

Arduino

- 2005년 이탈리아 IDII(Interaction Design Institute Ivrea)의 (Massimo Banzi)교수
- Open-source H/W, S/W platform
- Controller: Atmel AVR기반, Uno(Atmega 328), Mega(Atmega2560), Due(32bit ARM Cortex-M3)
- Board: Uno, Mega, Yun, Due
- Sketch: Arduino Development Tool (IDE: Integrated Development Environment)

Arduino series

ENTRY LEVEL	ARDUINO UNO	ARDUINO 101	ARDUINO PRO	ARDUINO PRO MINI	ARDUINO MICRO
	ARDUINO NANO	ARDUINO STARTER KIT	ARDUINO BASIC KIT	ARDUINO MOTOR SHIELD	
ENHANCED FEATURES	ARDUINO MEGA	ARDUINO ZERO	ARDUINO DUE	ARDUINO PROTO SHIELD	
INTERNET OF THINGS	ARDUINO YÚN	ARDUINO MKR1000	ARDUINO ETHERNET SHIELD	ARDUINO GSM SHIELD	
	ARDUINO WIFI SHIELD 101				
WEARABLE	ARDUINO GEMMA	LILYPAD ARDUINO USB	LILYPAD ARDUINO MAIN BOARD		
	LILYPAD ARDUINO SIMPLE	LILYPAD ARDUINO SIMPLE SNAP			
3D PRINTING	MATERIA 101				

BOARDS
MODULES
SHIELDS
KITS
ACCESSORIES
COMING NEXT

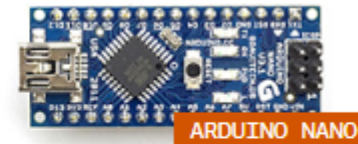
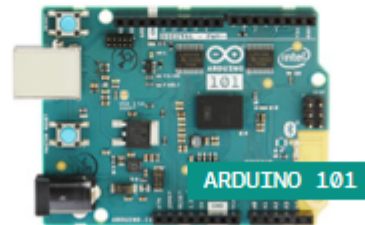
Genuino

ENTRY LEVEL	GENUINO UNO	GENUINO 101	GENUINO MICRO	GENUINO STARTER KIT
ENHANCED FEATURES	GENUINO MEGA	GENUINO ZERO		
INTERNET OF THINGS	GENUINO MKR1000			

BOARDS MODULES KITS COMING NEXT

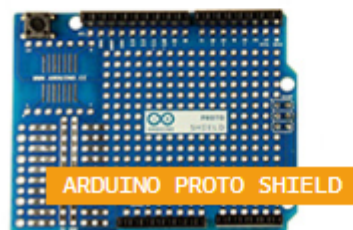
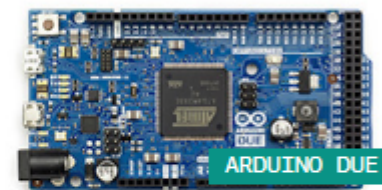
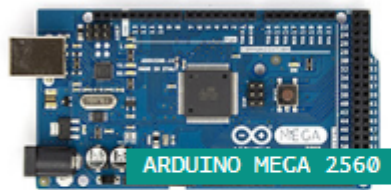
Entry Level

Get started with Arduino using Entry Level products: easy to use and ready to power your first creative projects. These boards and modules are the best to start learning and tinkering with electronics and coding. The StarterKit includes a book with 15 tutorials that will walk you through the basics up to complex projects.



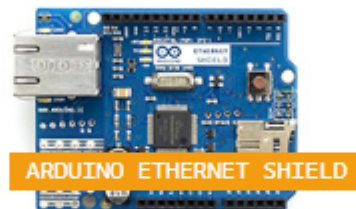
Enhanced Features

Experience the excitement of more complex projects choosing one of the boards with advanced functionalities, or faster performances.



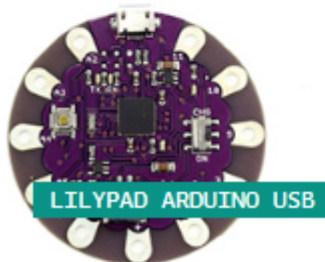
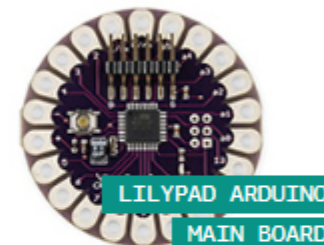
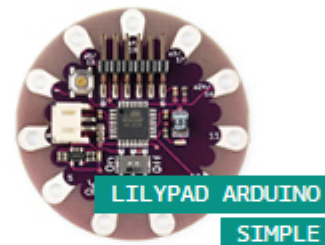
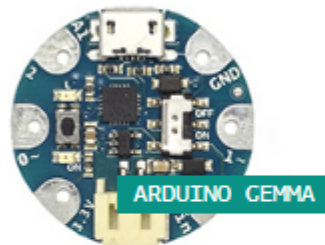
Internet of Things

Make connected devices easily with one of these IoT products and open your creativity with the opportunities of the world wide web.



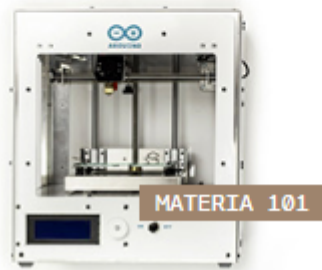
Wearable

Add smartness to your soft projects and discover the magic of sewing the power of electronics directly to textiles.



3D Printing

The Arduino approach to 3D printing is represented by Materia 101, a printer that allows you to start experimenting with this amazing technology in the easiest way.



Retired

Explore the history of Arduino with a journey through all the boards, accessories, shield, kits and documentation released since 2006.



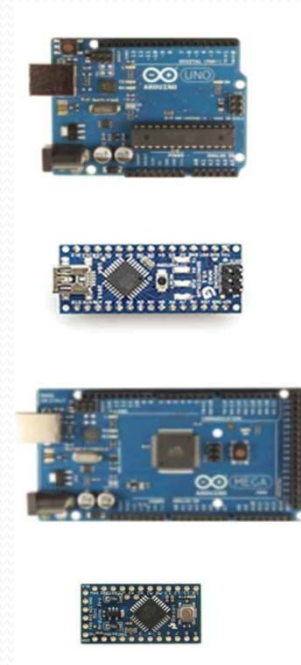
Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16MHz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168	5 V/7-9 V	16MHz	8/0	14/6	0.512	1	16	Mini-B	1
	ATmega328					1	2	32		
Ethernet	ATmega328	5 V/7-12 V	16MHz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16MHz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8MHz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
		5 V/5-12 V	16MHz							
LilyPad	ATmega168V	2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
	ATmega328V	V/2.7-5.5 V								
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8MHz	4/0	9/4	1	2.5	32	Micro	-
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
Yun	ATmega32u4	5 V	16MHz	12/0	20/7	1	2.5	32	Micro	1

Arduino Retired boards specs

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
BT	ATmega328P	5 V / 2.5-12 V	16 MHz	6/0	14/6	1	2	32	-	1
Esplora	ATmega32U4	5 V / 7-12 V	16 MHz	-	-	1	2.5	32	Micro	-
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4	1	2	32	Regular	-
Fio	ATmega328P	3.3 V / 3.7-7 V	8 MHz	8/0	14/6	1	2	32	Mini	1
Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-

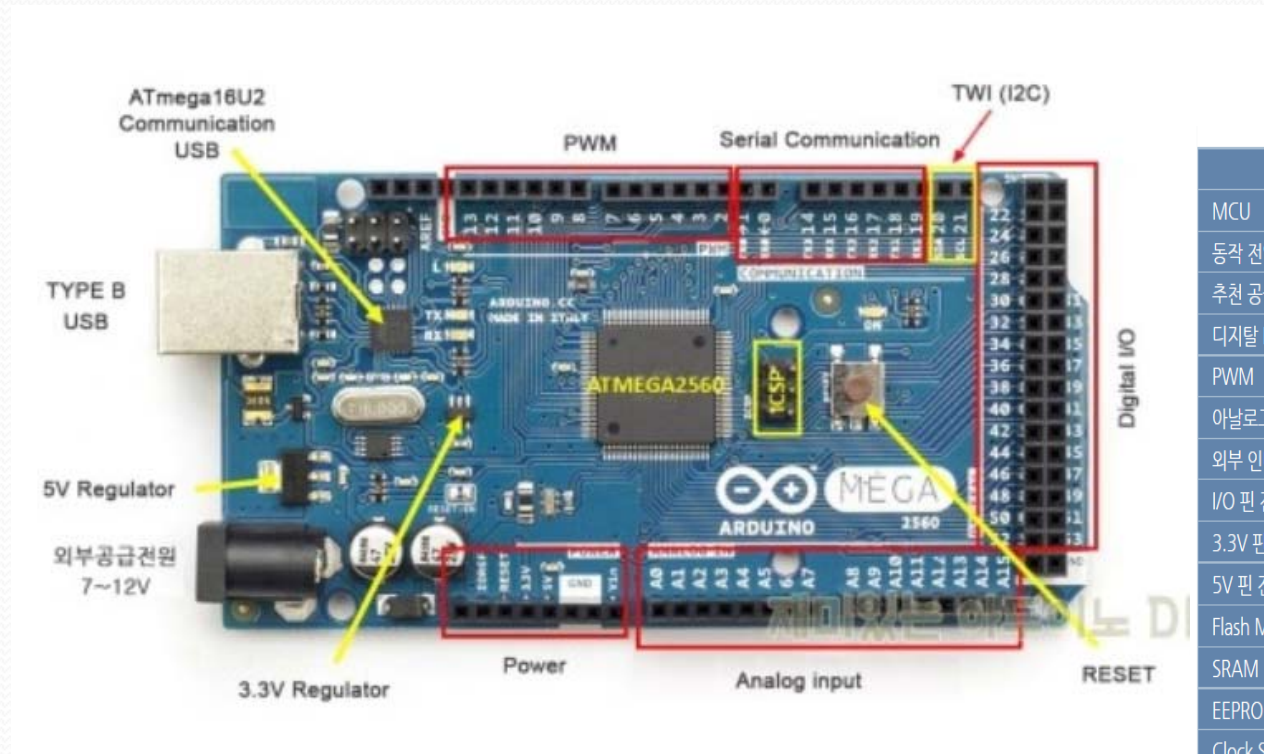
Arduino Board

- UNO : Standard Board
- Nano : UNO의 소형화 버전
- Pro/Pro mini : UNO 초소형화 버전
(3.3v/5v 버전이 별도로 존재)
- Mega2560 : UNO의 확장 버전
- Due (32bit, ARM Cortex-M3)
- Shield: Arduino 보드의 확장용 보드
(예: WiFi Shield, Ethernet Shield)



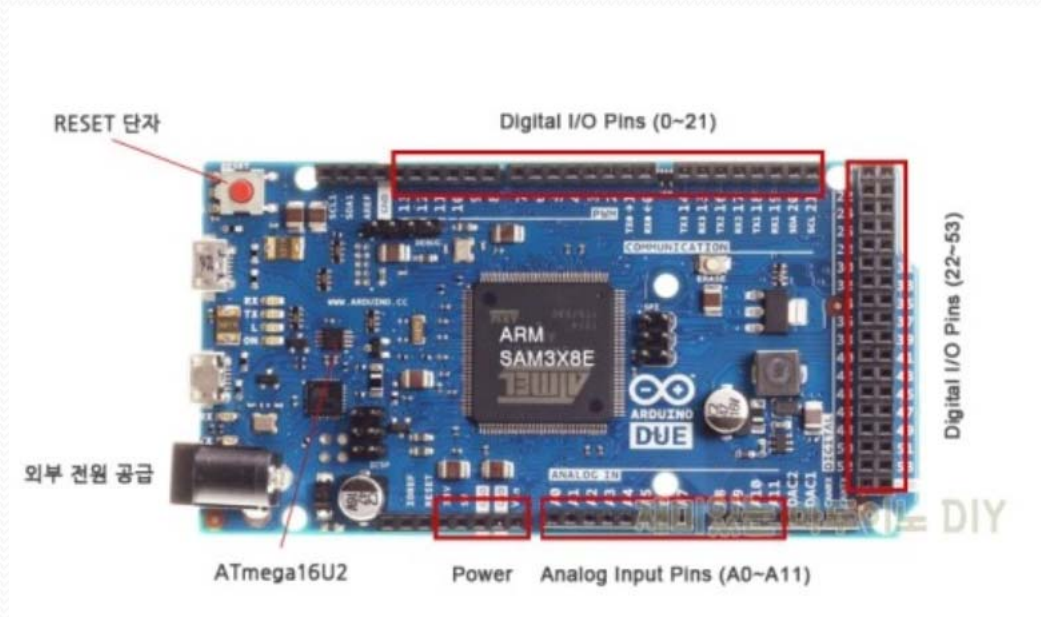
- Duemilanoves : UNO의 구버전
- Leonardo : UNO와 유사, 핀 배열과 개수가 약간 틀림
- MegaADK : 안드로이드 ADK 지원보드 (USB Host 기능)
- Micro : Leonardo 보드의 소형화 버전
- Pro micro : Nano와 유사하지만 3.3/5v 별도로 존재
- Zero : ARM core, 고성능 버전
- Tre, Yun : 아두이노 + 리눅스
- Lily Pad : 웨어러블용(의류용) 특화 보드
- FIO (Funnel IO) : 무선통신 특화 보드

Arduino Mega 2560



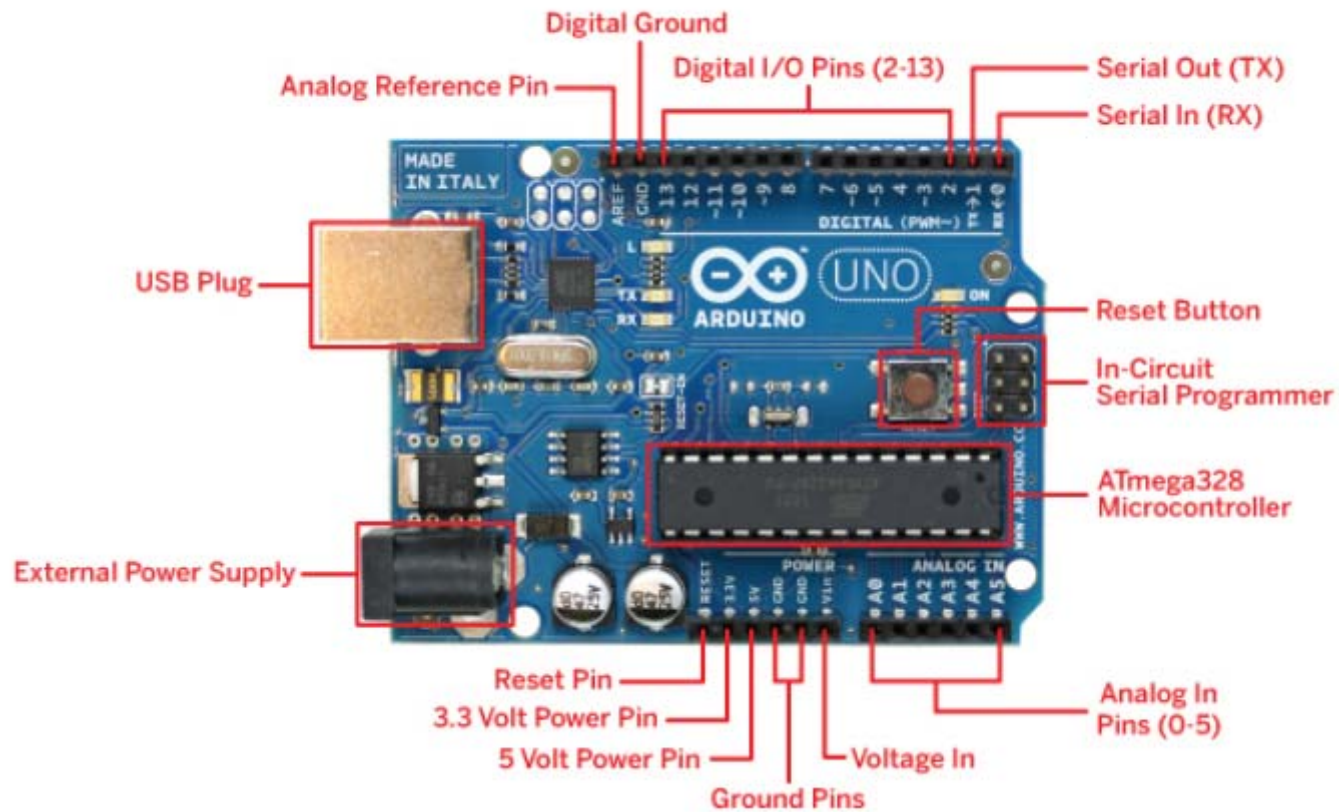
	MEGA 2560
MCU	ATmega2560
동작 전압	5 V
추천 공급 전압	7~12 V
디지털 I/O 핀	54 (PWM 15개 포함)
PWM	15
아날로그 입력 핀	16
외부 인터럽트	6개 (2, 3, 18, 19, 20, 21)
I/O 핀 전류	20 mA
3.3V 핀 전류	50 mA
5V 핀 전류	800 mA
Flash Memory	256 KB (부트로더 8KB 사용)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Arduino Due



	듀에 (Due)
MCU	AT91SAM3X8E
동작 전압	3.3V
추천 공급 전압	7~12V
디지털 I/O 핀	54 (PWM 12개 포함)
PWM	12
아날로그 입력 핀	12
아날로그 출력 핀	2 (DAC)
외부 인터럽트	모든 디지털 입출력 핀
I/O 핀 전류	130 mA
3.3V 핀 전류	800 mA
5V 핀 전류	800 mA
Flash Memory	512 KB
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz

Arduino UNO



ATmega328 Pin Mapping

Arduino function

reset
digital pin 0 (RX)
digital pin 1 (TX)
digital pin 2
digital pin 3 (PWM)
digital pin 4
VCC
GND
crystal
crystal
digital pin 5 (PWM)
digital pin 6 (PWM)
digital pin 7
digital pin 8

(PCINT14/RESET) PC6
(PCINT16/RXD) PD0
(PCINT17/TXD) PD1
(PCINT18/INT0) PD2
(PCINT19/OC2B/INT1) PD3
(PCINT20/XCK/T0) PD4
VCC
GND
(PCINT6/XTAL1/TOSC1) PB6
(PCINT7/XTAL2/TOSC2) PB7
(PCINT21/OC0B/T1) PD5
(PCINT22/OC0A/AIN0) PD6
(PCINT23/AIN1) PD7
(PCINT0/CLKO/CP1) PB0



PC5 (ADC5/SCL/PCINT13)
PC4 (ADC4/SDA/PCINT12)
PC3 (ADC3/PCINT11)
PC2 (ADC2/PCINT10)
PC1 (ADC1/PCINT9)
PC0 (ADC0/PCINT8)
GND
AREF
AVCC
PB5 (SCK/PCINT5)
PB4 (MISO/PCINT4)
PB3 (MOSI/OC2A/PCINT3)
PB2 (SS/OC1B/PCINT2)
PB1 (OC1A/PCINT1)

Arduino function

analog input 5
analog input 4
analog input 3
analog input 2
analog input 1
analog input 0
GND
analog reference
VCC
digital pin 13
digital pin 12
digital pin 11 (PWM)
digital pin 10 (PWM)
digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega 168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

아두이노 우노 Arduino UNO	아두이노 종류
ATmega328P	MCU종류
5V	작동 전압
7-12V	입력 전압(권장)
6-20V	입력 전압(최대)
14	디지털 입출력 핀
6	아날로그 입력 핀
6	PWM 출력 핀
20mA	입출력핀 당 최대전류
50mA	3.3V핀 최대전류
32KB*	플래시메모리
2KB	SRAM
1KB	EEPROM
16MHz	클럭 속도
68.6mm	길이
53.4mm	폭
25g	무게
USB B	프로그래밍 단자

Arduino 전원 공급

- USB 케이블
- 배럴잭 (DC 어댑터, 배터리와 배터리 홀더)
- VIN과 GND 를 통한 전원 공급 (일반 전선을 사용할 경우)
- 5V, 3.3V: 외부 모듈 전원 공급용

디지털 핀 (INPUT / OUTPUT)

- 5V 동작 (전압 조절은 불가)
- pin 0~13
- Input/output
- HIGH (5V, on), LOW (0V, off)
- GPIO (General Purpose Input Output)
- Pin setting: pinMode(pinNumber, mode)
- Input: digitalRead(pinNumber)
→ on, off = 5V, 0V = HIGH, LOW
- Output: digitalWrite(pinNumber, state)
→ on, off = 5V, 0V = HIGH, LOW

아날로그 핀 (INPUT)

- input 전용
- A0~A5 (A4, A5 핀은 I2C 핀으로 예약 (SDA, SCL))
- 들어오는 전압에 따라 구분된 값을 읽을 수 있음
- 0~5V 를 1024 단계로 구분된 값으로 변환(10bit resolution)
- ADC (Analog to Digital Converter)
- 초기화가 필요 없음 : input 모드만 가능
- analogRead(pinNumber)
- 입력된 전압에 따라 0~1023 사이의 값 출력

Hardware serial

- 0, 1 Pin(RX, TX)

SPI 통신용 Pin

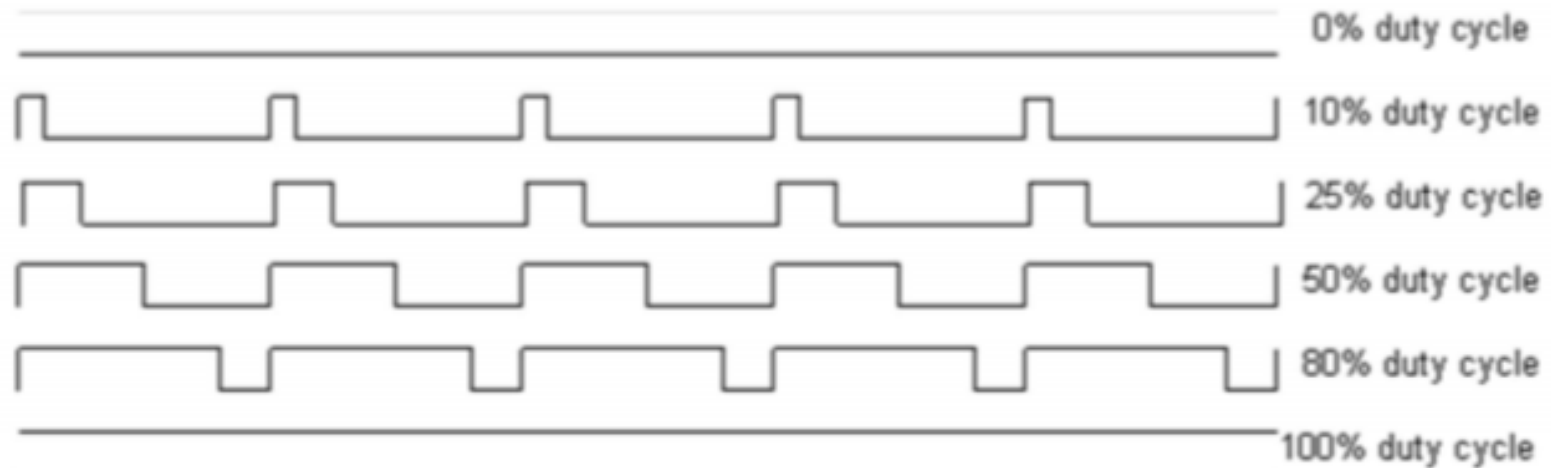
- 10, 11, 12, 13 (SS, MOSI, MISO, SCK)

PWM (Pulse Width Modulation) Pin

- Output: **analogWrite(9, 출력값)**
- Pin 3,5,6,9,10,11 (UNO 보드: ~ 표시): Digital I/O Pin 사용
- 아두이노는 출력 전압을 조절하는 능력이 없음
→ LED 밝기 조절? : 출력 전압 조절을 PWM으로 사용

PWM Theory

- analogWrite() 함수 사용 → analog 출력
- 5V 출력을 계속 유지하는 것이 아니라 on/off 를 빠르게 교번
- on/off 간격(pulse width)에 따라 평균전압이 변하는 효과
- 아두이노에서는 500Hz 주파수, 2ms 간격으로 이런 작업을 수행
- 256 단계로 구분해서 출력이 가능 (0~255, 8bit)
- 파형으로 본 PWM



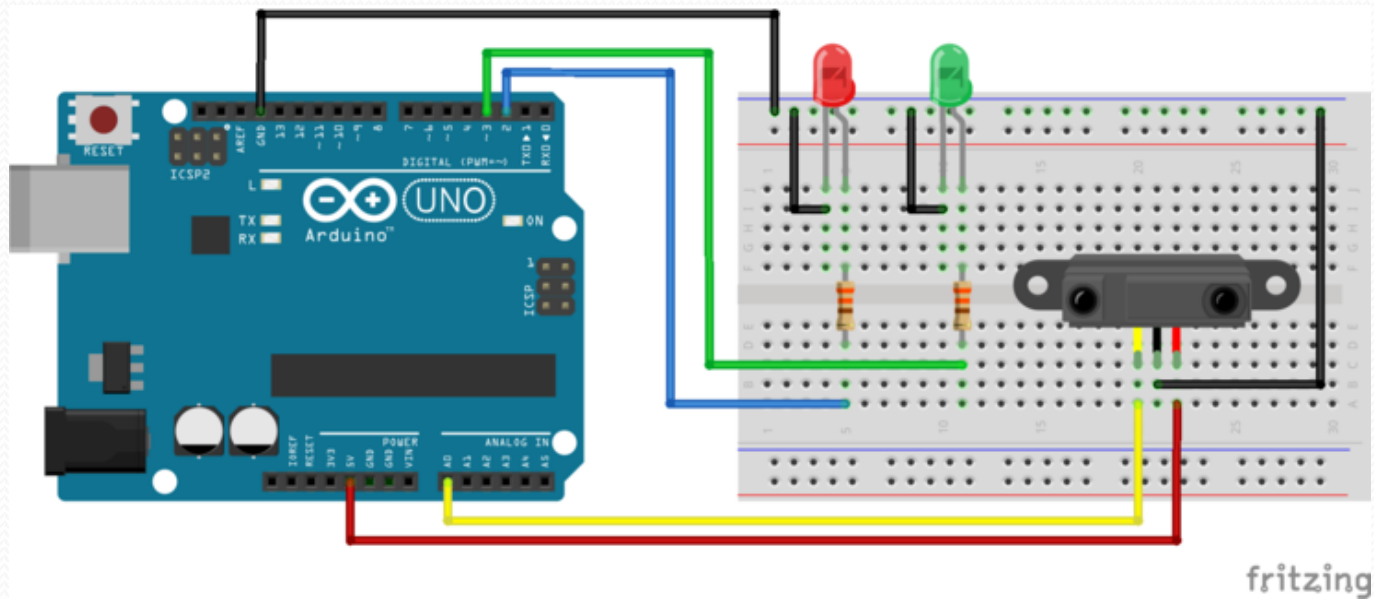
참고: AVR/MCS-51 명령어 체계

- ATmega128의 PORTA에 I/O 명령어 Setting
 - PORTA를 출력으로 지정
DDRA register set to 0xFF (all pins output)
DDRA = 0xFF;
 - PORTA에 출력할 값 지정
PORTA register에 출력할 값을 지정
PORTA = 0xAA; // 1010 1010 B

MCS-51 Families

P1 = 0xAA;

Arduino UNO 결선 예



Arduino UNO에 USB 결선 예



Arduino 명령어

- 명령어()
- 명령어(x,y)
- Digital I/O
 - pinMode(pin, mode): input, output
 - digitalWrite(pin, value): high, low → output
 - digitalRead(pin): high, low → input
- Analog I/O
 - analogWrite(pin, value): 다음 analogWrite 명령어가 올 때까지 특정 duty cycle PWM output
 - → speed control
 - analogRead(pin): analog signal input
- Serial
 - Serial.begin(speed): communication speed
 - Serial.available(): byte 크기로 serial data read 후 반환
 - Serial.flush(): data 비움
 - Serial.println(val): ASCII Code 형식으로 출력
 - Serial.read(): data input
- map(val, 0, 1000, min, max);
 - → analogRead(pin)의 return 값을 min ~ max로 mapping
 - → map(val, 0, 1000, 0, 100);

- **Math**
- $\min(x,y)$ = 두 수중 최소값 반환
- $\max(x,y)$ = 두 수중 최대값 반환
- $\text{abs}(x)$ = 입력된 값의 절대값을 반환
- $\text{constrain}(x,a,b)$ = 첫 번째 인자 값을 두 번째와 세 번째 인자 값의 사이 값으로 제한
- $\text{map}(\text{value}, \text{fromLow}, \text{fromHigh}, \text{toLow}, \text{toHigh})$ = 첫 번째 인자 값을 지정된 범위로 선형 사상하여 반환
- $\text{pow}(\text{base}, \text{exponent})$ = 입력한 밑과 지수 값으로 거듭제곱하여 값을 반환
- $\text{sqrt}(x)$ = 입력한 값을 루트 계산하여 반환

- **Random**
- $\text{randomSeed}(\text{seed})$ = 의사 난수생성기를 초기화
- $\text{random}()$ = 난수값을 받아 반환

- **Time**

- `millis()` = 현재 프로그램이 시작된 이후의 경과시간을 밀리 초 단위로 반환
- `micors()` = 현재 프로그램이 시작된 이후의 경과시간을 마이크로 초 단위로 반환
- `delay()` = 지정된 밀리 초 시간만큼 프로그램을 일시 중지
- `delayMicroseconds()` = 지정된 마이크로 초 시간만큼 프로그램을 일시 중지

- **Bits and Bytes**

- `lowByte(x)` = 입력된 값의 최하위 바이트를 추출
- `highByte(x)` = 입력된 값의 최상위 바이트를 추출(또는 두 번 째로 작은 바이트를 반환)
- `bitRead(x,n)` = 주어진 데이터의 n번째 비트를 읽어 반환
- `bitWrite(x,n,b)` = 주어진 데이터의 n번째 비트에 b값(0 또는 1)을 입력
- `bitSet(x,n)` = 주어진 데이터의 n번째 비트를 1로 설정
- `bitClear(x,n)` = 주어진 데이터의 n번째 비트를 0으로 설정
- `bit(n)` = 지정된 비트위치(n번째)에 해당하는 비트 값을 계산하여 반환

- **Interrupt**

- `interrupts()` = `nointerrupts()`에 의해 금지된 인터럽트의 발생을 허용
- `nointerrupts()` = 인터럽트의 발생을 금지, 나중에 `interrupts()`을 통해 다시 발생을 허용

- **Advanced IO**

- `tone(pin, frequency)` = 입력된 핀 번호를 입력된 주파수를 가지는 square wave를 생성, 시간이 입력되지 않았을 경우 `notone()` 함수가 출력될 때까지 지속
- `noTone(pin)` = 특정 핀에서 `tone()`함수에 의한 square wave 생성을 중지
- `shiftOut(dataPin, clockPin, bitOrder, value)` = 입력된 핀 번호로 입력된 값을 출력순서에 맞게 출력, 하나 의 비트가 출력된 이후에는 `clockPin`으로 펄스가 출력
- `shiftIn(dataPin, clockPin, bitOrder)` = 특정 핀(`dataPin`)으로부터 데이터를 입력 받아 비트 순서에 맞게 정렬 후 바이트 단위로 반환
- `pulsein(pin, value)` = 특정 핀으로부터 `pluse(High, Low)`를 읽어서 `pulse`의 길이를 마이크로 초 단위로 반환

- **trigonomatic function**

- `sin(rad)` = 입력된 라디안타입의 각도 값에 대한 사인 값을 반환
- `cos(rad)` = 입력된 라디안타입의 각도 값에 대한 코사인 값을 반환
- `tan(rad)` = 입력된 라디안타입의 각도 값에 대한 탄젠트 값을 반환

pinMode()

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

digitalWrite()

```
int ledPin = 13;                // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

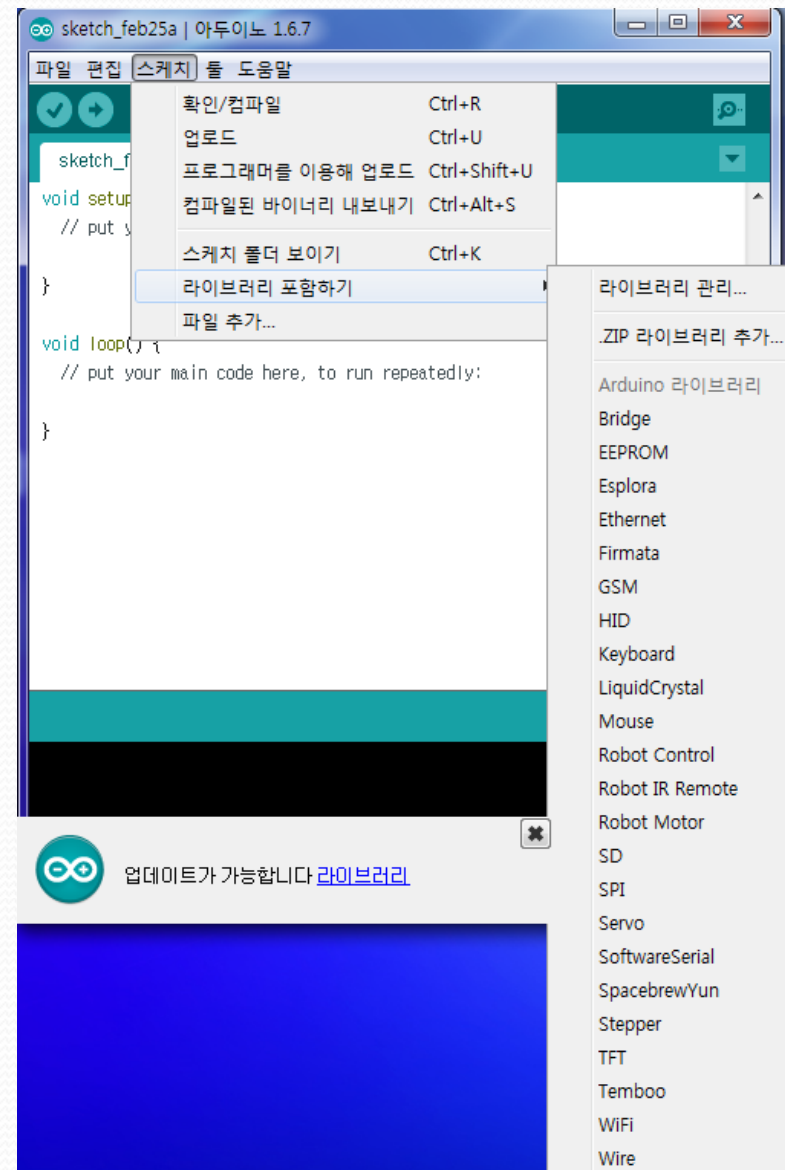

digitalRead()

```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;    // variable to store the read value

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
  pinMode(inPin, INPUT);  // sets the digital pin 7 as input
}

void loop()
{
  val = digitalRead(inPin); // read the input pin
  digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

Library



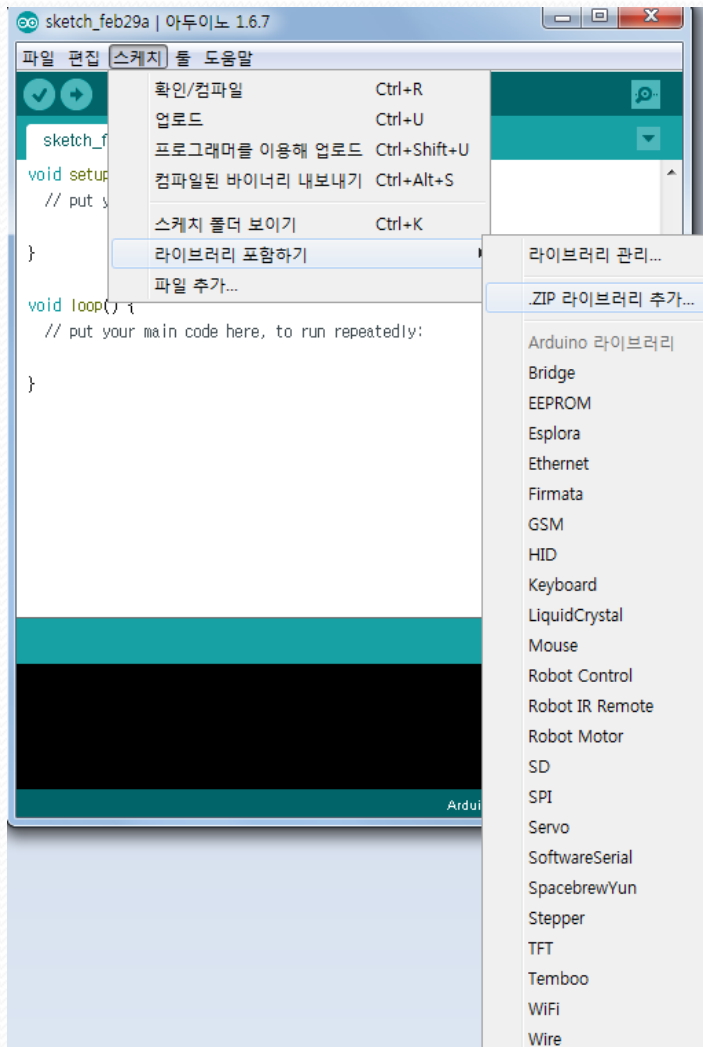
Libraries

- Sensor, Motor, Module 등 을 쉽게 사용할 수 있도록 지원
- 일반적으로 ZIP 압축 파일로 구성
- 기본 헤더 파일: `Arduino.h` //include 하지 않아도 됨
- 헤더 파일 include 한 뒤, 초기화 해서 사용
- Ex) `#include <SoftwareSerial.h>`
- `SoftwareSerial mySerial(2, 3) // Rx, Tx pin`

- 라이브러리 폴더
- 외부 라이브러리 : `[C:\Program Files (x86)\Arduino\libraries]`
- 내부 라이브러리 (기본 라이브러리) :
`[C:\Program Files (x86)\Arduino\hardware\arduino\avr\libraries]`

- 외부 라이브러리 추가
- <https://www.arduino.cc/en/Guide/Libraries> 참조

Library Include



컴퓨터 > 로컬 디스크 (C:) > Program Files > Arduino > hardware > arduino > avr > cores > arduino

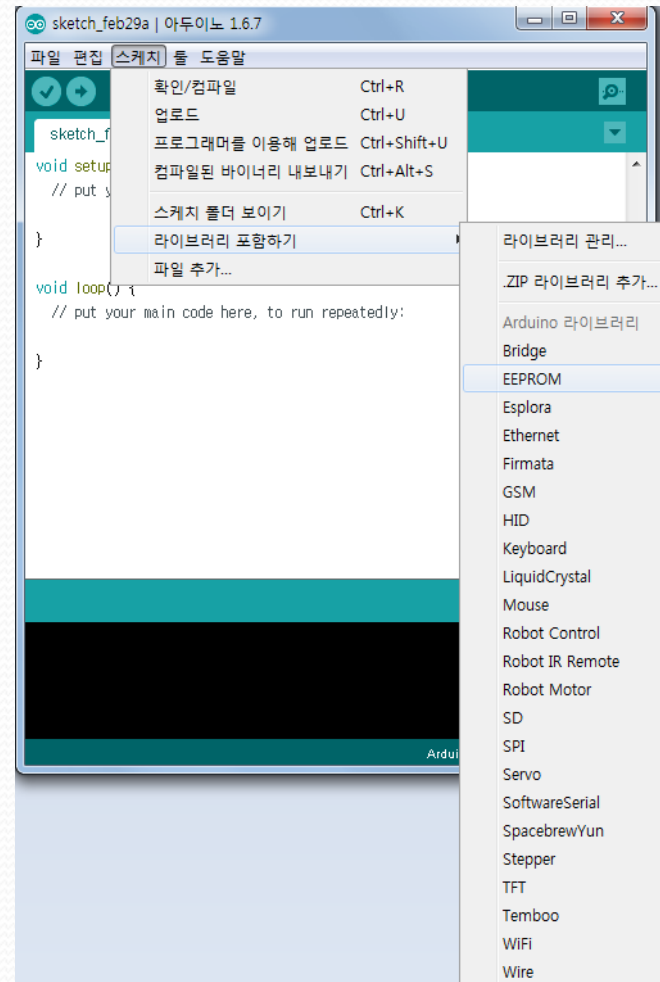
열기(F) 보기(V) 도구(T) 도움말(H)

열기(F) 보기(V) 도구(T) 도움말(H)

이름	수정된 날짜	유형	크기
abi.cpp	2015-12-18 오전...	CPP 파일	2KB
Arduino.h	2015-12-18 오전...	H 파일	8KB
binary.h	2015-12-18 오전...	H 파일	11KB
CDC.cpp	2015-12-18 오전...	CPP 파일	8KB
Client.h	2015-12-18 오전...	H 파일	2KB
HardwareSerial.cpp	2015-12-18 오전...	CPP 파일	8KB
HardwareSerial.h	2015-12-18 오전...	H 파일	6KB
HardwareSerial_private.h	2015-12-18 오전...	H 파일	5KB
HardwareSerialIO.cpp	2015-12-18 오전...	CPP 파일	3KB
HardwareSerialL.cpp	2015-12-18 오전...	CPP 파일	3KB

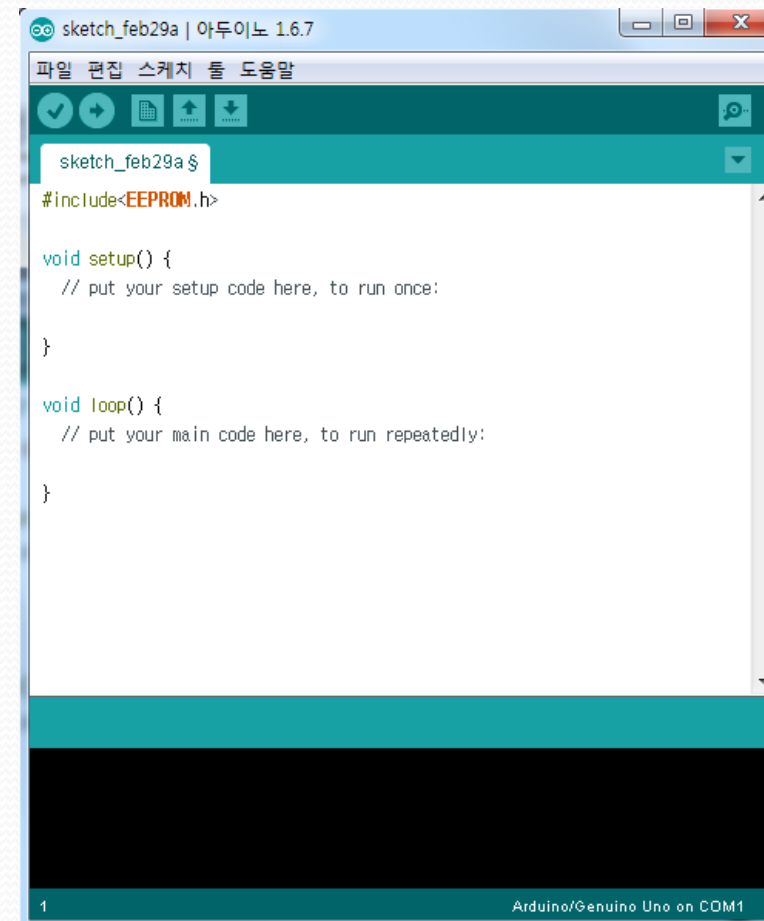
Library example

- 라이브러리 포함하기
- 원하는 라이브러리 선택
- 소스 코드에서 확인
- `#include <Keyboard.h>`



Library 직접 입력 하기

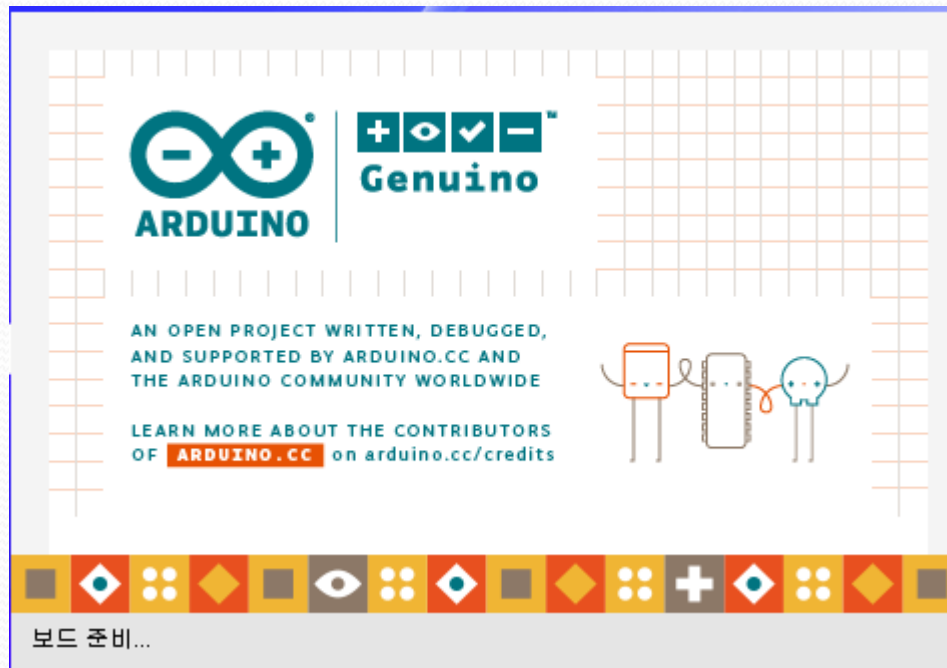
- Include 구문을 직접 입력
- `#include <Keyboard.h>`
- 컴파일해서 확인



변수선언

Name	Size*	Range*
char	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	1byte	true or false
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)

Arduino Software (IDE) – Sketch



- <https://www.arduino.cc> -> Software
- 해당 OS의 file을 down 받아서 실행
- Windows 경우 하단의 JUST DOWNLOAD -> 실행

Access the Online IDE



ARDUINO WEB EDITOR

Start coding online with the [Arduino Web Editor](#), save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards. The Arduino Web Editor is one of the [Arduino Create](#) platform's tools.

[Try It Now](#)
[Getting Started](#)



Download the Arduino IDE



ARDUINO 1.8.5

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

Windows app Requires Win 8.1 or 10

[Get](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

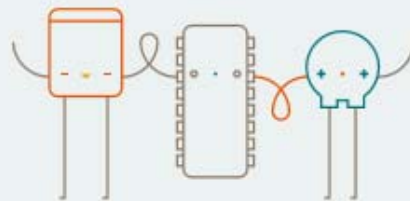


[Home](#) [Buy](#) [Software](#) [Products](#) [Learning](#) [Forum](#) [Support](#) [Blog](#)

[LOG IN](#) [SIGN UP](#)

Support the Arduino Software

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



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **13,573,494** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

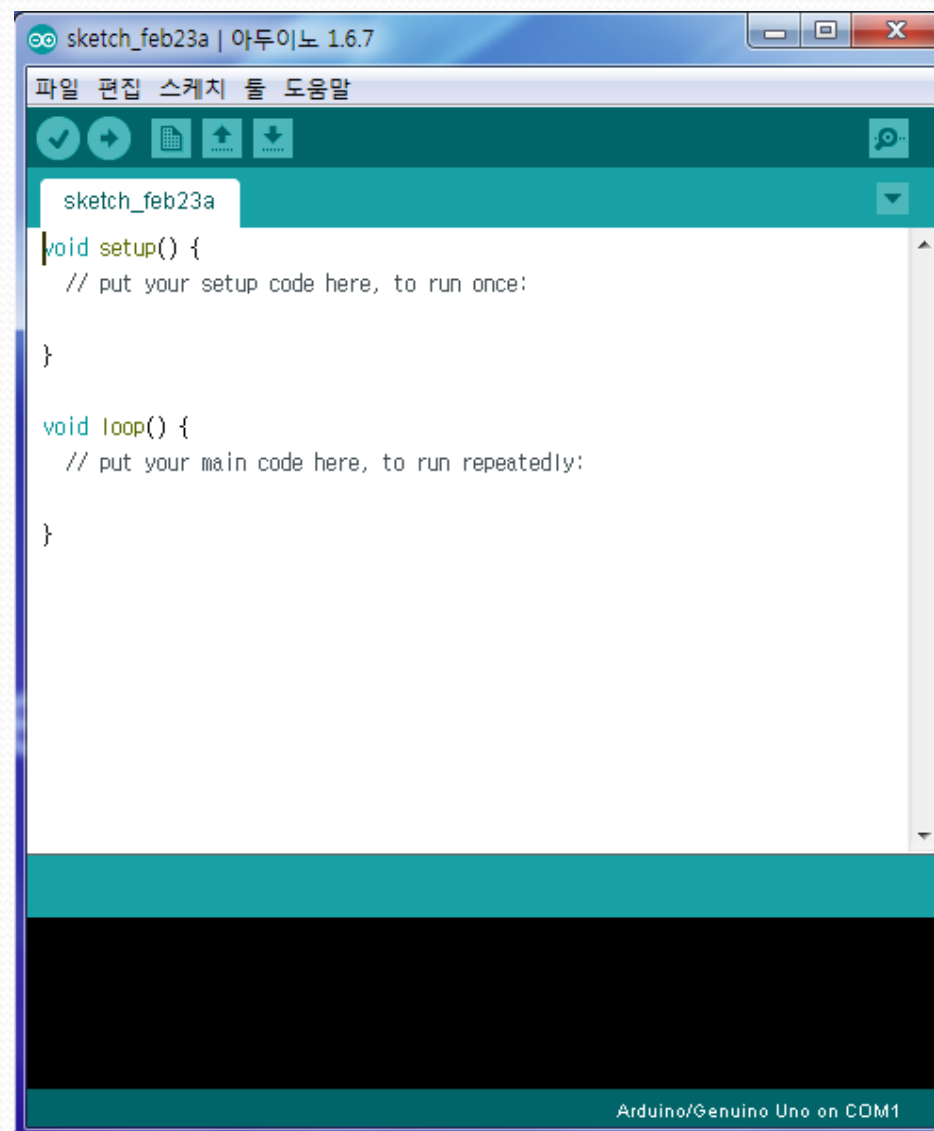
\$25

\$50

OTHER

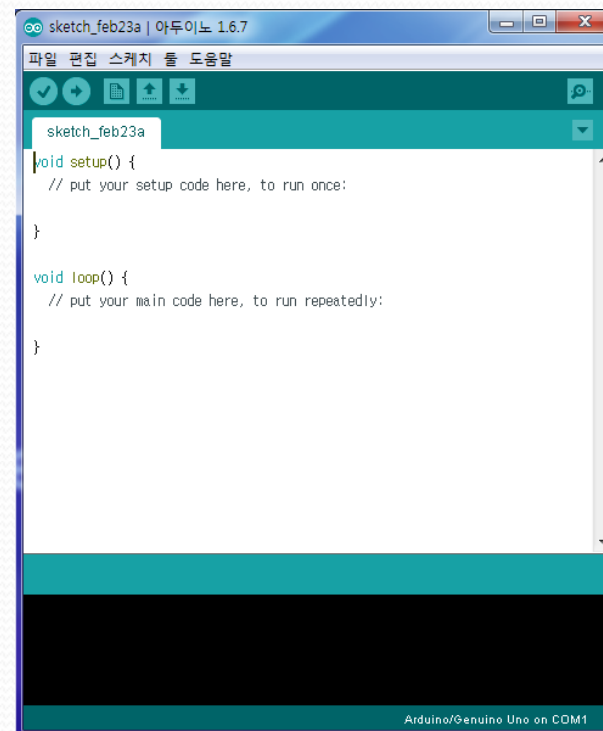
[JUST DOWNLOAD](#)

[CONTRIBUTE & DOWNLOAD](#)



Sketch

- 아두이노를 동작시키기 위한 Editor/Downloader
 - C/C++ 문법에 기초
 - 탭 지원 (파일 분할 가능)
 - ino 파일 형식으로 저장 (구버전 pde)
 - 작성한 파일을 저장할 때 폴더명, main 파일명이 일치해야 함 → 폴더 자동생성
-
- Menu
 - 컴파일, 업로드 버튼
 - 시리얼 모니터 버튼
 - 소스 수정 영역
 - 메시지 영역
 - 컴파일 버튼
 - setup() 함수
 - 최초 1회만 실행
 - loop() 함수
 - 무한반복 실행

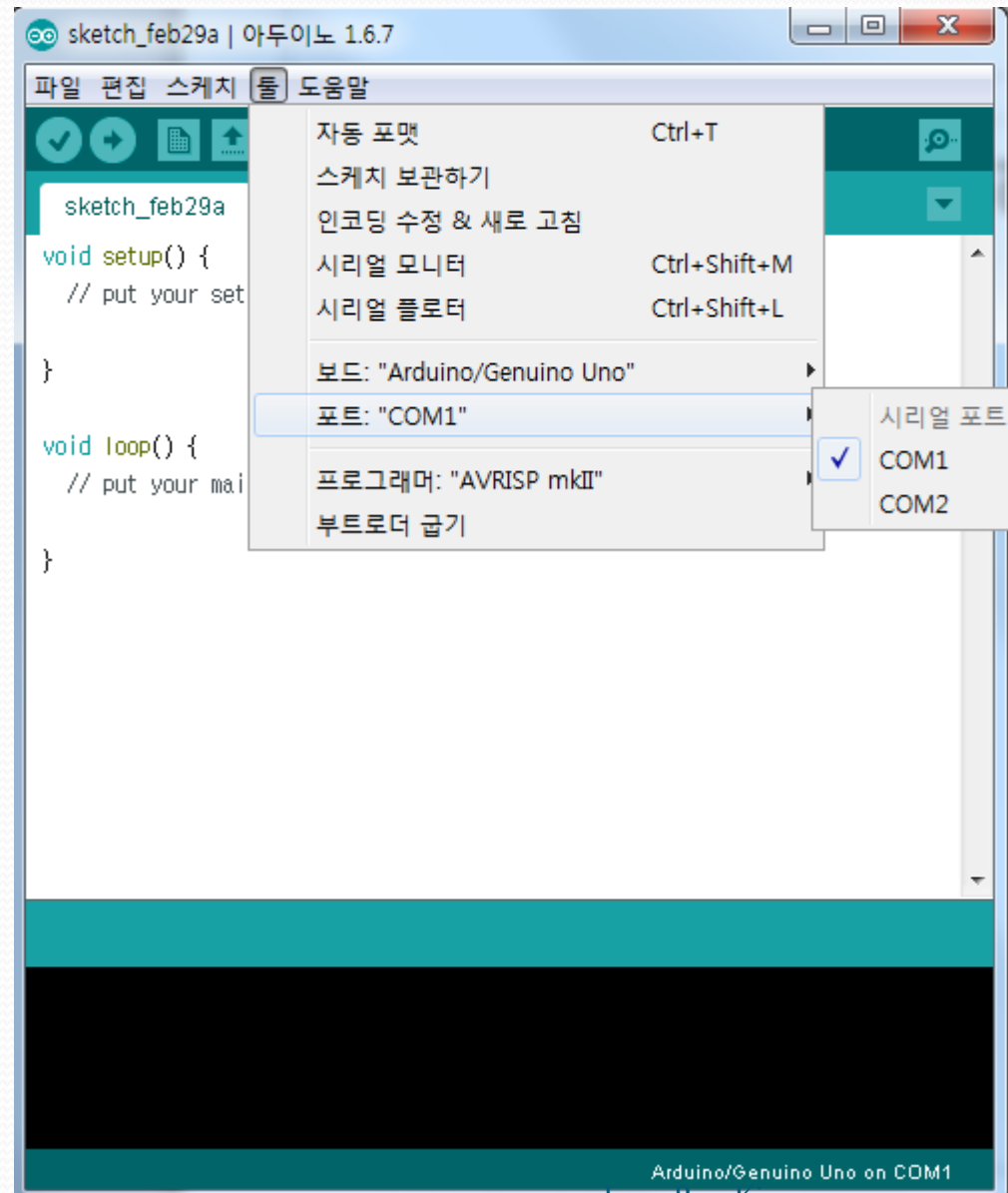


Com Port 선택

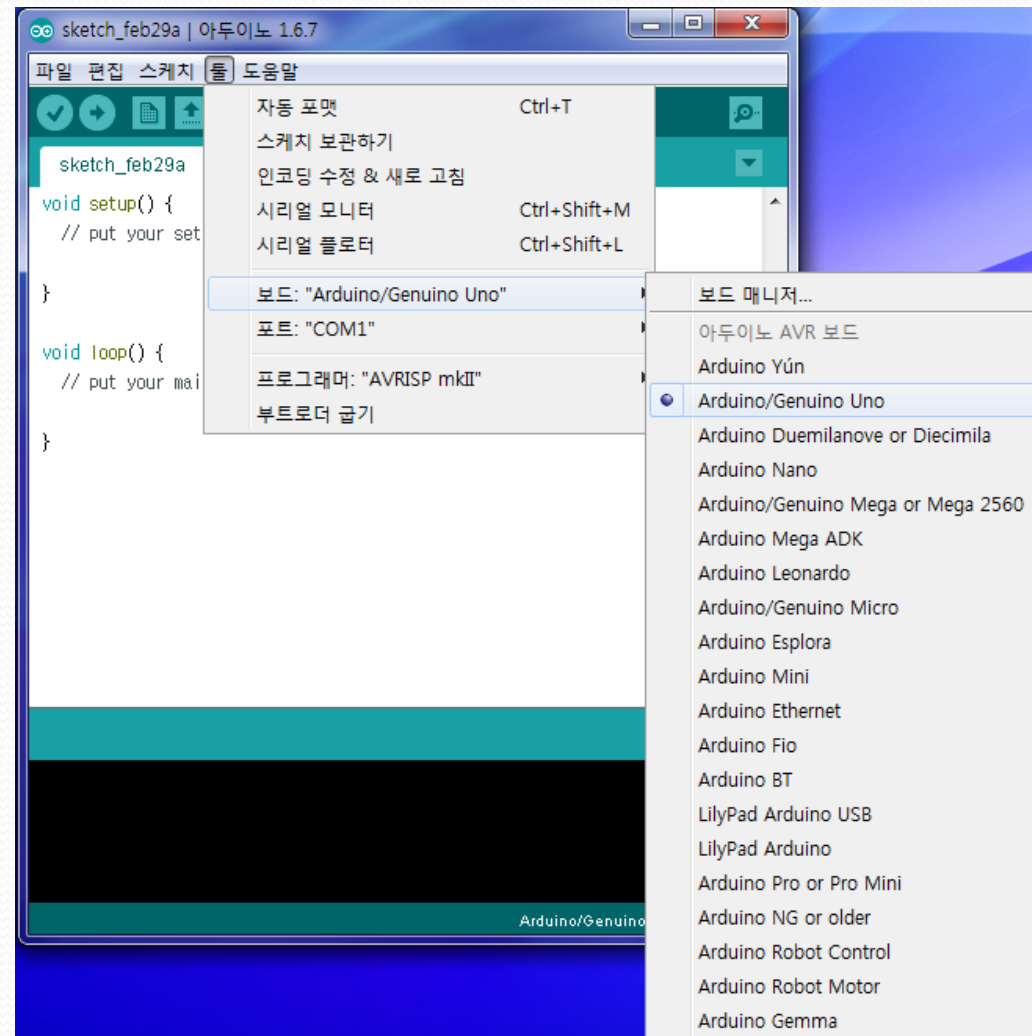
* Usb 뽑으면 없어지는
포트가 연결 포트임

포트확인

제어판 -> 시스템 및 보안
-> 시스템 -> 장치관리자
-> 포트



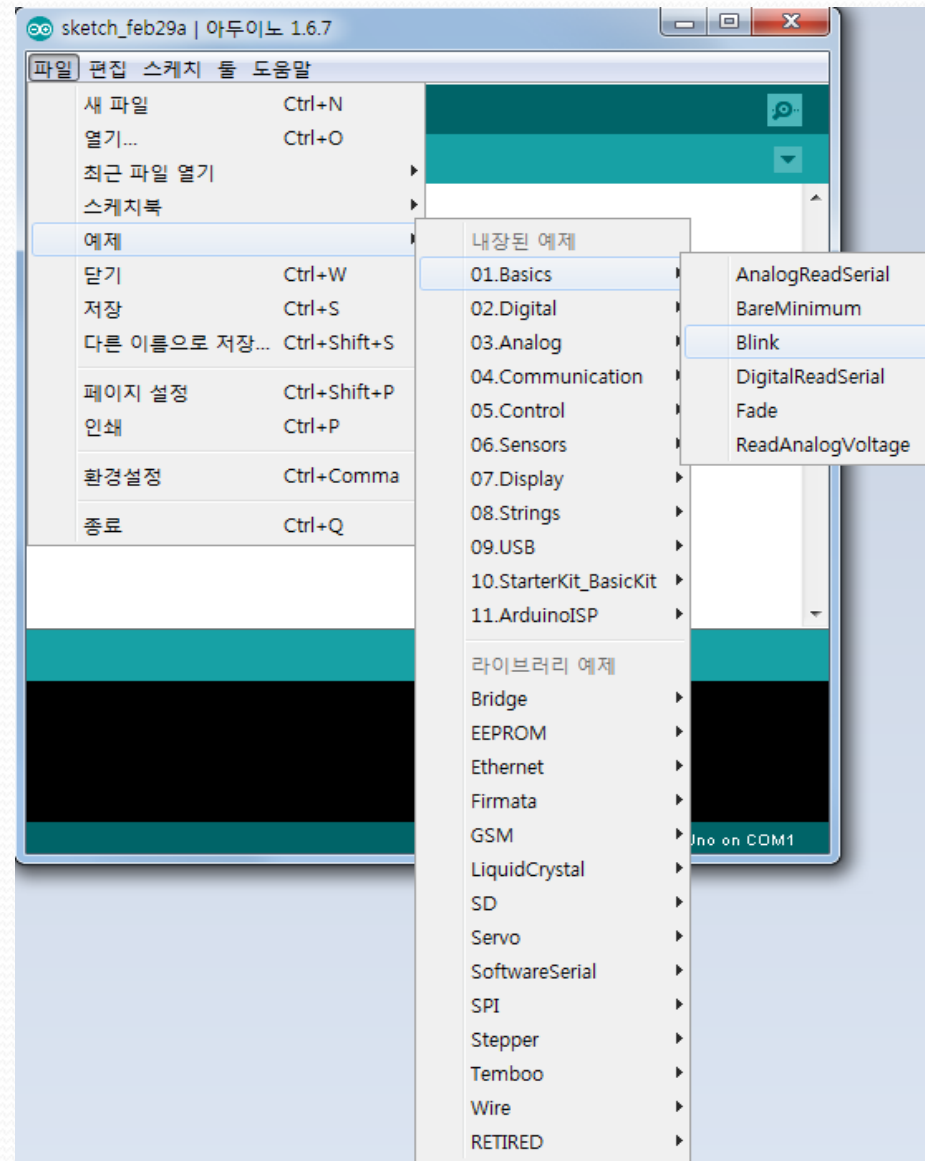
보드선택



Example Test – Blink

- 아두이노에 기본으로 내장된 LED를 제어
- 아두이노에 내장된 LED는 내부적으로 D13 핀에 연결되어 있음
- 아두이노에 기본으로 내장된 LED를 1초간 on / 1초간 off 반복
- Blink Program 이용

파일 ->예제 ->
01 Basics->Blink



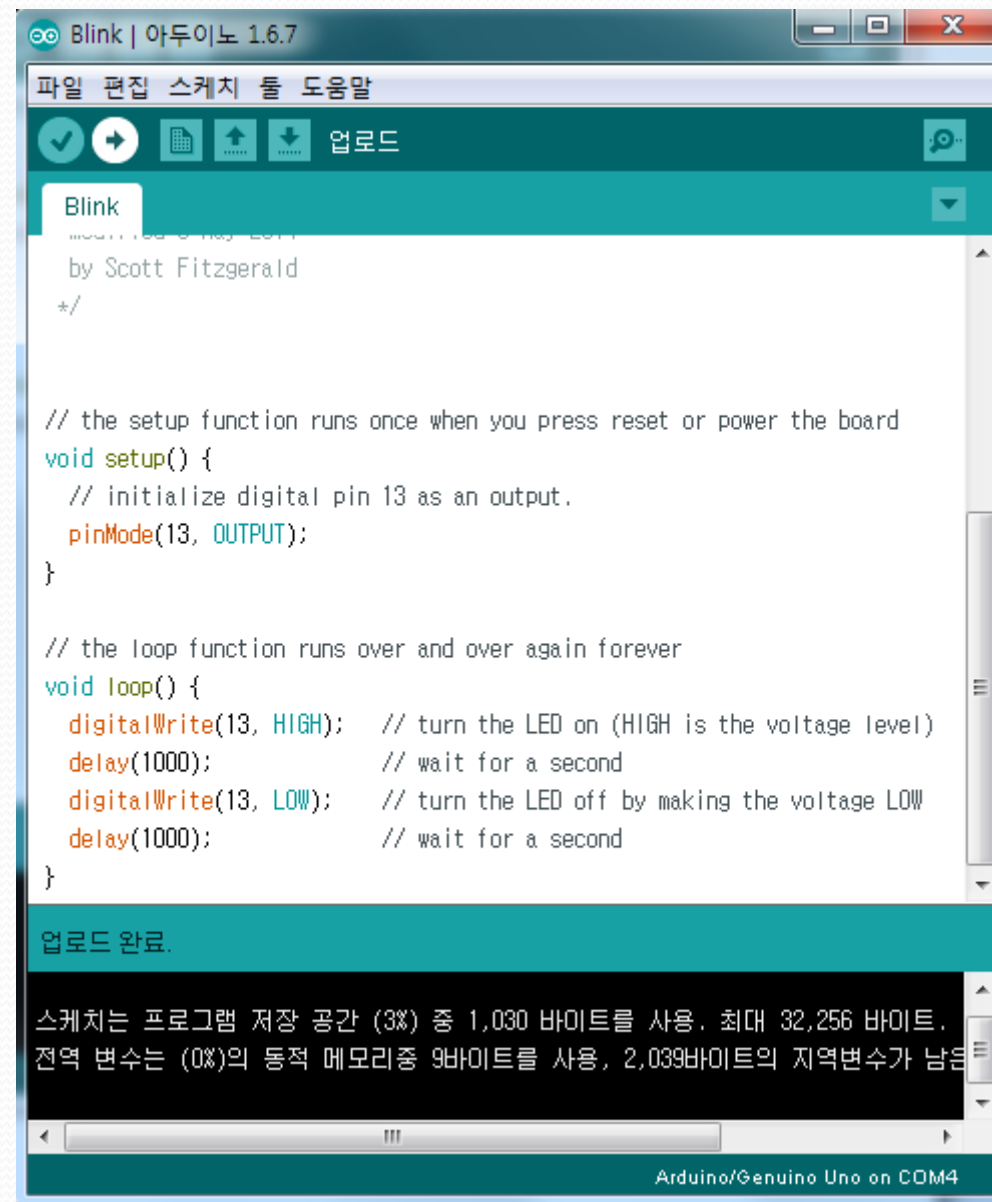
A screenshot of the Arduino IDE interface. The title bar reads "Blink | 아두이노 1.6.7". The menu bar includes "파일", "편집", "스케치", "도움말". The toolbar contains icons for opening, saving, compiling, uploading, and a search icon. The "Blink" sketch is selected in the left sidebar. The main editor displays the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

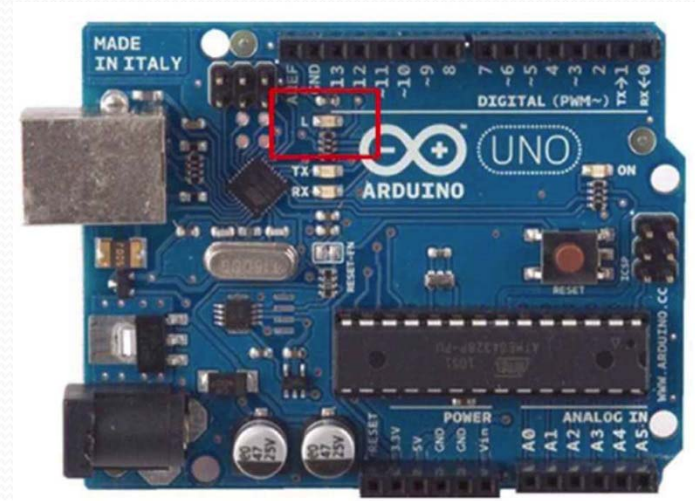
The status bar at the bottom indicates "Arduino/Genuino Uno on COM4".

Upload



Upload 후 결과

- 업로드 버튼 클릭 후
 - 보드 중앙의 TX/RX LED가 깜빡임
 - 메시지 영역에 “업로드 완료” 출력되면 정상
 - 결과: LED 가 1초 간격으로 깜박임
- Challenge 1)
 - LED가 0.5초씩 더 빠르게 깜박이도록 프로그램을 수정 한다.
- Challenge 2)
 - LED가 계속 켜져 있게 프로그램을 수정한다
- Challenge 3)
 - LED가 1초 ON → 1초 OFF → 3 초 ON → 1초 OFF → 5 초 ON → 1초 OFF 반복 되도록 프로그램을 수정 한다.



Arduino Shield

- 아두이노 쉴드
 - 확장 보드
 - 약 300여 종류
 - Uno에 맞게 제작 Mega 2560 호환
- LCD, 이더넷, 와이파이, SD카드 쉴드 등



Language Structure

- Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.
- setup()
- loop()
- Control Structures
 - if
 - if...else
 - for
 - switch case
 - while
 - do... while
 - break
 - continue
 - return
 - goto

Further Syntax

; (semicolon)

{ } (curly braces)

// (single line comment)

/ * */ (multi-line comment)

#define

#include

Arithmetic Operators

= (assignment operator)

+ (addition)

- (subtraction)

* (multiplication)

/ (division)

% (modulo)

Comparison Operators

== (equal to)

!= (not equal to)

< (less than)

> (greater than)

<= (less than or equal to)

>= (greater than or equal to)

Boolean Operators

&& (and)

|| (or)

! (not)

Pointer Access Operators

* dereference operator

& reference operator

Bitwise Operators

& (bitwise and)

| (bitwise or)

^ (bitwise xor)

~ (bitwise not)

<< (bitshift left)

>> (bitshift right)

Compound Operators

++ (increment)

-- (decrement)

+= (compound addition)

-= (compound subtraction)

*= (compound multiplication)

/= (compound division)

%= (compound modulo)

&= (compound bitwise and)

|= (compound bitwise or)

Variables

- Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
integer constants
floating point constants



Data Types

void

boolean

char

unsigned char

byte

int

unsigned int

word

long

unsigned long

short

float

double

string – char array

String – object

array

Conversion

char()

byte()

int()

word()

long()

float()

Variable Scope & Qualifiers

variable scope

static

volatile

const

Utilities

sizeof()

PROGMEM

Functions

- Digital I/O
 - pinMode()
 - digitalWrite()
 - digitalRead()
- Analog I/O
 - analogReference()
 - analogRead()
 - analogWrite() – *PWM*
- Due & Zero only
 - analogReadResolution()
 - analogWriteResolution()

Advanced I/O

tone()

noTone()

shiftOut()

shiftIn()

pulseIn()

Time

millis()

micros()

delay()

delayMicroseconds()

Math

min()

max()

abs()

constrain()

map()

pow()

sqrt()

Trigonometry

[sin\(\)](#)

[cos\(\)](#)

[tan\(\)](#)

Characters

[isAlphaNumeric\(\)](#)

[isAlpha\(\)](#)

[isAscii\(\)](#)

[isWhitespace\(\)](#)

[isControl\(\)](#)

[isDigit\(\)](#)

[isGraph\(\)](#)

[isLowerCase\(\)](#)

[isPrintable\(\)](#)

[isPunct\(\)](#)

[isSpace\(\)](#)

[isUpperCase\(\)](#)

[isHexadecimalDigit\(\)](#)

Random Numbers

randomSeed()

random()

Bits and Bytes

lowByte()

highByte()

bitRead()

bitWrite()

bitSet()

bitClear()

bit()

External Interrupts

attachInterrupt()

detachInterrupt()



Interrupts

[interrupts\(\)](#)

[noInterrupts\(\)](#)

Communication

[Serial](#)

[Stream](#)

USB (32u4 based boards and
Due/Zero only)

[Keyboard](#)

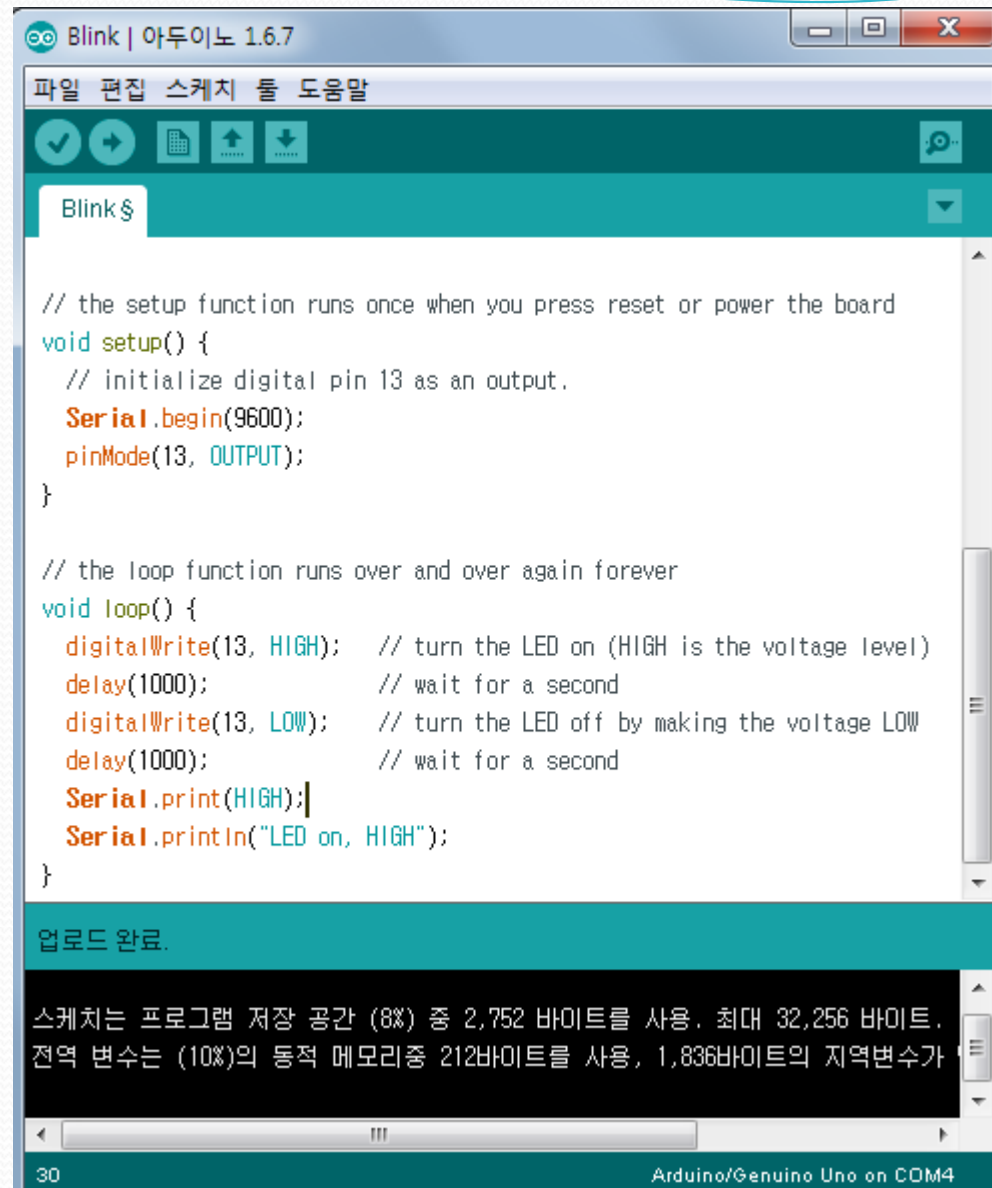
[Mouse](#)

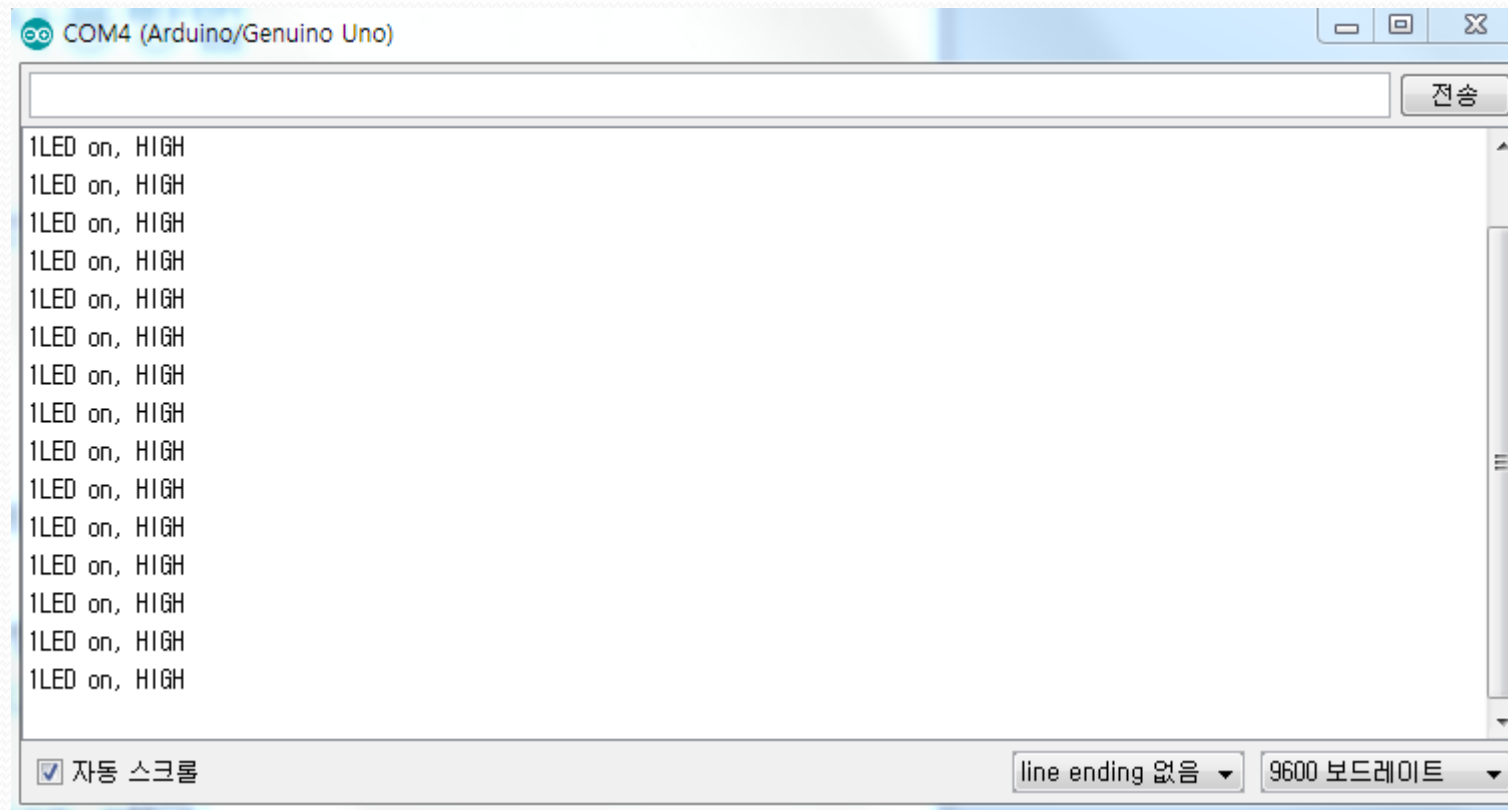
Serial Communication

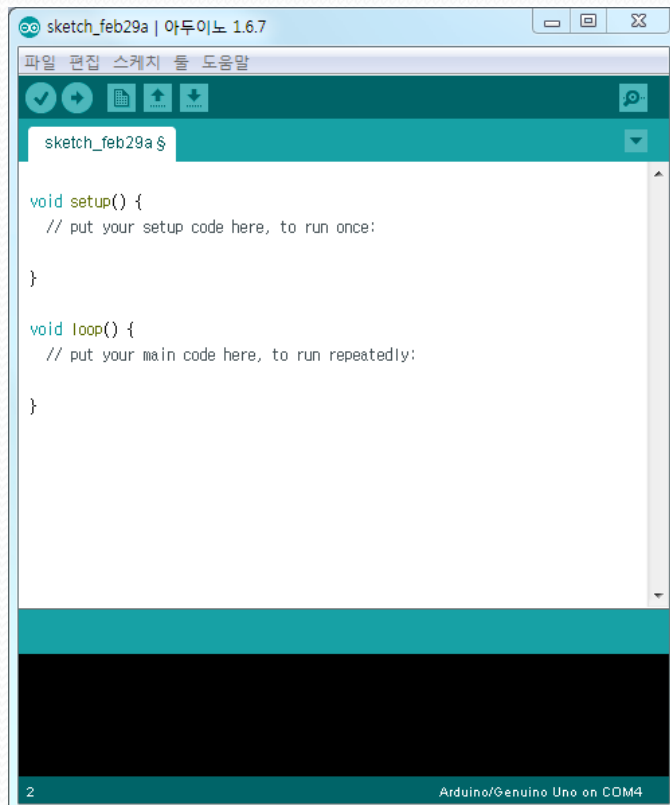
- USB Serial 통신
 - USB를 이용해서 PC와 아두이노 통신
 - Serial 통신을 사용 (Hardware serial)
 - Serial 통신은 2개의 데이터 통신용 핀을 사용 (Tx, Rx)
 - UNO의 경우 digital 0번, 1번 핀이 이 용도로 예약되어 있음
-
- Serial Monitor
 - 아두이노와 데이터를 주고 받기 위해 사용하는 툴
 - 주로 아두이노 디버깅, 데이터 입력을 위해 사용
 - Serial monitor를 이용한 디버깅 메시지를 보는 것이 거의 유일한 디버깅 방법이므로 중요

Serial Monitor 사용방법

- 아두이노 연결된 상태
- 보드 종류, COM 포트 선택
- 메시지 입력창
- 수신 메시지 표시창
- 통신 속도 설정
 - Baud Rate: 기본 9600 bps
- 툴 → 시리얼 모니터 선택 → 별도의 모니터에 출력 값 표시됨
- Serial Monitor 설정 방법
 - `Serial.begin(9600);`
- 메시지
 - `Serial.println("LED on, off");`







Fritzing is open source, free software. Please consider [donating to Friends-of-Fritzing e.V.](#) before downloading the app.

Fritzing is a non-profit organization devoted to making creative use of electronics accessible to everyone.

☒ No Donation

☐ € 10

☐ € 25

☐ € 50

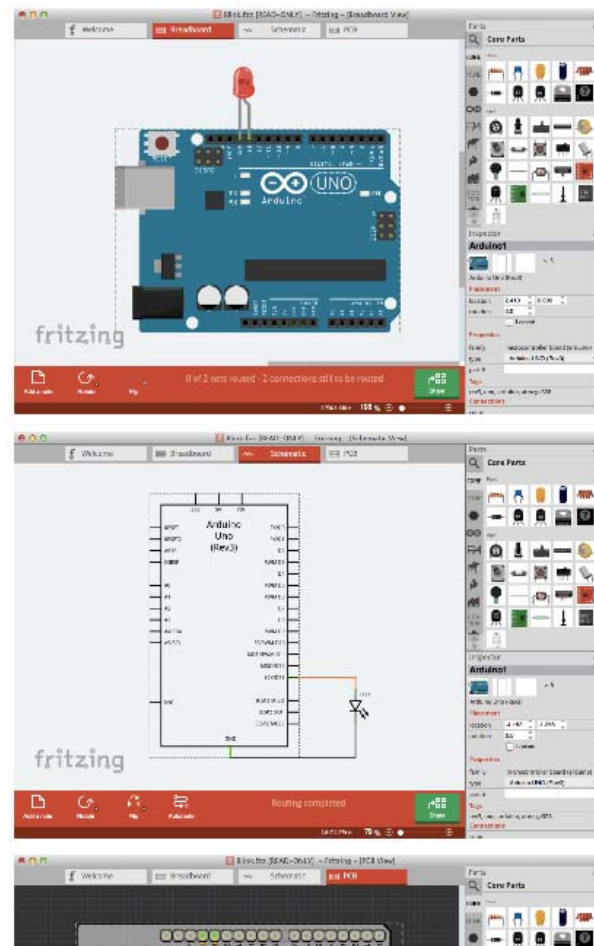
Download

Version **0.9.3b** was released on **June 2, 2016**.

Downloaded 1118497 times.

See [what's new](#) and the [known issues](#).

Read the [installation instructions](#) below



GO

FAQ ABOUT CONTACT

**Digi-Key™
Korea**

No Min Order. Instant
Availability. Same Day
Shipping. Order Now!

digikey.kr



Blog

[New fritzing release 0.9.3b!](#)
Jun. 3, 2016

[New Book: "Fritzing for Inventors"](#)
Dec. 6, 2015

[A new fritzing discussion forum](#)
Nov. 8, 2015

[More posts...](#)

Projects

[Mic-engine](#)
Forset456

[no1](#)
xmass

