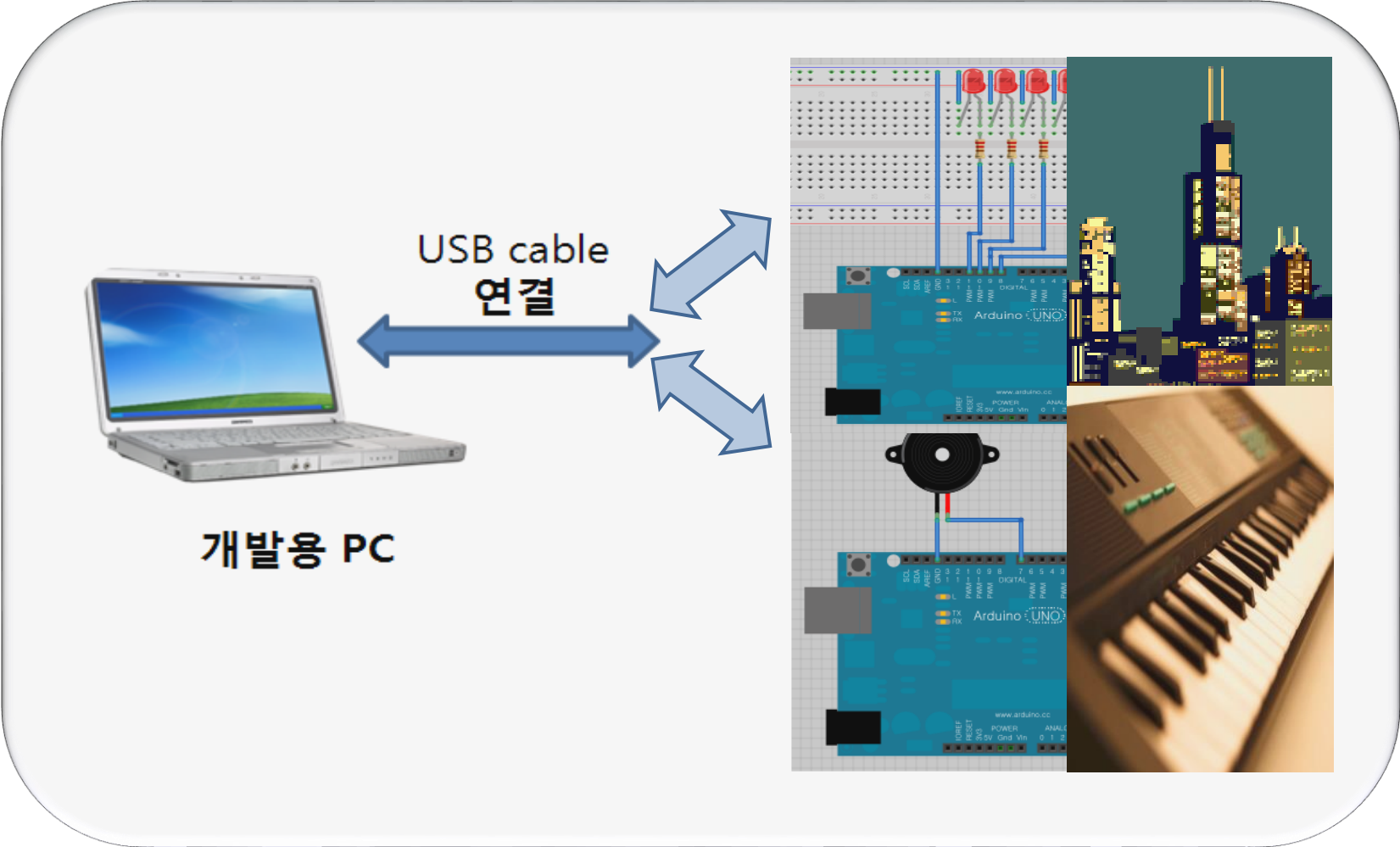


# 아두이노 보드로 제어하기

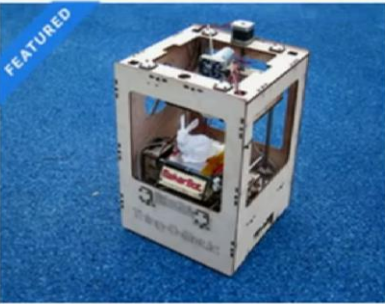


# 수업내용

- Arduino 소개
- Arduino 개발환경 구축하기
- Arduino로 제어하기

# 아두이노 소개

Digital designs for real, physical objects. A Universe of Things!



## Thing-O-Matic 3D Printer



Created by [makerbot](#)

Created on Dec 9, 2010

Featured on Jan 24, 2011

Fully Automatic

With the automated build platform it printing.

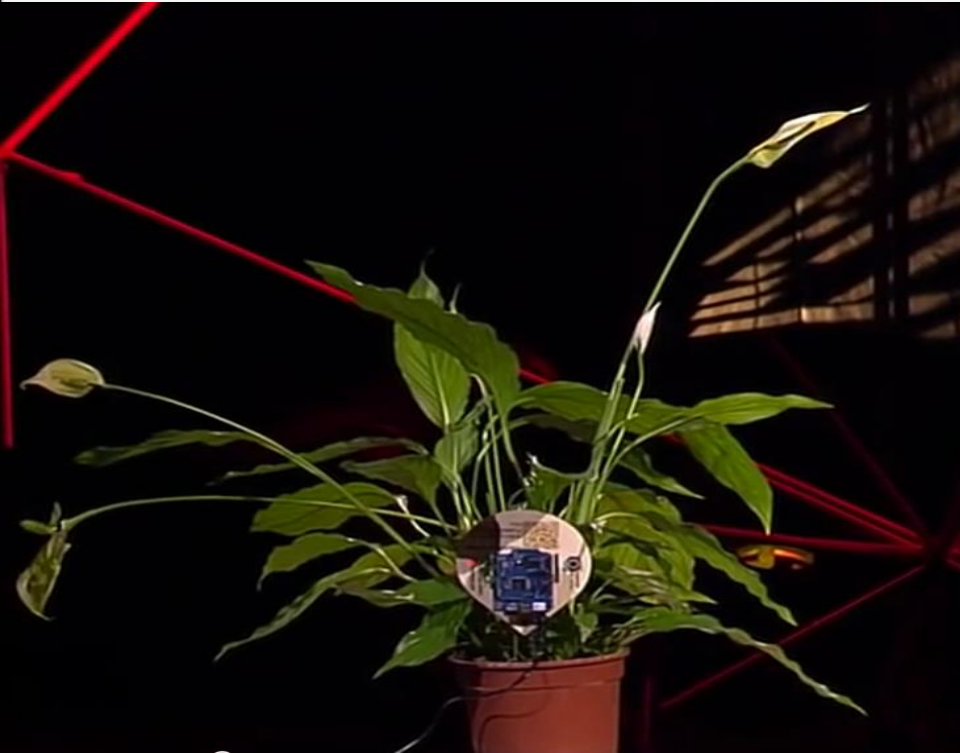
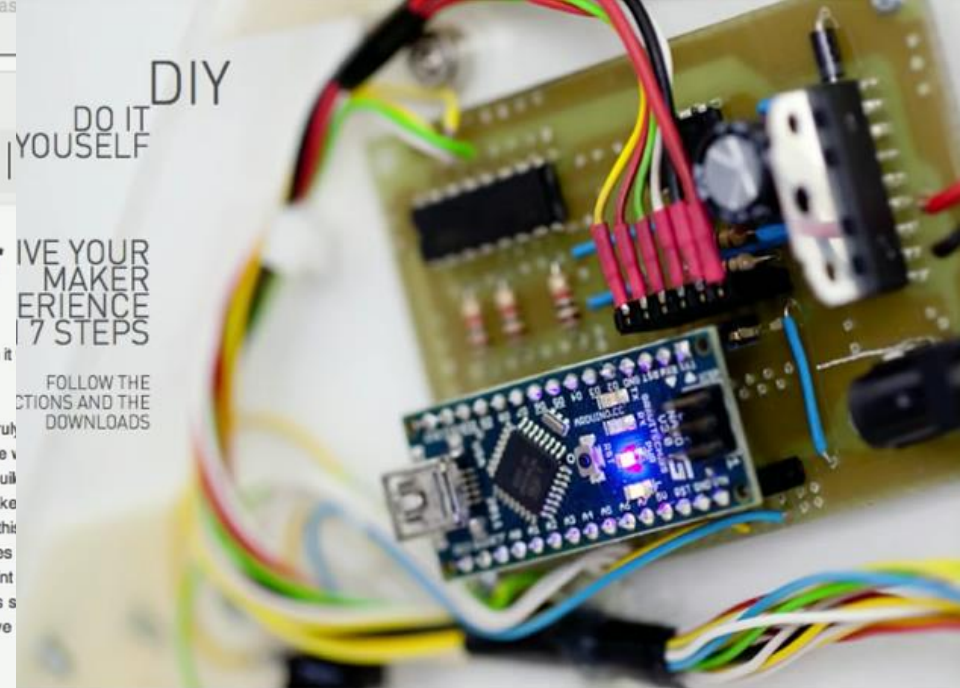
The automated build platform is truly it allows you to have a print queue v clear the build surface between buil after another. This turns your Make sits on your desktop. Best of all, thi you hit print, and the machine does 100 butterflies? Easy. Want to print problem. Want to start a business s you've designed? Awesome. Have you design new things.

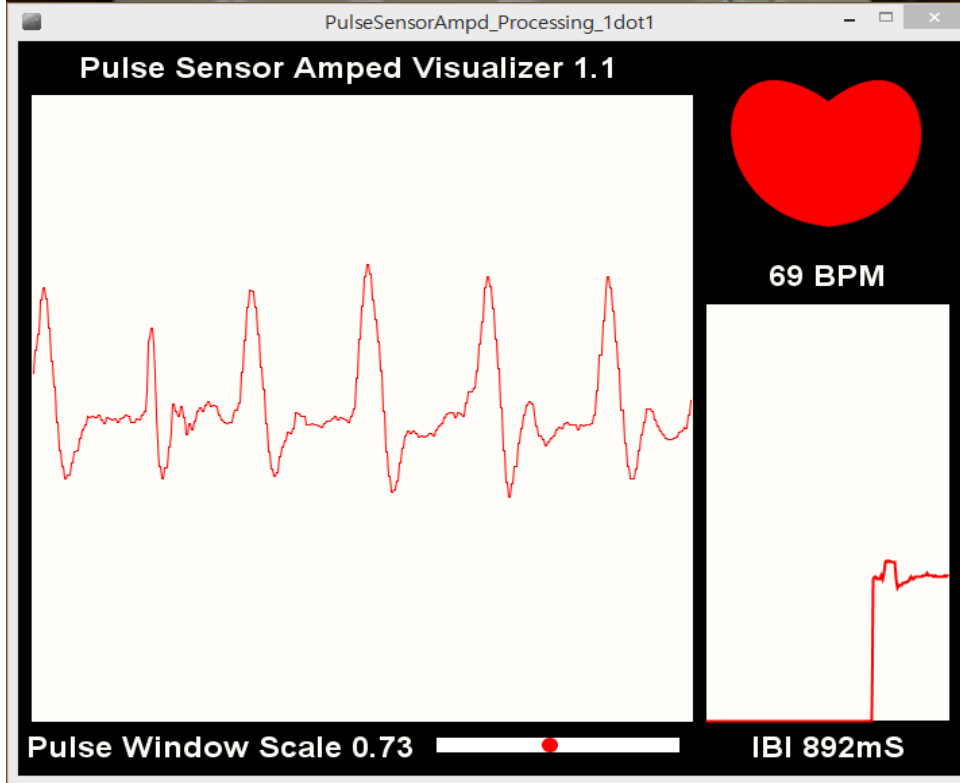
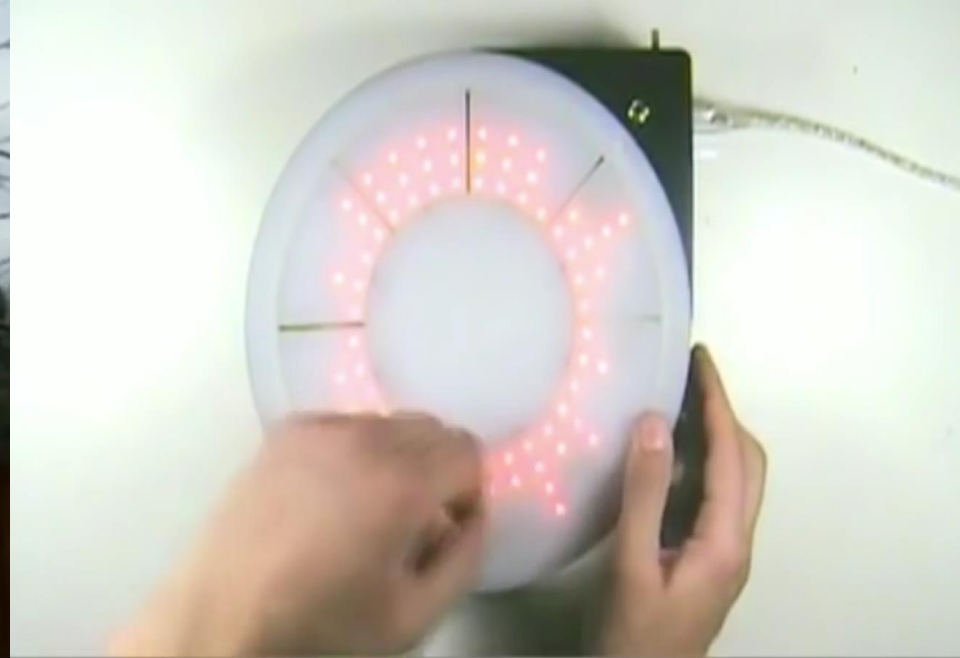


DIY  
DO IT  
YOURSELF

GIVE YOUR  
MAKER  
EXPERIENCE  
7 STEPS

FOLLOW THE  
INSTRUCTIONS AND THE  
DOWNLOADS





**instructables** share what you make

sign up or login or [Log In](#)

[Arduino](#) [ACE QUICK FIXES](#) [Woodworking](#)


Create Explore Contests More

Google Custom Search


all photos step by step video ebooks guides answers contests

editors' picks popular recent rating views zeitgeist uncommented random


- All
- Food
- Living
- Outside
- Play
- Technology
- 3D Printing
- Apple
- Arduino
- Art
- Assistive Tech
- Audio
- Cell Phones




AC Arduino dimming circuit  
by gabriellelevine



Honey Bee Counter  
by hydronics



Gas detector / indicator (USB powered) with arduino



# 아두이노 특징

- USB를 통해 컴파일 및 업로드를 쉽게 할 수 있음
- 초보 수준의 사용자에서부터 전문적인 지식을 갖추고 복잡한 기능을 구현하기를 원하는 개발자들도 사용할 있도록 다양한 기능들을 가지고 있고 확장성도 제공함
- 하드웨어와 소프트웨어가 무료인 오픈 시스템-아두이노 보드의 회로도가 공개되어 있으므로, 누구나 직접 보드를 만들고 수정 가능, 아두이노 보드를 사용한 많은 예제 코드도 공개됨
- 비교적 가격이 저렴하고, Window를 비롯해 맥 OS X, 리눅스와 같은 여러 OS를 모두 지원

# Sketch(아두이노 프로그램)

- 아두이노 보드를 구동해주는 드라이버 외에 아두이노 보드의 기능을 실현하여 주는 응용 s/w
- 초기에 주로 아티스트와 디자이너가 사용했기 때문에 '아이디어를 빠르고 쉽게 실현할 수 있는 방법'이라는 의미로 '스케치'라는 용어가 사용됨
- 스케치(sketch)와 프로그램(program)이라는 용어는 같은 의미로 사용

# 하드웨어의 종류



Arduino Due



Arduino Leonardo



Arduino Uno R3



Mega Pro Mini 3.3



Pro Micro



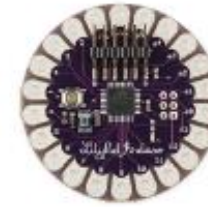
RedBoard



Arduino Mega 2560 R3



Arduino Pro



LilyPad Main



Lily Pad Simplified



Mega Pro 5V



Arduino Fio

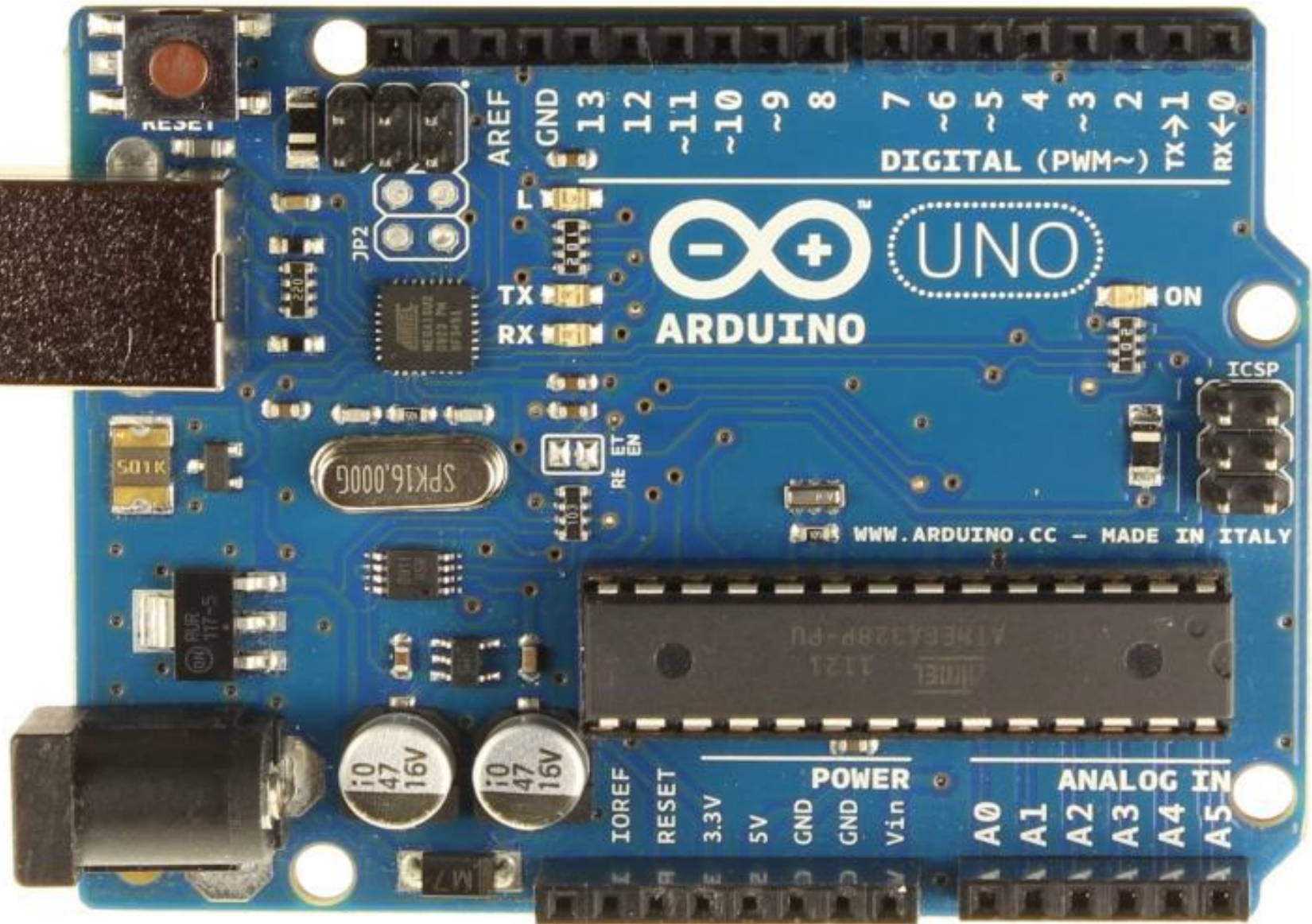


Arduino Mini

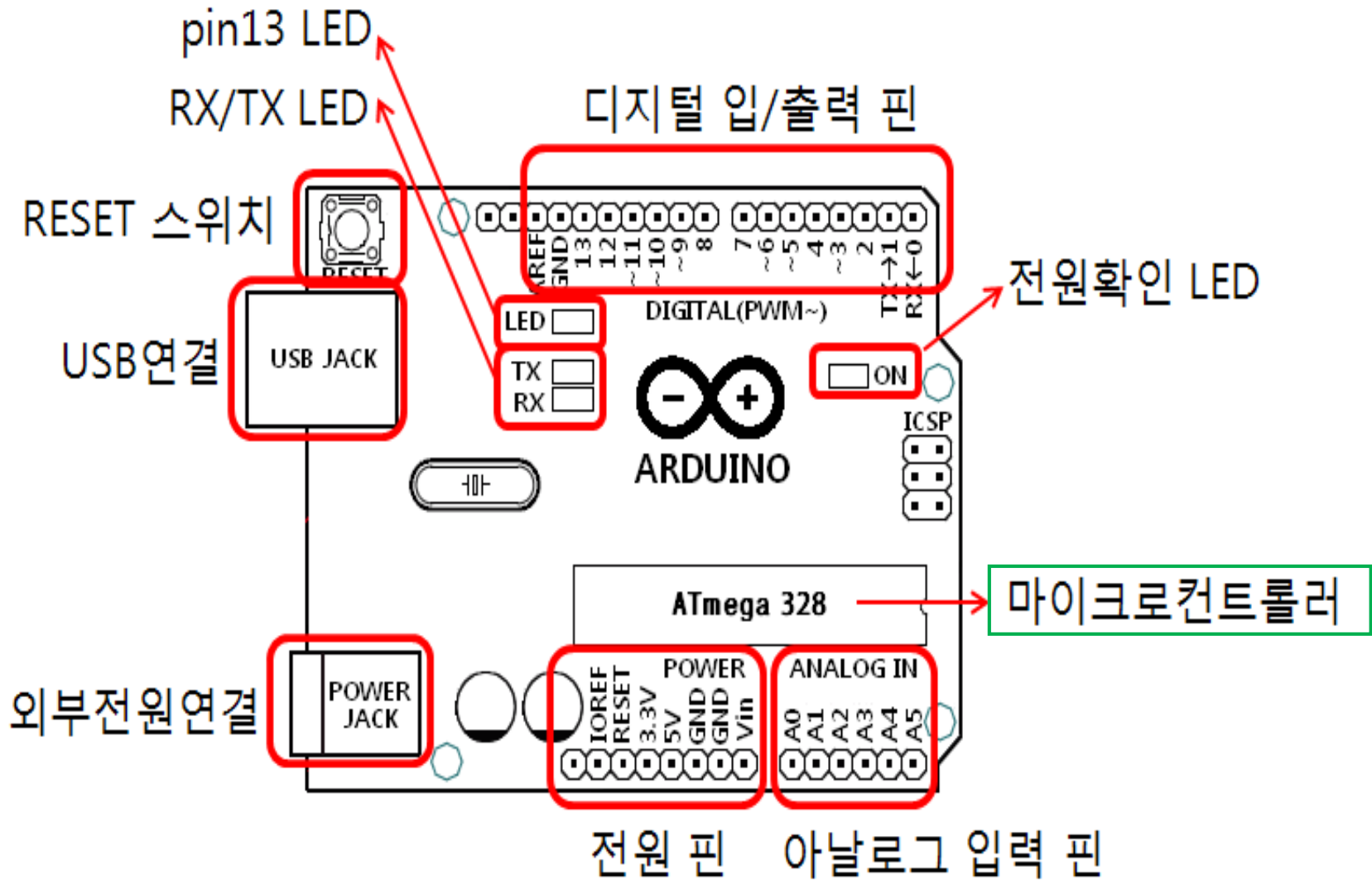




# Arduino Uno Board



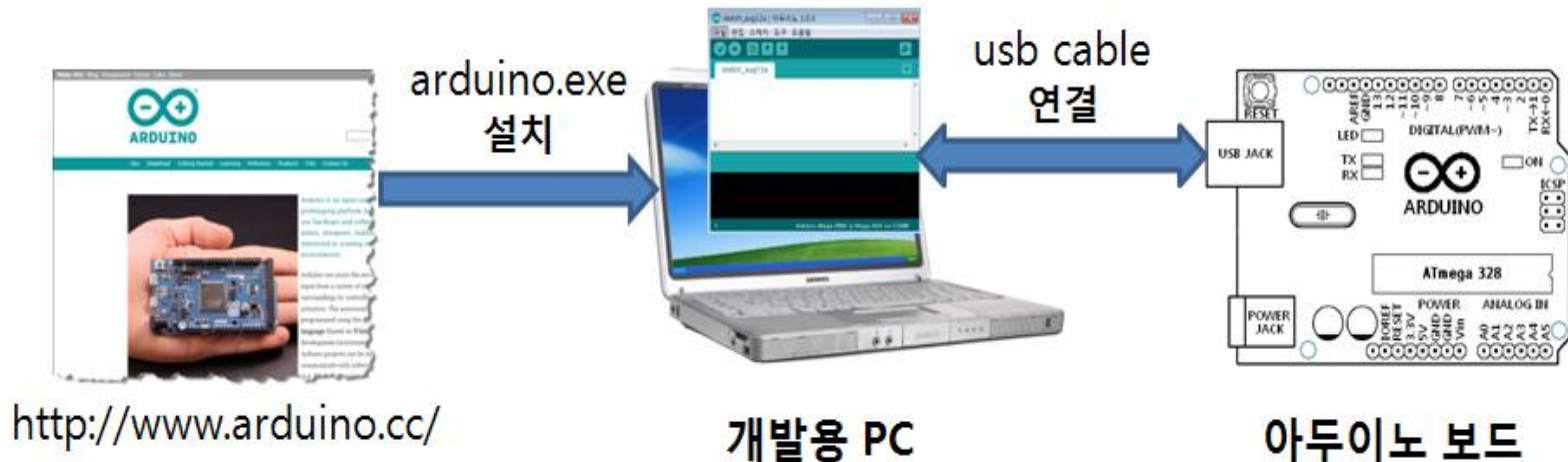
# UNO 보드의 핀과 기능



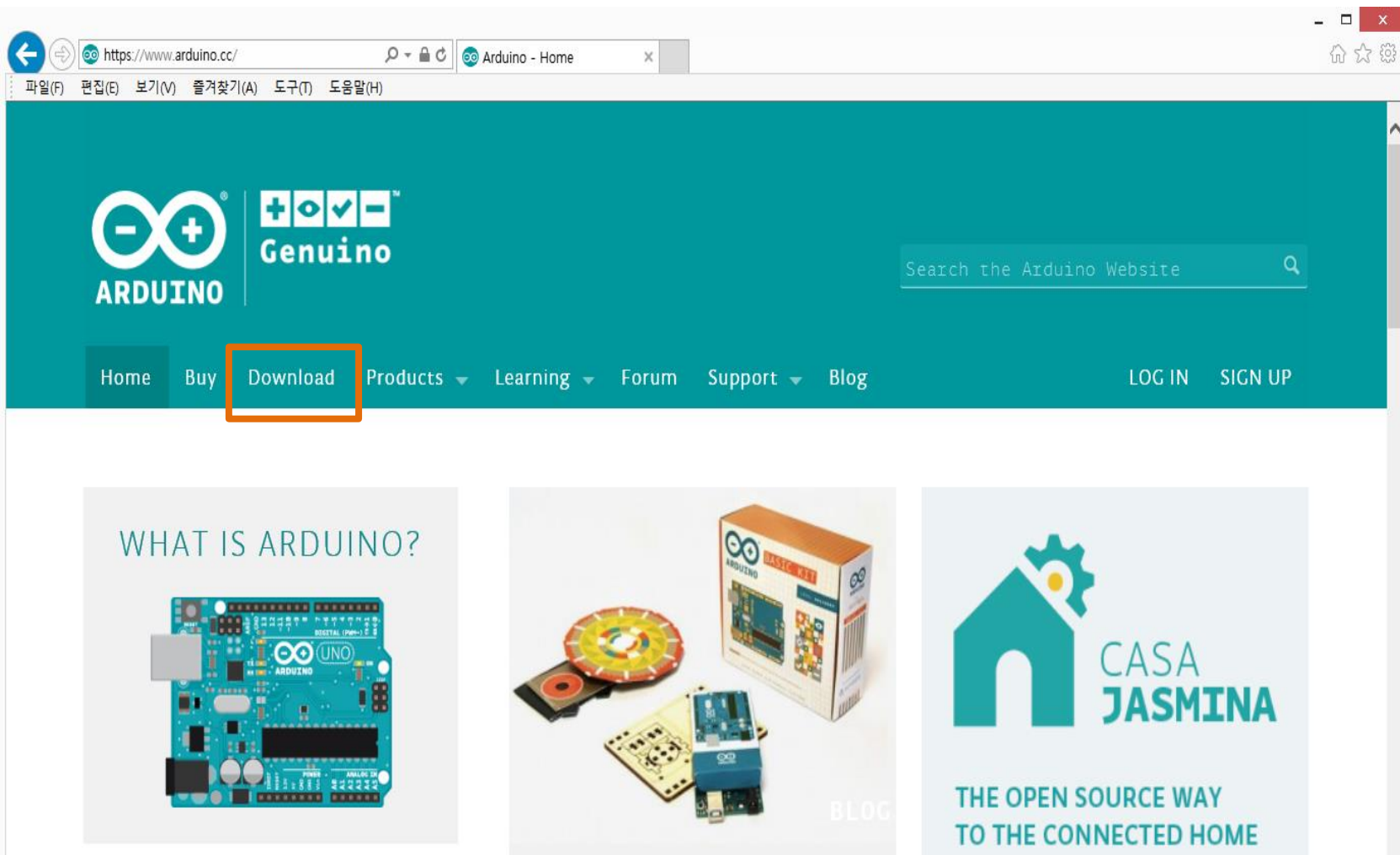
# Arduino 개발환경 구축

# 아두이노 개발환경

- 아두이노를 사용하기 위해서는 아두이노 보드를 위한 개발환경을 아두이노 개발용 PC에 구축 필요
- 개발환경이 구축된 PC는 아두이노 보드에서 수행할 스케치(sketch, 프로그램)를 작성하여, USB 케이블 등으로 연결된 아두이노 보드에 업로드



# 1. Arduino개발자 홈페이지에서 Download (<http://www.arduino.cc>)




# 2. ARDUINO 1.0.6 버전으로 사용하는 컴퓨터의 OS에 맞는 개발환경 프로그램 선택

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

Arduino - Software

Buy Download Products Learning Forum Support Blog LOG IN SIGN UP

## Download the Arduino Software



### ARDUINO 1.6.5

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

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

- Windows Installer
- Windows ZIP file for non admin install
- Mac OS X 10.7 Lion or newer
- Linux 32 bits
- Linux 64 bits

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

AROUND SOFTWARE  
**HOURLY BUILDS**

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

[Windows](#)  
[Mac OS X](#) (Mac OSX Lion or later)  
[Linux 32 bit](#), [Linux 64 bit](#)

LAST UPDATE  
22 July 2015 0-15:16 GMT

AROUND 1.0.6 / 1.5.x / 1.6.x  
**PREVIOUS RELEASES**

Download the [previous version of the current release](#), the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

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

### 3. Previous version of the current release 선택

ARDUINO 1.0.6 / 1.5.x / 1.6.x

## PREVIOUS RELEASES

Download the [previous version of the current release](#), the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

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

# 4. ARDUINO 1.0.6버전 windows installer 선택

The screenshot shows the Arduino website's 'Previous IDE Releases' page. The browser address bar displays 'https://www.arduino.cc/en/Main/OldSoftwareReleases#previous'. The page features a teal navigation bar with the Arduino logo and links for 'Buy', 'Download', 'Products', 'Learning', 'Forum', 'Support', and 'Blog'. On the right side of the navigation bar are 'LOG IN' and 'SIGN UP' buttons. The main content area is titled 'Previous IDE Releases' and contains two columns of release information. The right column is for 'ARDUINO 1.0.6' and includes a 'Windows Installer' link that is highlighted with an orange rectangular box. Below it are links for 'Windows ZIP file for non admin install', 'Mac OS X', 'Linux 32 bits', 'Linux 64 bits', and 'Source'. The left column is for 'ARDUINO 1.6.4' and includes similar links for 'Windows Installer', 'Windows ZIP file for non admin install', 'Mac OS X 10.7 Lion or newer', 'Linux 32 bits', 'Linux 64 bits', and 'Source'.

https://www.arduino.cc/en/Main/OldSoftwareReleases#previous

Arduino - OldSoftwareRele... x

Buy Download Products Learning Forum Support Blog LOG IN SIGN UP

## Previous IDE Releases

### ARDUINO 1.6.4

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.  
[See the release notes.](#)

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

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits

**Source**

### ARDUINO 1.0.6

Classic Arduino IDE, to be used with any Arduino board, but Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.  
[See the release notes.](#)

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

**Mac OS X**

**Linux** 32 bits  
**Linux** 64 bits

**Source**



# 5. 실행 클릭한 후 프로그램 다운로드 후 설치

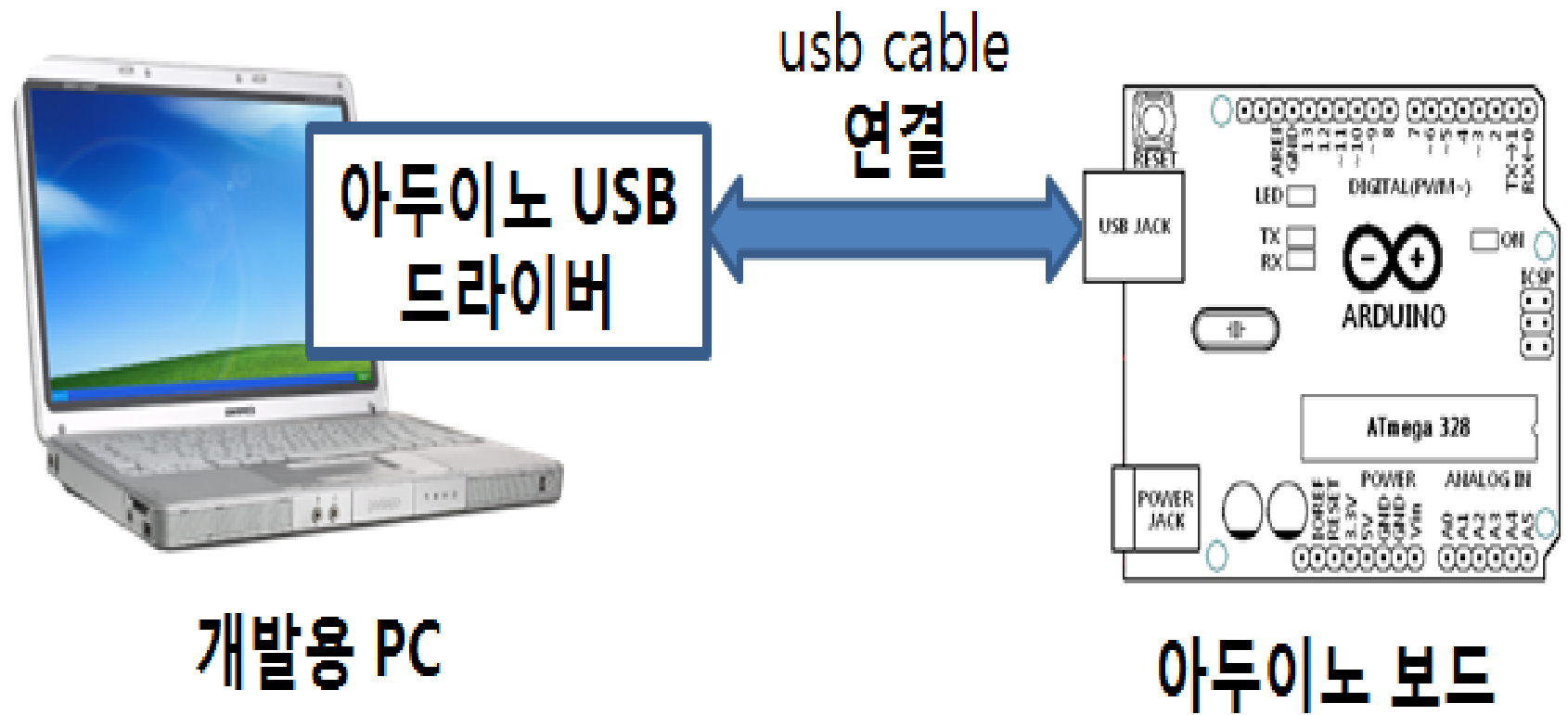
The screenshot shows the Arduino website's 'Previous IDE Releases' page. The browser address bar displays 'https://www.arduino.cc/en/Main/OldSoftwareReleases#previous'. The page features a teal navigation bar with the Arduino logo and links for 'Buy', 'Download', 'Products', 'Learning', 'Forum', 'Support', and 'Blog'. The main content area is titled 'Previous IDE Releases' and contains two columns of release information.

**ARDUINO 1.6.4**  
Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.  
[See the release notes.](#)  
**Windows** Installer  
**Windows** ZIP file for non admin install  
**Mac OS X** 10.7 Lion or newer  
**Linux** 32 bits  
**Linux** 64 bits  
**Source**

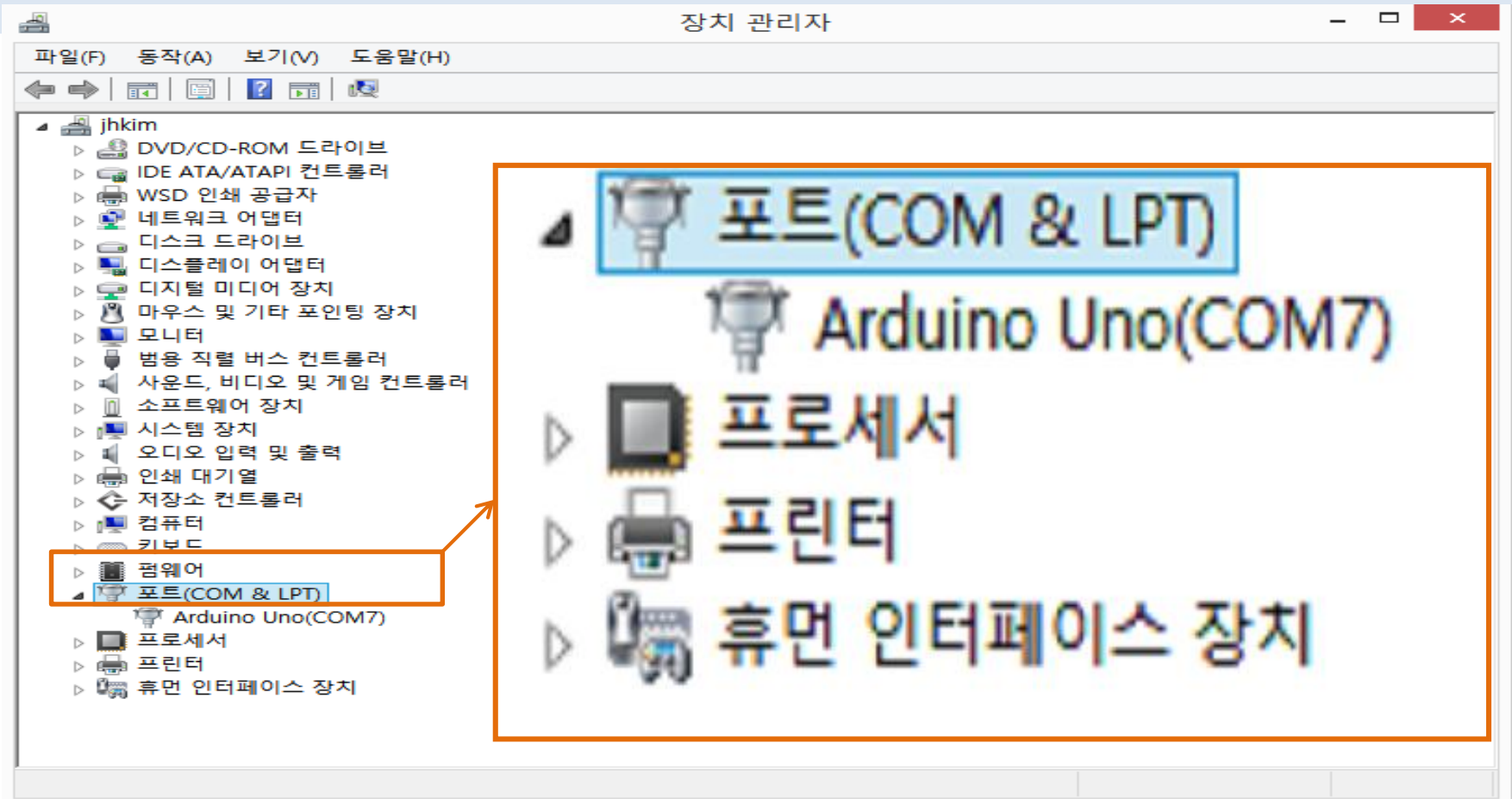
**ARDUINO 1.0.6**  
Classic Arduino IDE, to be used with any Arduino board, but Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.  
[See the release notes.](#)  
**Windows** Installer  
**Windows** ZIP file for non admin install  
**Mac OS X**  
**Linux** 32 bits  
**Linux** 64 bits  
**Source**

A Windows download dialog box is open at the bottom of the screen, showing the file 'downloads.arduino.cc의 arduino-1.0.6-windows.exe(52.5MB)을(를) 실행하거나 저장하시겠습니까?' with buttons for '실행(R)', '저장(S)', and '취소(C)'.

# 6. 아두이노 보드를 USB 케이블로 컴퓨터에 연결

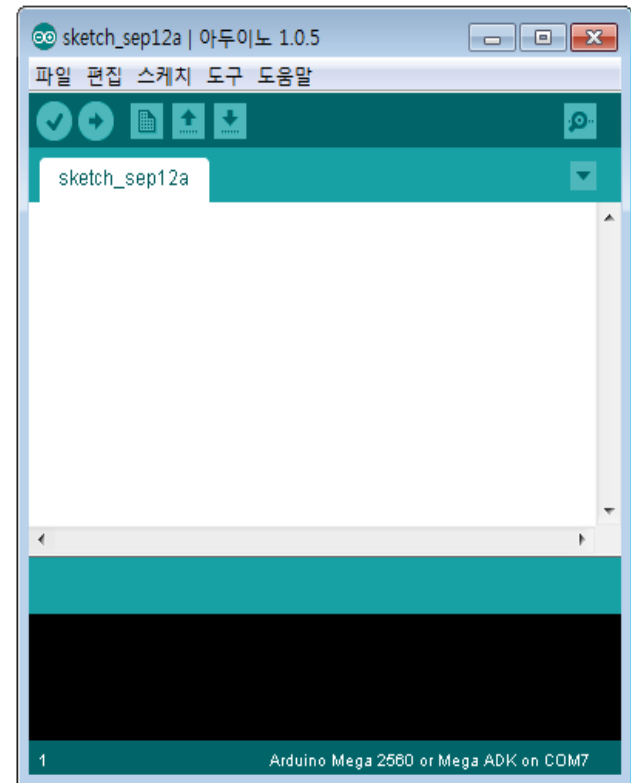
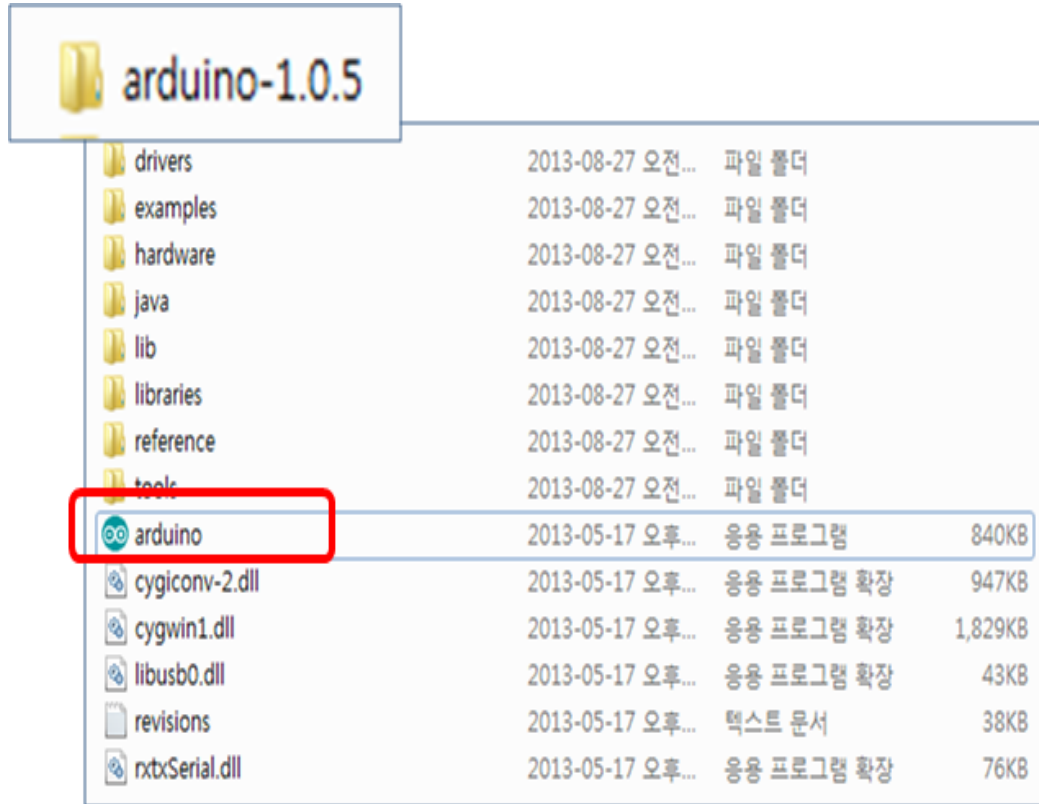


# 7. 장치 관리자에서 아두이노 보드 드라이버 설치 확인

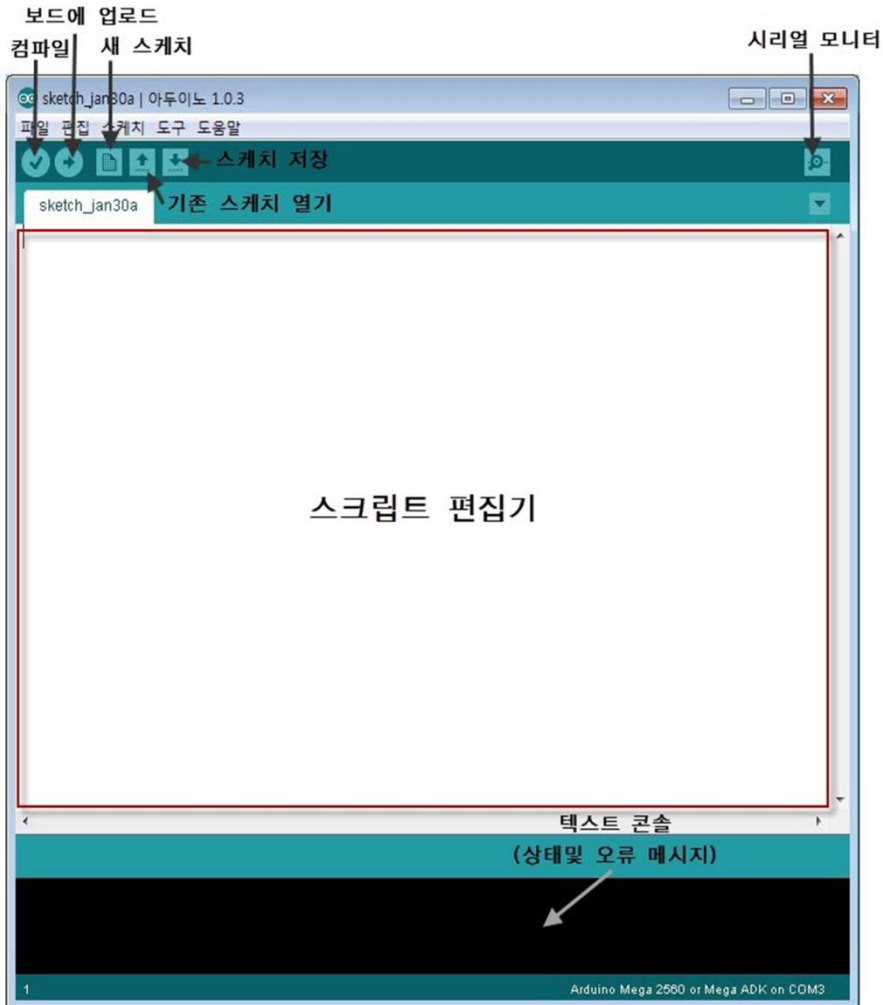


\*\*\* 장치 관리자에서 드라이버를 찾지 못하는 경우 수동으로 지정해 줌  
(프로그램이 설치된 폴더의 drivers라는 폴더에서 찾아서 선택해 줌)

# 8. 압축을 해제한 후 생성되는 Arduino.exe 파일, 또는 단축아이콘(바탕화면)을 실행



# 아두이노 개발환경 툴바



확인(컴파일) 업로드 새파일 열기 저장 시리얼 모니터

파일	편집	스케치	도구	도움말
새 파일				
열기...				
스케치 복사				
예제				
닫기				
저장				
다른 이름으로 저장...				
업로드				
프로그래머를 이용해 업로드				
페이지 설정				
인쇄				
환경 설정				
종료				

편집	스케치	도구	도움말
취소			
다시 실행			
잘라내기			
복사			
포함을 위해 복사			
HTML로 복사			
붙여넣기			
모두 선택			
주석추가/주석삭제			
틀여쓰기 추가			
틀여쓰기 삭제			
찾기...			
다음 찾기			
이전 찾기			
찾기위해 선택된 부분을 사용			

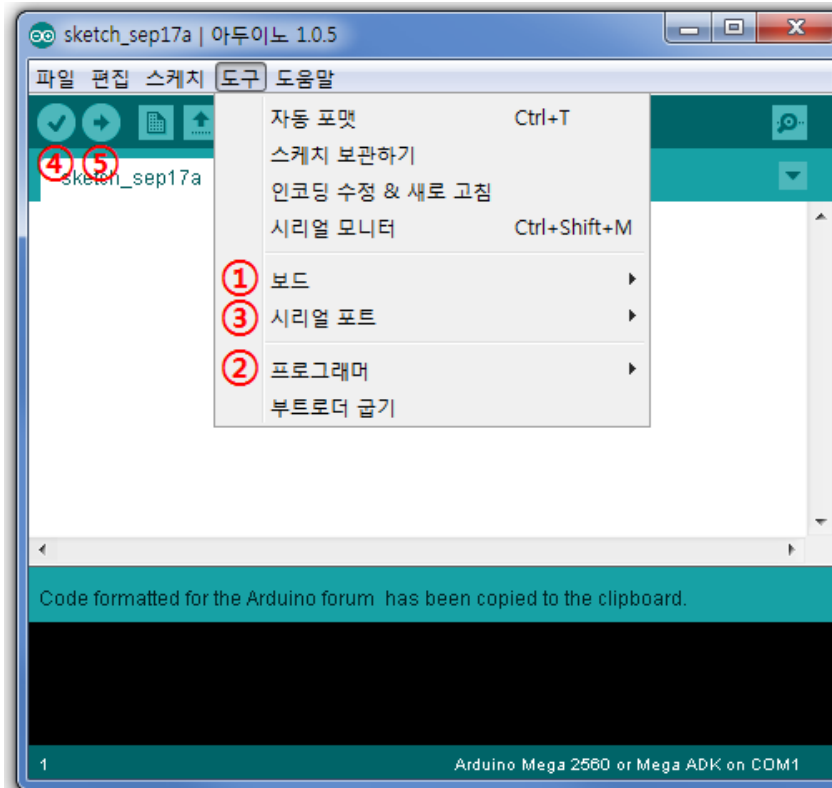
스케치	도구	도움말
확인 / 컴파일		
스케치 숨더 보기		
파일 추가...		
라이브러리 가져오기...		

도구	도움말
자동 포맷	
스케치 보관하기	
인코딩 수정 & 새로 고침	
시리얼 모니터	
보드	
시리얼 포트	
프로그래머	
부트로더 굽기	

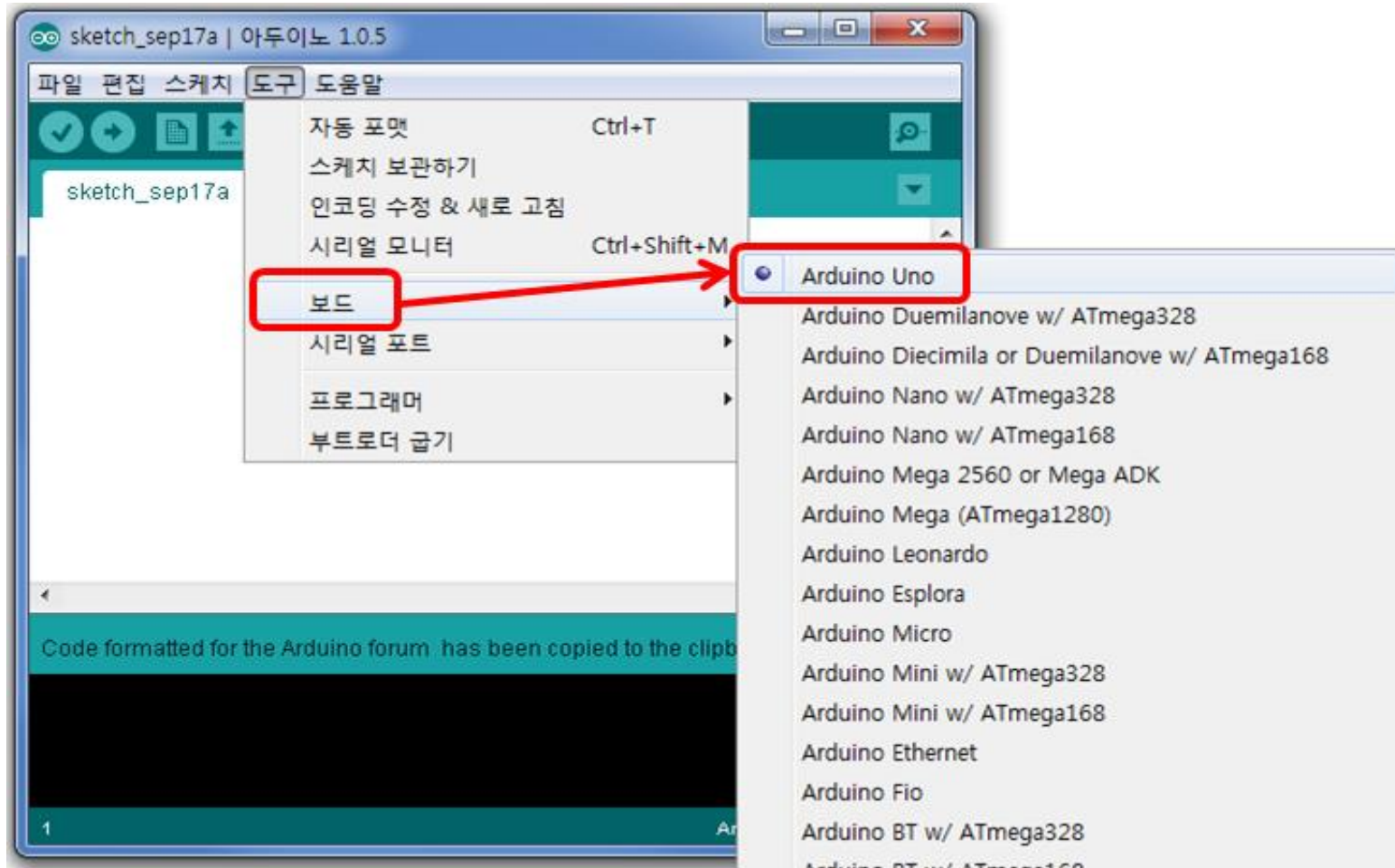
도움말
시작하기
환경
문제해결
참조
참조에서 찾기
자주 묻는 질문
Arduino.cc를 방문
아두이노 정보

# 업로드 환경 설정

- 개발용 PC에서 작성한 스케치를 아두이노 보드에 업로드하기 위하여, 아두이노 개발환경에서 사용할 보드와 프로그래머를 선택
- 아두이노 개발환경은 선택한 보드와 프로그래머에 맞게 컴파일하고 업로드

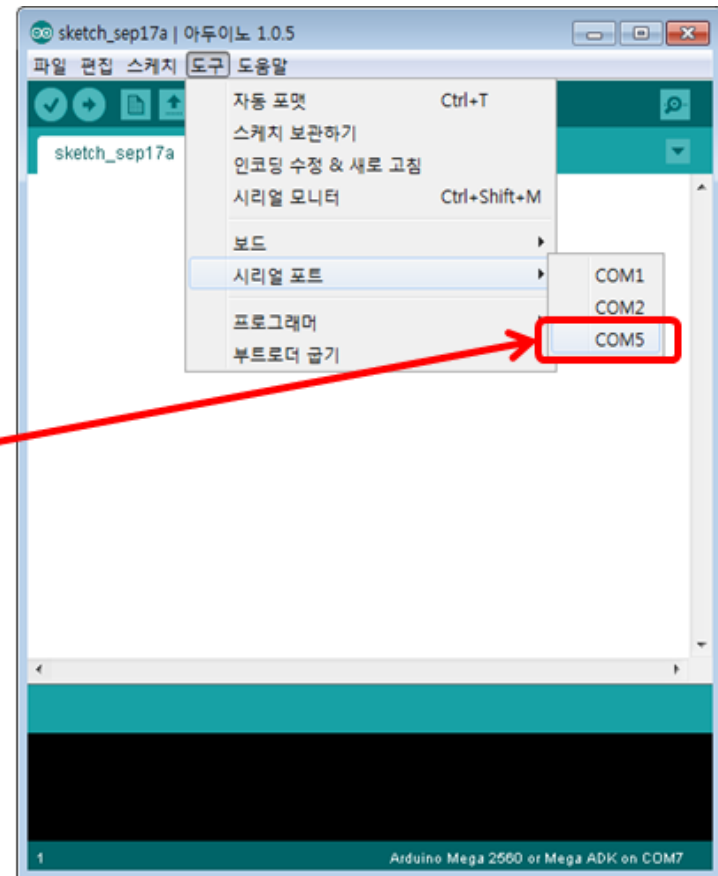
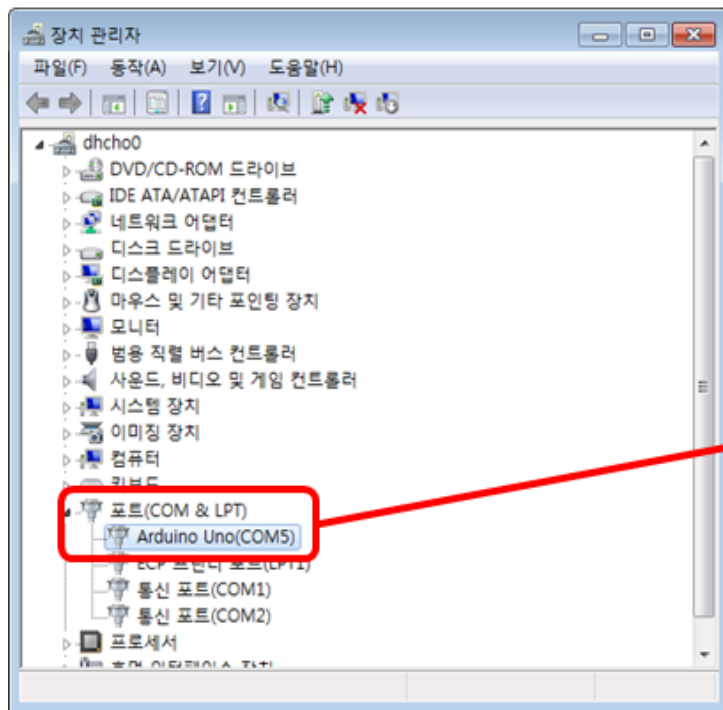


# 환경 설정-아두이노 보드 선택



# 환경 설정-시리얼포트 확인

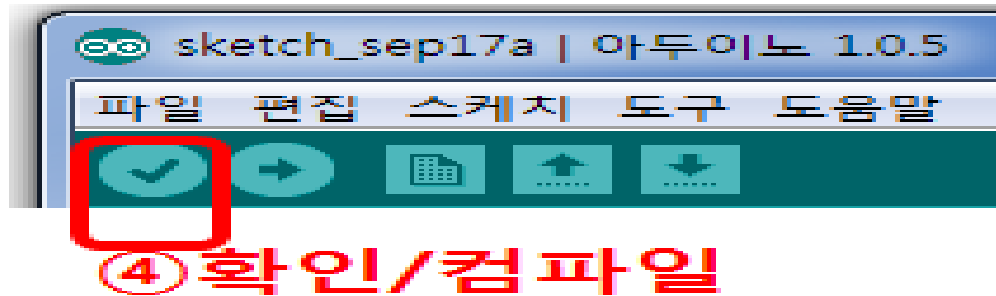
: 아두이노 보드가 연결된 포트인지 확인 후 선택



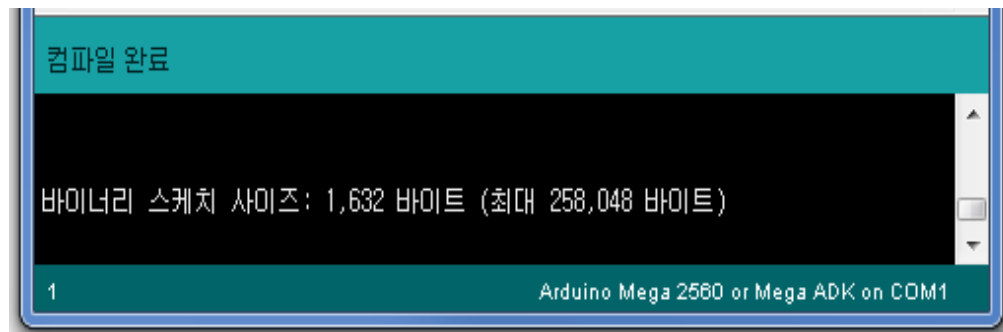


# 프로그램 컴파일

- 상단 툴바의 확인/컴파일을 누르면, 작성한 스케치가 선택한 아두이노 보드에 맞는 형식으로 컴파일

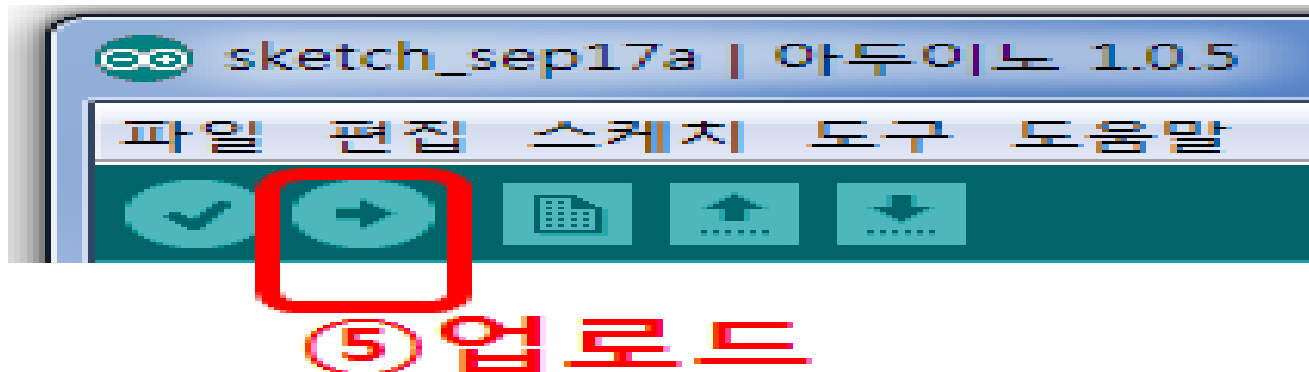


- 만일 스케치에 오류가 있거나 보드선택이 잘못되면 스케치창 아래 부분에 오류 메시지 표시
- 오류가 없이 컴파일이 완료 메시지



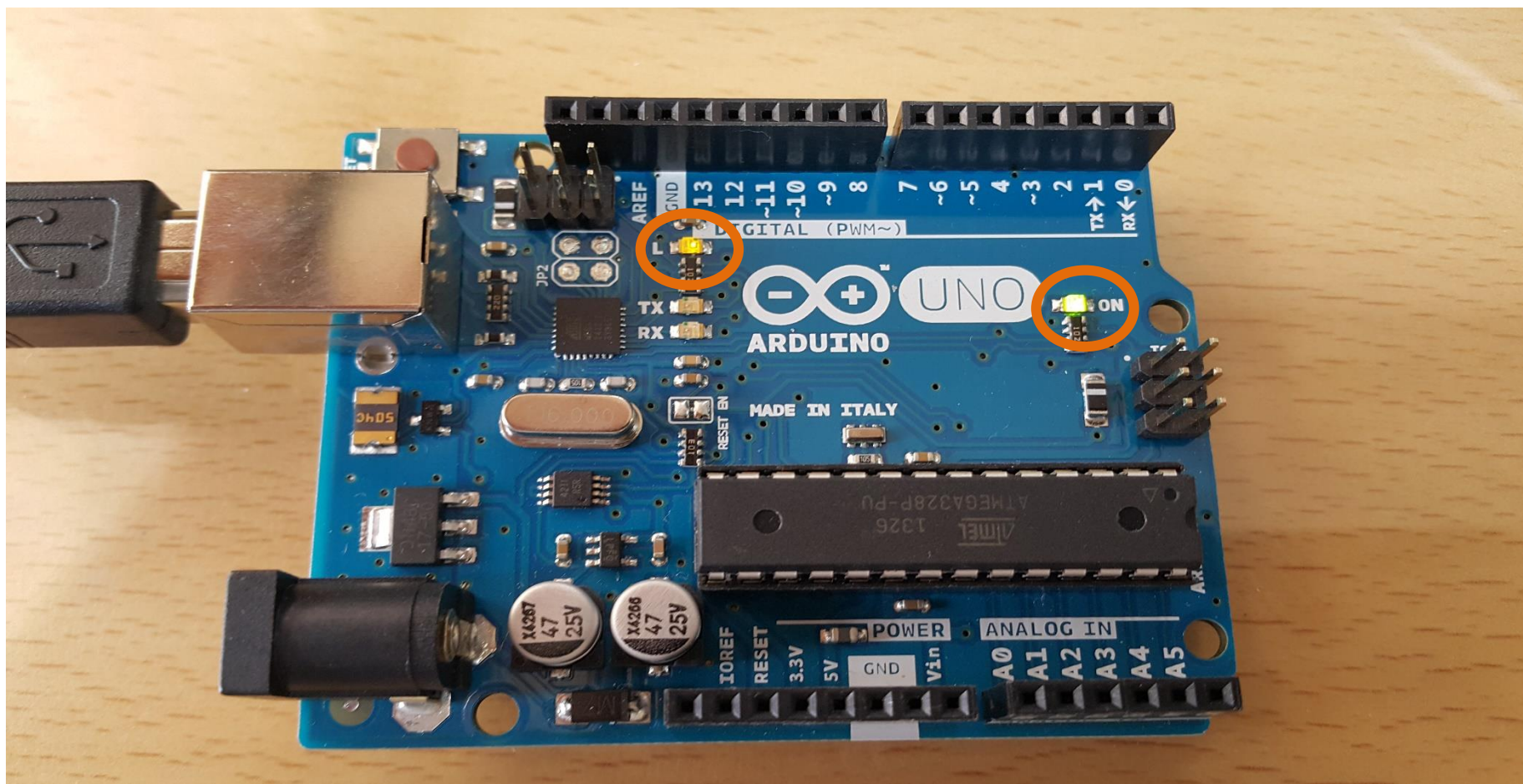
# 프로그램 업로드

- 확인/컴파일에서 오류가 발생하지 않으면, 개발용 PC에 USB 케이블로 연결된 아두이노 보드에 컴파일된 코드를 전송하여 저장



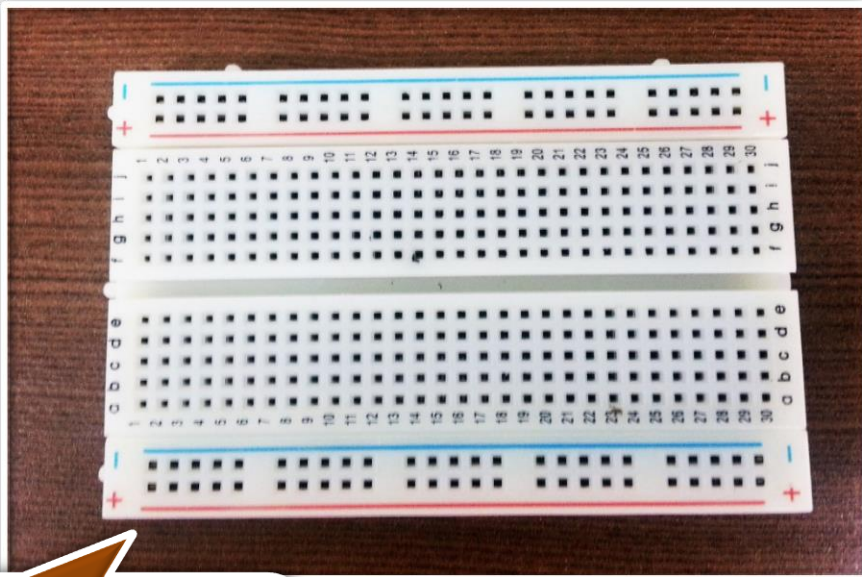
# 개발환경 설치 완료 확인

: 아두이노 보드의 13번 핀에 연결된 황색 LED(황색)와 ON(연두색)에 불이 켜짐



# 아두이노로 LED 제어하기

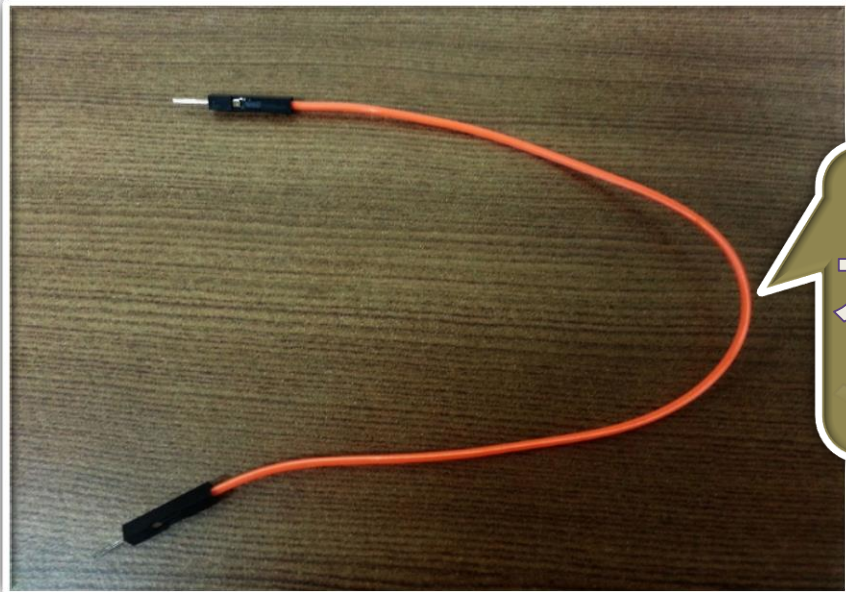
뱅판



저항

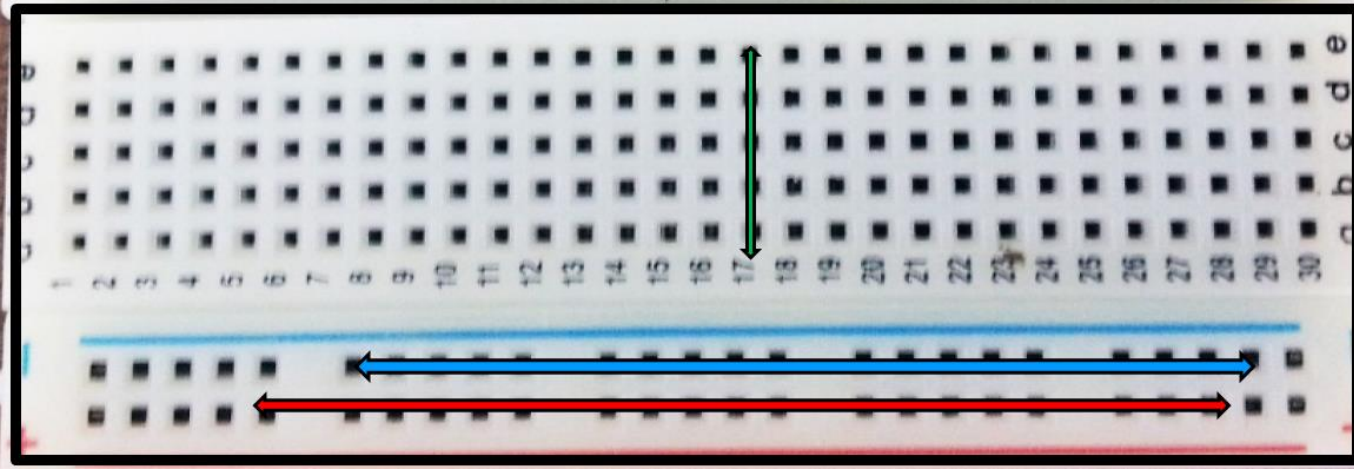
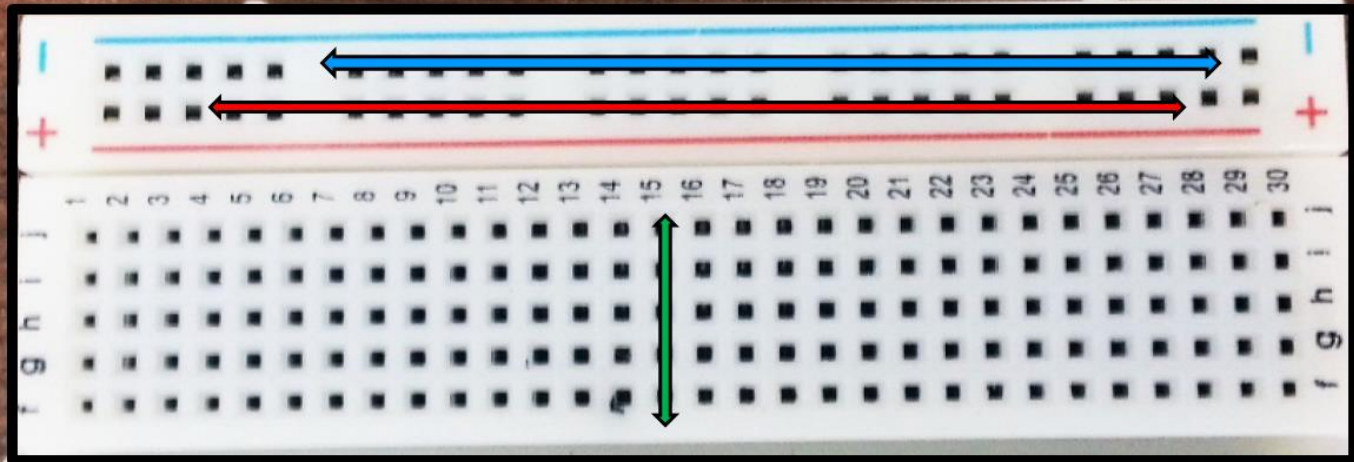
LED

부저

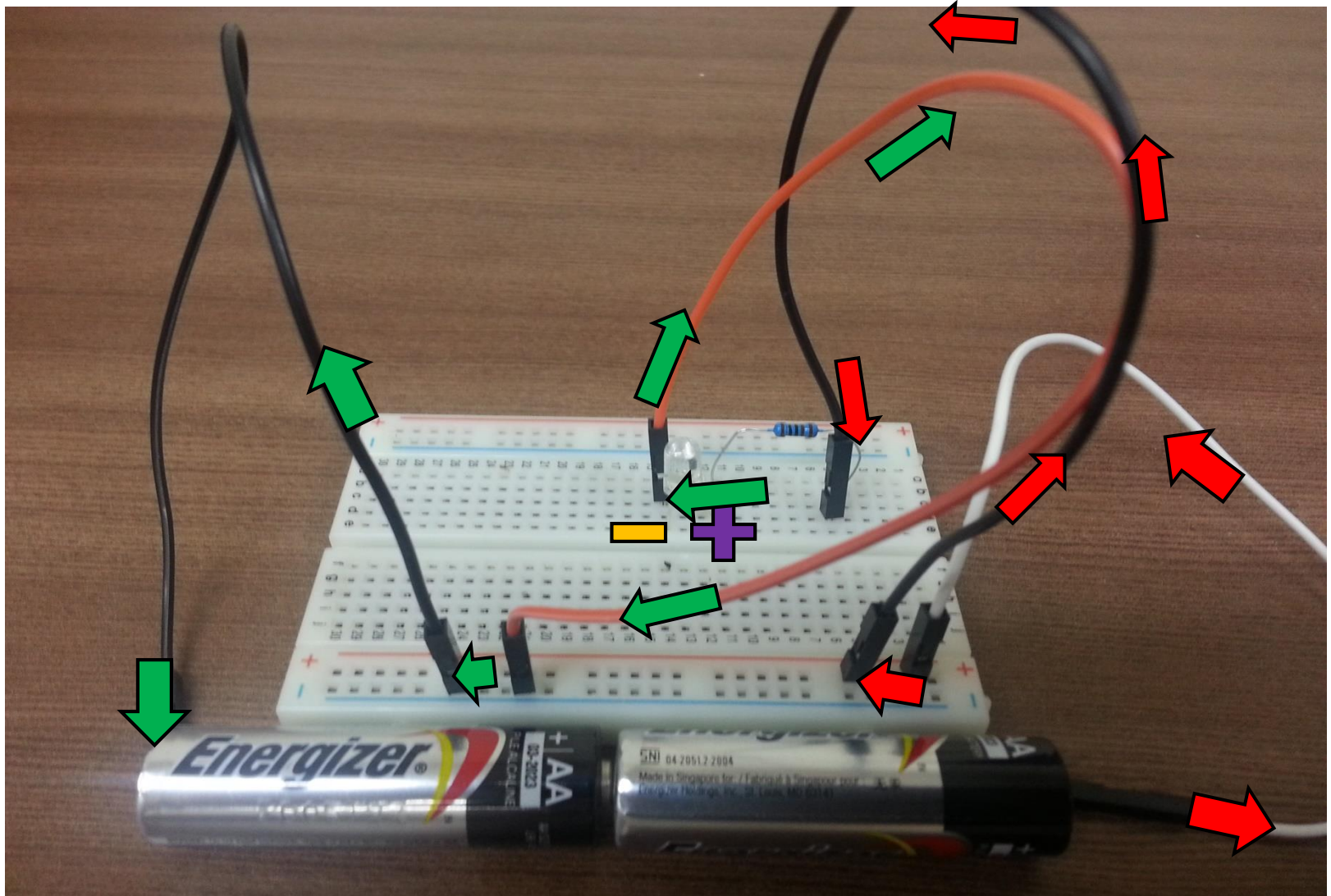


전선

# 브레드보드 사용법



# LED 1개 건전지로 연결하기



# 실습시 주의사항

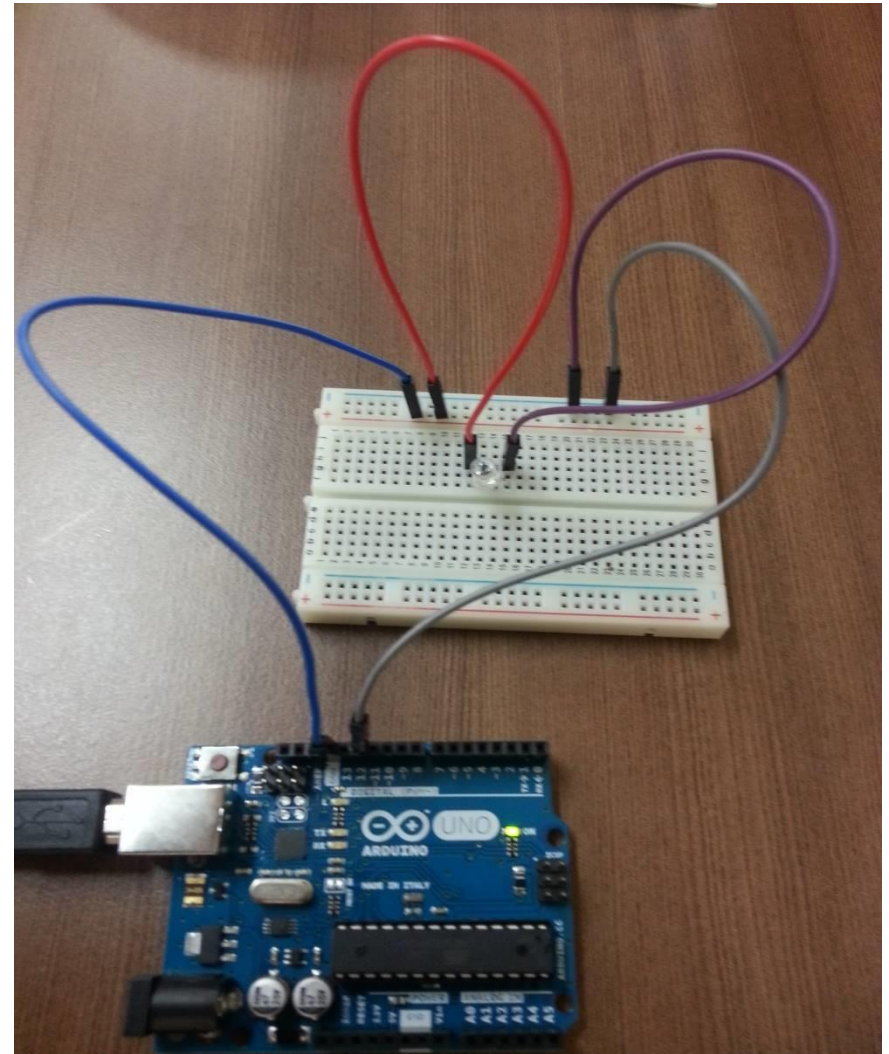
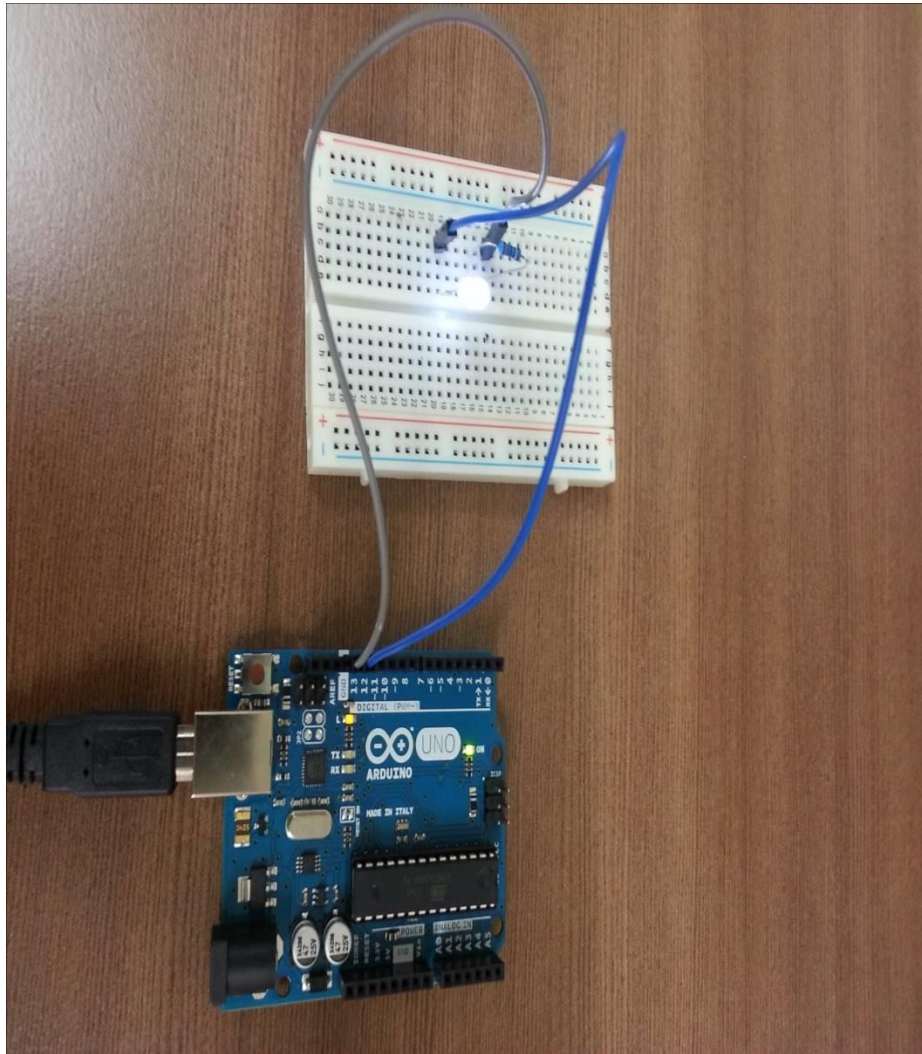
1. 아두이노 보드의 5V 핀과 GND 핀을 바로 연결하지 않기
2. 브레드보드에 배선 연결 작업을 할 때는 항상 전원을 연결하지 않은 상태에서 작업하기



# 저항 읽는 방법

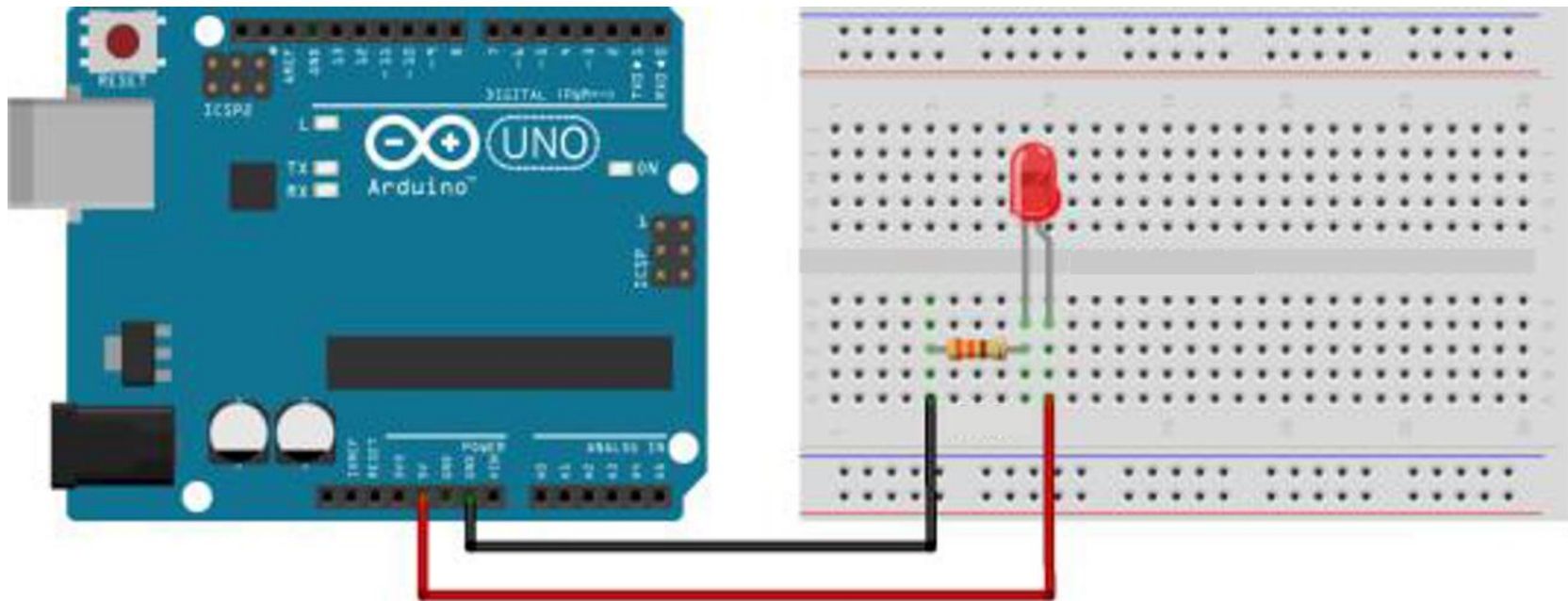
색명 (컬러 코드)		제1색대	제2색대	제3색대	제4색대
		제1숫자	제2숫자	승수	공칭저항값 허용 치(오차)
검은색		0	0	1	
갈색		1	1	10	1%
적색		2	2	100	2%
주황색		3	3	1000	
노랑색		4	4	10000	
녹색		5	5	100000	0.5% *
청색		6	6	1000000	
보라색		7	7	10000000	
회색		8	8	100000000	
흰색		9	9	1000000000	
금색				0.1	5%
은색				0.01	10%
무색					20%

# LED 1개 ON/OFF 연결 방법



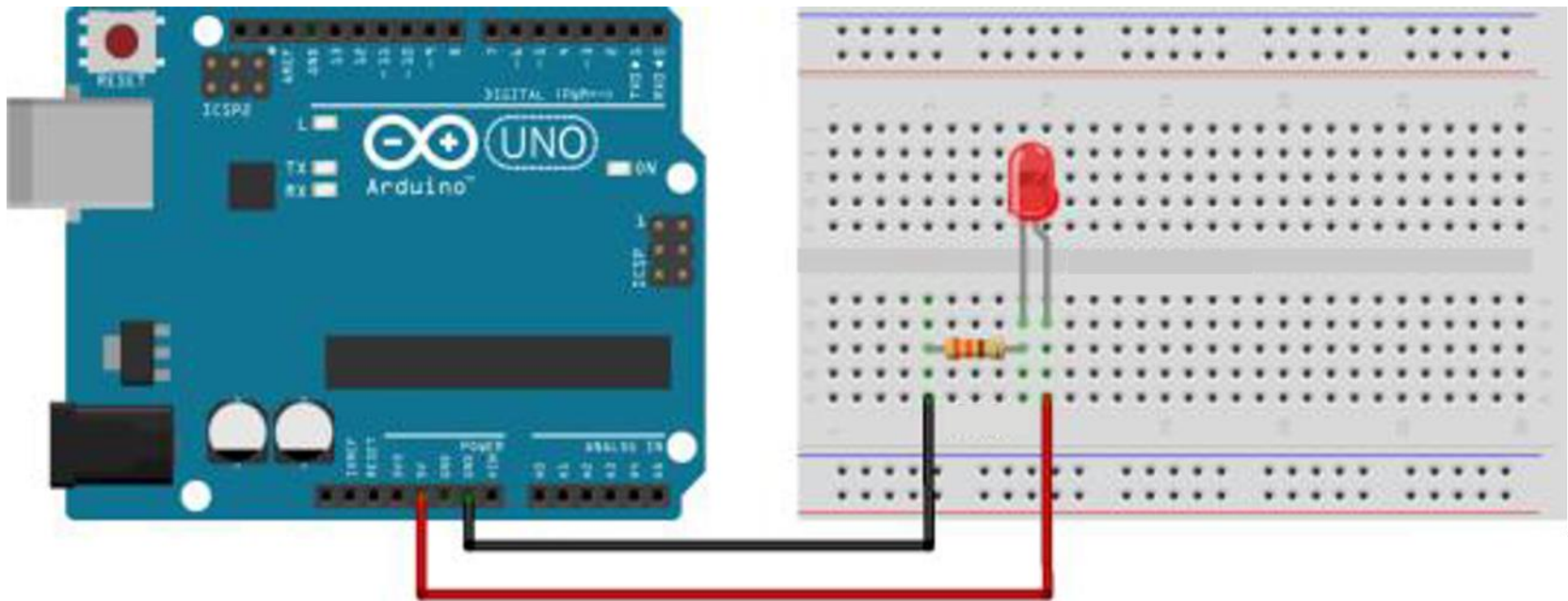
# 회로 구성 방법

- 전류는 5V (+) 핀에서 흘러 나와서, LED의 긴 핀(+)으로 들어가서 저항을 거쳐서 GND(-)로 들어간다.  
이 때 LED에 전류가 흐르기 때문에 LED가 켜진다.



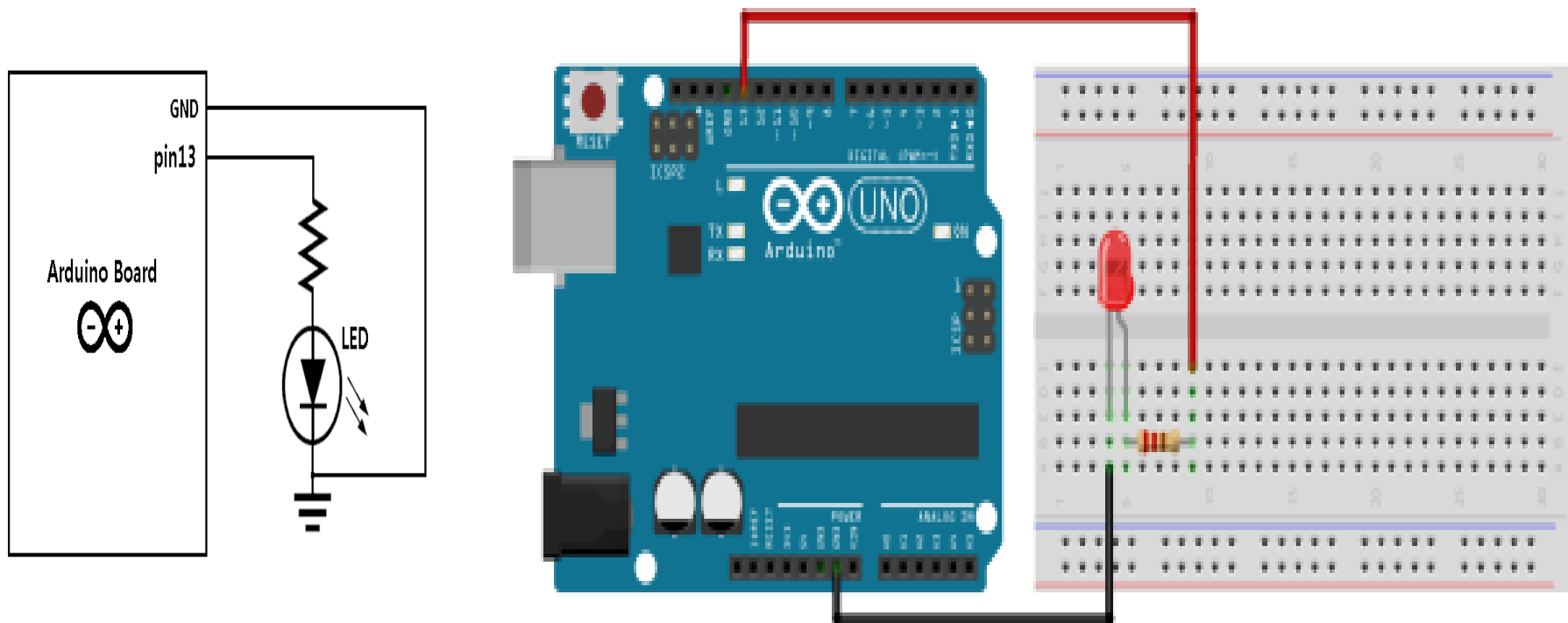
# 스케치 없이 LED 1개 켜기

1. LED의 두 핀 중에서 길이가 긴 핀(+)을 아두이노 보드의 5V 핀에 연결함
2. LED의 두 핀 중 짧은 핀(-)을 저항의 한쪽 핀에 연결함
3. 저항의 다른 한 쪽 핀을 아두이노 보드의 GND핀과 연결함
4. PC와 아두이노 보드를 USB 케이블로 연결함



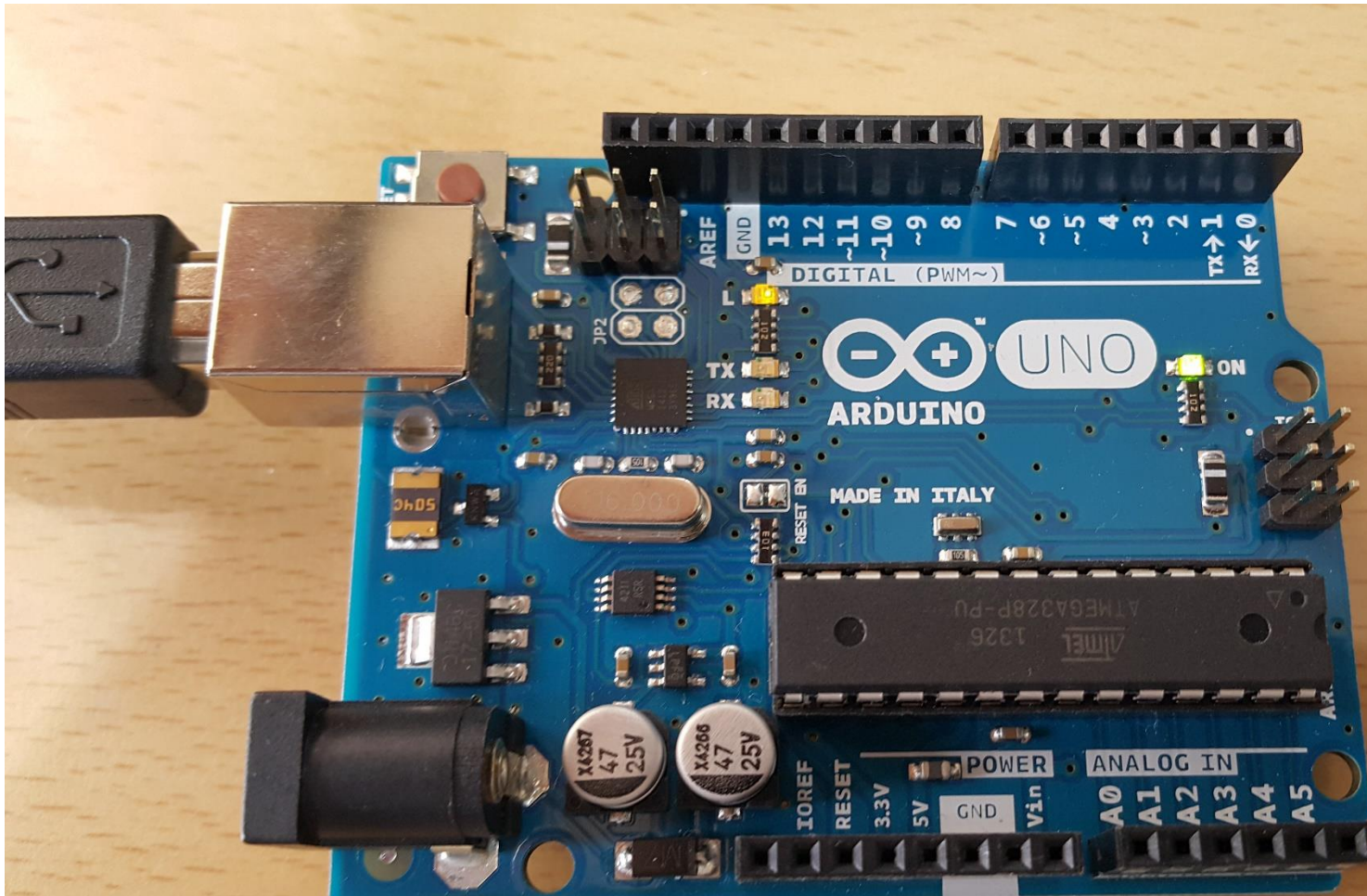
# 디지털입출력 13번의 스케치 프로그램

- 아두이노 보드 외부에 LED를 연결하여 제어
- 디지털입출력 핀 13번에 LED 연결



# 디지털입출력 13번 연결 화면

디지털입출력 핀 13번에 연결한 LED와 ON에 불이 켜짐



# 예제 "Blink" 사용한 스케치 업로드

The image shows the Arduino IDE interface with the 'Sketches' menu open. The 'Blink' example is selected. The IDE window title is 'Blink | 아두이노 1.0.5-r2'. The menu items are as follows:

- 파일 (File)
- 편집 (Edit)
- 스케치 (Sketches)
- 도구 (Tools)
- 도움말 (Help)

The 'Sketches' menu items include:

- 새 파일 (New) Ctrl+N
- 열기... (Open...) Ctrl+O
- 스케치북 (Sketchbook)
- 예제 (Examples) - expanded
  - 01.Basics
  - 02.Digital
  - 03.Analog
  - 04.Communication
  - 05.Control
  - 06.Sensors
  - 07.Display
  - 08.Strings
  - 09.USB
  - 10.StarterKit
  - ArduinoISP
  - EEPROM
  - Esplora
  - Ethernet
  - Firmata
  - GSM
  - LiquidCrystal
  - Robot\_Control
  - Robot\_Motor
  - SD
  - Servo
  - SoftwareSerial
  - SPI
  - Stepper
  - TFT
  - WiFi
  - Wire
- 닫기 (Close) Ctrl+W
- 저장 (Save) Ctrl+S
- 다른 이름으로 저장... (Save As...) Ctrl+Shift+S
- 업로드 (Upload) Ctrl+U
- 프로그래머를 이용해 업로드 (Upload Using Programmer) Ctrl+Shift+U
- 페이지 설정 (Page Setup) Ctrl+Shift+P
- 인쇄 (Print) Ctrl+P
- 환경 설정 (Preferences) Ctrl+Comma
- 종료 (Quit) Ctrl+Q

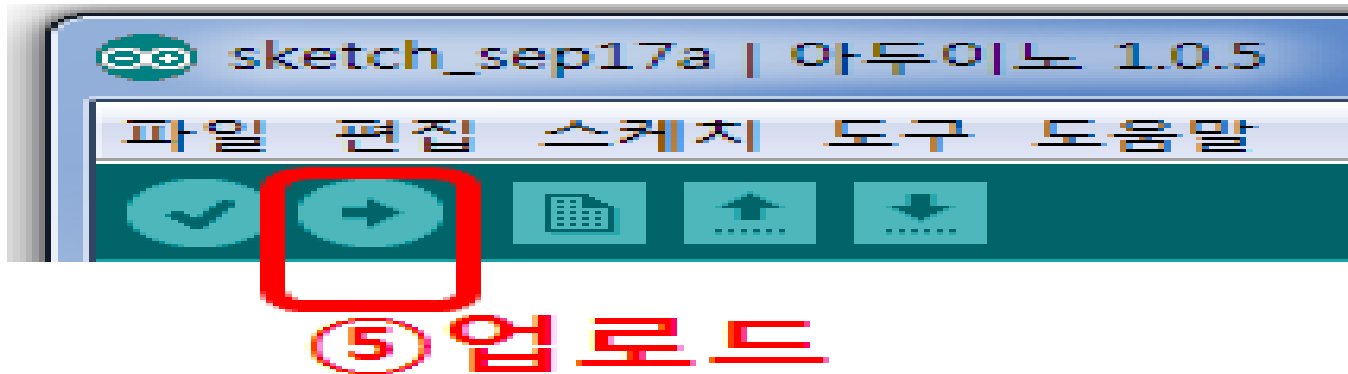
The code editor shows the following code:

```
delay(1000);  
digitalWrite(led, LOW); // turn the LED  
delay(1000); // wait for a s  
}
```

The status bar at the bottom indicates: 컴파일 완료 (Compilation complete), 바이너리 스케치 사이즈: 1,084 바이트 (최대 32 (Maximum sketch size: 1,084 bytes (maximum 32...)), and the line number 10.

# 프로그램 업로드

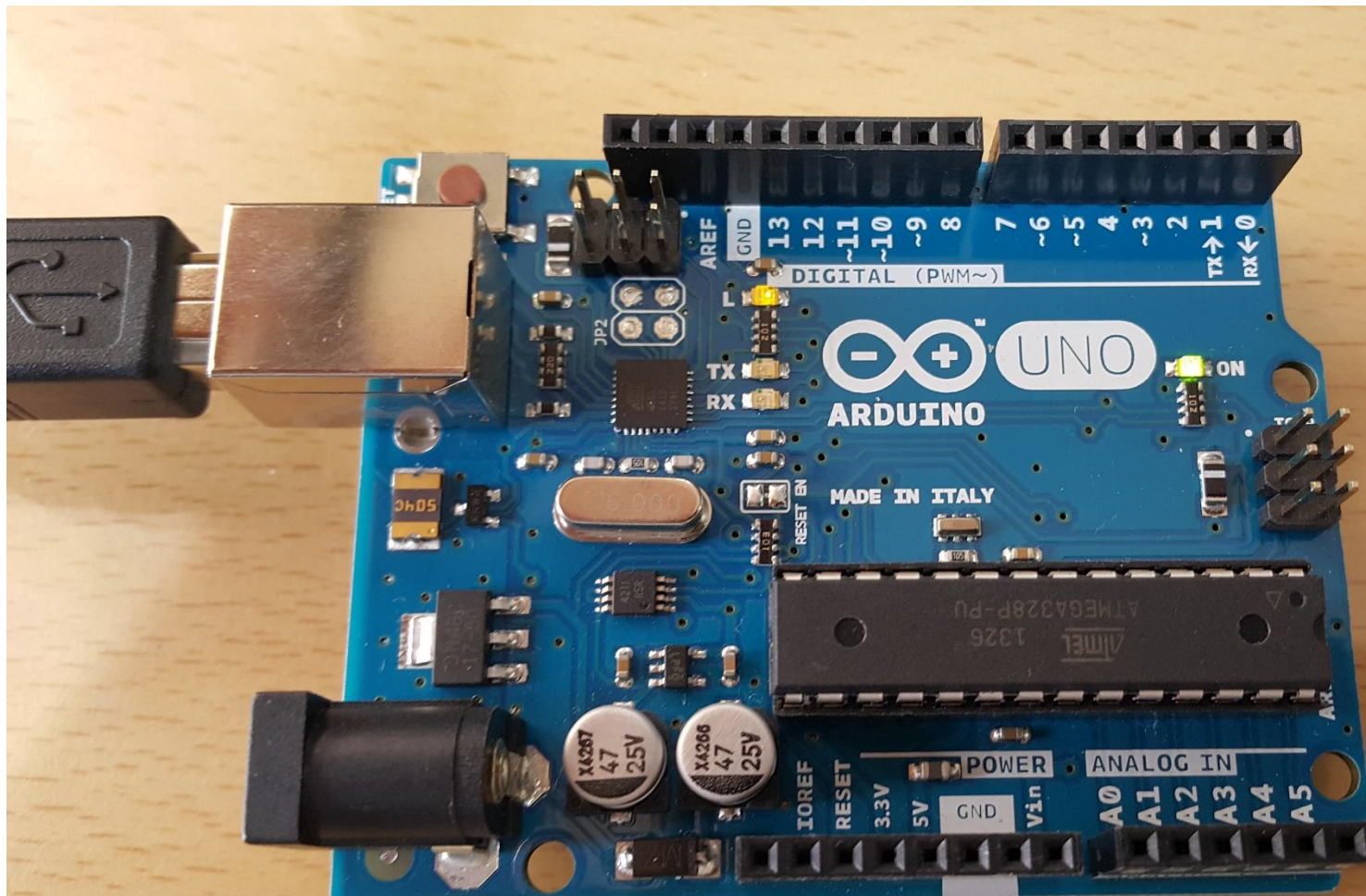
- Blink 예제 프로그램 아두이노 보드에 업로드





# 디지털입출력 13번이 연결된 화면

디지털입출력 핀 13번에 연결한 LED가 약 1초 간격으로 깜박임



# 예제 "Blink" 스케치 프로그램

```
Blink | 아두이노 1.0.5-r2
파일 편집 스케치 도구 도움말
Blink
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

컴파일 완료

바이너리 스케치 사이즈: 1,084 바이트 (최대 32,256 바이트)

10 Arduino Uno on COM4
```

"delay(1000)"에서 보다  
"delay(500)"에서 빠르게 LED가 깜박임

# "Blink" 예제 분석

<code>void setup() {</code>	아두이노 보드의 핀 설정 <code>setup()</code> 는 제일 먼저 실행되는 함수임
<code>pinMode(13, OUTPUT);</code>	<code>pinMode()</code> 는 핀의 입출력(핀의 상태) 결정 디지털입출력 핀 13번을 출력으로 설정
<code>}</code>	<code>setup()</code> 함수 종료
<code>void loop() {</code>	<code>loop()</code> 함수의 시작, 내부의 명령이 무한 반복
<code>digitalWrite(13, HIGH);</code>	디지털입출력 핀 13번에 HIGH("1") 출력
<code>delay(1000);</code>	1000ms(1초) 동안 시간지연
<code>digitalWrite(13, LOW);</code>	디지털입출력 핀 13번에 LOW("0") 출력
<code>delay(1000);</code>	1000ms 동안 시간지연
<code>}</code>	<code>loop()</code> 함수의 끝

# ON/OFF 속도 조절

- LED의 깜박이는 속도 조절

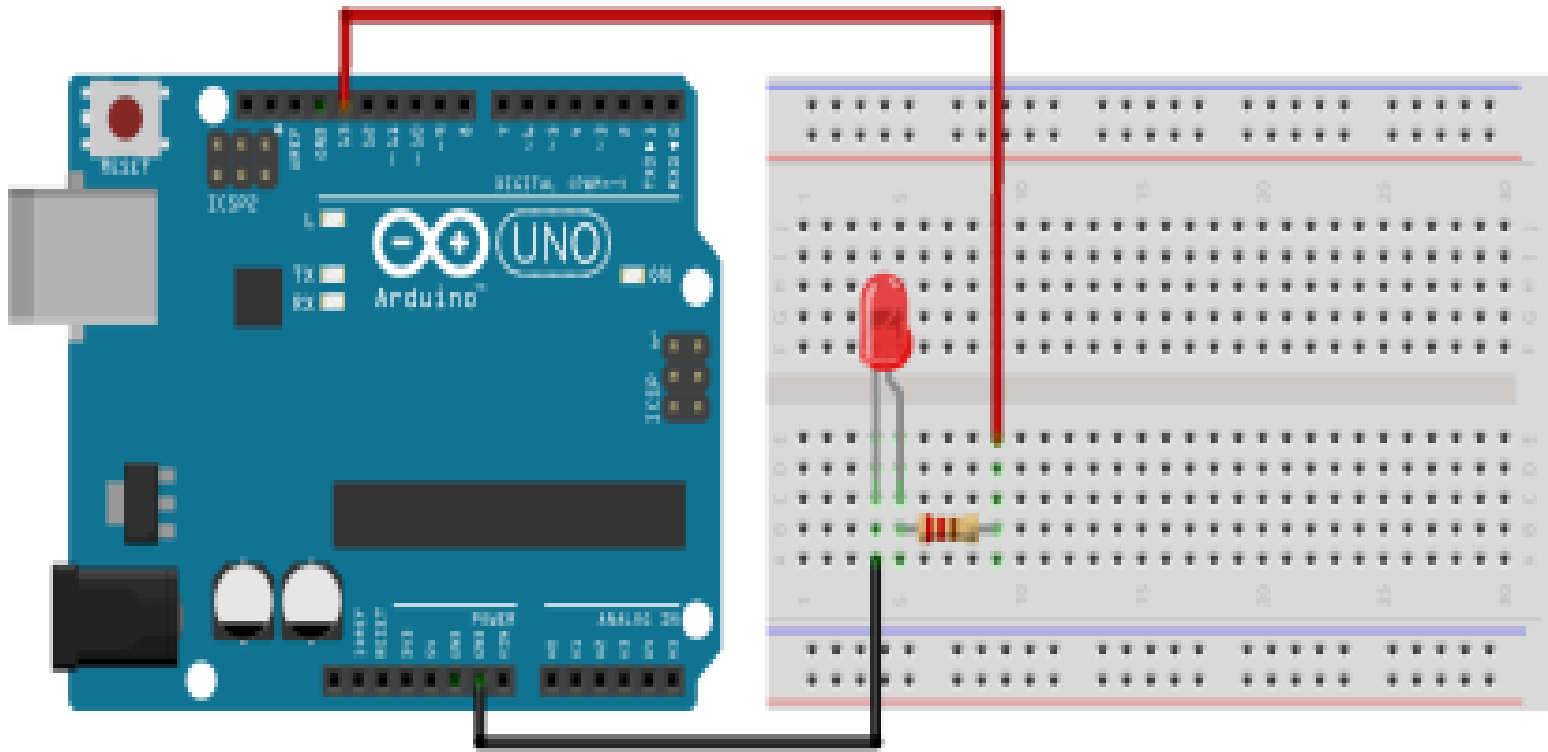
- 주요개념

- delay(ms)함수에서 변수 ms는 **밀리초(milli second)** 의미
  - "delay(500)" 0.5초의 시간지연을 시키는 동작

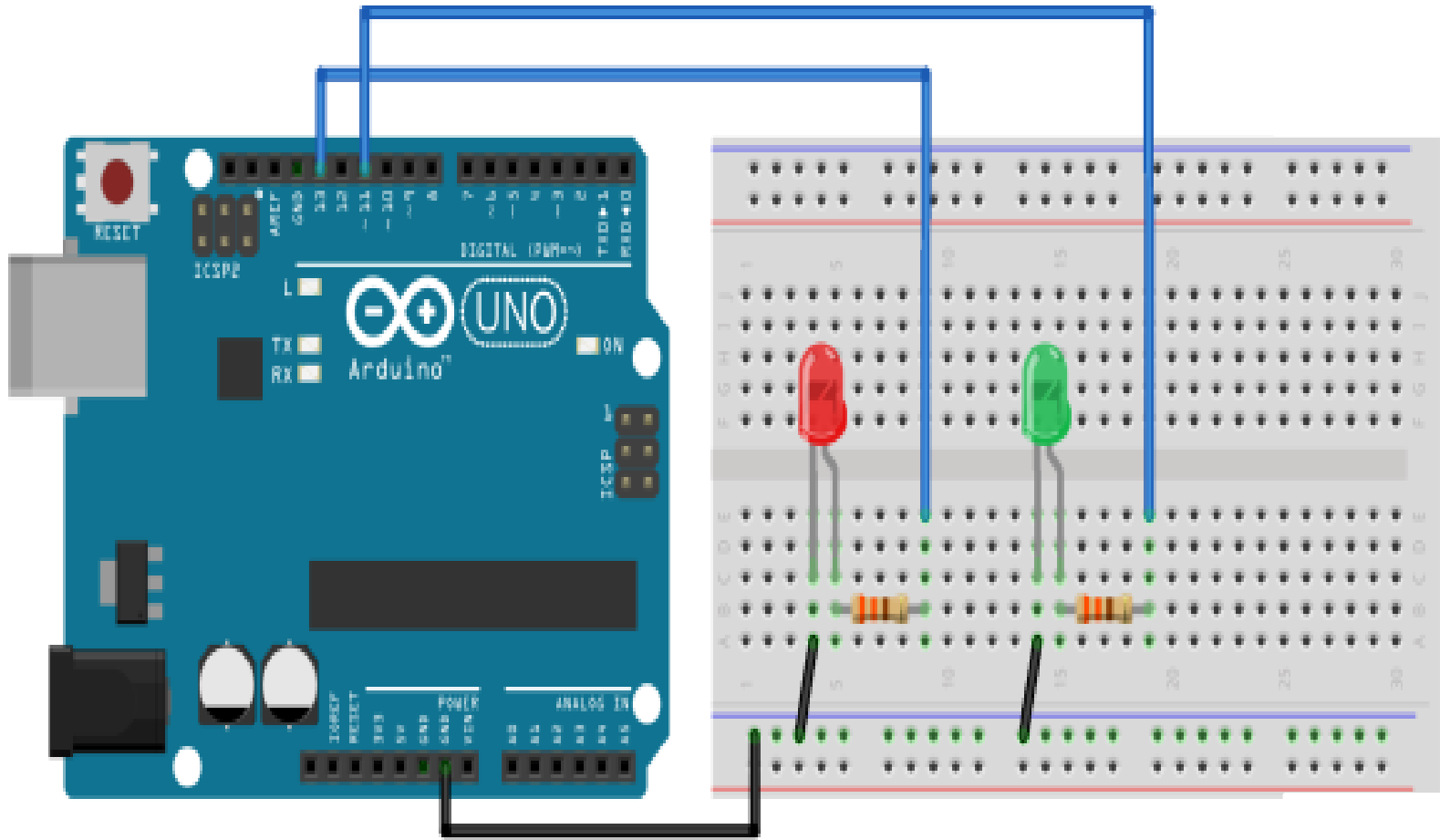
```
delay(500);
```

시간의 단위이며, 1밀리세컨드는 1000분의 1초를 나타낸다.

아두이노 5V 핀에 연결된 선-> 13번 핀  
->10번으로 변경하기->스케치 프로그램 수정하기



# LED 두 개가 교차하며 깜박거리게 만들기

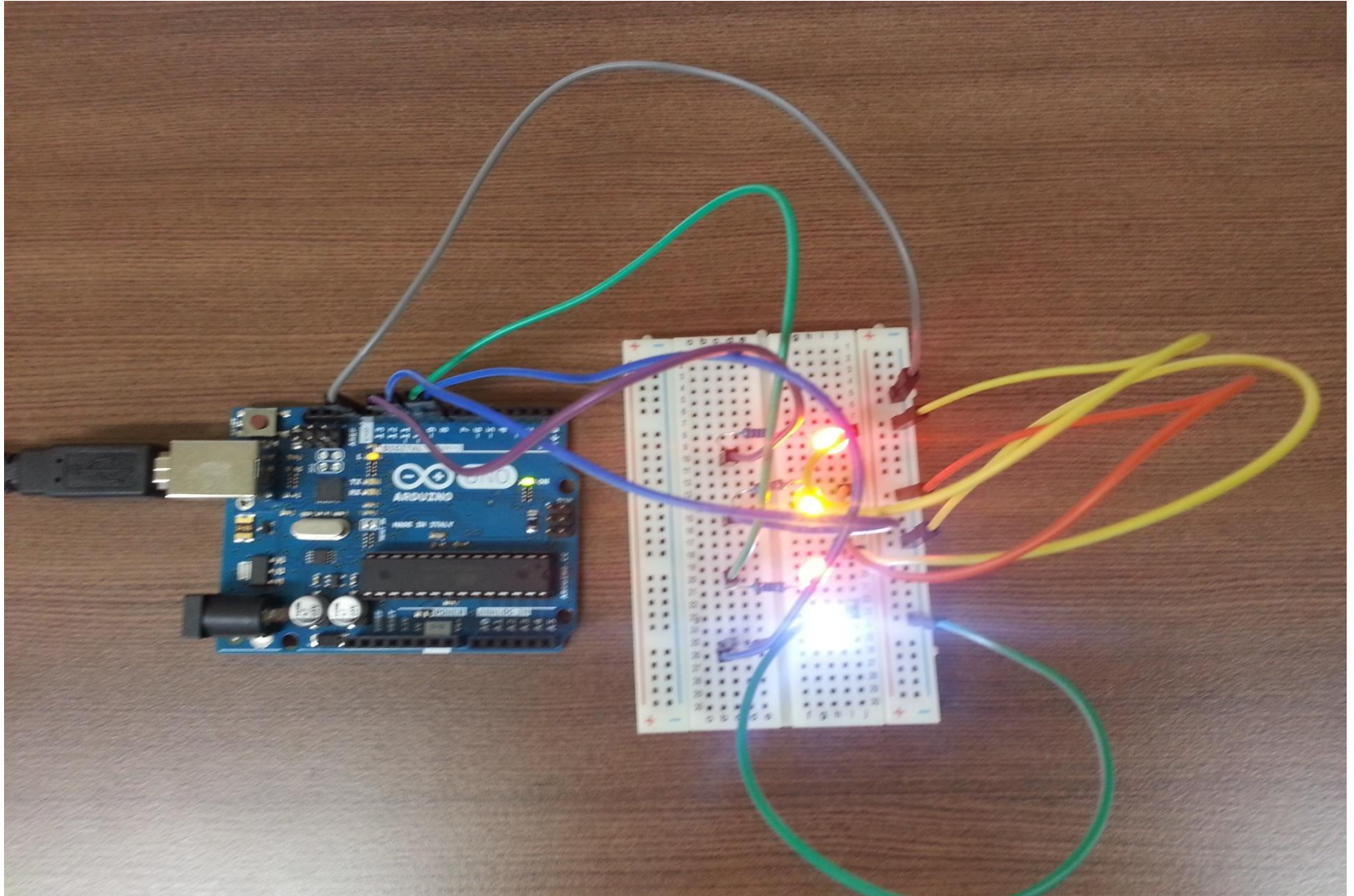


회로 연결: 13번 핀과 11번 핀에 LED와 저항을 각각 연결해 준다.

# LED 두 개가 교차하며 깜박거리게 만들기

```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(11, OUTPUT);  
}  
void loop() {  
  digitalWrite(13, HIGH);  
  digitalWrite(11, LOW);  
  delay(1000);  
  digitalWrite(13, LOW);  
  digitalWrite(11, HIGH);  
  delay(1000);  
}
```

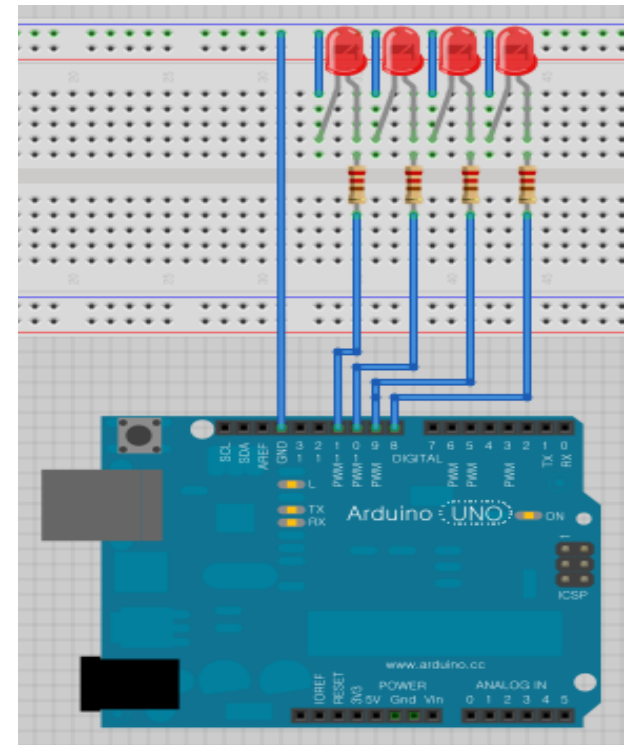
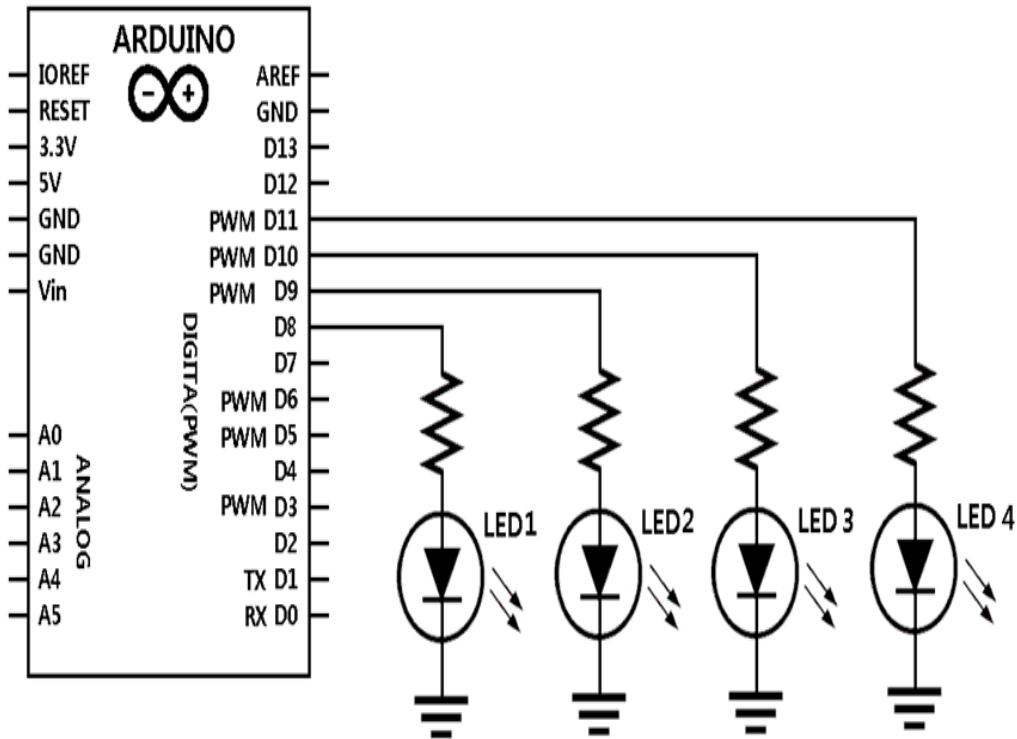
# LED 4개의 ON-OFF 제어





# LED 4개 on/off 동시 제어

- 여러 개의 LED의 on/off 상태 제어
- 디지털 입출력 핀 4개에 LED 4개 연결



# LED 4개의 ON-OFF 제어

- 4개의 LED 동시 ON-OFF



```
void setup() {  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
  pinMode(10, OUTPUT);  
  pinMode(11, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(8, HIGH);  
  digitalWrite(9, HIGH);  
  digitalWrite(10, HIGH);  
  digitalWrite(11, HIGH);  
  delay(1000);  
  
  digitalWrite(8, LOW);  
  digitalWrite(9, LOW);  
  digitalWrite(10, LOW);  
  digitalWrite(11, LOW);  
  delay(1000);  
}
```

# LED 4개의 ON-OFF 제어

파일 편집 스케치 도구 도움말



Blink\_4 \$

```
This example code is in the public domain.
*/

// Pin 10 has an LED connected on most Arduino boards.
// give it a name:

// the setup routine runs once when you press reset:
void setup() {

  // initialize the digital pin as an output.
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(8, HIGH); // turn the LED on (HIGH is the voltage level)
  digitalWrite(9, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(11, HIGH);

  delay(1000); // wait for a second
  digitalWrite(8, LOW); // turn the LED off by making the voltage LOW
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);

  delay(1000); // wait for a second
}
```

# 아날로그 신호 입출력

# 아날로그 신호 입력

## ○ analogRead()

✦ 인자 → 핀 번호

✦ 아날로그 신호 읽어 옴

✦ 아날로그 신호를 10 비트로 디지털화

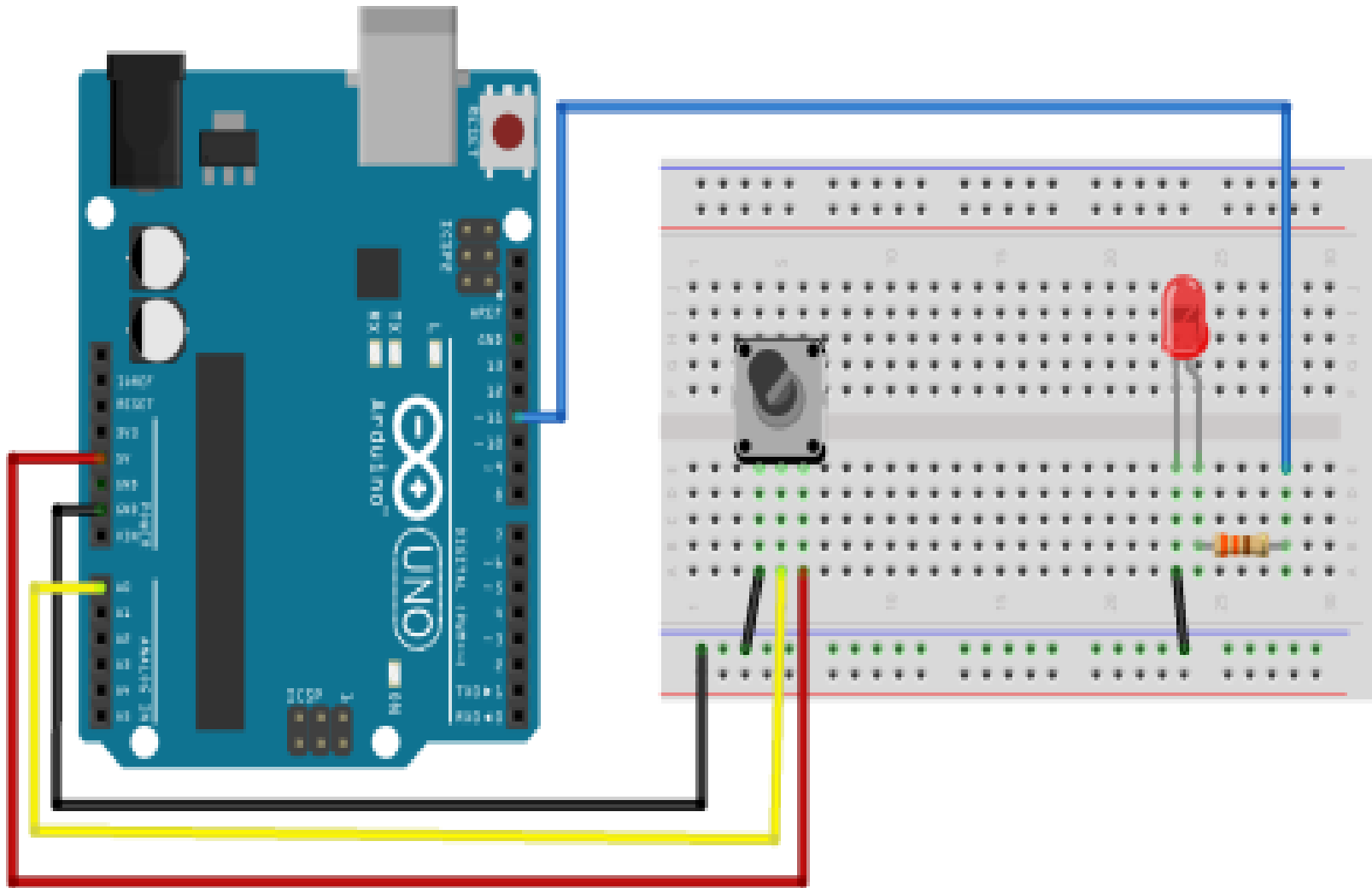
○ 0~5[V] → 0 ~ 1023

✦ 디지털 값으로 아날로그 전압 값 계산

○  $\text{sensorValue} * (5.0/1024.0)$

○  $\text{sensorValue} * (5.0/1023.0)$

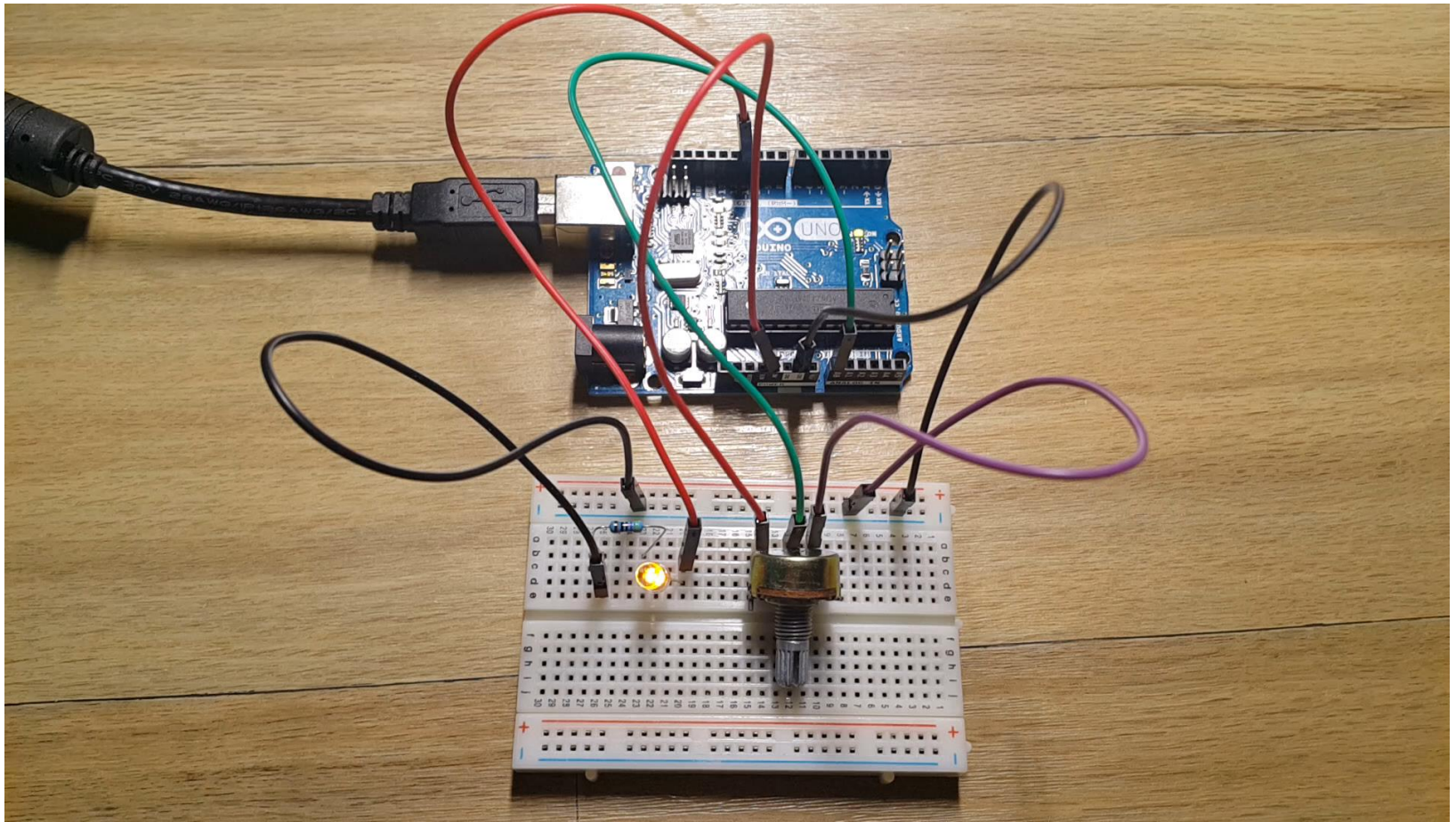
# 가변저항으로 LED깜박임 속도 변경



# 스케치 프로그램

```
void setup() {  
  pinMode(11, OUTPUT);  
}  
void loop() {  
  int sensorValue = analogRead(A0);  
  digitalWrite(11, HIGH);  
  delay(sensorValue);  
  //sensorValue = analogRead(A0);  
  digitalWrite(11, LOW);  
  delay(sensorValue);  
}
```

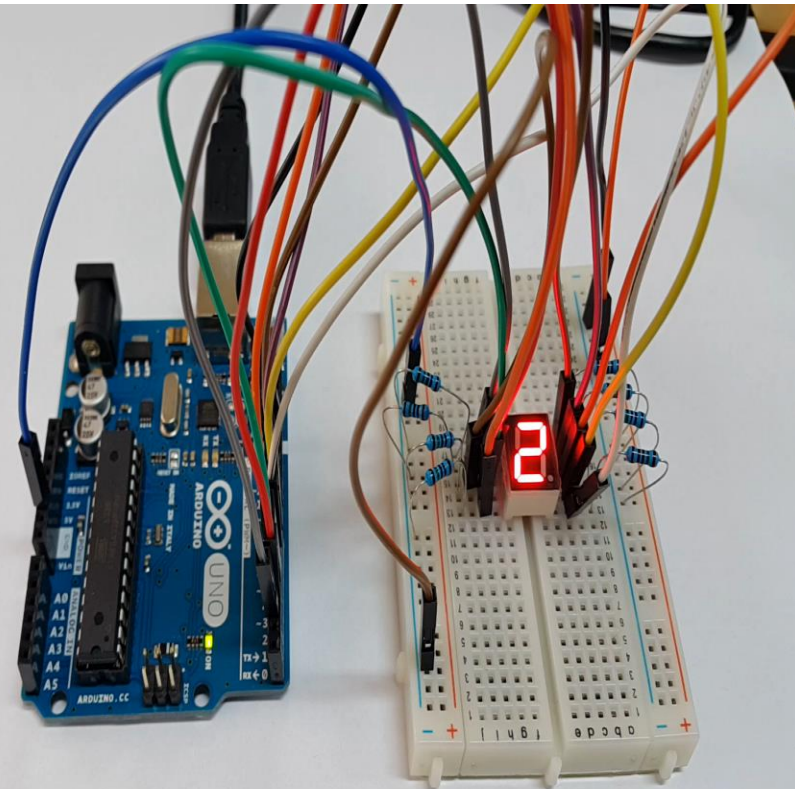
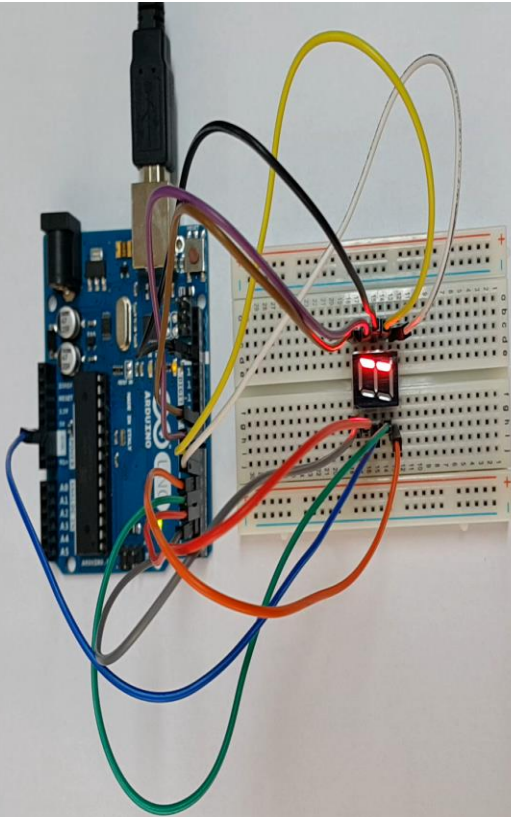
# 스케치 프로그램 실행 장면





# 7 세그먼트 구동하기

- 준비물: 아두이노 Uno, 공통 캐소드 Type MT03911AR 1개, 브레드보드, 470 $\Omega$  저항 8개



# 7 세그먼트 구동 설계하기

## 프로그램 작성

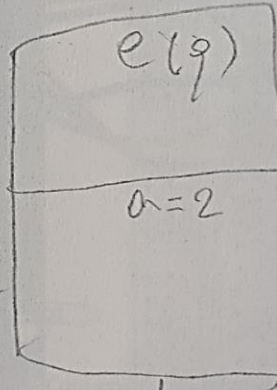
cathod 33

아두이노 스케치에 아래와 같은 코드를 작성한다.

```
const int a=2;
const int b=3;
const int c=4;
const int d=5;
const int e=9;
const int f=8;
const int g=7;
const int dot=6;
```

2 a  
3 b  
GND  
4 c  
5 d

b=3  
c=4



e 9  
f 8  
GND  
g 7  
dot  
dot(6)

```
void display_number(int n)
{
```

# 7 세그먼트 스케치 프로그램(led\_7segment.ino)

```
const int _a=2;
const int _b=3;
const int _c=4;
const int _d=5;
const int _e=9;
const int _f=8;
const int _g=7;
const int _dot=6;
```

```
void display_number(int n)
{
    switch(n)
    {
```

case 0:

```
digitalWrite(_e,HIGH);
digitalWrite(_b,HIGH);
digitalWrite(_c,HIGH);
digitalWrite(_d,HIGH);
digitalWrite(_g,HIGH);
digitalWrite(_f,HIGH);
digitalWrite(_a,LOW);
digitalWrite(_dot,LOW);
break;
```

case 1:

```
digitalWrite(_e,LOW);
digitalWrite(_b,LOW);
digitalWrite(_c,LOW);
digitalWrite(_d,LOW);
digitalWrite(_g,HIGH);
digitalWrite(_f,HIGH);
digitalWrite(_a,LOW);
digitalWrite(_dot,LOW);
break;
```

# 7 세그먼트 구동 스케치 프로그램

case 2:

```
digitalWrite(_e,HIGH);  
digitalWrite(_b,LOW);  
digitalWrite(_c,HIGH);  
digitalWrite(_d,HIGH);  
digitalWrite(_g,LOW);  
digitalWrite(_f,HIGH);  
digitalWrite(_a,HIGH);  
digitalWrite(_dot,LOW);  
break;
```

case 3:

```
digitalWrite(_e,HIGH);  
digitalWrite(_b,LOW);  
digitalWrite(_c,LOW);  
digitalWrite(_d,HIGH);  
digitalWrite(_g,HIGH);  
digitalWrite(_f,HIGH);  
digitalWrite(_a,HIGH);  
digitalWrite(_dot,LOW);  
break;
```

case 4:

```
digitalWrite(_e,LOW);  
digitalWrite(_b,HIGH);  
digitalWrite(_c,LOW);  
digitalWrite(_d,LOW);  
digitalWrite(_g,HIGH);  
digitalWrite(_f,HIGH);  
digitalWrite(_a,HIGH);  
digitalWrite(_dot,LOW);  
break;
```

case 5:

```
digitalWrite(_e,HIGH);  
digitalWrite(_b,HIGH);  
digitalWrite(_c,LOW);  
digitalWrite(_d,HIGH);  
digitalWrite(_g,HIGH);  
digitalWrite(_f,LOW);  
digitalWrite(_a,HIGH);  
digitalWrite(_dot,LOW);  
break;
```

```

case 6:
    digitalWrite(_e,HIGH);
    digitalWrite(_b,HIGH);
    digitalWrite(_c,HIGH);
    digitalWrite(_d,HIGH);
    digitalWrite(_g,HIGH);
    digitalWrite(_f,LOW);
    digitalWrite(_a,HIGH);
    digitalWrite(_dot,LOW);
    break;
case 7:
    digitalWrite(_e,HIGH);
    digitalWrite(_b,HIGH);
    digitalWrite(_c,LOW);
    digitalWrite(_d,LOW);
    digitalWrite(_g,HIGH);
    digitalWrite(_f,HIGH);
    digitalWrite(_a,LOW);
    digitalWrite(_dot,LOW);
    break;
case 8:
    digitalWrite(_e,HIGH);
    digitalWrite(_b,HIGH);
    digitalWrite(_c,HIGH);
    digitalWrite(_d,HIGH);
    digitalWrite(_g,HIGH);
    digitalWrite(_f,HIGH);
    digitalWrite(_a,HIGH);
    digitalWrite(_dot,LOW);
    break;

```

```

case 9:
    digitalWrite(_e,HIGH);
    digitalWrite(_b,HIGH);
    digitalWrite(_c,LOW);
    digitalWrite(_d,HIGH);
    digitalWrite(_g,HIGH);
    digitalWrite(_f,HIGH);
    digitalWrite(_a,HIGH);
    digitalWrite(_dot,LOW);
    break;
case 10:
    digitalWrite(_e,LOW);
    digitalWrite(_b,LOW);
    digitalWrite(_c,LOW);
    digitalWrite(_d,LOW);
    digitalWrite(_g,LOW);
    digitalWrite(_f,LOW);
    digitalWrite(_a,LOW);
    digitalWrite(_dot,HIGH);
    break;
}
}

```

```

void setup()
{
    pinMode(_a,OUTPUT);
    pinMode(_b,OUTPUT);
    pinMode(_c,OUTPUT);
    pinMode(_d,OUTPUT);
    pinMode(_e,OUTPUT);
    pinMode(_f,OUTPUT);
    pinMode(_g,OUTPUT);
    pinMode(_dot,OUTPUT);
};

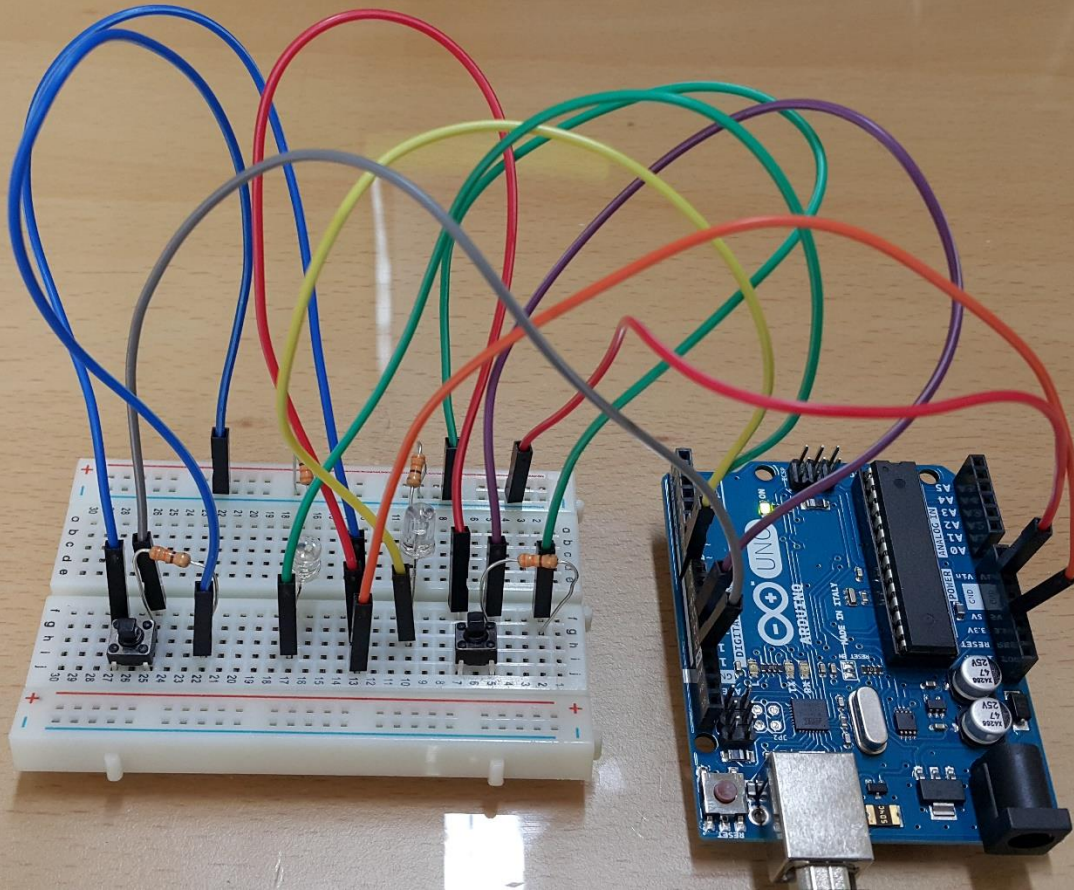
void loop()
{
    int i;

    for(i=0;i<11;i++)
    {
        display_number(i);
        delay(1000);
    }
}

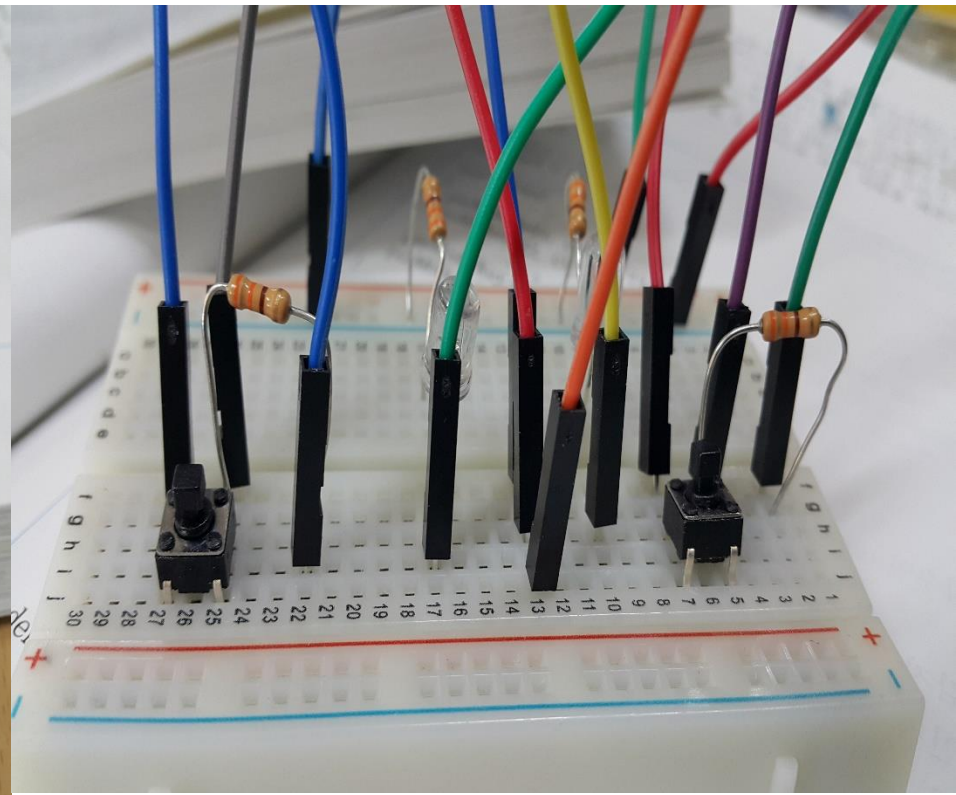
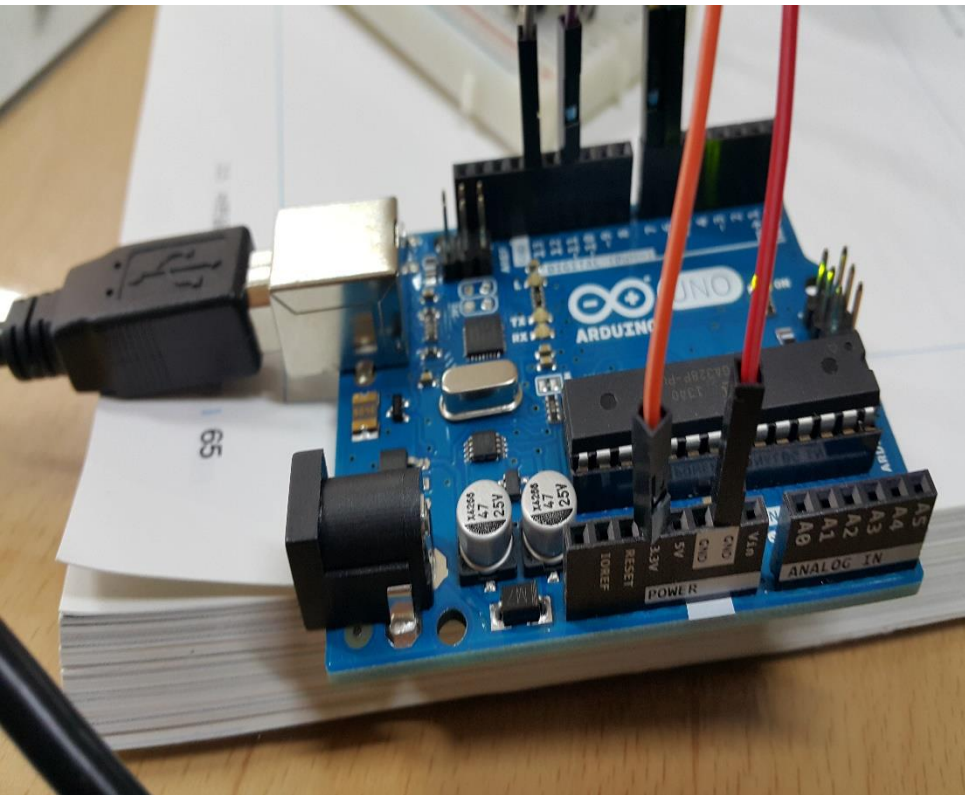
```

# 버튼으로 Led 제어하기

- 준비물: 아두이노 Uno, Led 2개, 1K저항 2개, 10K저항2개, 브레드보드, 푸시 버튼 2개



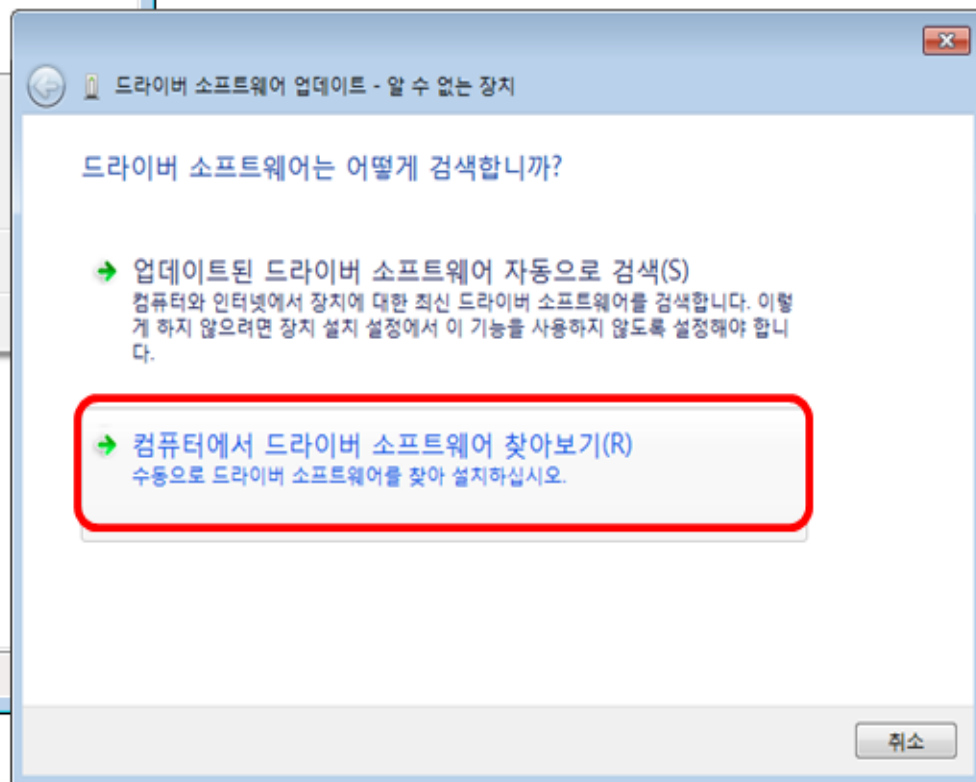
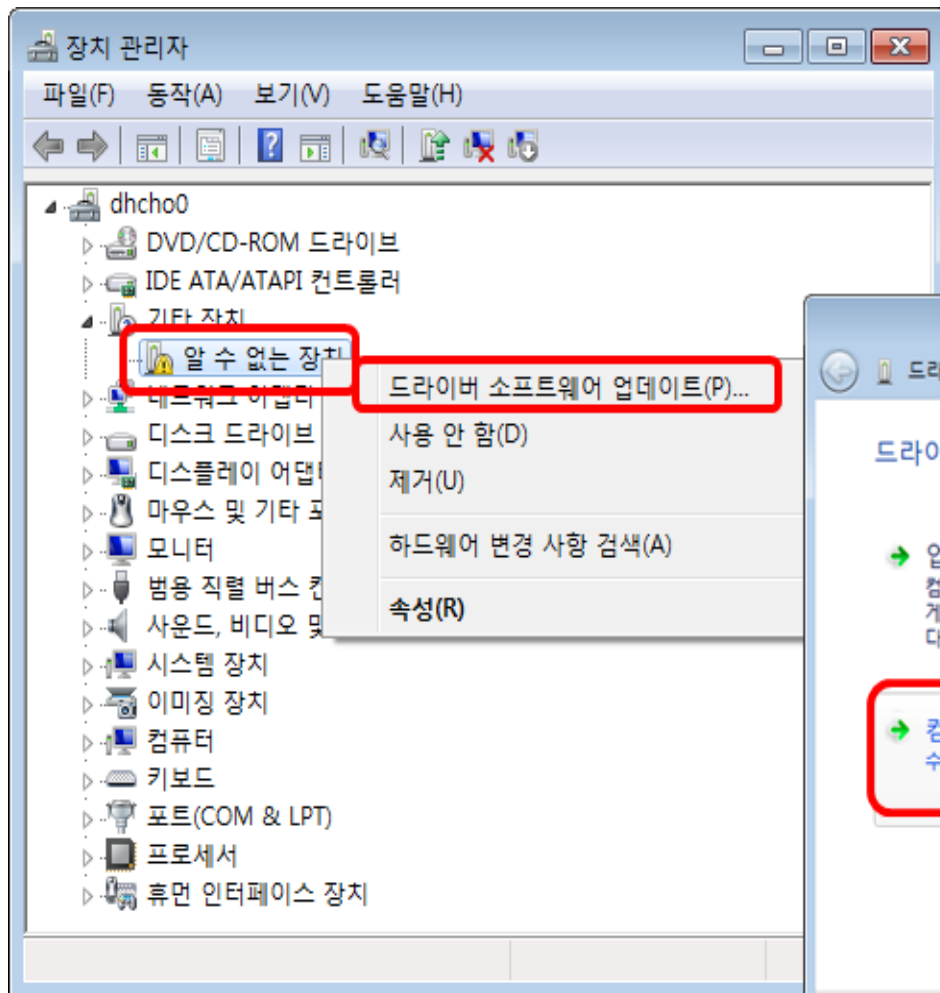
# 버튼으로 Led 제어하기(65쪽)



# 아두이노 UNO 보드 드라이버 설치 방법

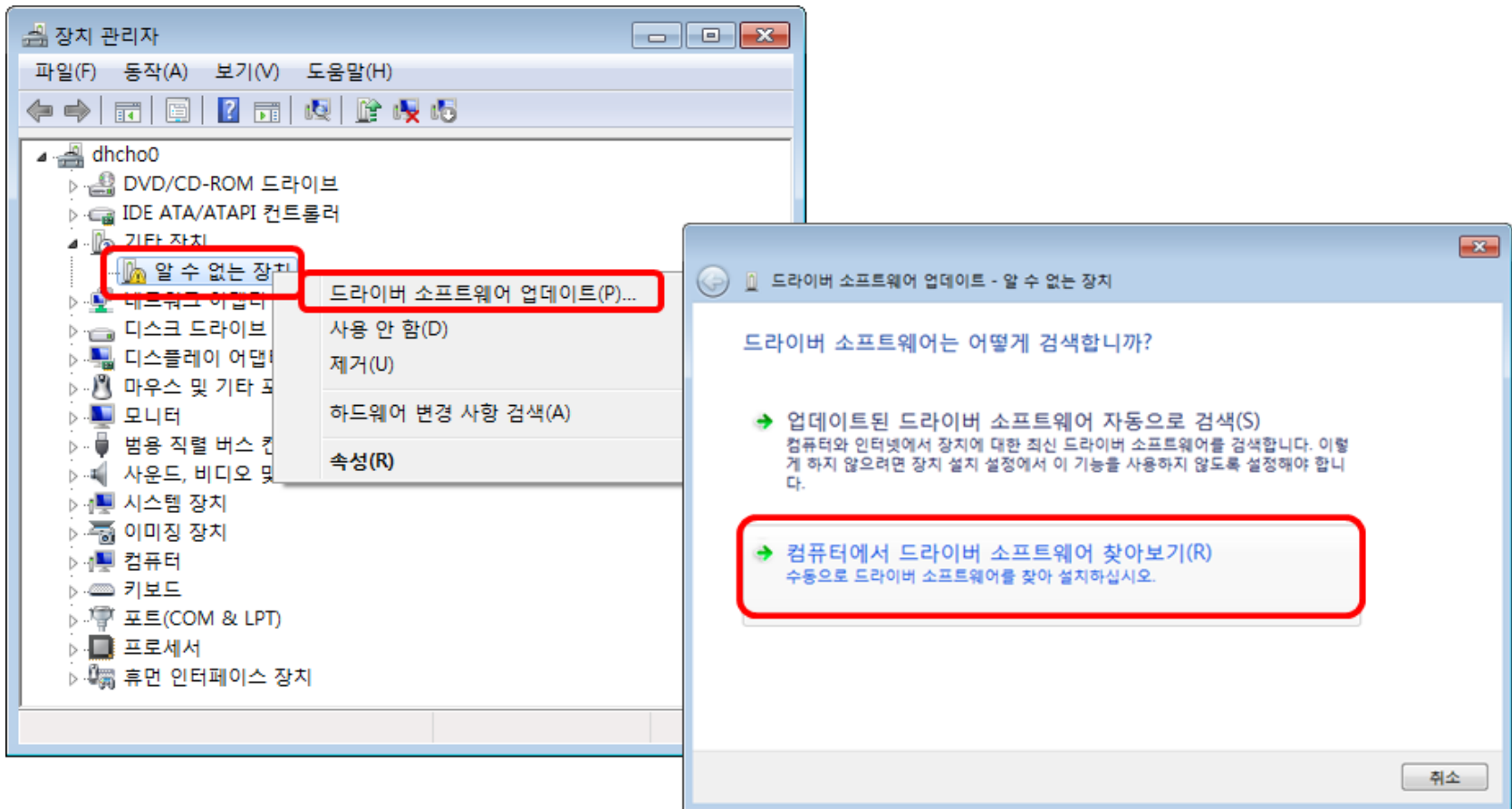


# 1. 제어판-시스템 및 보안-장치관리자를 찾는다.

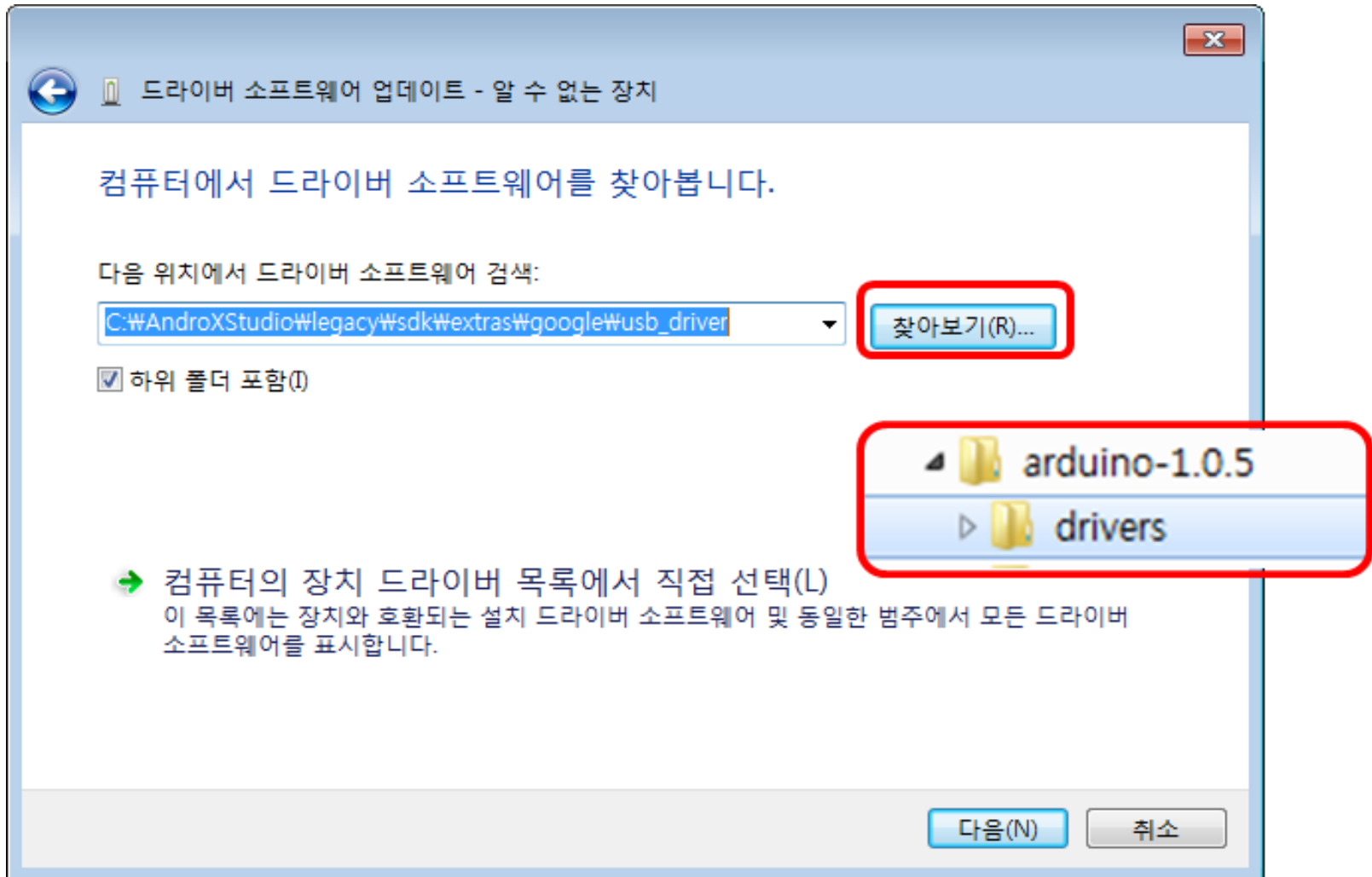


## 2. 드라이버 소프트웨어 업데이트를 한다.

: 장치 관리자-> 알 수 없는 장치->마우스 오른쪽 버튼->  
"드라이버 소프트웨어 업데이트"를 선택->개발용 PC에서 드라이버  
소프트웨어 찾아보기(R)를 선택



### 3. 개발환경에서 설치한 아두이노 프로그램이 있는 폴더를 찾아서 그 밑에 있는 drivers 폴더 선택



## 4. drivers 폴더 선택 후 '설치'버튼 클릭



## 5. 설치 완료 창 확인

