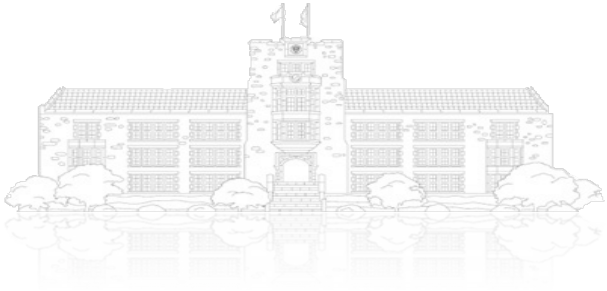
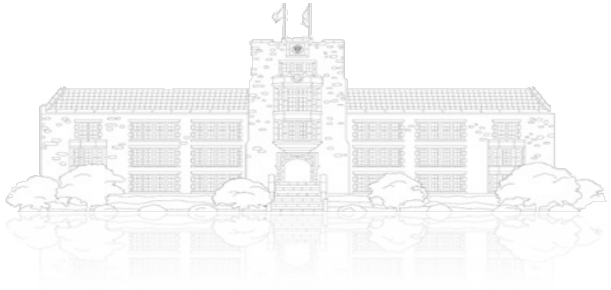
Maker Space Lecture
Week 1 – What is Arduino?



YONSEI, where we make *history*

연세대학교 기계공학과 우성하
sungha.woo@yonsei.ac.kr

https://github.com/SunghaWoo/Arduino-Seminar



YONSEI, where we make *history*

CONTENTS

1. Introduction to Arduino

2. Arduino Grammar

3. LED Tutorials

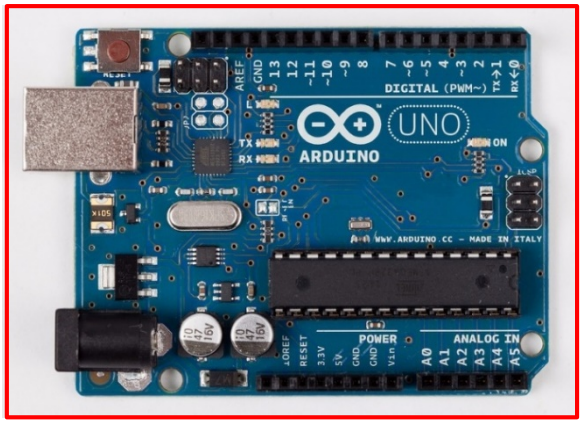
4. References

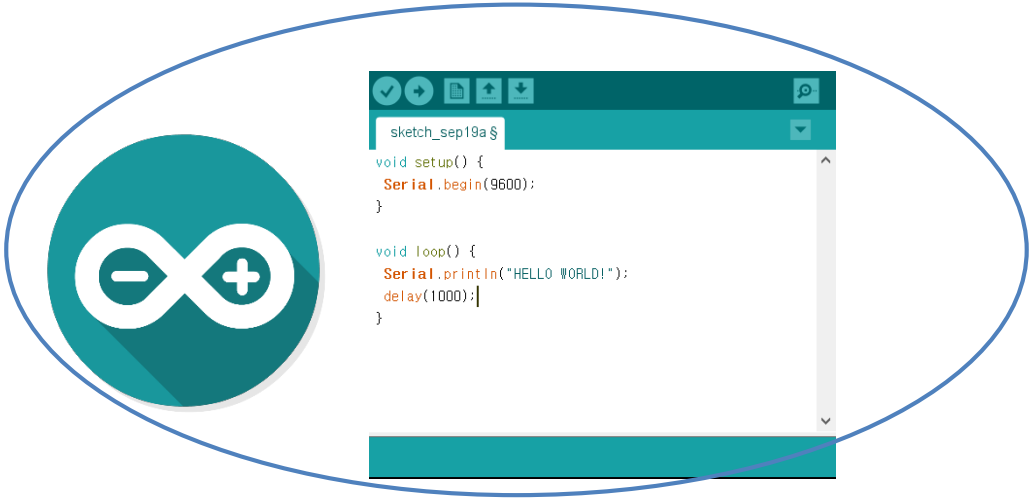
Definition of Arduino

Easy to share code, library, and even hardware!

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

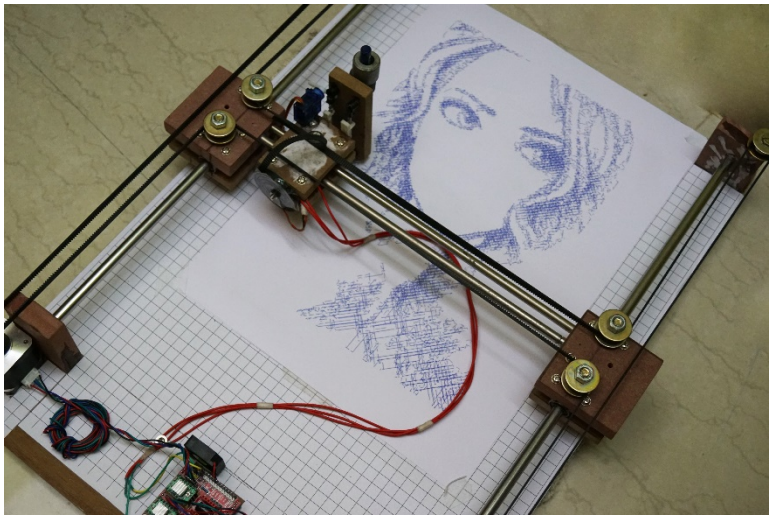
To use it, you use the Arduino programming language, and the Arduino Software(IDE).



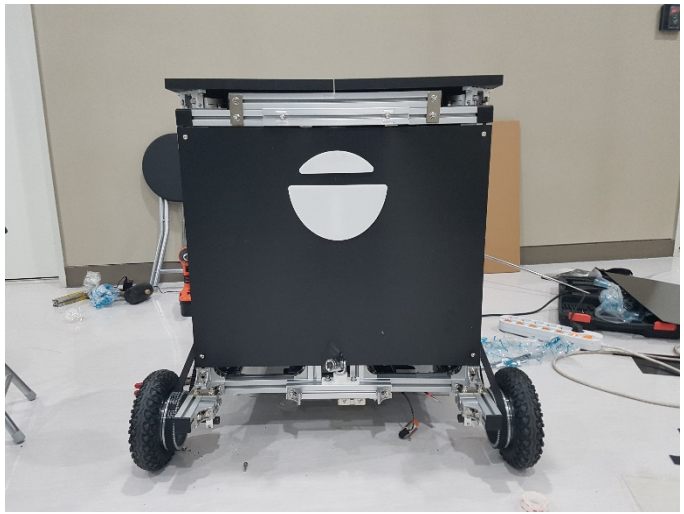


Applications of Arduino

https://www.youtube.com/watch?v=S6bjZ5kTxyM

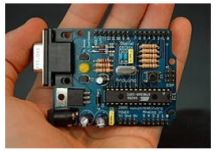


Arduino CNC Drawing Machine
Index-http://www.arnabkumardas.com/product/arduino-cnc-drawing-machine/

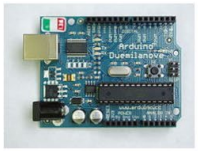


BOB-Delivery Robot with Table and Chairs

Types of Arduino



Arduino RS232[32]
(male pins)



Arduino Duemilanove[34]
(rev 2009b)

Arduino Uno R2[35][36]

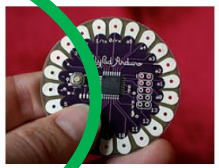
Arduino Uno SMD
R3[37]

Arduino Leonardo[38]

Arduino Pro[39]
(No USB)

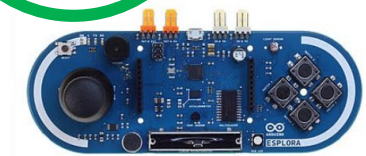
Arduino Mega[40]

Arduino
Nano[41]
(
footprint
)



Arduino LilyPad 00[42]
(rev 2007) (No USB)

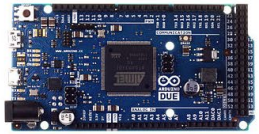
Arduino Robot[43]



Arduino Esplora[44]

Arduino Ethernet[45]
(AVR + W5100)

Arduino Yun[46]
(AVR + AR9331)

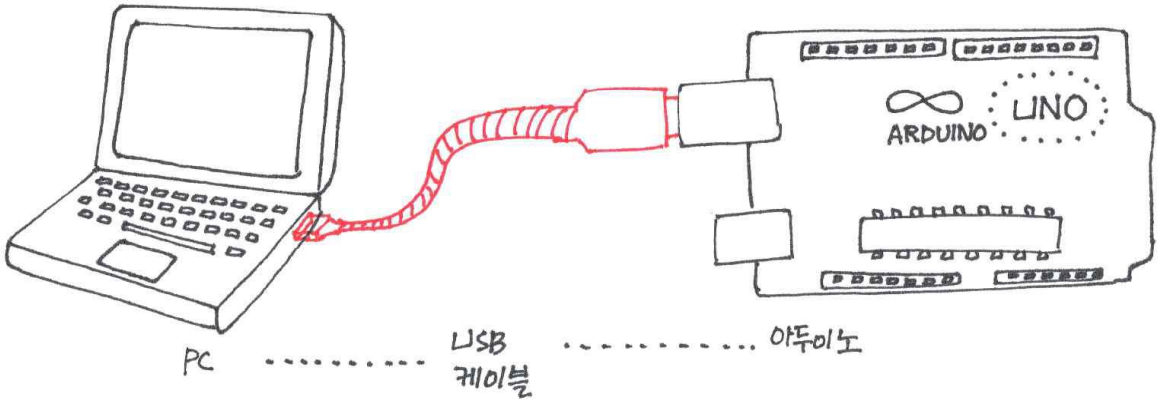


Arduino Due[47]
(ARM Cortex-M3 core)

Index-https://en.wikipedia.org/wiki/Arduino

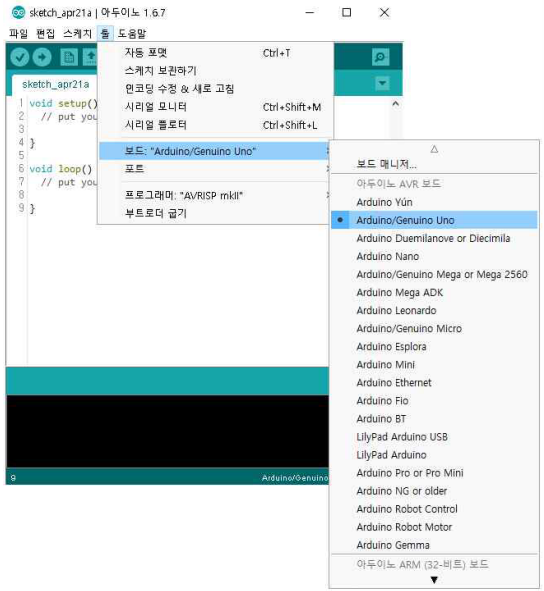
Arduino Connection

Step 1.
Connect as the picture

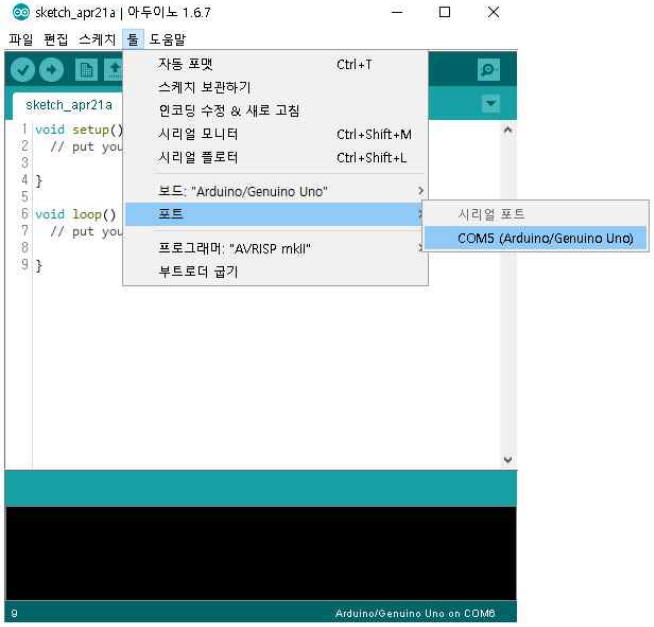


Arduino Connection

Step 2.
Tool-Board setting



Step 3.
Tool-Port setting



Basic Structure

setup()
: Initial setting function

loop()
: Real working & repeated function

Practice 0 - HELLO WORLD!

Serial.begin()

: Start serial communication

Serial.println(“”)

: Print sentence on the serial monitor and add a newline

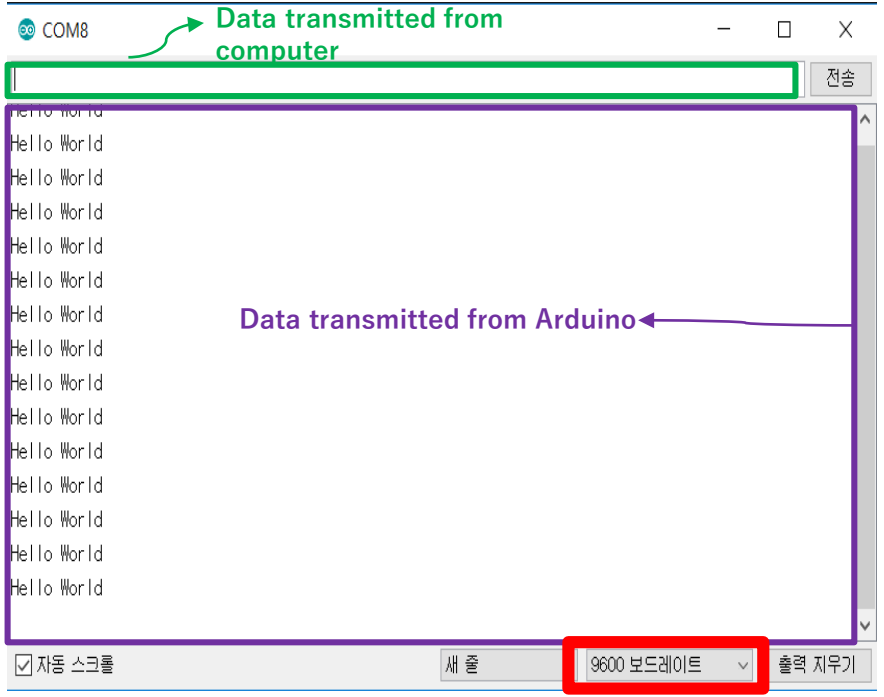
delay()

: Wait () milli seconds

Serial Communication?
https://m.blog.naver.com/yuyyulee/220301424499

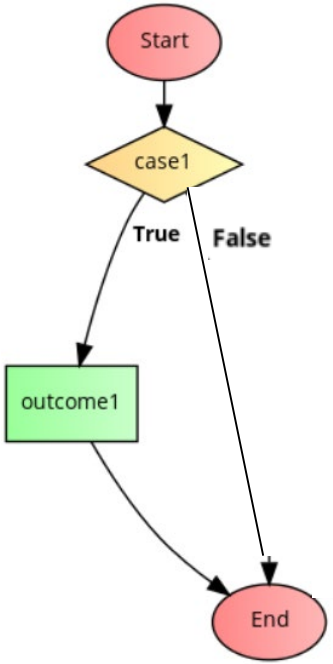
1 : Upload 2 : Click Serial monitor

3 : Set the baud rate



if statement

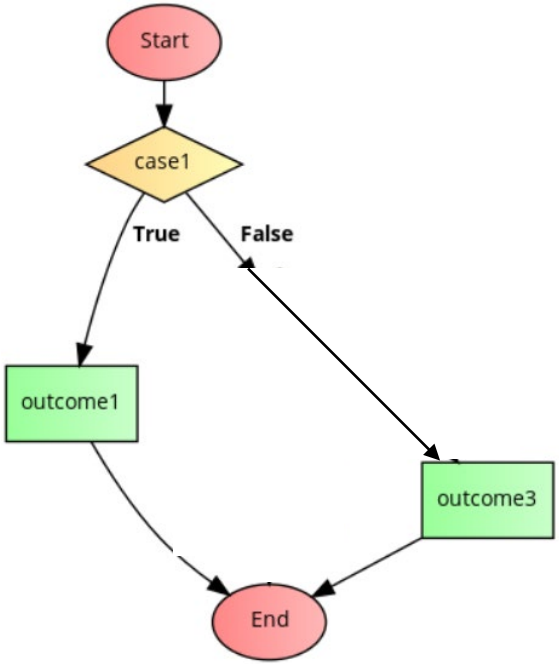
```
if(case1){
  outcome1
}
```



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

if statement

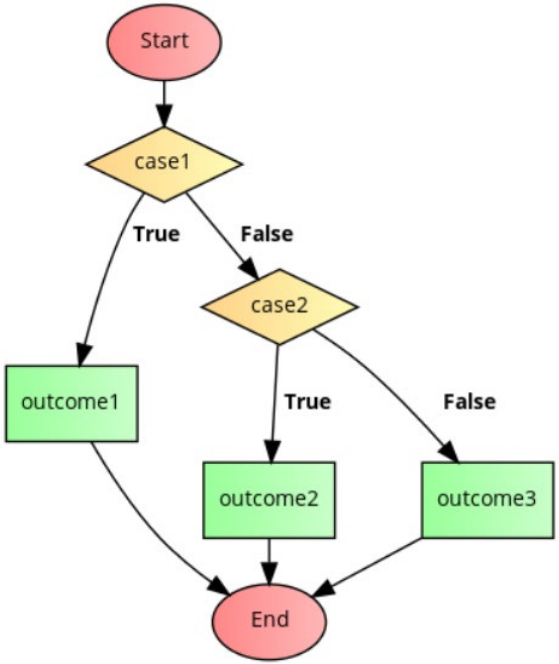
```
if(case1){
  outcome1
}

else(){
  outcome3
}
```



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

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’

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

Function

sketch_jan16a §

```
return_type function_name( parameter_list ) {
  body of the function
  return output;
}
```

확인

sketch_apr16a §

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

return_type
The data type of "the value the function returns"(=output)
You can use "int", "float", "char", "void" …
When performing the desired operations without returning a value, the return_type is the "void"

function_name
The actual name of the function
NO 한글 & "Built In functions' names"

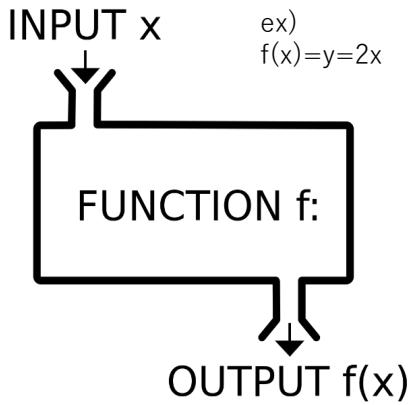
parameters
The input to use for the desired operations
A function may contain no parameters → ( )

return
Return the desired value after executing the function
The output's data type must be the same as return_type
When return_type is the "void", omit "return" or use "return;"

{} Paragraph

; End a statement

Blockcomment // for one line /* */ for multiple lines



Practice 1

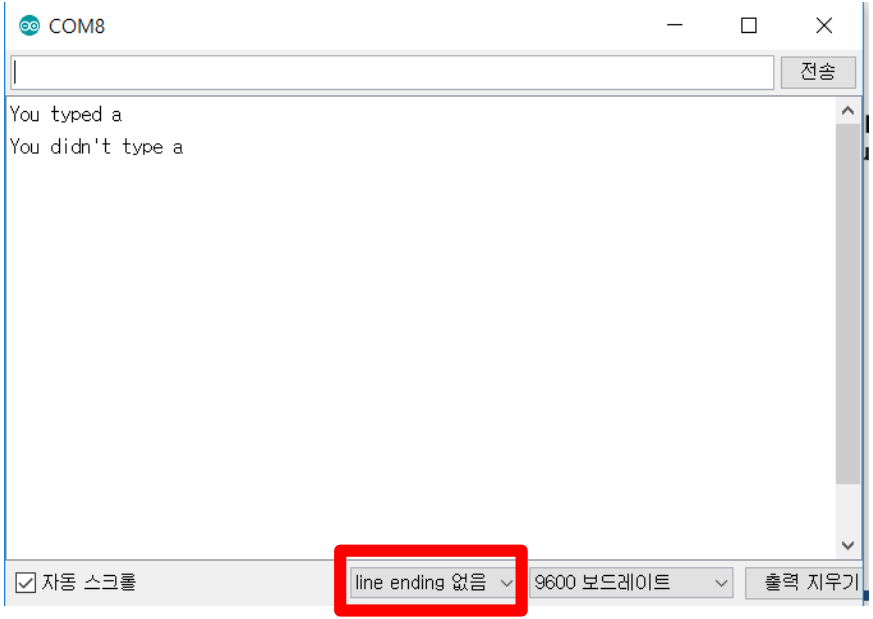

```
void setup() {
  Serial.begin(9600);
}

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");
    }
  }
}
```

Serial.available()
: Get the number of bytes available for reading from serial port

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

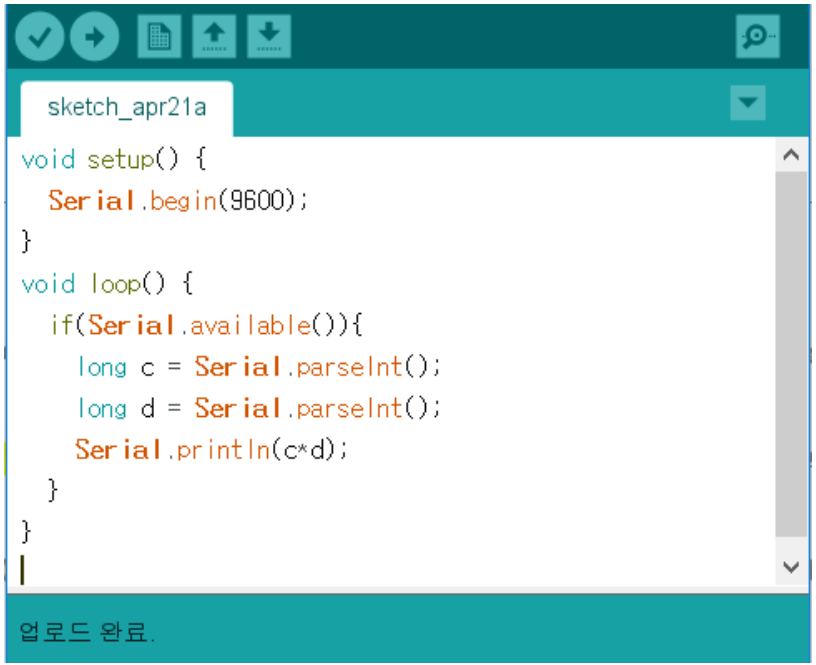
Serial. read()
: Reads incoming serial data and returns the first byte of incoming serial data available
(or -1 if no data is available)

Practice 1



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'

Practice 2 – About Numbers

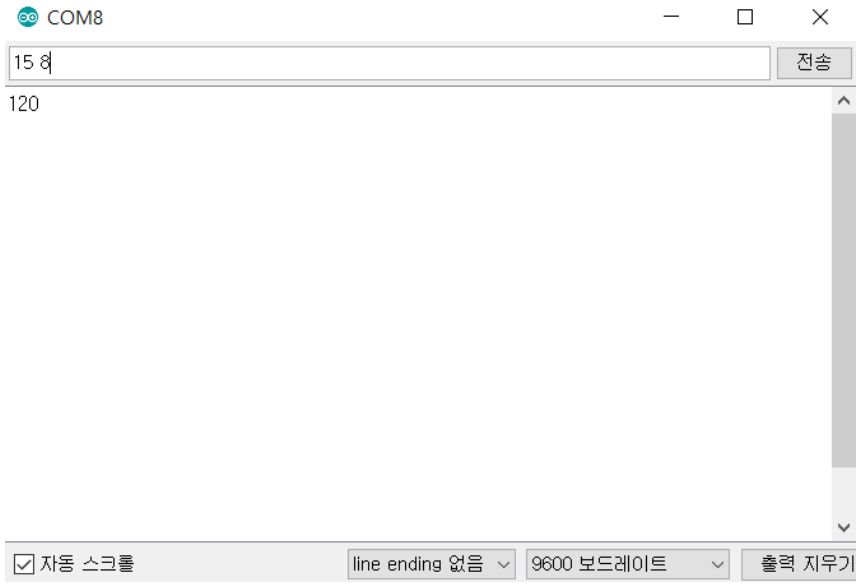


Serial.parseint()
: We cannot use Serial.read

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

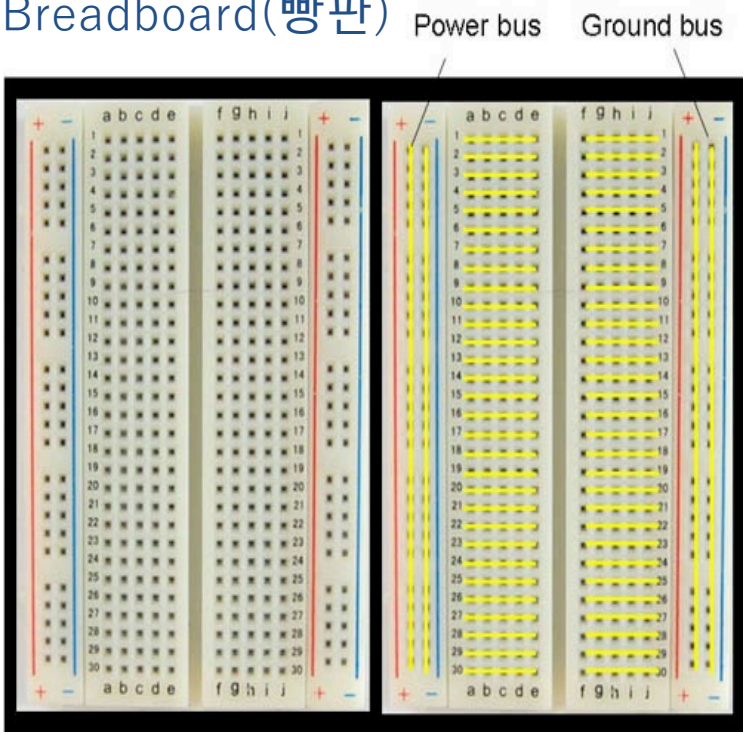
: if input is number, returns long type number, if not, returns 0

Practice 2 – About Numbers



Practice 3 – User-defined Function

Breadboard(빵판)

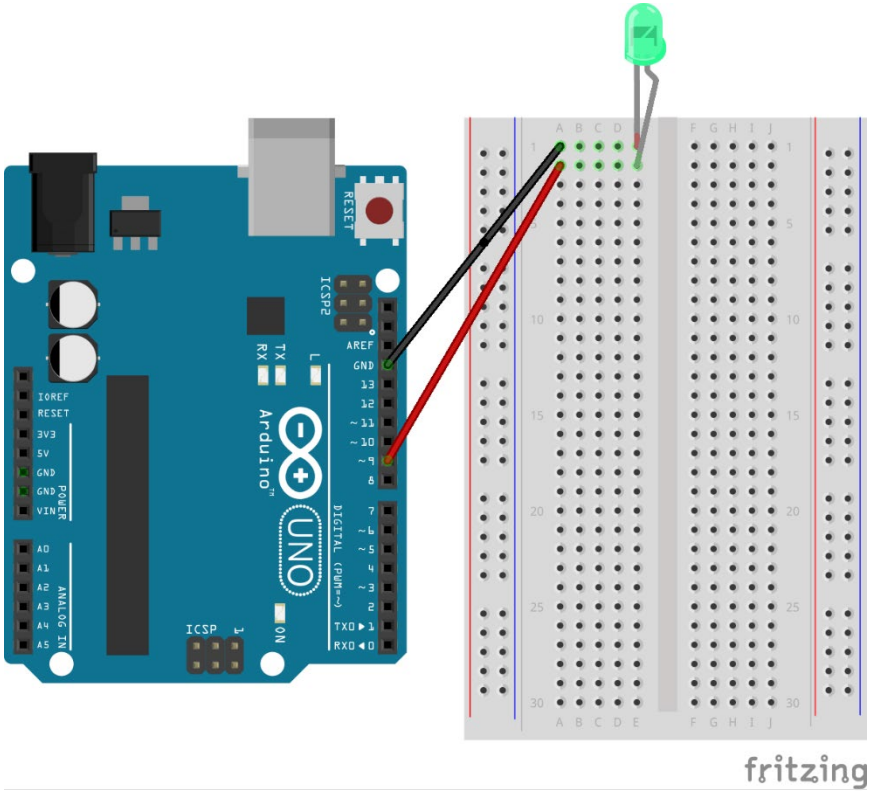


Directions for the use

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!

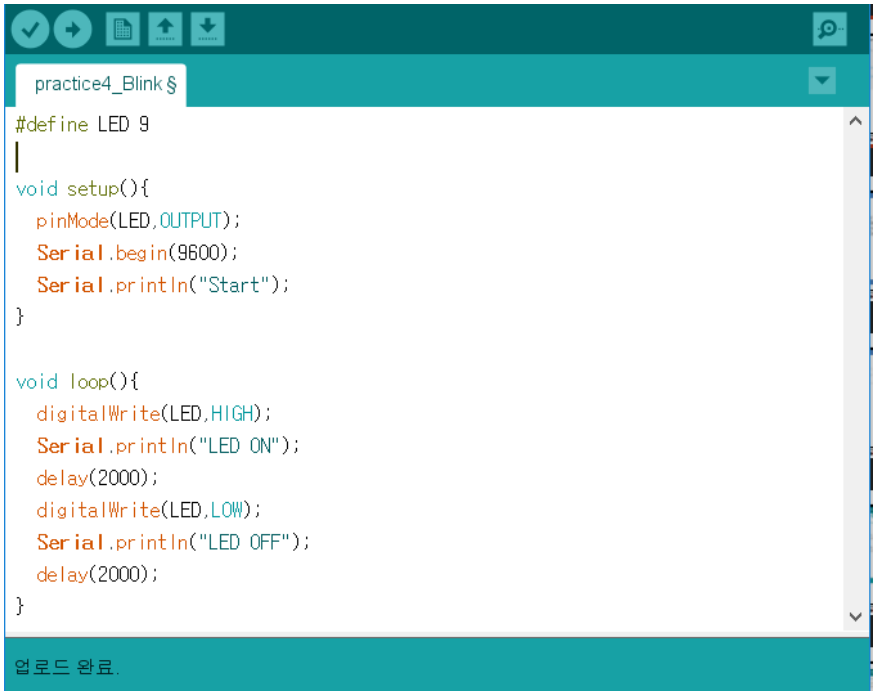
and you want to disconnect electronic components or jumper cables..
→Disconnect +(power) first!



Led

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

Practice 4 – Blink



#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 OUPUT

digitalWrite()

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

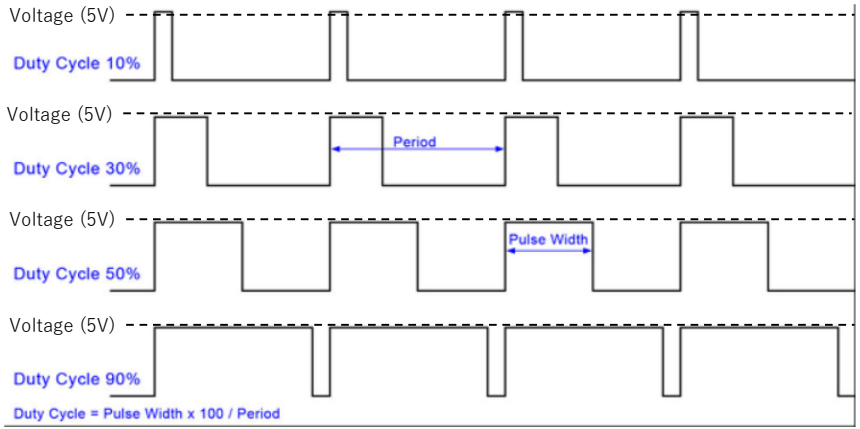
: digitalWrite(pin, mode)
pin – the pin number
value – HIGH or LOW

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

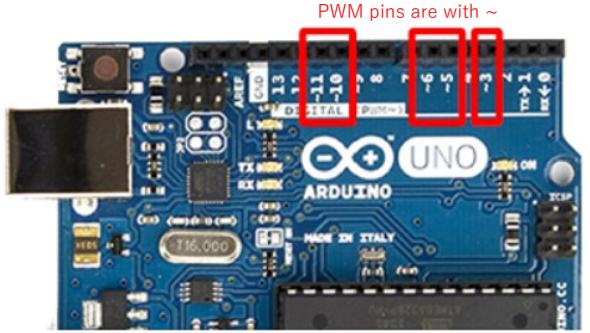
: ON or OFF (interpreted in shorthand as 1 or 0)

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.



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()

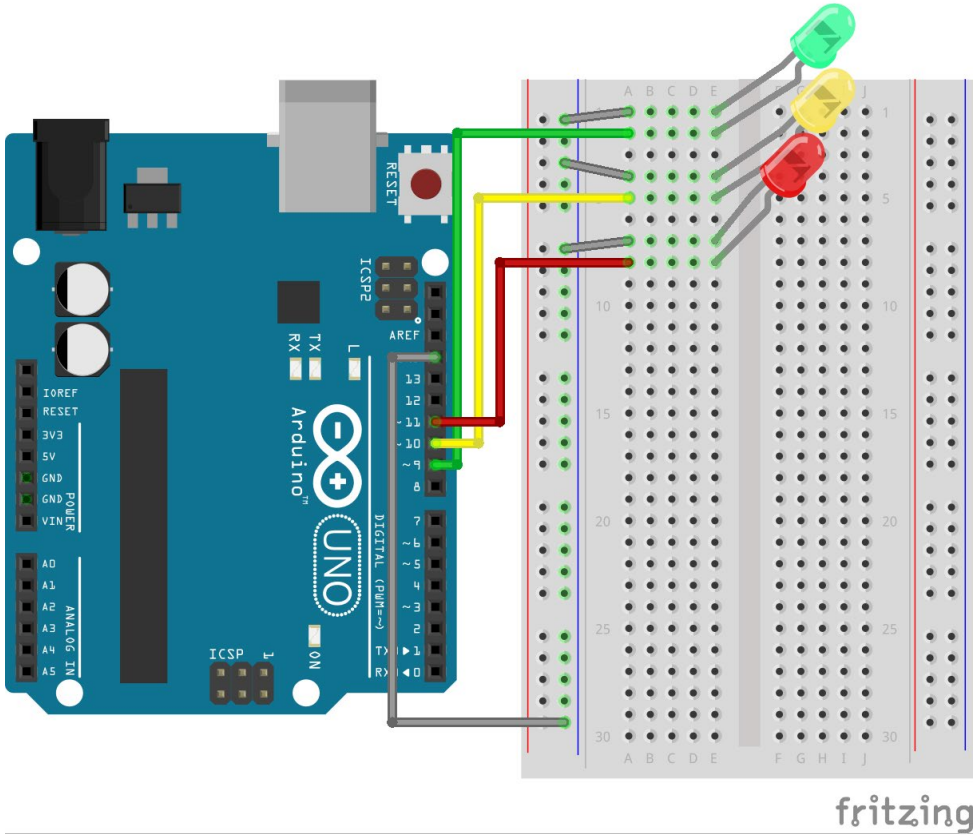
: Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds

: 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


RESET
ICSP2
AREF
TX
RX
L
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
Arduino
UNO
DIGITAL (PWM~)
ICSP
ON
A B C D E
F G H I J
fritzing

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);
}
```

References

https://codingrun.com/65
https://forum.arduino.cc/
https://dokkodai.tistory.com/188

If you have any questions, then email me (sungha.woo@yonsei.ac.kr)
You can also refer to the articles on https://www.arduino.cc/reference/ko/