



Maker Space Lecture

Week 1 — What is Arduino?



YONSEI, where we make *history*



연세대학교
Yonsei Where we make History



YONSEI, where we make *history*

CONTENTS

1. Introduction to Arduino
2. Arduino Grammar
3. LED Tutorials
4. References



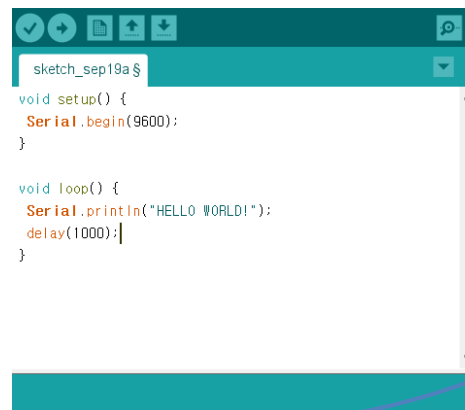
1
2
3
4

Definition of Arduino

→ Easy to share code, library, and even hardware!

Arduino is an **open-source** electronics platform based on easy-to-use **hardware** and **software**.

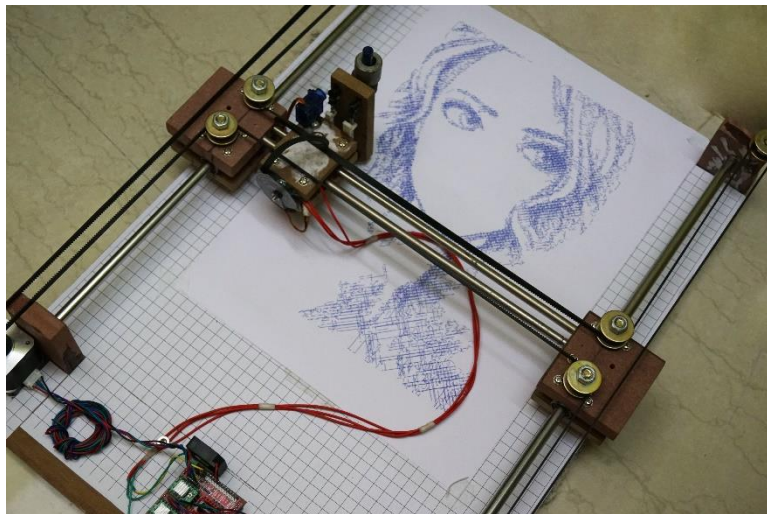
To use it, you use the Arduino programming language, and the **Arduino Software(IDE)**.



1
2
3
4

Applications of Arduino

<https://www.youtube.com/watch?v=swoiluHrq4c&t=167s>



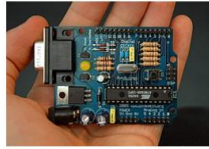
Arduino CNC Drawing Machine

Index-<http://www.arnabkumardas.com/product/arduino-cnc-drawing-machine/>



BOB-Delivery Robot with Table and Chairs

Types of Arduino



Arduino RS232^[32]
(male pins)



Arduino Diecimila



Arduino Duemilanove^[34]
(rev 2009b)



Arduino Uno R2^[35]



Arduino Uno SMD
R3^[37]



Arduino Leonardo^[38]



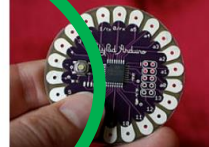
Arduino Pro^[39]
(No USB)



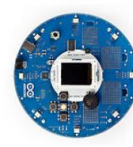
Arduino Mega^[40]



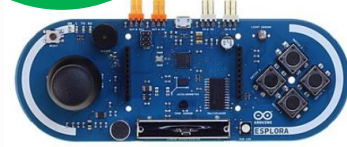
Arduino
Nano^[41]
(50
footprint
)



Arduino LilyPad^[42]
(rev 2007) (No USB)



Arduino Robot^[43]



Arduino Esplora^[44]



Arduino Ethernet^[45]
(AVR + W5100)



Arduino Yun^[46]
(AVR + AR9331)



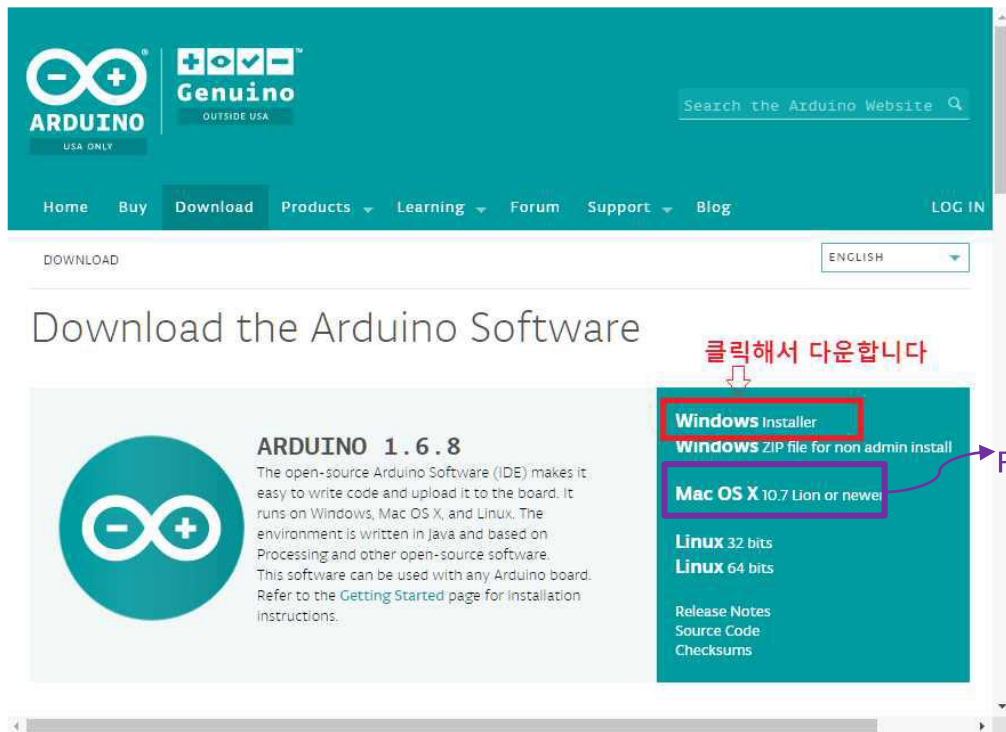
Arduino Due^[47]
(ARM Cortex-M3 core)



1
2
3
4

Arduino Download

Step 1.
<https://www.arduino.cc/en/Main/Software>



ARDUINO 1.6.8

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



1
2
3
4

Arduino Download



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](#)



JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Share



Step 2.
Click 'JUST DOWNLOAD'

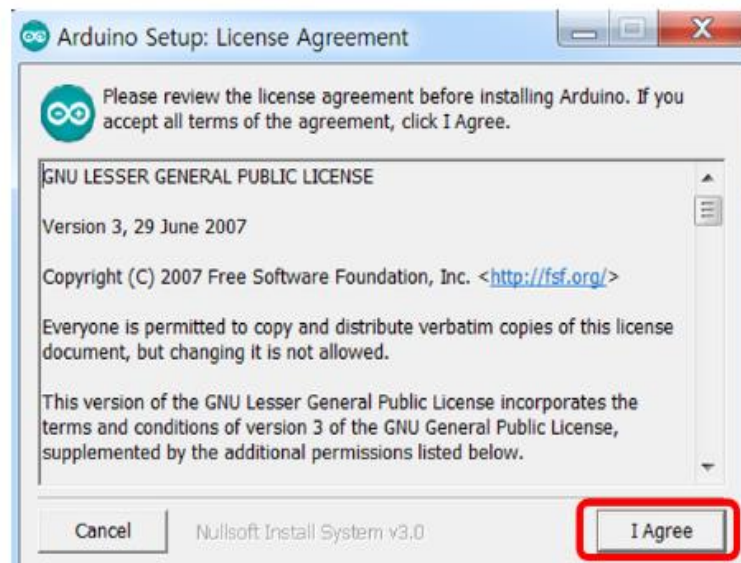
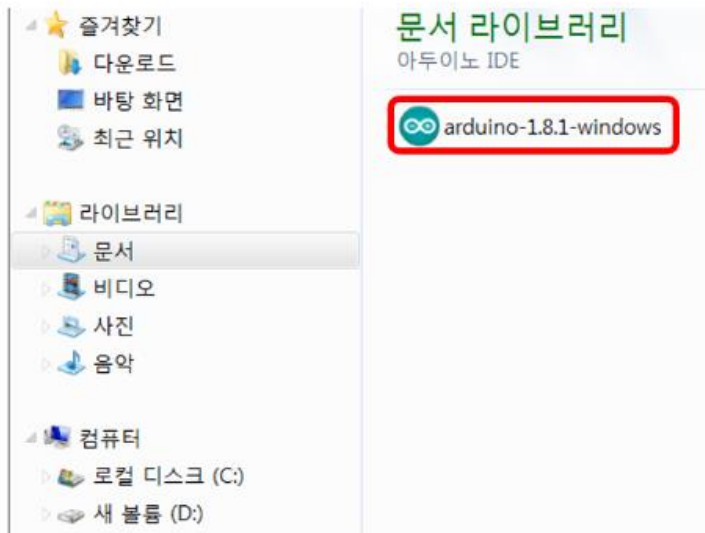


1
2
3
4

Arduino Download

Step 3.

Double click the icon and click 'I Agree'

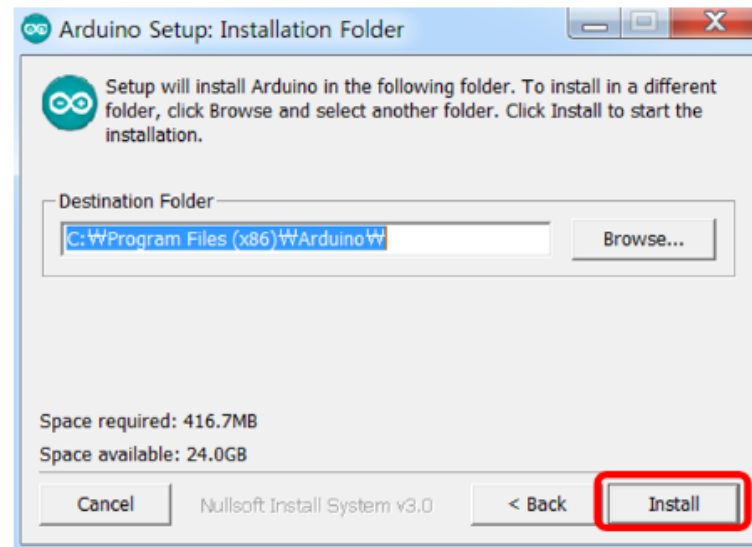
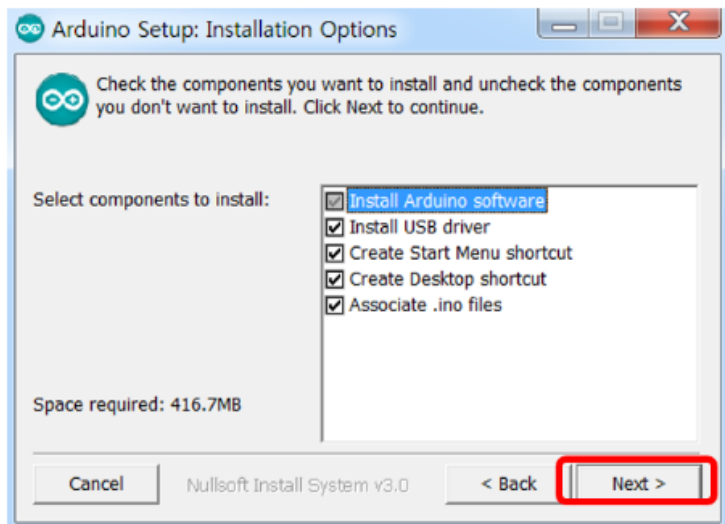




1
2
3
4

Arduino Download

Step 4.
Click 'Next' and Install

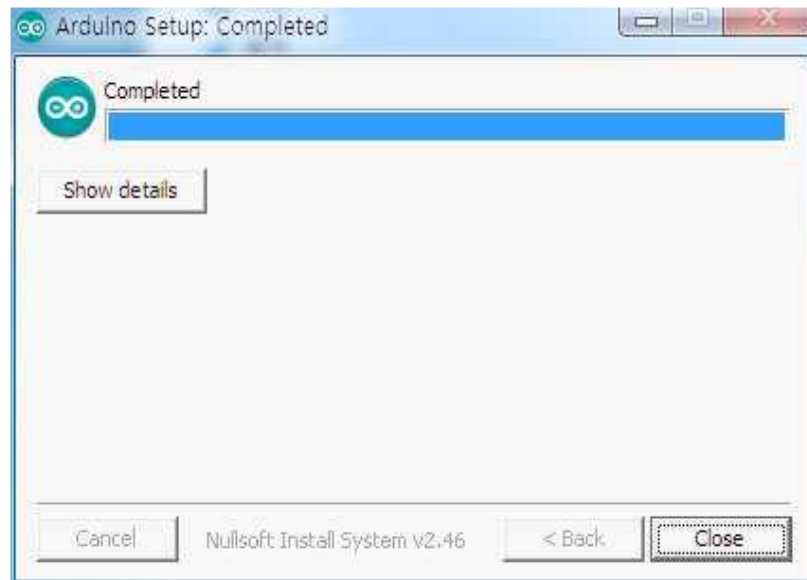
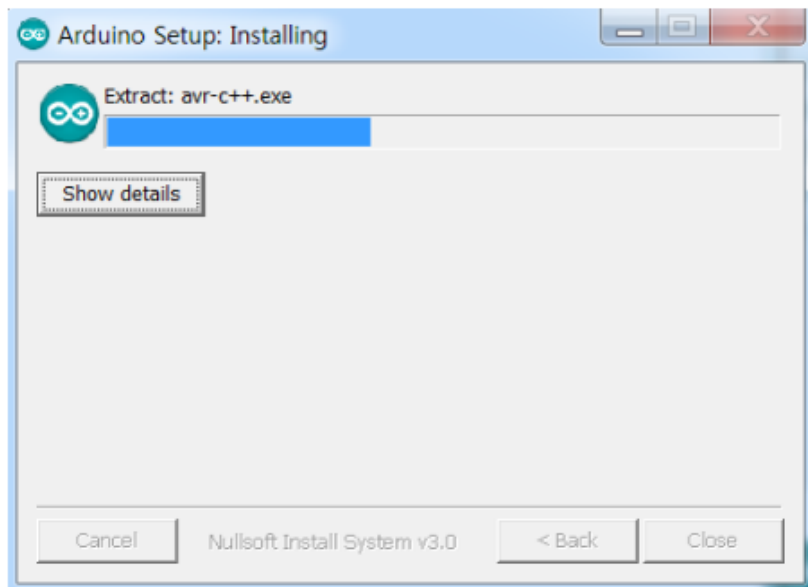




1
2
3
4

Arduino Download

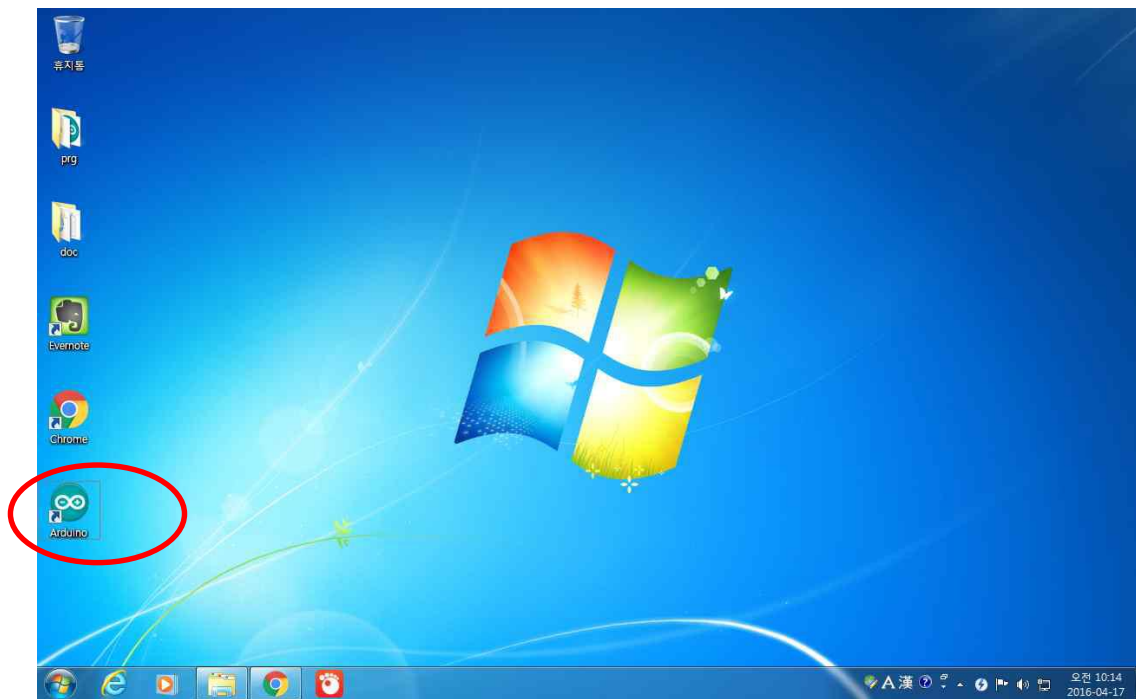
Step 5.
Click 'Close'



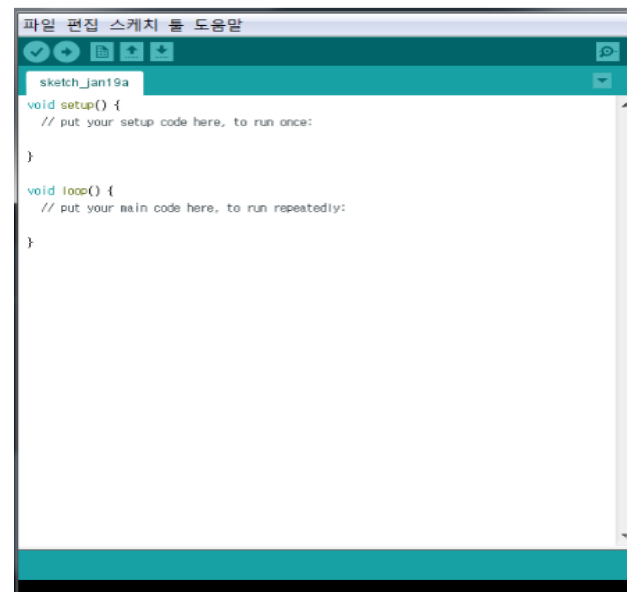


1
2
3
4

Arduino Download



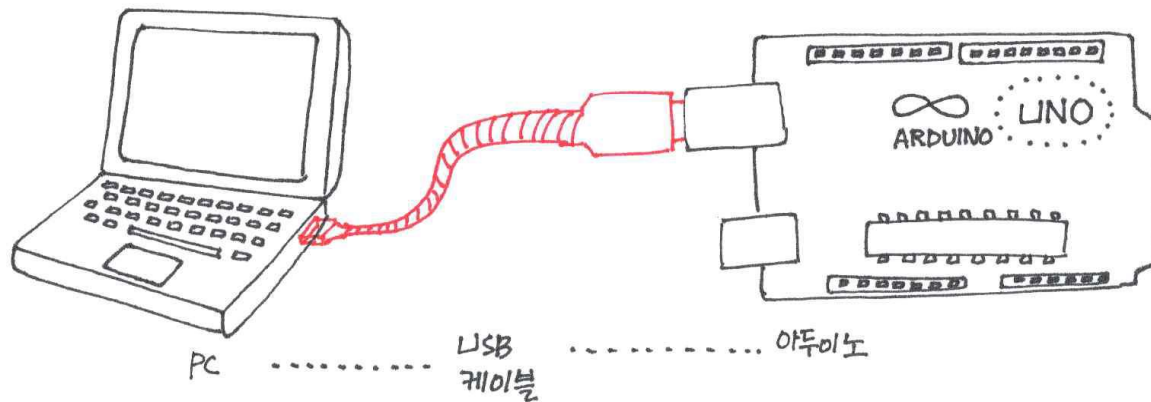
Step 6.
Double Click Arduino icon



1
2
3
4

Arduino Connection

Step 1.
Connect as the picture

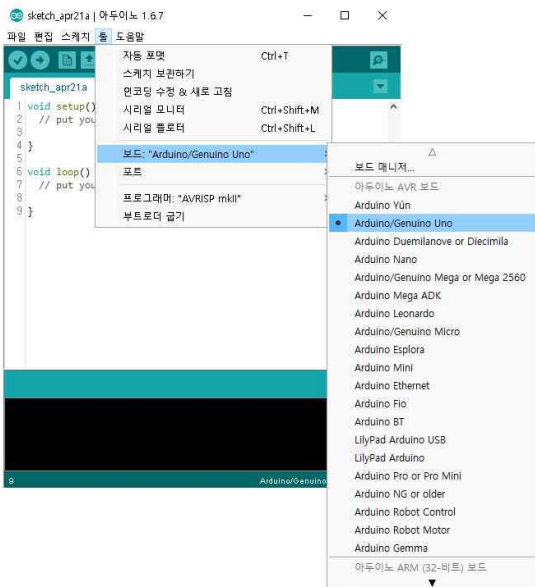




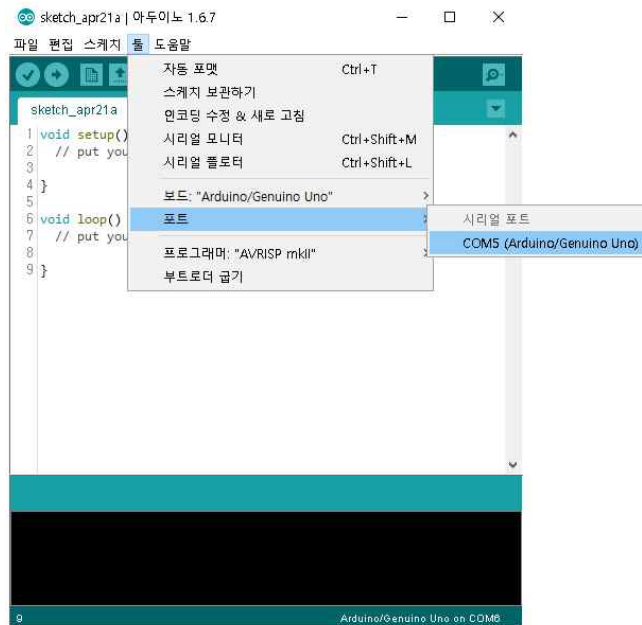
1
2
3
4

Arduino Connection

Step 2. Tool-Board setting



Step 3. Tool-Port setting





1
2
3
4

Basic Structure

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

setup()

: Initial setting function

loop()

: Real working & repeated function



1
2
3
4

Practice 1 - HELLO WORLD!



```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("HELLO WORLD!");  
  delay(1000);  
}
```

Serial.begin()

: Start serial communication

Serial.println("")

: Print sentence on the serial monitor
and add a newline

delay()

: Wait () milli seconds

Serial Communication?

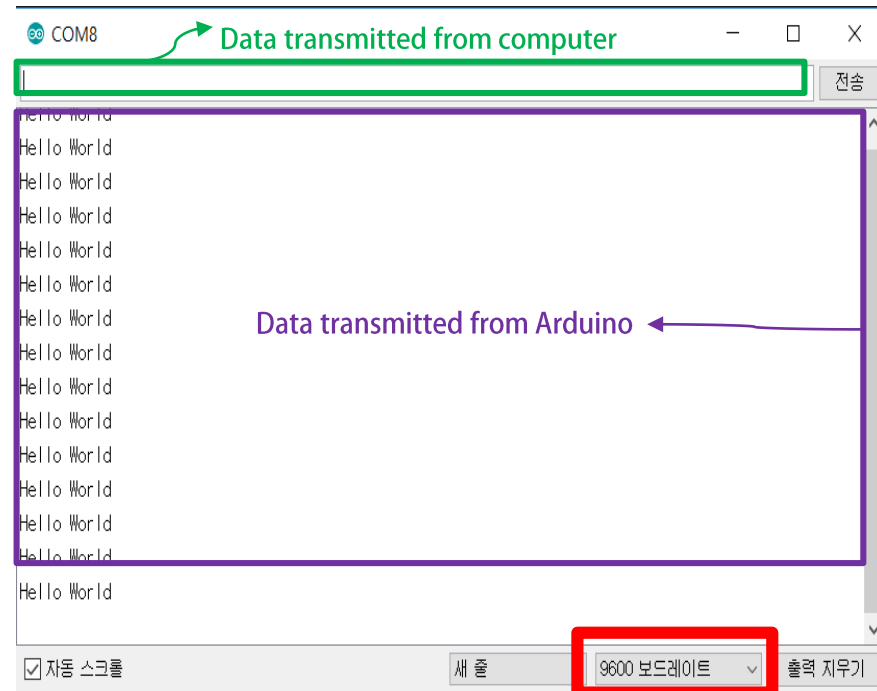
<https://m.blog.naver.com/yuyyulee/220301424499>



1
2
3
4

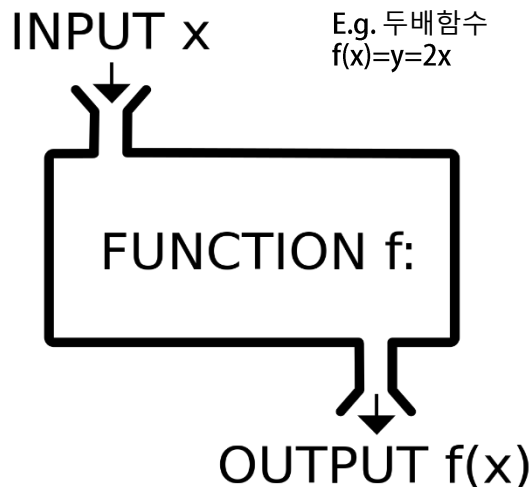
1 : Upload 2 : Click Serial monitor

3 : Set the baud rate



1
2
3
4

Function



```
sketch_apr22a $
나가는값의자료형 함수이름(들어오는값){
  처리를 위한 명령문들...
}
```

```
sketch_apr16a $
void happy() {
  //주석이어서 아무 효과 없어요~
  /*
   이렇게 쓰면
   여러 줄을 주석으로 쓸 수 있습니다.
   */
  Serial.println("HELLO WORLD!");
}
```

`}` Paragraph

`;` End a statement

Blockcomment

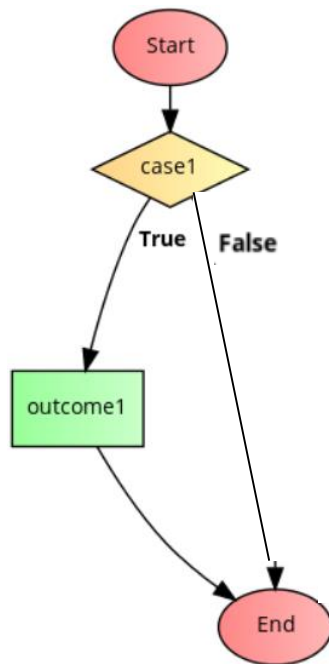
`//` for one line

`/**/` for multiple lines

1
2
3
4

if statement

```
if(case1){
    outcome1
}
```



-if the condition in the () is true, then the following outcome in the {} will run

-you can skip 'else' and 'else if' if you want

-else if can be written as many times as you want

cf.

1 means 'ON' , 'True' , and 'HIGH'

0 means 'OFF' , 'False' , and 'LOW'

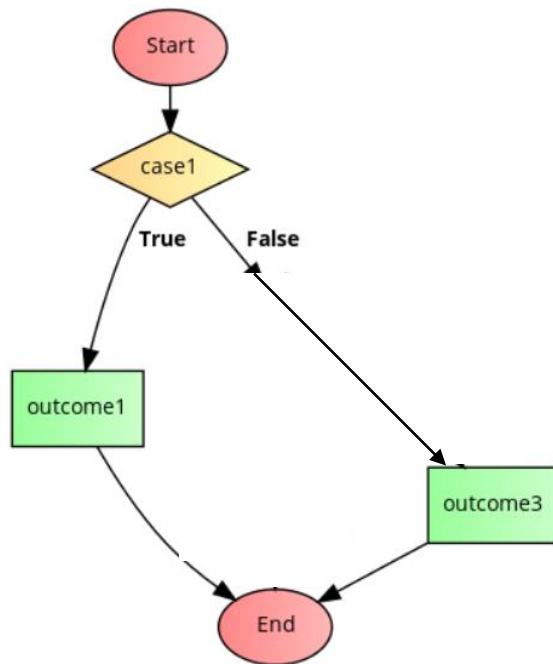


1
2
3
4

if statement

```
if(case1){
    outcome1
}
```

```
else(){
    outcome3
}
```



-if the condition in the () is true, then the following outcome in the {} will run

-you can skip 'else' and 'else if' if you want

-else if can be written as many times as you want

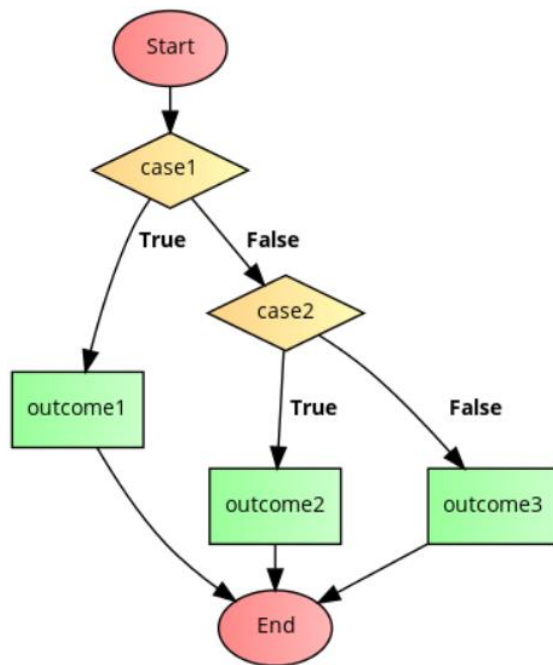
cf.

1 means 'ON' , 'True' , and 'HIGH'
0 means 'OFF' , 'False' , and 'LOW'

1
2
3
4

if statement

```
if(case1){
    outcome1
}
else if(case2){
    outcome2
}
else(){
    outcome3
}
```



-if the condition in the () is true, then the following outcome in the {} will run

-you can skip 'else' and 'else if' if you want

-else if can be written as many times as you want

cf.

1 means 'ON' , 'True' , and 'HIGH'
0 means 'OFF' , 'False' , and 'LOW'



1
2
3
4

variable

-A variable is a place to store a piece of data. It has a **name**, a **value**, and a **type**. For example, this statement

```
int pin = 13;
```

 is a typical variable declaration(선언) format.

And this creates a variable whose name is **pin**, whose value is **13**, and whose type is **int**.

-**variable name** can be written with A~Z, a~z, 0~9, _ (but number cannot written at the first)

-**type of variable** integer(정수형) : int, long
real number(실수형) : float
character(문자형) : char

(you should put a proper value according to the variable)

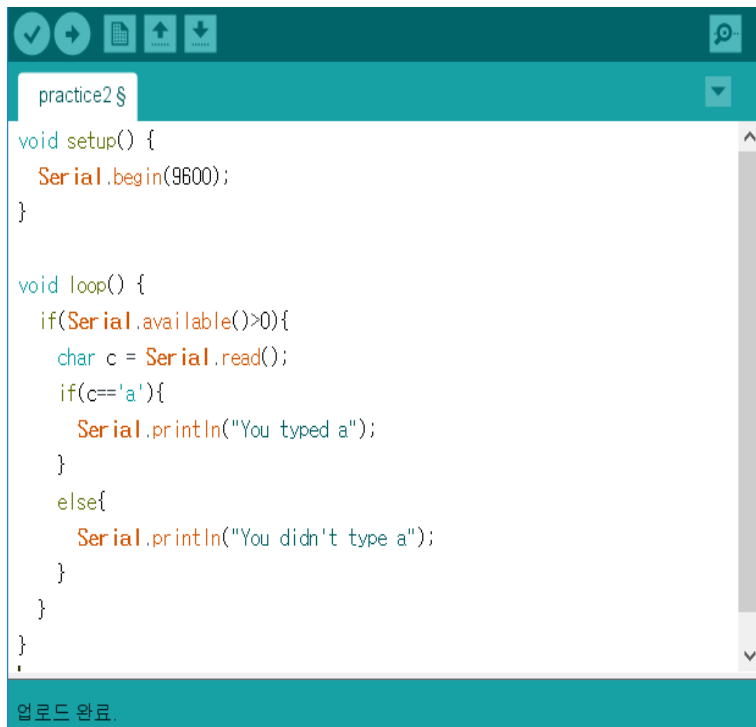
Type?

<http://www.3demp.com/community/boardDetails.php?cbID=212>



1
2
3
4

Practice 2



```
practice2 $  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(Serial.available()>0){  
    char c = Serial.read();  
    if(c=='a'){  
      Serial.println("You typed a");  
    }  
    else{  
      Serial.println("You didn't type a");  
    }  
  }  
}
```

업로드 완료.

Serial.available()

: Get the number of bytes available for reading from serial port

: If data does exist, then its value is more than zero

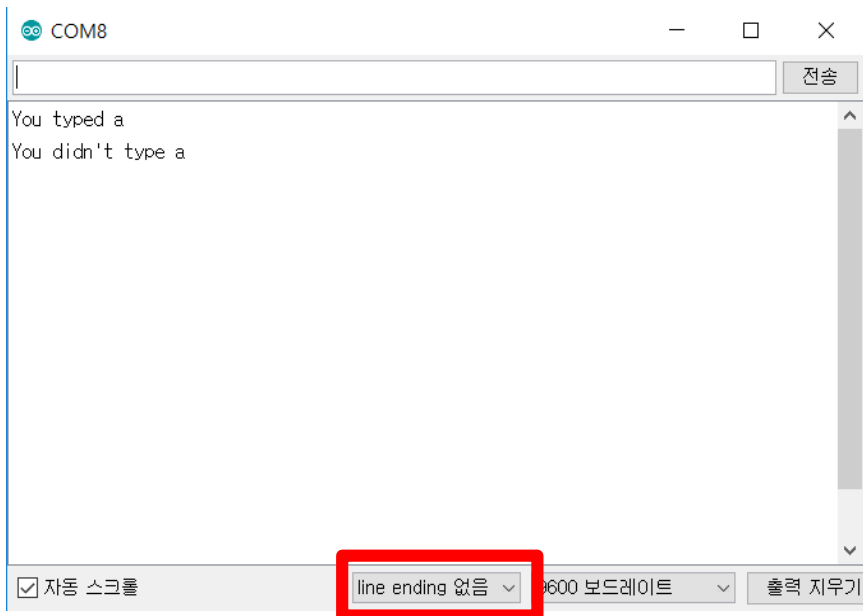
Serial.read()

: Reads incoming serial data and returns the first byte of incoming serial data available (or -1 if no data is available)



1
2
3
4

Practice 2



Error case :

-If you type any character and see that Serial.available() runs again as you see in the image to the left, then you should click 'no line ending'



1
2
3
4

Practice 3 — About Numbers



```
sketch_apr21a
void setup() {
  Serial.begin(9600);
}
void loop() {
  if(Serial.available()){
    long c = Serial.parseInt();
    long d = Serial.parseInt();
    Serial.println(c*d);
  }
}
```

업로드 완료.

Serial.parseInt()

: We cannot use Serial.read

: Simply click 'space bar' between integers when you write the numbers

: if input is number, returns long type number, if not, returns 0



1
2
3
4

Practice 3 — About Numbers

COM8

15 8

전송

120

☒ 자동 스크롤

line ending 없음 ▾

9600 보드레이트 ▾

출력 지우기

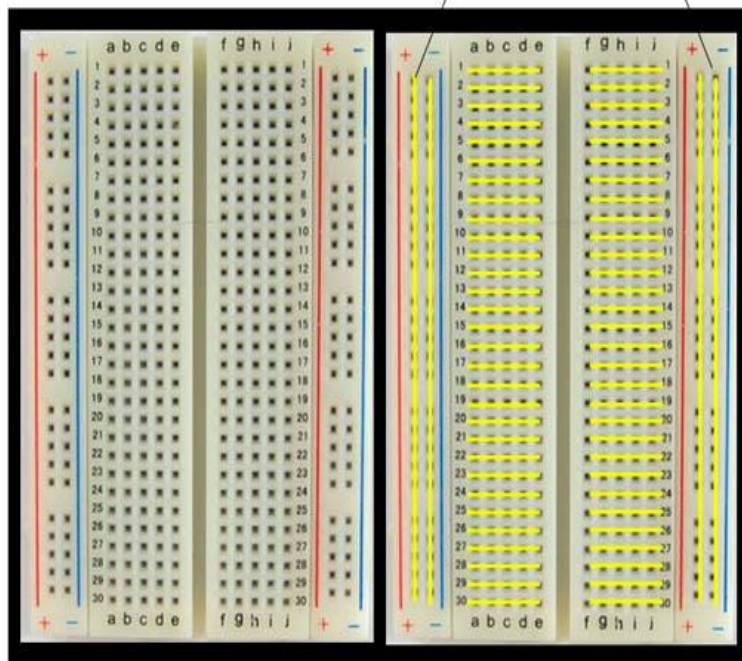


1
2
3
4

Breadboard(빵판)

Power bus

Ground bus



Directions for the use

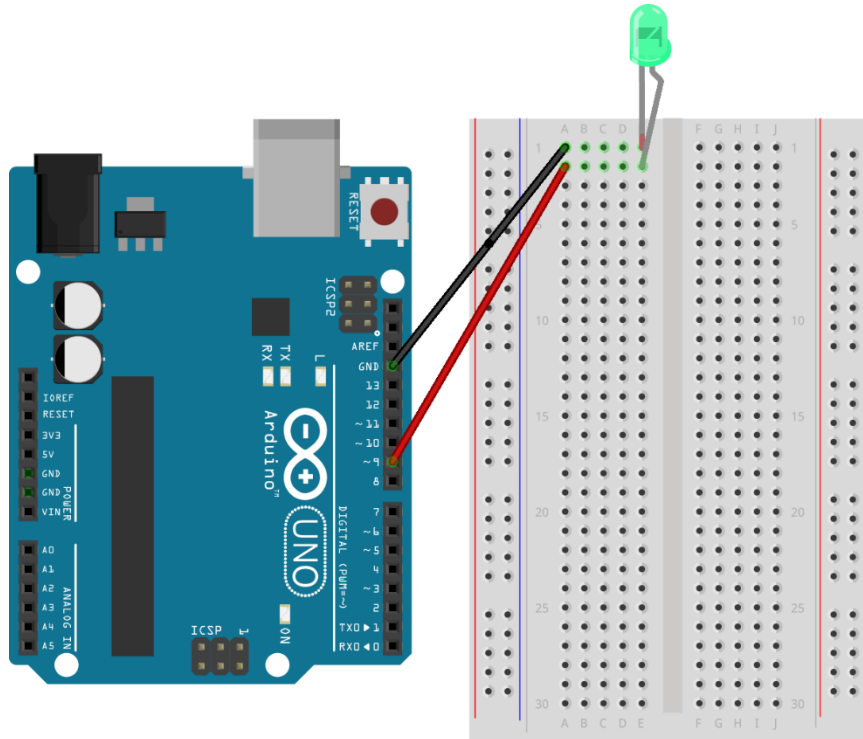
Connection

: Always disconnect an Arduino from external power sources (laptop, battery, etc.) before making an electrical circuit.

: When the Arduino is connected with external power sources
and you want to connect electronic components or jumper cables..
→Connect —(ground) first!

and you want to disconnect electronic components or jumper cables..
→Disconnect +(power) first!

1
2
3
4



fritzing

Led

Long lead : + (plus)
Short lead : - (minus)



1
2
3
4

Practice 4 — Blink

```
practice4_Blink $
#define LED 9
|
void setup(){
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("Start");
}

void loop(){
  digitalWrite(LED,HIGH);
  Serial.println("LED ON");
  delay(2000);
  digitalWrite(LED,LOW);
  Serial.println("LED OFF");
  delay(2000);
}

업로드 완료.
```

#define

: Give a name to a constant
: **#define** constantName value

pinMode()

: Configures the specified pin to behave either as an input or an output

: pinMode(pin, mode)

pin — the number of the pin whose mode you wish to set

mode — **INPUT** or **OUTPUT**

digitalWrite()

: Write a HIGH or a LOW value to a digital pin

: digitalWrite(pin, mode)

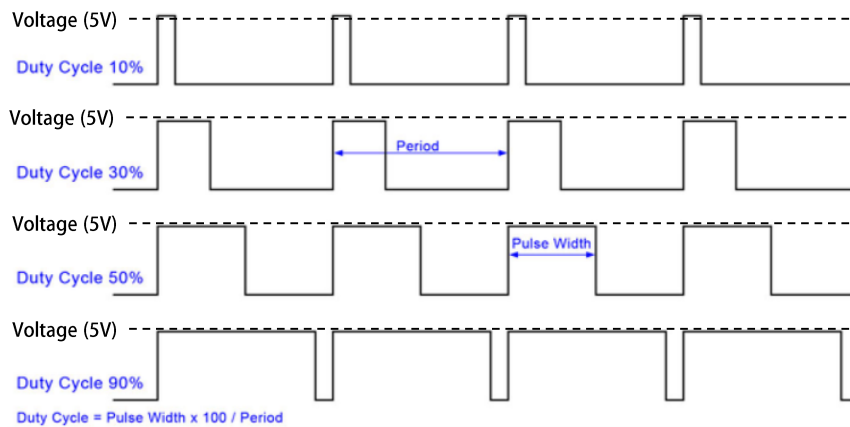
pin — the pin number

value — **HIGH** or **LOW**

PWM (Pulse Width Modulation)

1. 한 주기(Period)안에서 신호가 'ON' 상태인 시간을 지속시간 (Pulse Width)
2. 'ON' 시간과 'OFF' 시간의 비율을 Duty Cycle.

** 주기의 경우(t) 1/f 로 표현된다. $t=1/f$ (주파수)



Digital signals

: ON or OFF (interpreted in shorthand as 1 or 0)

Analog signals

: infinite number of positions between 0 and 1

PWM

: A way to control analog devices with a digital output. You can output a modulating signal from a digital device such as an Arduino to drive an analog device.





1
2
3
4

for statement

-The for statement is used to repeat a block of statements enclosed in curly braces. An increment counter is usually used to increment and terminate the loop. The for statement is useful for any repetitive operation.

-The initialization happens first and exactly once. Each time through the loop, the condition is tested; if it is true, the statement block, and the increment are executed. Then the condition is tested again. When the condition becomes false, the loop ends

```
: for(initialization; condition; increment){  
    statement (s);  
}
```

Examples

```
: for(int i=0; i<=255; i++){  
    statement (s);  
}
```

```
: for(int i=255; i>=0; i--){  
    statement (s);  
}
```

cf.
i++ is equal to i=i+1;
i-- is equal to i=i-1;



Practice 5 — Brightness

```
practice5_Brightness
#define LED 9
void setup() {
  //pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("Start");
}
void loop() {
  for(int i=0; i<=255; i++){
    analogWrite(LED,i);
    Serial.print("LED : ");
    Serial.println(i);
    delay(10);
  }
  /*for(int i=255; i>=0; i--){
    analogWrite(LED,i);
    Serial.print("LED : ");
    Serial.println(i);
    delay(10);
  }*/
}
```

analogWrite()

: Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds

: analogWrite(pin, value)

pin — the pin to write to

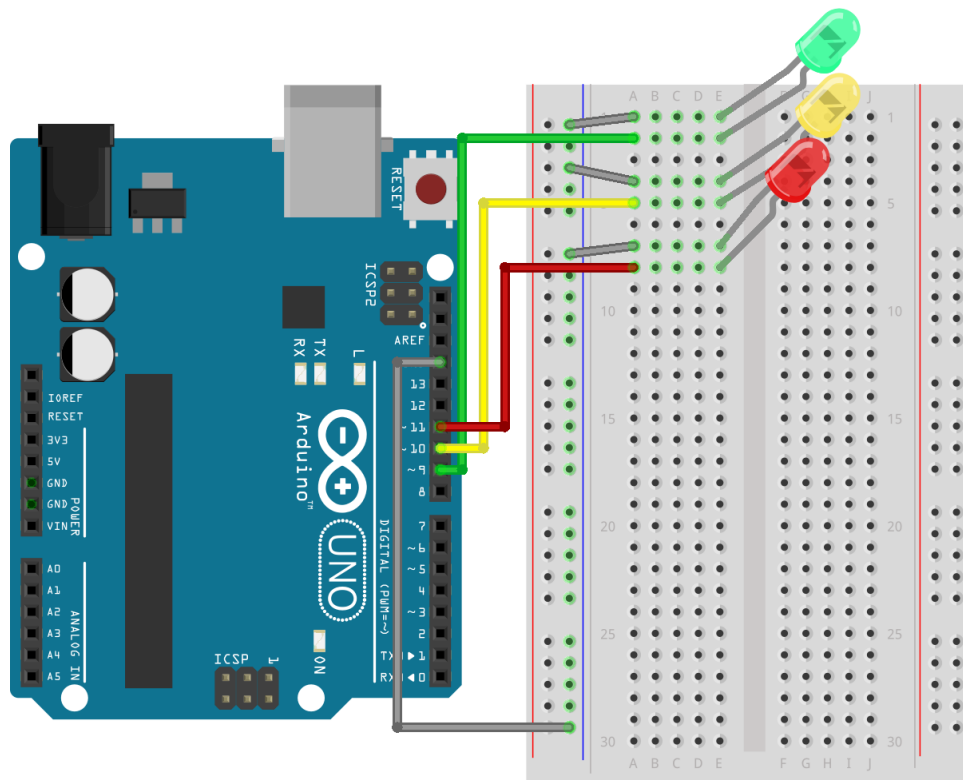
value — the duty cycle between 0 (always off) and 255 (always on)

Serial.print("")

: Print sentence on the serial monitor (no new line)



1
2
3
4



fritzing



1
2
3
4

Practice 6 — Traffic light

```
practice6_trafficlight $
```

```
#define Green 9
#define Yellow 10
#define Red 11
void setup() {
  pinMode(Green, OUTPUT);
  pinMode(Yellow, OUTPUT);
  pinMode(Red, OUTPUT);
  digitalWrite(Green, LOW);
  digitalWrite(Yellow, LOW);
  digitalWrite(Red, LOW);
  Serial.begin(9600);
  Serial.println("Start");
}
```

```
void loop() {
  digitalWrite(Green, HIGH);
  delay(3000);
  digitalWrite(Green, LOW);
  digitalWrite(Yellow, HIGH);
  delay(1000);
  digitalWrite(Yellow, LOW);
  digitalWrite(Red, HIGH);
  delay(3000);
  digitalWrite(Red, LOW);
}
```



1

2

3

4

References

- 누구나 따라하는 아두이노 프로그램, 바람
- Roboin Arduino Seminar PPT 2017~2019

If you have any questions, then email me (aaa3004@yonsei.ac.kr) or refer to the articles on <https://www.arduino.cc/reference/ko/>