

비교 연산자

- Relational Operators

기호	뜻	예제	값
>	큰가	$6 > 4$	참(1)
\geq	같거나 큰가	$7 \geq 6$	참(1)
<	작은가	$9 < 8$	거짓(0)
\leq	작거나 같은가	$5 \leq 5$	참(1)
$=\!=$	같은가	$3 \!=\! 4$	거짓(0)
$!\!=$	같지않다	$8.0 \!=\! 8$	거짓(0)

참 = 1

거짓 = 0

if (1) 이어지는 중괄호 안에 있는 작업수행 { }

if (0) 이어지는 중괄호 건너 뛴다 { }

2) Digital input 2 : Push button



Push button



2) Digital input 2 : Push button

준비물 :

아두이노 보드 1개

버튼 스위치 1개

LED 1개

220 Ω 저항 1개, 10K Ω 저항 1개

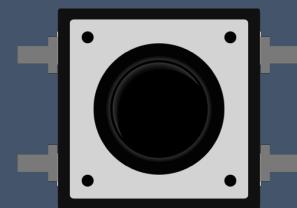
브래드 보드 1개

점퍼선

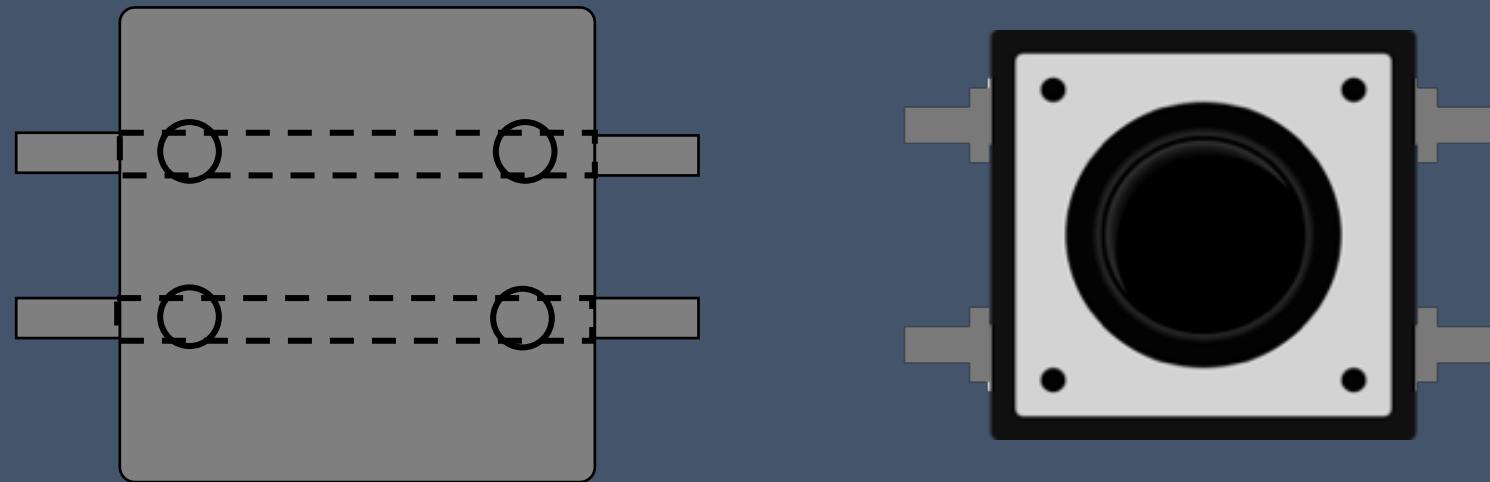
시리얼 모니터

풀다운

풀업



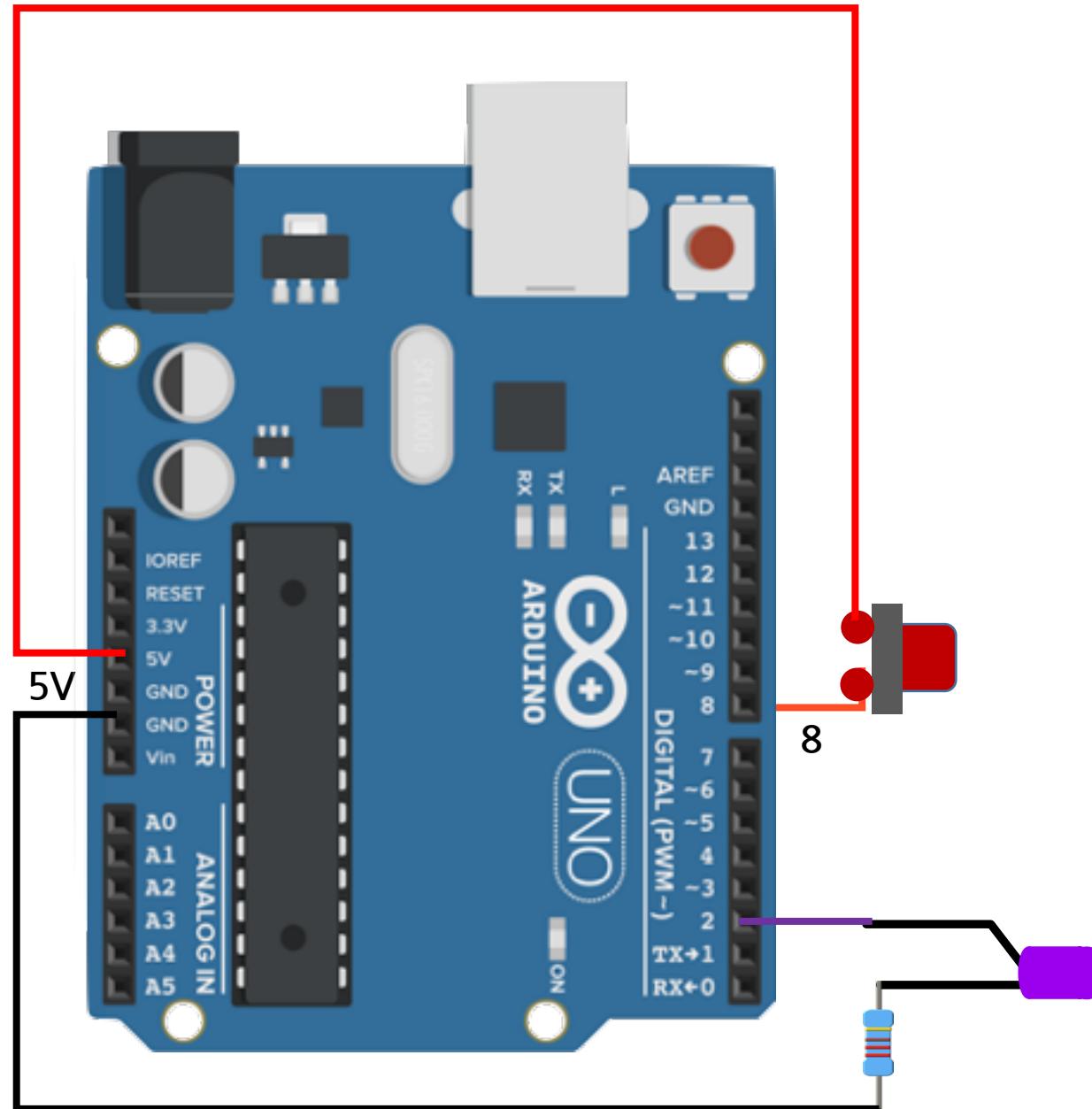
Digital input



푸시 버튼

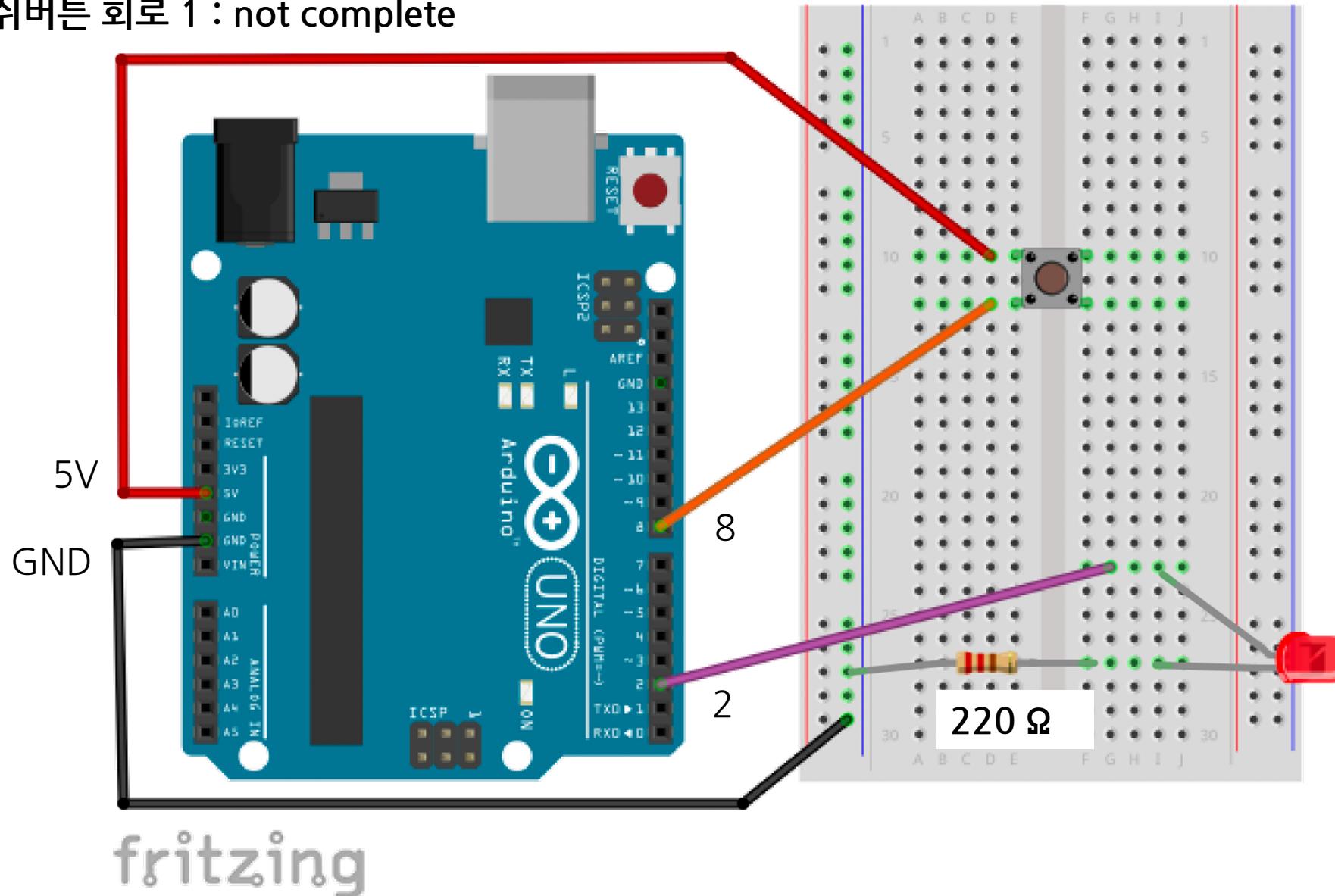
Digital input 2 : Push button

푸쉬버튼 회로 1 : not complete

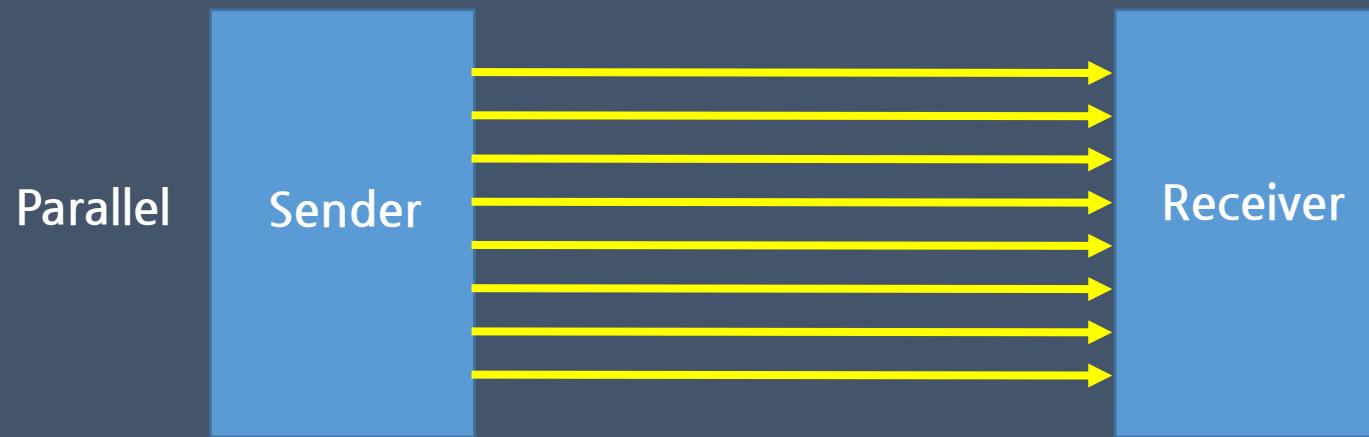
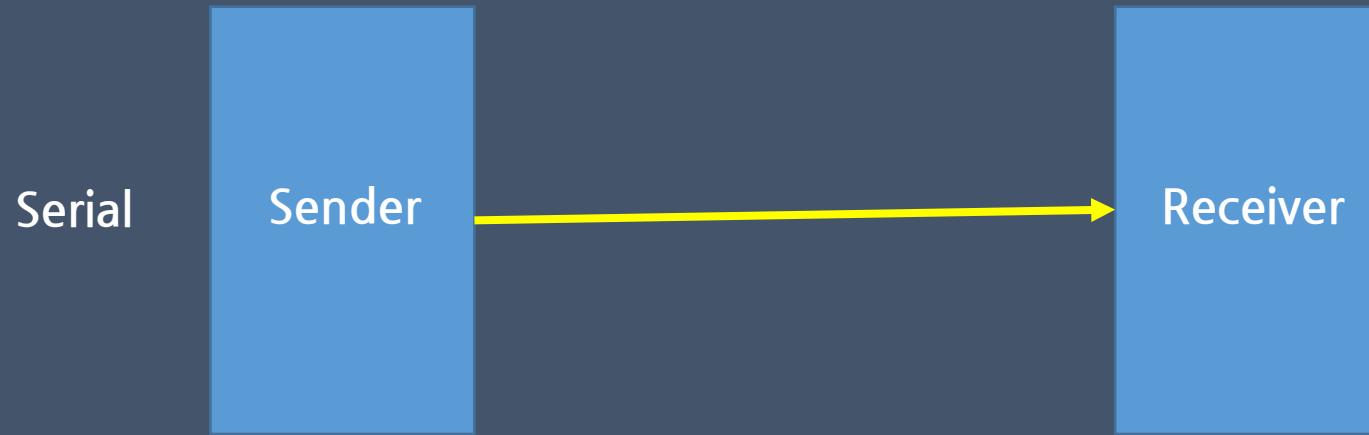


Digital input 2 : Push button

푸쉬버튼 회로 1 : not complete



Serial vs Parallel : 장단점



Baud Rate vs bps

Baud Rate : Number of signal change per second

Baud Rate \simeq bps

2) Digital input 2 : Push button

Sketch with Serial print

The screenshot shows the Arduino IDE interface with a sketch titled "Serial_Monitor". The sketch code is as follows:

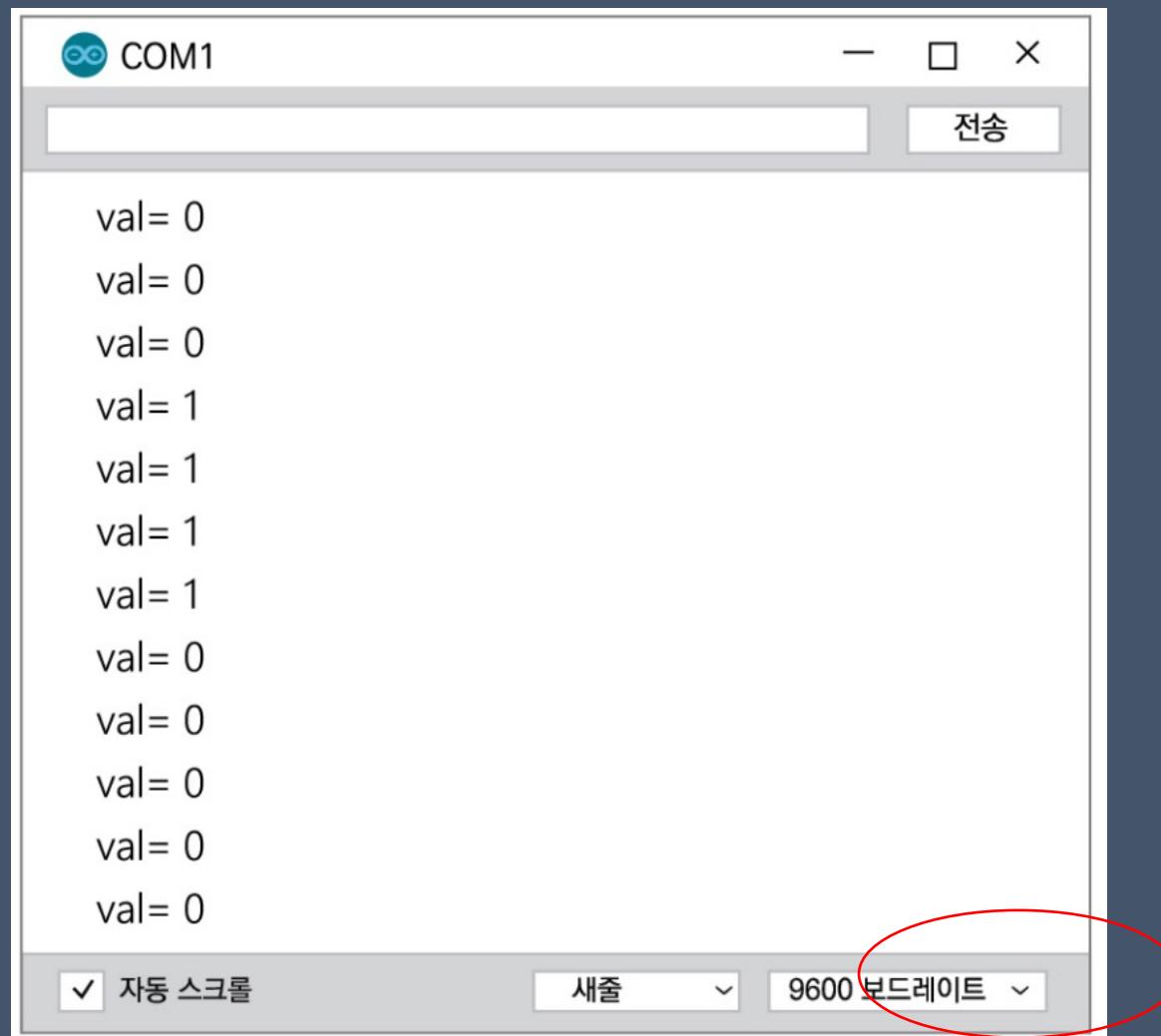
```
1 // Serial Monitor Example
2
3 void setup() {
4     pinMode(2, OUTPUT);
5     pinMode(8, INPUT);
6     Serial.begin(9600); ←
7 }
8
9 void loop() {
10    int val=digitalRead(8);
11
12    Serial.print("val= ");
13    Serial.println(val); ←
14    delay(10);
15
16    if(val==1) {
17        digitalWrite(2, HIGH);
18    }
19    else { digitalWrite(2, LOW); }
20 }
```

Annotations in the image include:

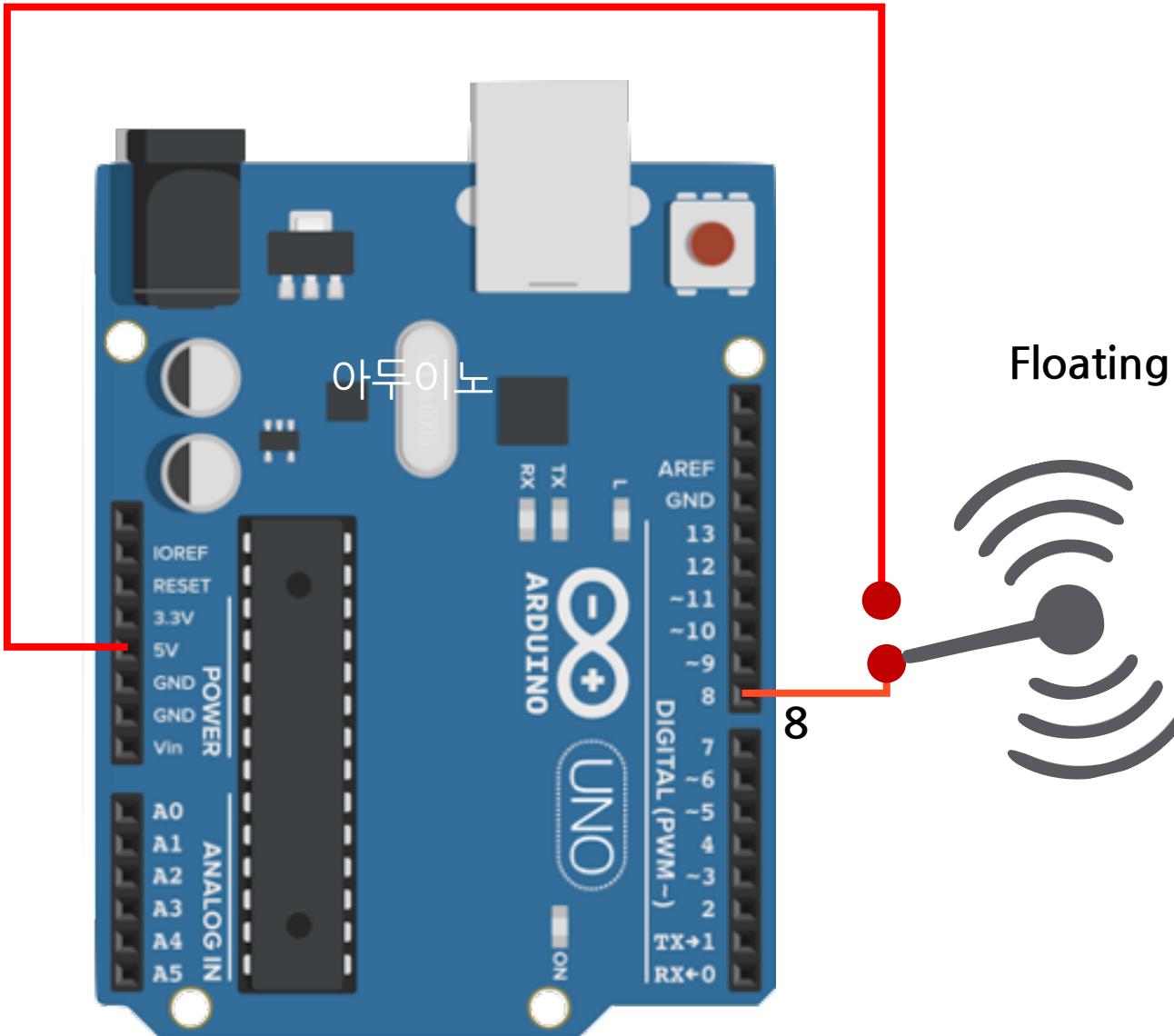
- A red arrow points from the top of the "Serial.begin(9600)" line to the "Serial.begin(9600)" icon in the toolbar.
- A red arrow points from the top of the "Serial.print("val= ") ;" line to the "Serial.print" icon in the toolbar.
- A red arrow points from the top of the "Serial.println(val) ;" line to the "Serial.println" icon in the toolbar.

2) Digital input 2 : Push button

버튼을 누르지도 않았는데 1 이 프린트 되는 현상 발생



2) Digital input 2 : Push button



2) Digital input 2 : Push button

28.2 DC Characteristics

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 1.8\text{V}$ to 5.5V (unless otherwise noted)

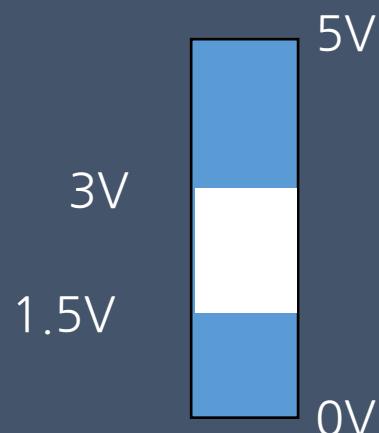
Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage, except XTAL1 and <u>RESET</u> pin	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage, except XTAL1 and <u>RESET</u> pins	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$		$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$	$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL1}	Input Low Voltage, XTAL1 pin	$V_{CC} = 1.8\text{V} - 5.5\text{V}$	-0.5		$0.1V_{CC}^{(1)}$	V
V_{IH1}	Input High Voltage, XTAL1 pin	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$		$0.8V_{CC}^{(2)}$ $0.7V_{CC}^{(2)}$	$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL2}	Input Low Voltage, <u>RESET</u> pin	$V_{CC} = 1.8\text{V} - 5.5\text{V}$	-0.5		$0.1V_{CC}^{(1)}$	V
V_{IH2}	Input High Voltage, <u>RESET</u> pin	$V_{CC} = 1.8\text{V} - 5.5\text{V}$		$0.9V_{CC}^{(2)}$	$V_{CC} + 0.5$	V
V_{IL3}	Input Low Voltage, <u>RESET</u> pin as I/O	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH3}	Input High Voltage, <u>RESET</u> pin as I/O	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$		$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$	$V_{CC} + 0.5$ $V_{CC} + 0.5$	V

2) Digital input 2 : Push button

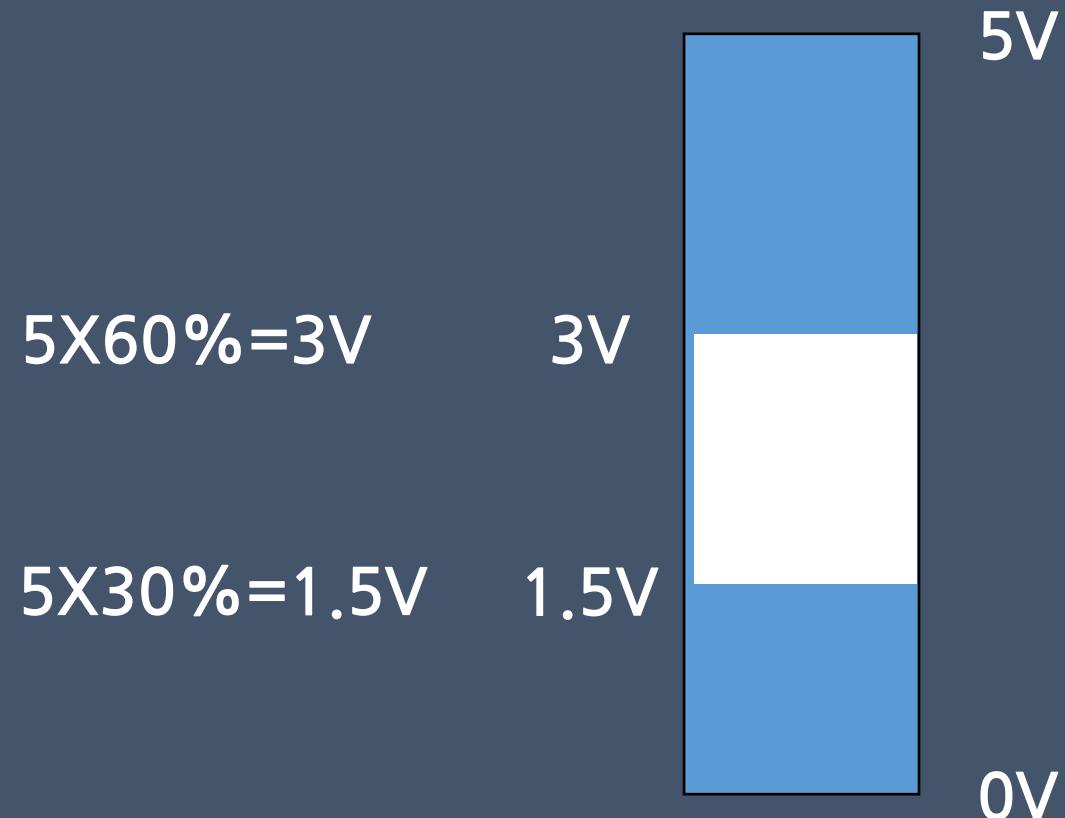
Symbol	Parameter	Condition	Min.	Typ.	Max.
V_{IL}	<u>Input Low Voltage, except XTAL1 and RESET pin</u>	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$
V_{IH}	<u>Input High Voltage, except XTAL1 and RESET pins</u>	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$

Input Low Voltage = $0.3 \times 5V = 1.5V$: Low MAX

Input HIGH Voltage = $0.6 \times 5V = 3.0V$: High Min

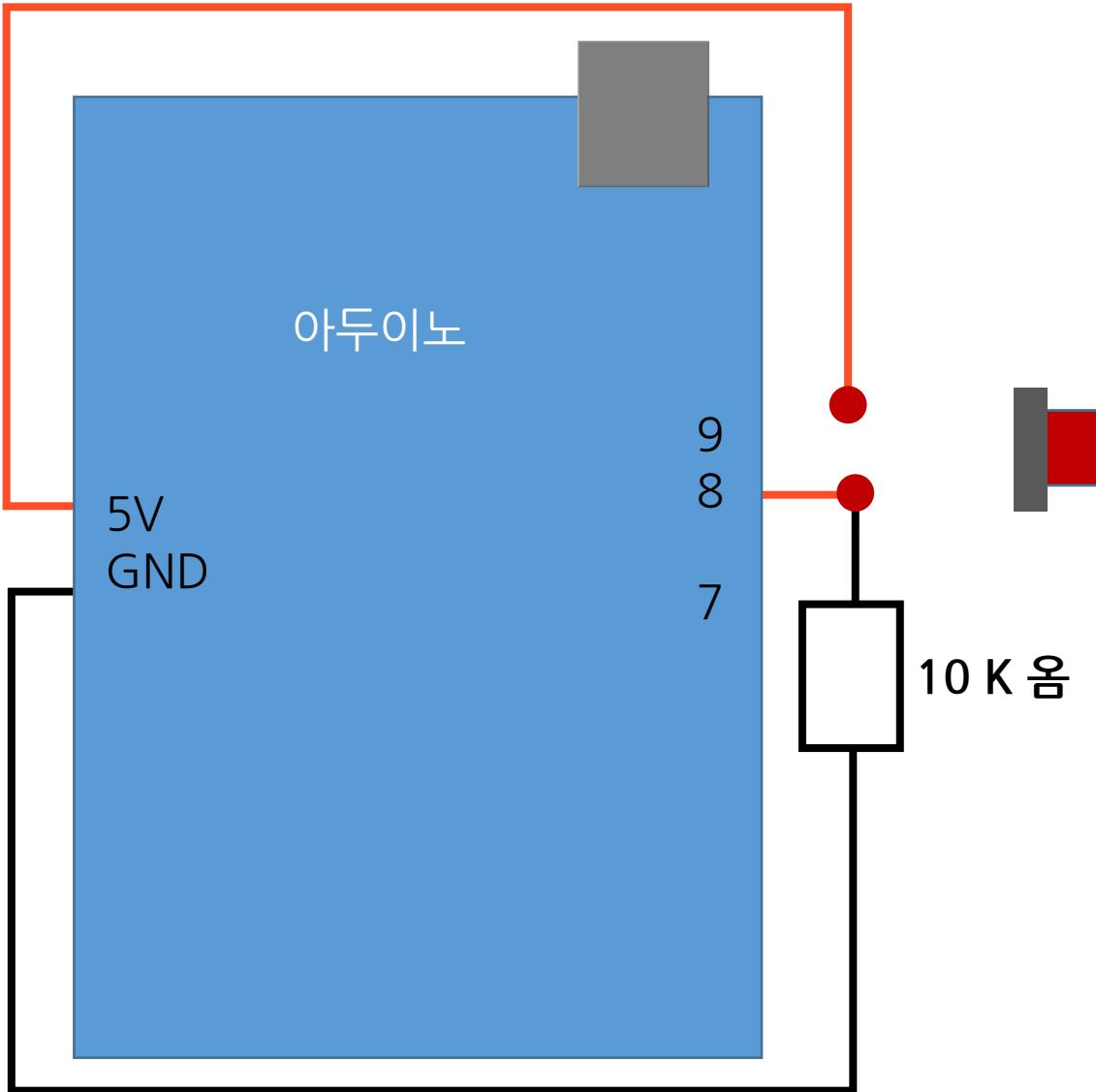


2) Digital input 2 : Push button

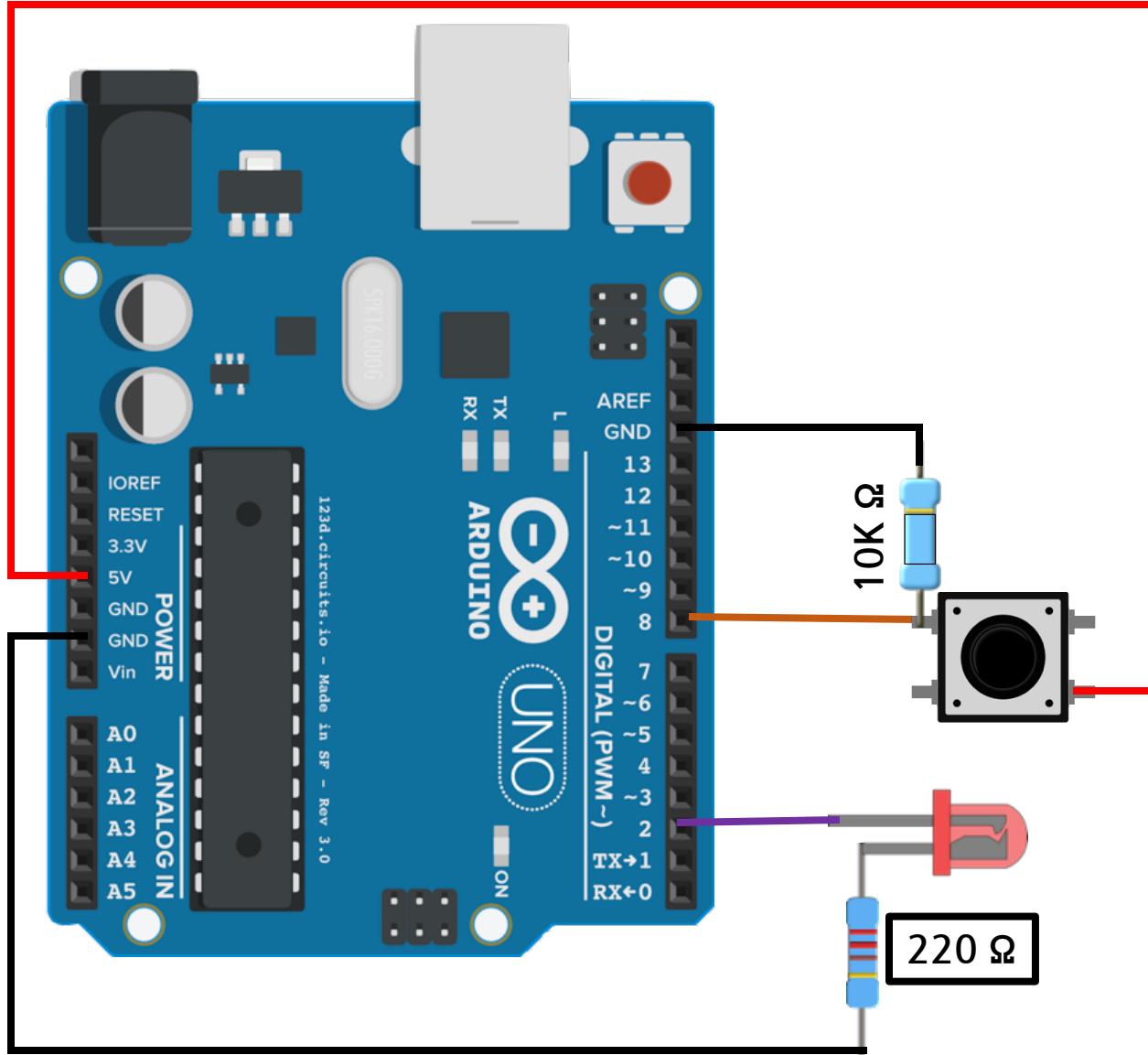


2) Digital input 2 : Push button

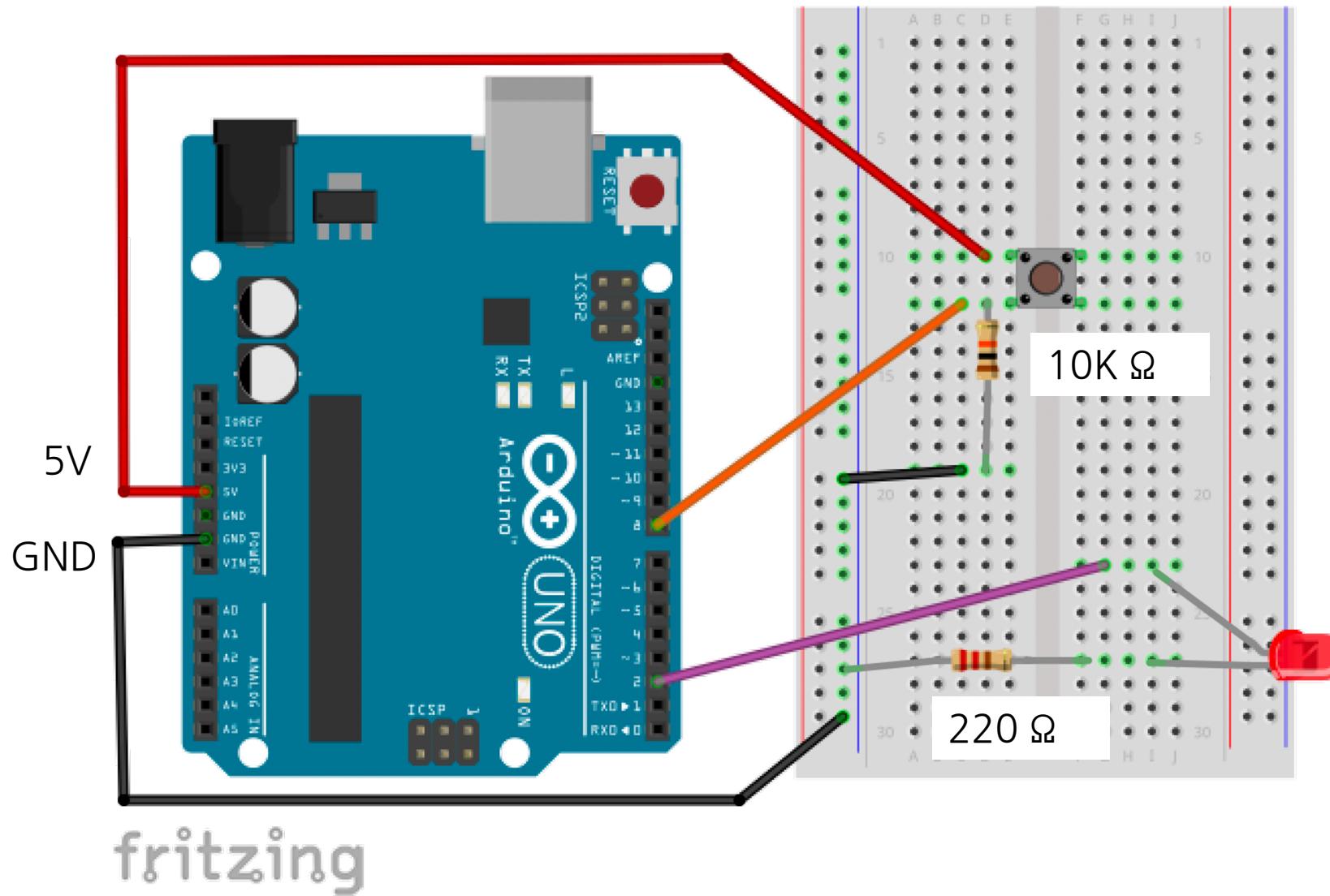
Pull Down 회로



PULL Down circuit



PULL Down circuit



2) Digital input 2 : Push button

Serial monitor

The image displays two identical screenshots of the Arduino Serial Monitor window, side-by-side. Both windows are titled 'COM1' and show a list of received data. The data consists of the string 'val= ' followed by either '0' or '1'. In the left window, all 12 lines of data show 'val= 0'. In the right window, all 12 lines of data show 'val= 1'. Each window has a 'Transmit' button labeled '전송' in Korean at the top right. At the bottom, there are checkboxes for '자동 스크롤' (Auto Scroll), a dropdown menu for '새줄' (Newline), and a dropdown menu for '9600 보드레이트' (9600 Baud Rate).

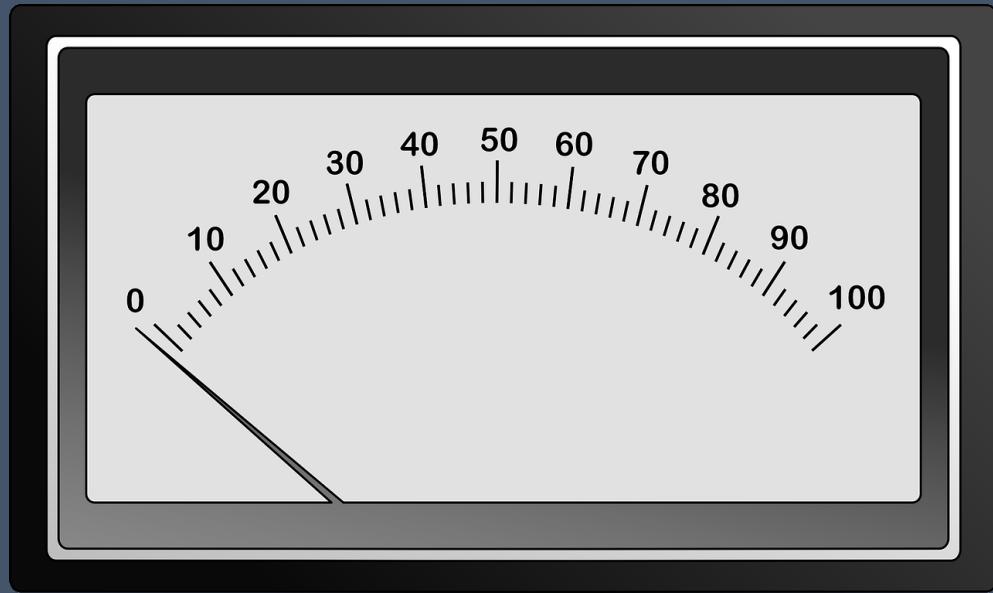
Window	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10	Line 11	Line 12
Left (Digital Input 0)	val= 0	val= 0	val= 0									
Right (Digital Input 1)	val= 1	val= 1	val= 1									

Ch3. Digital INPUT/Analog INPUT

1) Digital INPUT

2) Analog INPUT

2) Analog INPUT



3) Analog INPUT

3-1) Analog INPUT 1 : Potentiometer :

준비물 :

아두이노 보드 1개

포텐시오미터 1개

LED 1개

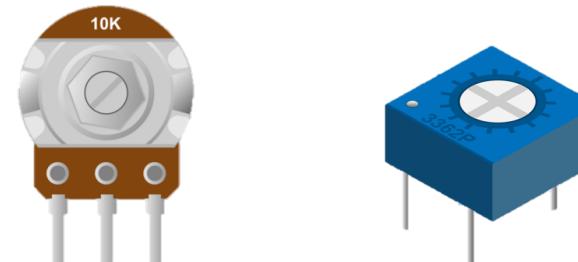
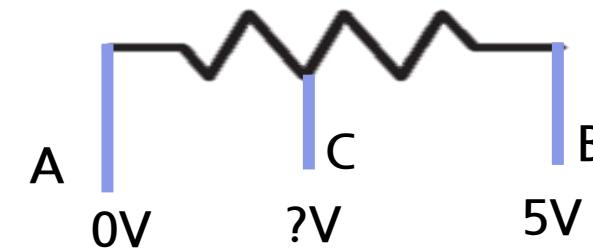
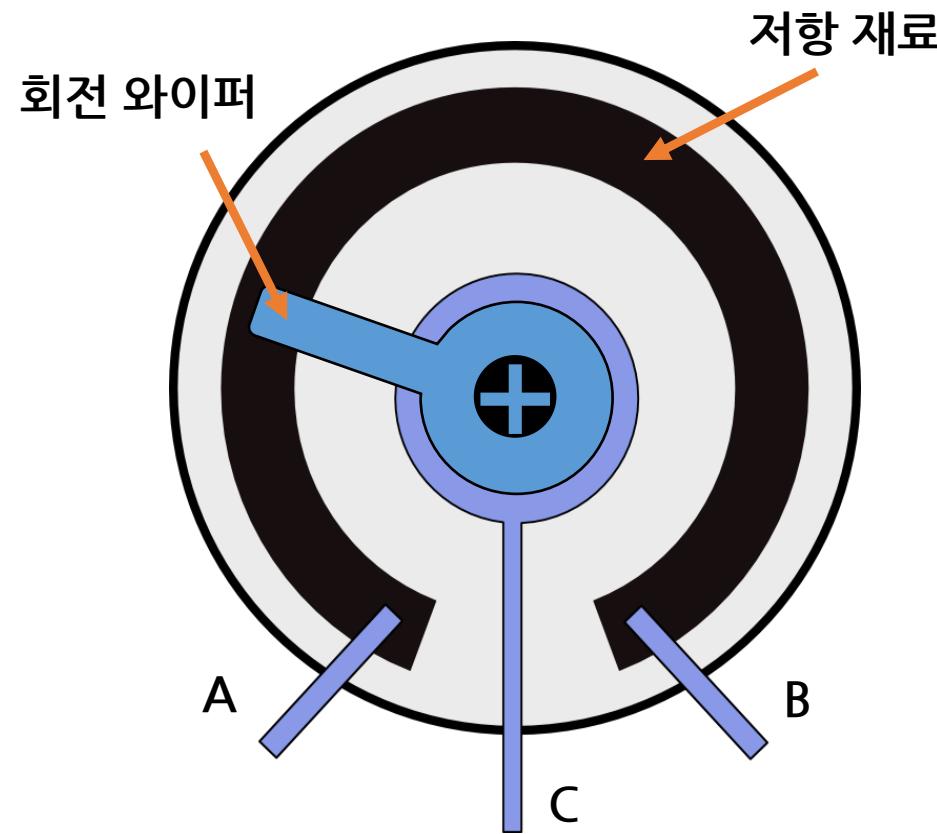
220 Ω 저항 1개

브래드 보드 1개

점퍼선



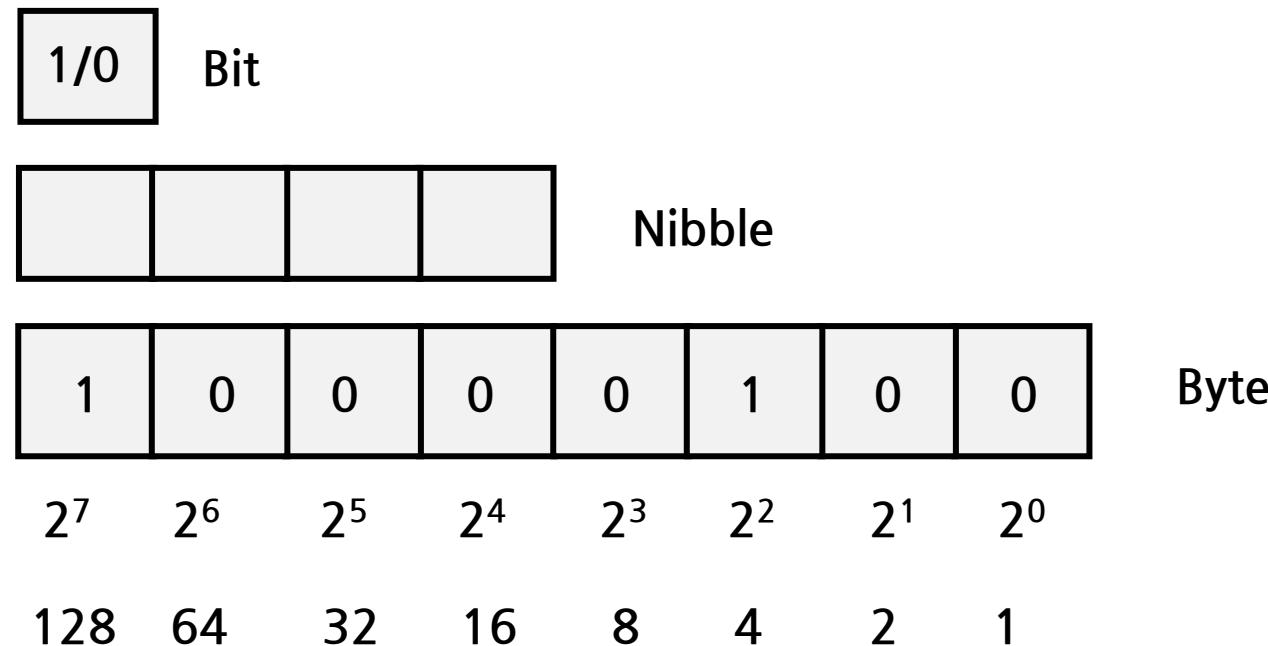
3-1) Analog INPUT 1 : potentiometer



Potentiometer (가변저항) 내부

3-1) Analog INPUT 1 : potentiometer

10 bit Analog register

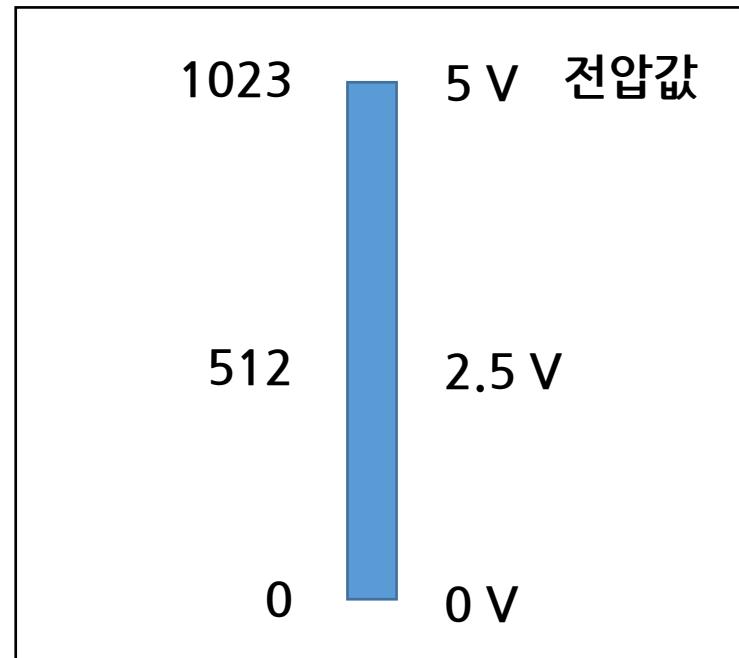


8 비트로 나타낼수 있는 최대 10 진수 숫자는 ?

10 비트로 나타낼수 있는 최대 10 진수 숫자는 ?

3-1) Analog INPUT 1 : potentiometer

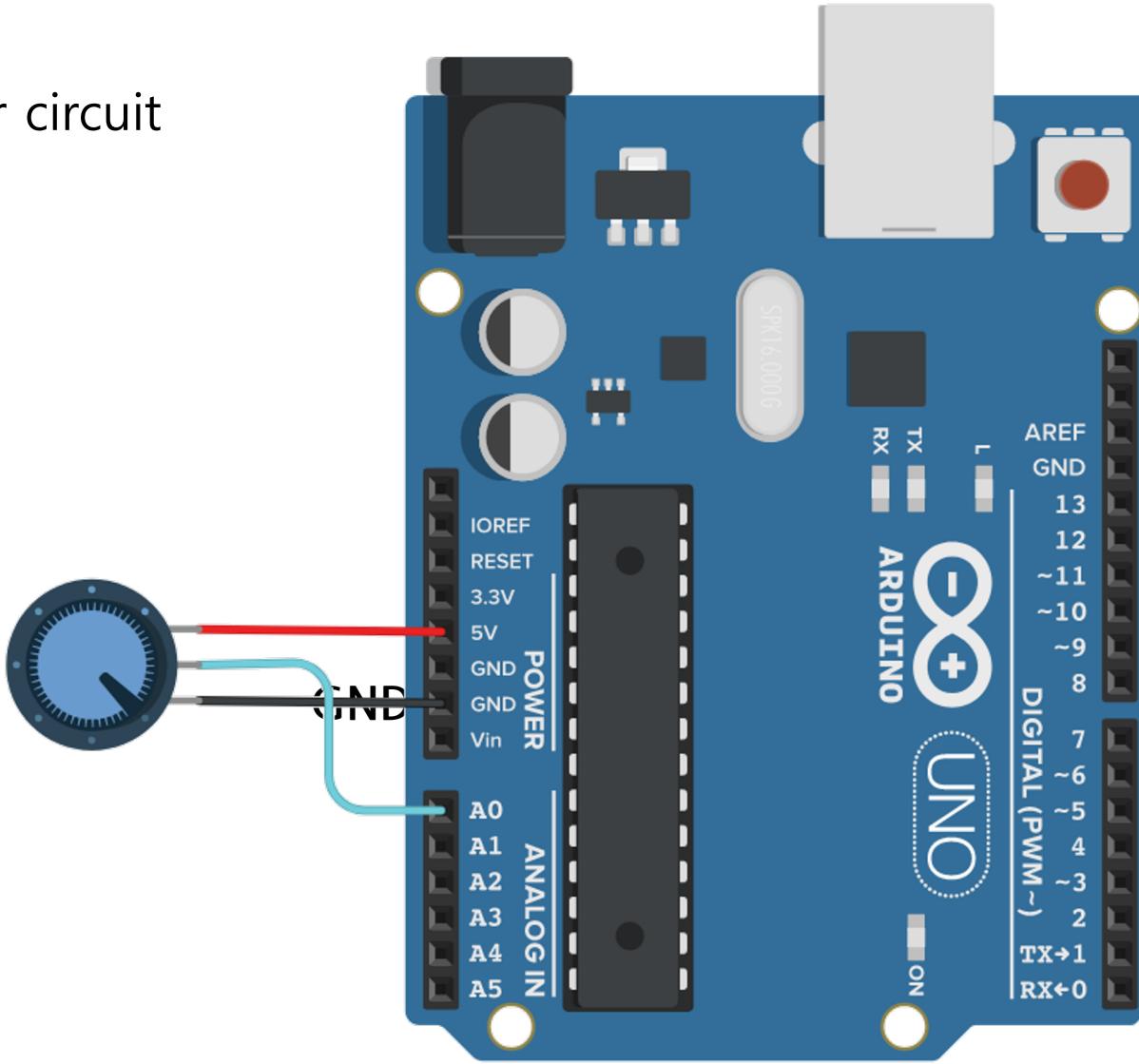
Analog number vs voltage



$$\text{Volt} = (\text{analog input}/1023) \times 5$$

3-1) Analog INPUT 1 : potentiometer

Potentiometer circuit



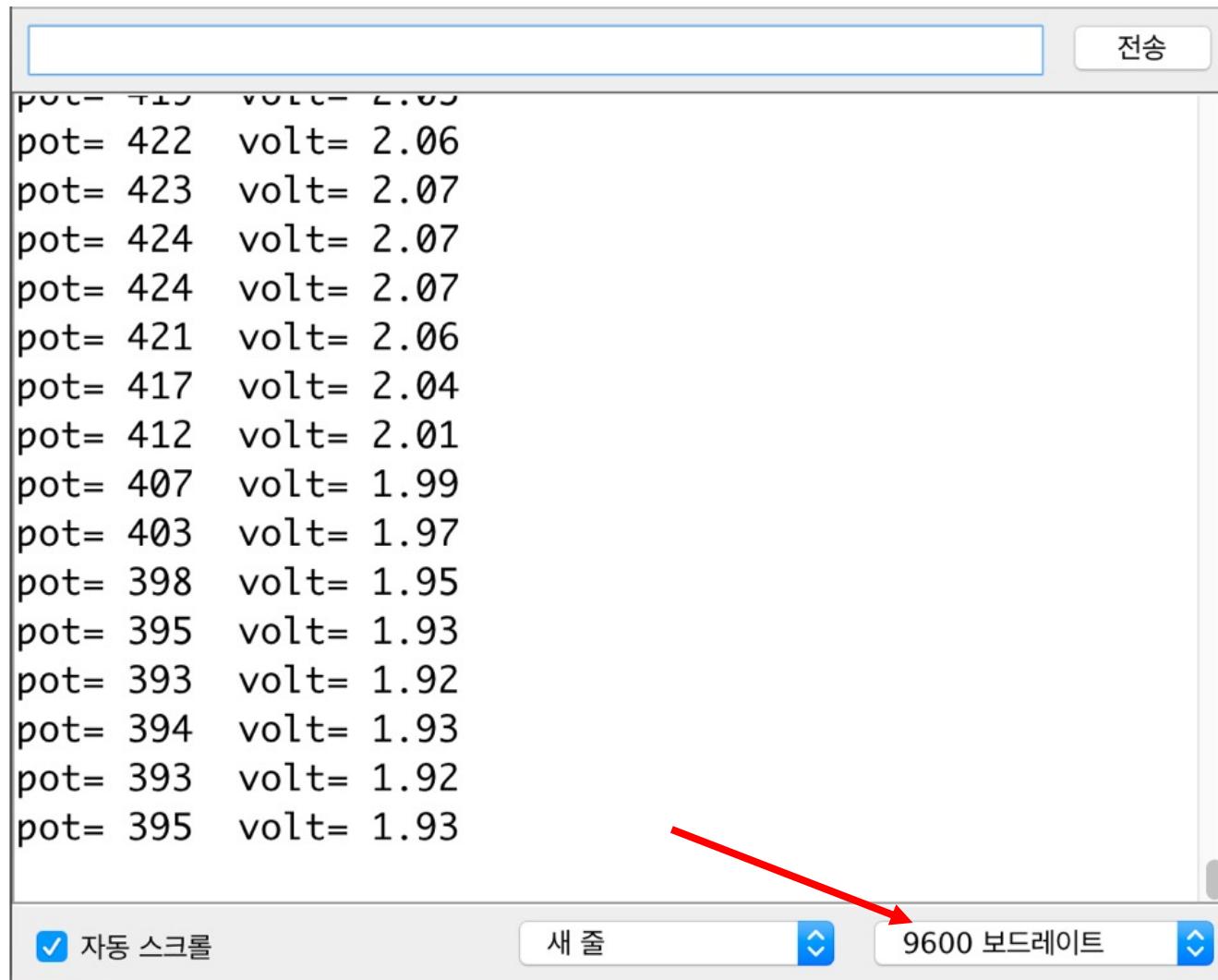
3-1) Analog INPUT 1 : potentiometer

Potentiometer Sketch



```
Analog_Read_A0
1 // analog read from A0
2
3 void setup() {
4     Serial.begin(9600) ;
5 }
6
7 void loop() {
8     int pot= analogRead(A0) ;
9     Serial.print("pot= ") ;
10    Serial.print(pot) ;
11
12    float volt=(pot/1023.0)*5.0 ;
13    Serial.print("  volt= ") ;
14    Serial.println(volt) ;
15    delay(50) ;
16 }
```

3-1) Analog INPUT 1 : potentiometer



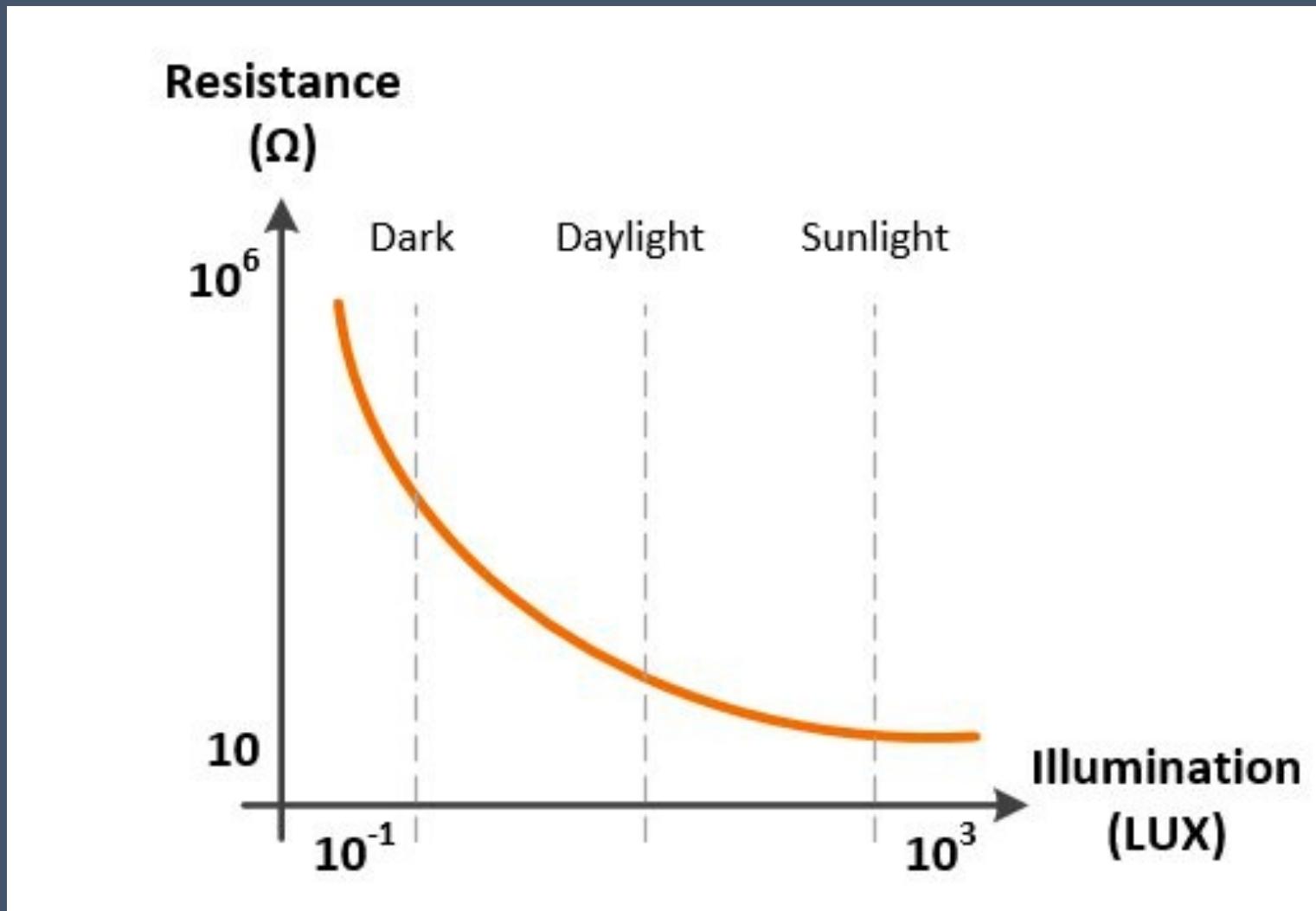
```
pot= 422  volt= 2.06
pot= 423  volt= 2.07
pot= 424  volt= 2.07
pot= 424  volt= 2.07
pot= 421  volt= 2.06
pot= 417  volt= 2.04
pot= 412  volt= 2.01
pot= 407  volt= 1.99
pot= 403  volt= 1.97
pot= 398  volt= 1.95
pot= 395  volt= 1.93
pot= 393  volt= 1.92
pot= 394  volt= 1.93
pot= 393  volt= 1.92
pot= 395  volt= 1.93
```

자동 스크롤 새 줄 9600 보드레이트

2-2) Analog INPUT 2 : Light sensor



3-2) Analog INPUT 2 : Light sensor



Illumination vs resistance of LDR

3-2) Analog INPUT 2 : Light sensor

빛의 세기에 따라 저항값이 변하는 센서

준비물 :

아두이노 보드 1개

CDS sensor 1개

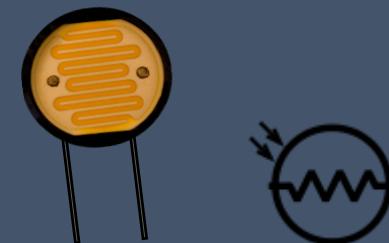
LED 1개

220 Ω, 2K Ω 저항 각 1개

포텐시오미터 1개

브래드 보드 1개

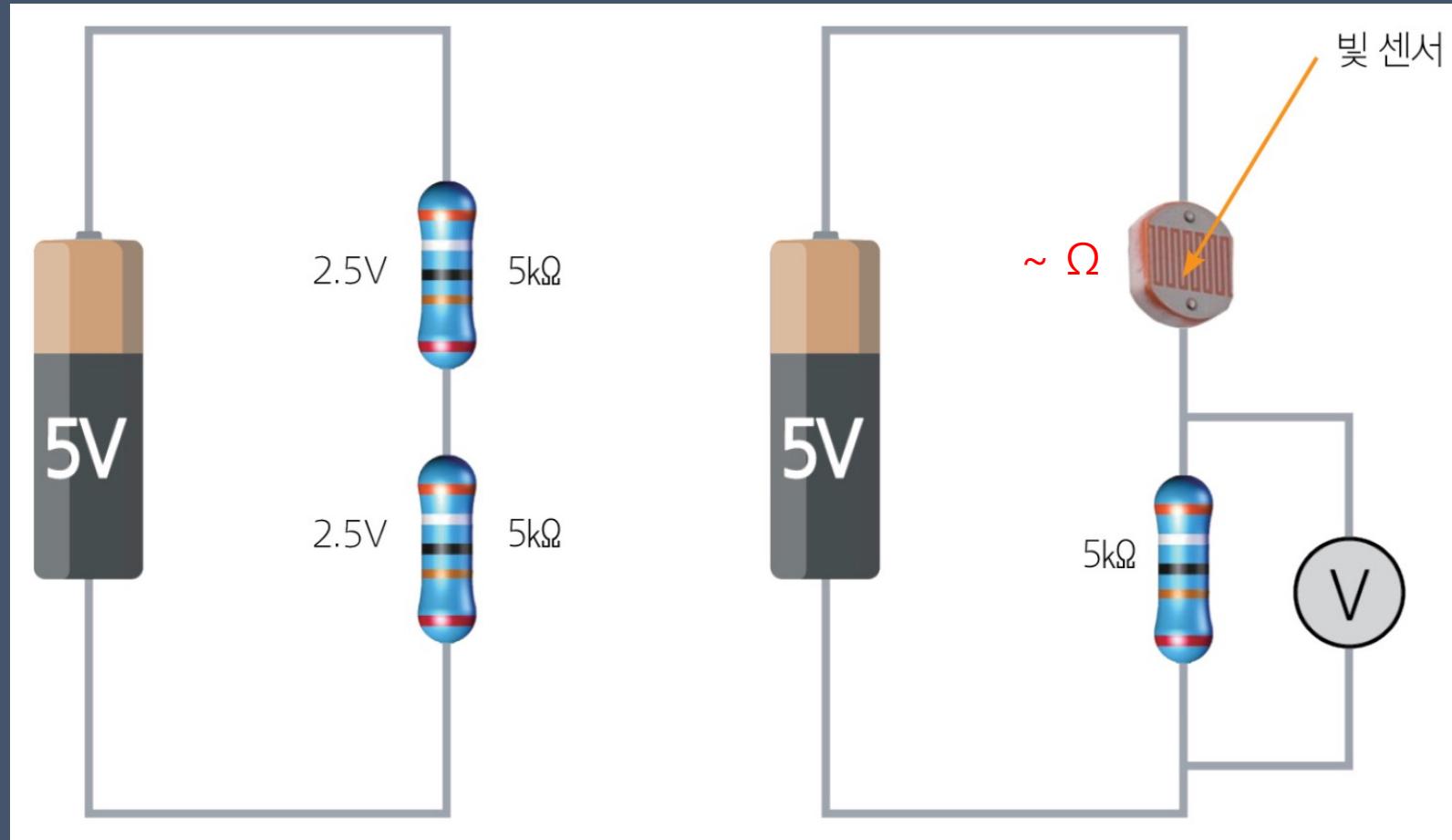
점퍼선



CDS 빛 센서 심볼

3-2) Analog INPUT 2 : Light sensor

Voltage divider



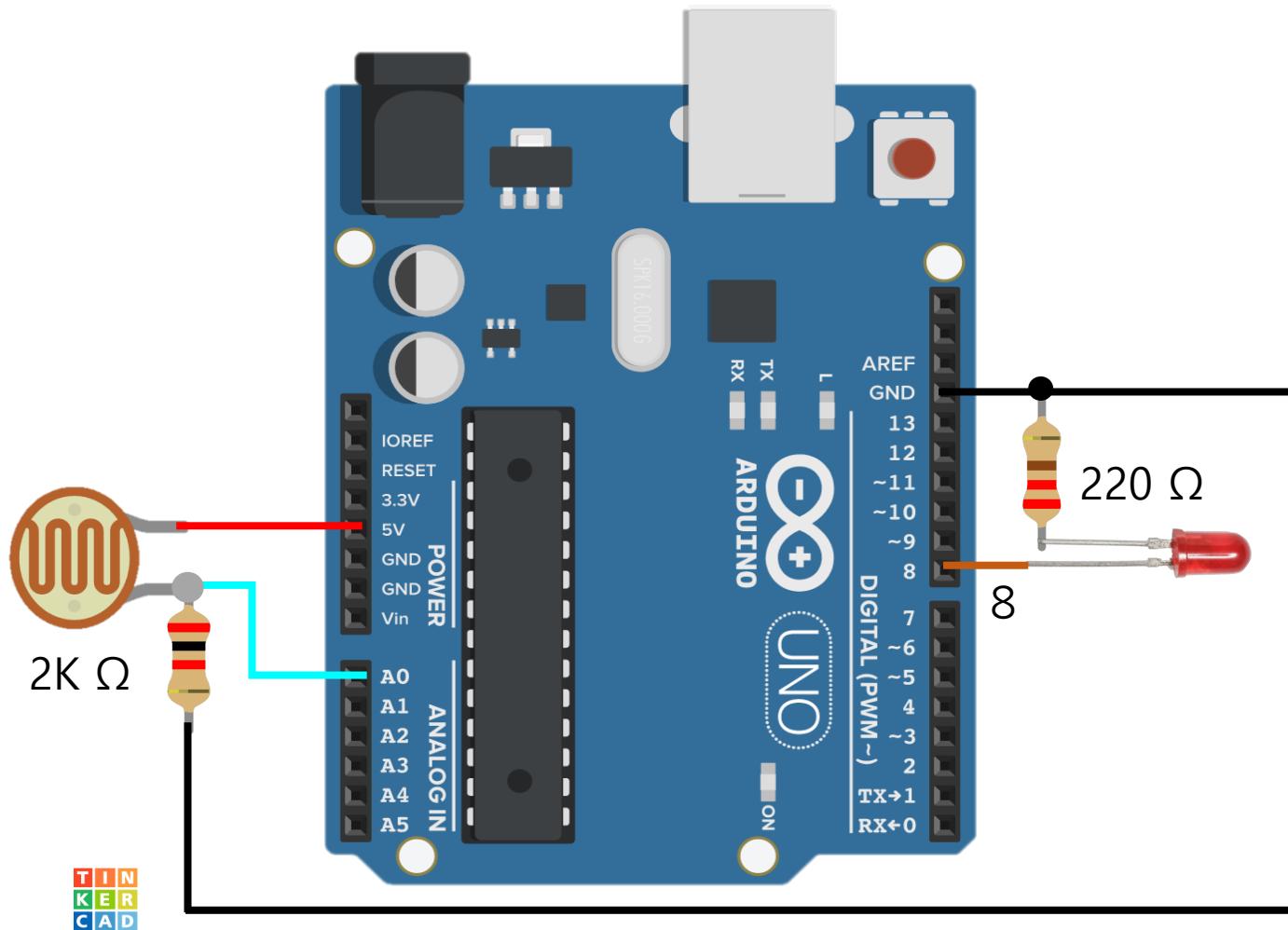
3-2) Analog INPUT 2 : Light sensor

Task 2 :

빛 센서 시그널을 아날로그 A0 에서 받고, 아날로그 값과 환산된 Volt 값을 Serial monitor 에 프린트한다,
그리고 2.5 V 이하 일때 디지털 9 번핀에 연결된 LED 를 켜는 회로와 스케치를 만든다.

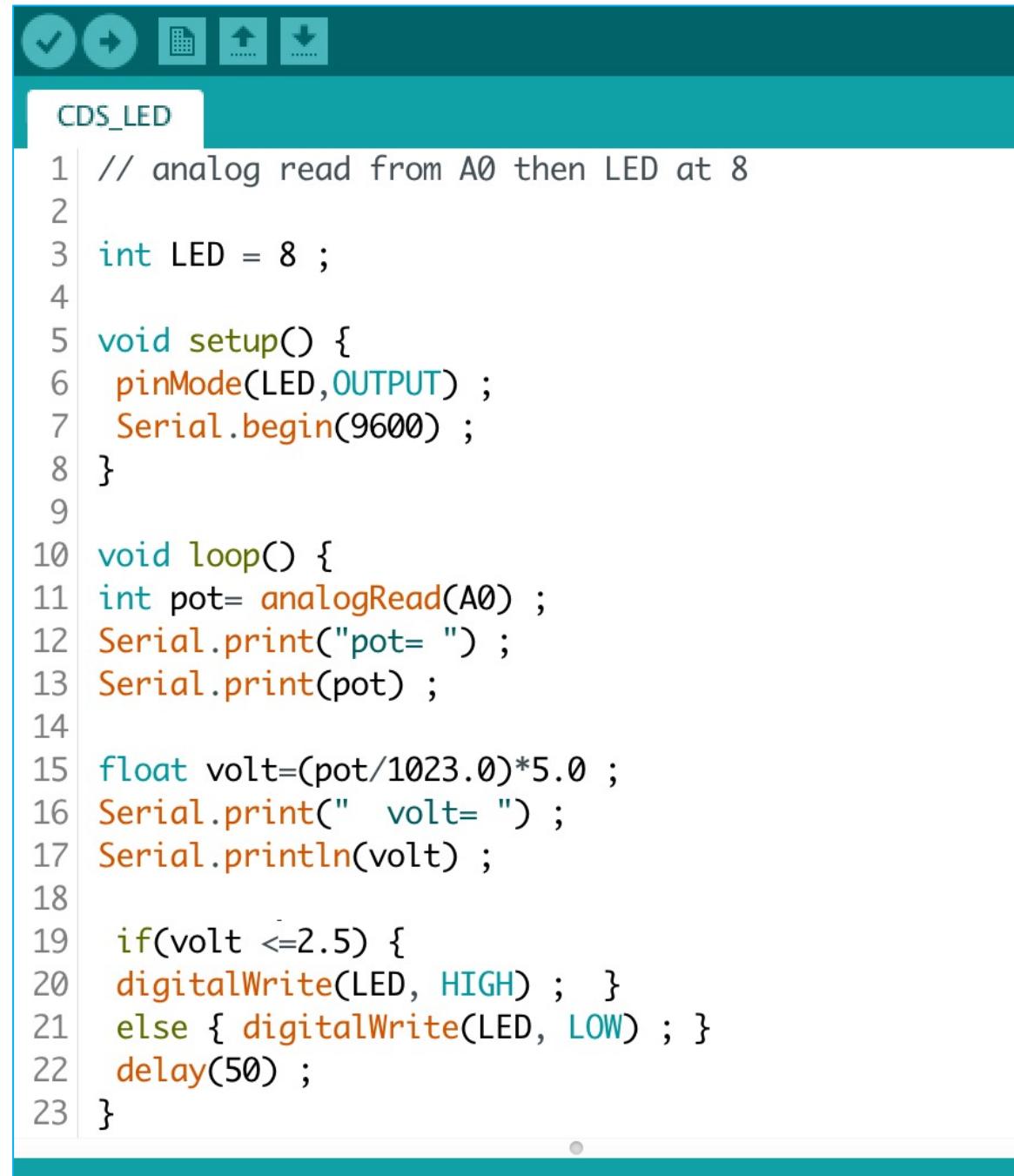
3-2) Analog INPUT 2 : Light sensor

LDR circuit



3-2) Analog INPUT 2 : Light sensor

LDR Sketch



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Shows the file name "CDS_LED".
- Tool Buttons:** Includes a checkmark, a right arrow, a file icon, and three up/down arrows.
- Code Area:** Displays the following C++ code for an Arduino sketch:

```
1 // analog read from A0 then LED at 8
2
3 int LED = 8 ;
4
5 void setup() {
6     pinMode(LED,OUTPUT) ;
7     Serial.begin(9600) ;
8 }
9
10 void loop() {
11     int pot= analogRead(A0) ;
12     Serial.print("pot= ") ;
13     Serial.print(pot) ;
14
15     float volt=(pot/1023.0)*5.0 ;
16     Serial.print("  volt= ") ;
17     Serial.println(volt) ;
18
19     if(volt <=2.5) {
20         digitalWrite(LED, HIGH) ; }
21     else { digitalWrite(LED, LOW) ; }
22     delay(50) ;
23 }
```

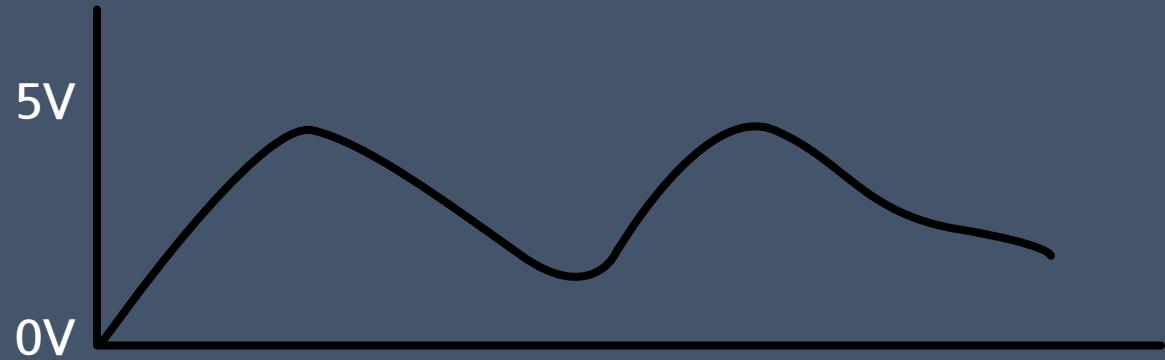

Ch4. Analog OUTPUT

- 1) Principle of PWM
- 2) PWM 1 LED



1) Analog Output

How to make analog volt from Arduino



Convert digital volt to analog volt



2) Principle of PWM

PWM(Pulse Width Modulation)



펄스

준비물 :

아두이노 우노 1개

LED 1개, 220 Ω 저항 1개

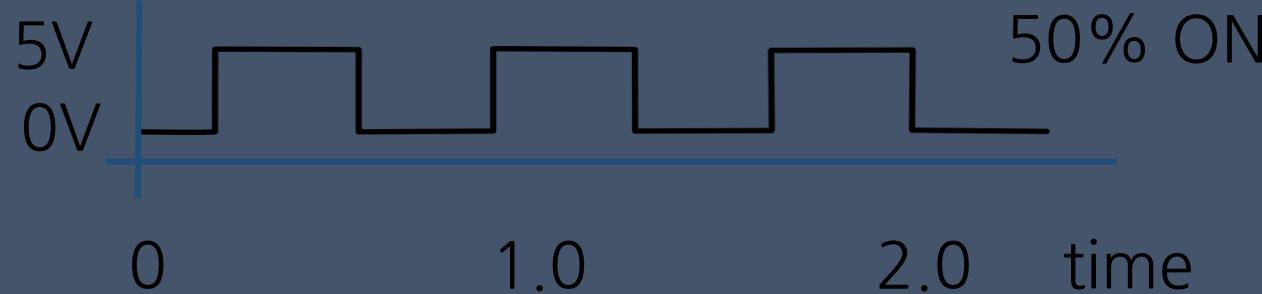
점퍼 케이블

브래드보드

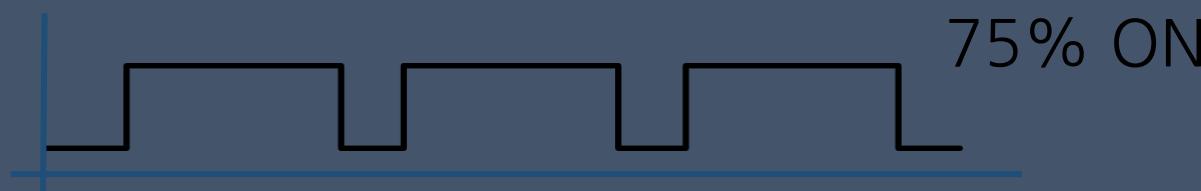
아두이노 디지털 핀은 8 비트

8 비트를 10 진수로 표시하면 ?

2) Principle of PWM



2.50V



3.75V



1.25V

2) Principle of PWM

`analogWrite(9, 255)` 를 하면 9 번 핀에서 5V 가 출력된다.

255 의 50% 는 127.5, 정수로 표시하면 128 이다.

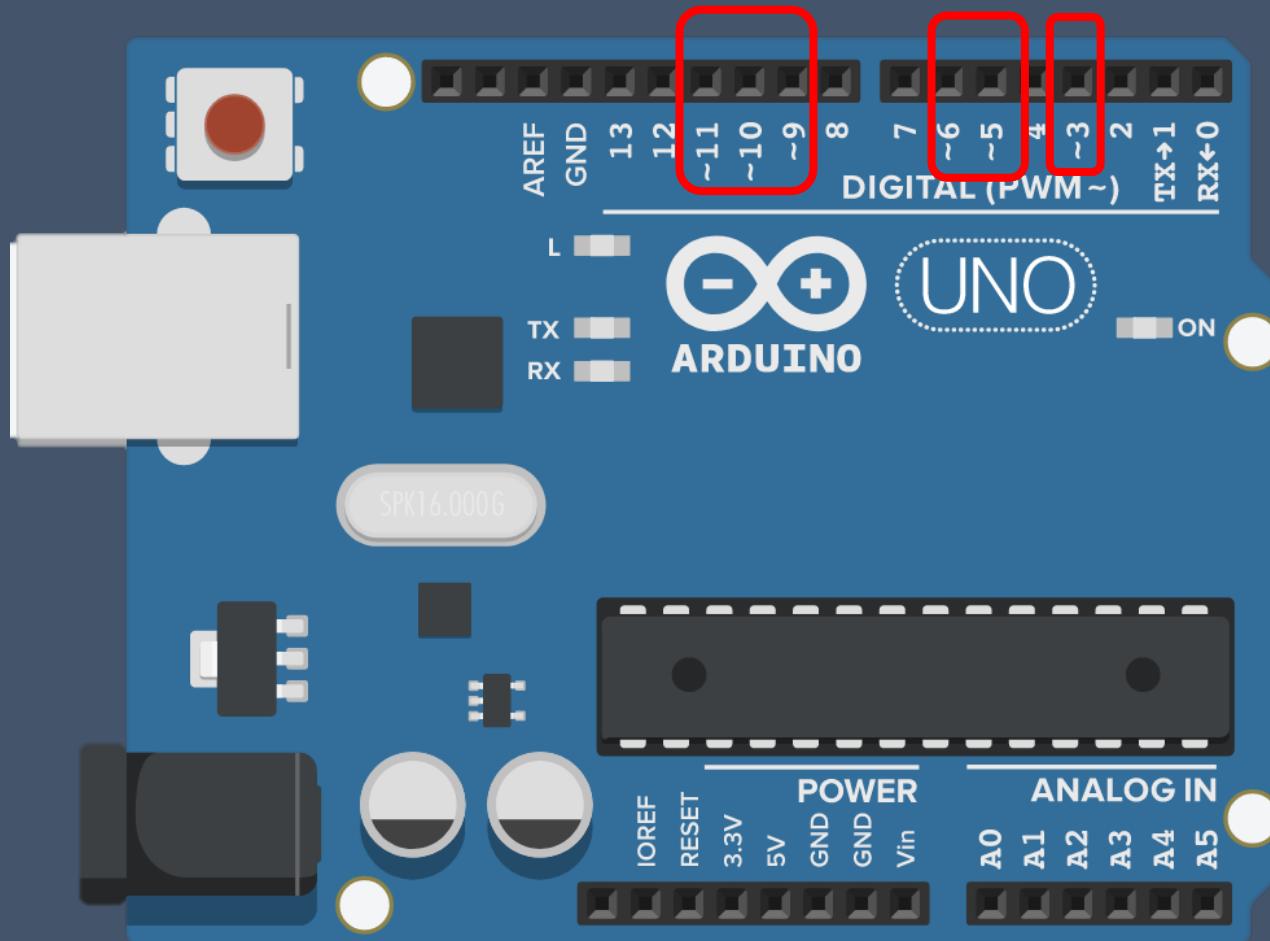
`analogWrite(9, 128)` 를 하면 9 번 핀에서 2.5V 가 출력된다.

`analogWrite(9, 0)` 를 하면 9 번 핀에서 0V 가 출력된다.

2) Principle of PWM

아날로그 출력을 할 수 있는 핀 앞에는 ~ 표시가 있다.

우노는 3, 5, 6, 9, 10, 11 핀이다.



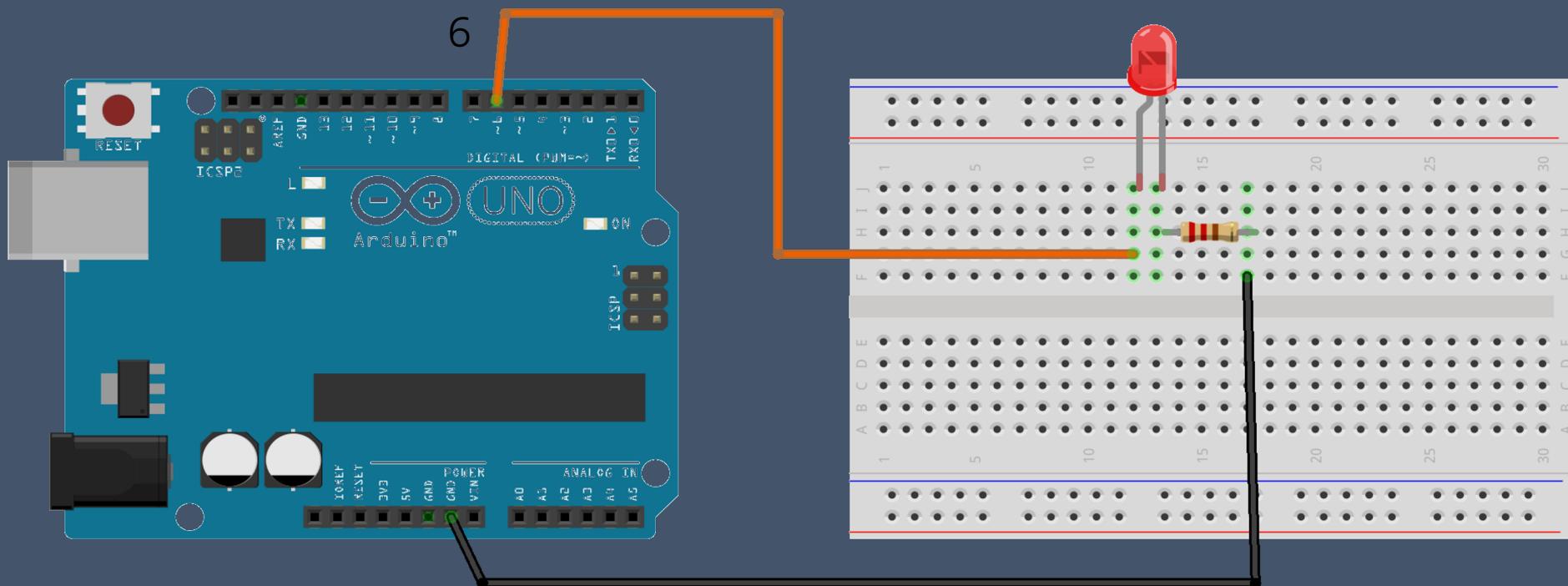
2) PWM 1 LED

프로젝트 :

PWM 을 사용하여 디지털 6번 핀에 연결한 LED 의 밝기를 0 부터
최대까지 점진적으로 밝게하고 어둡게 하기를 반복한다.

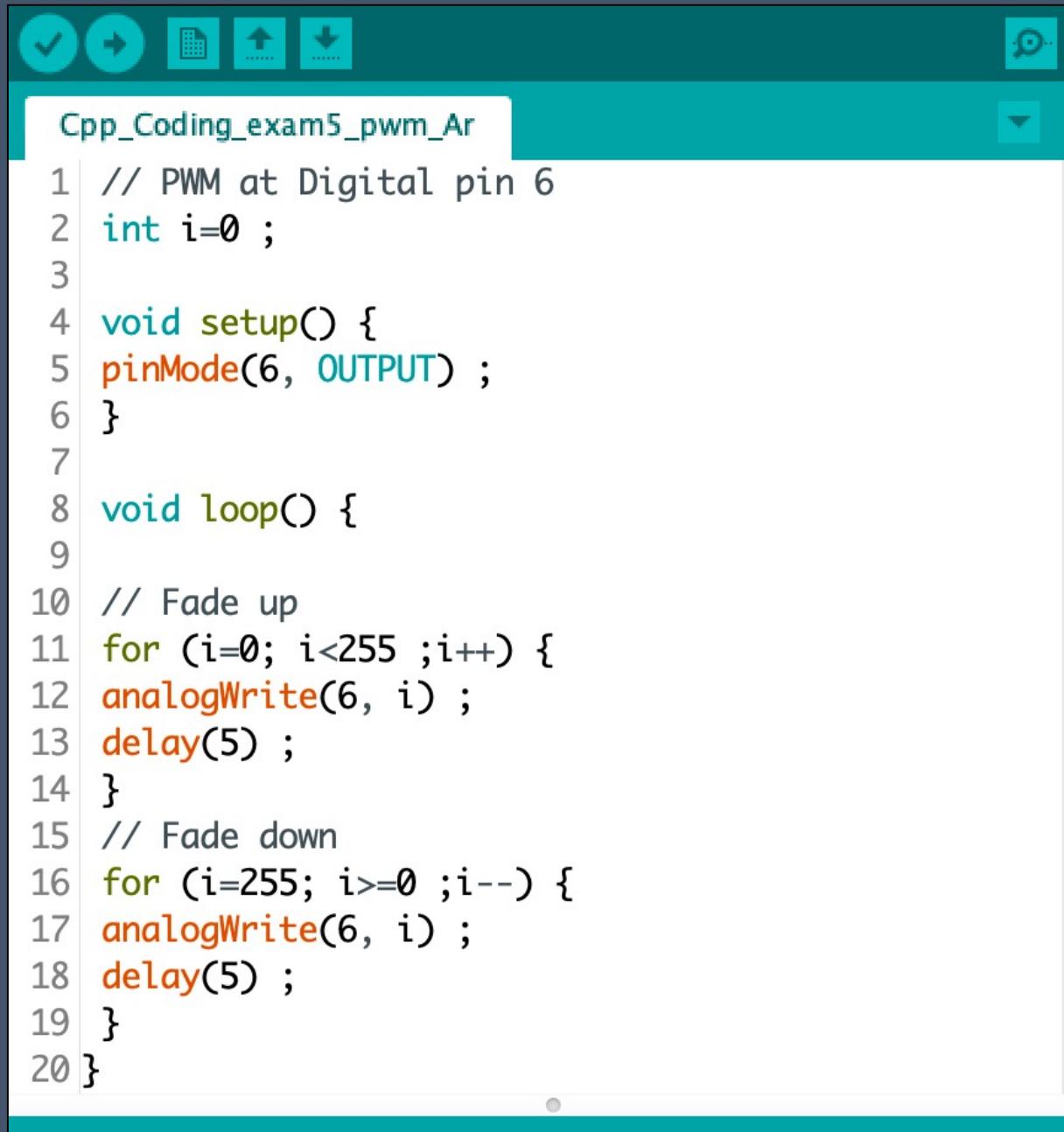
2) PWM 1 LED

PWM at D6



fritzing

2) PWM 1 LED



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Cpp_Coding_exam5_pwm_Ar
- Toolbar:** Includes icons for file operations (checkmark, plus, save, upload, download), a magnifying glass for search, and a gear for settings.
- Code Area:** Displays the following C++ code for an Arduino sketch:

```
1 // PWM at Digital pin 6
2 int i=0 ;
3
4 void setup() {
5   pinMode(6, OUTPUT) ;
6 }
7
8 void loop() {
9
10 // Fade up
11 for (i=0; i<255 ;i++) {
12   analogWrite(6, i) ;
13   delay(5) ;
14 }
15 // Fade down
16 for (i=255; i>=0 ;i--) {
17   analogWrite(6, i) ;
18   delay(5) ;
19 }
20 }
```

i++ 는 $i=i+1$ 의 줄임

i-- 는 $i=i-1$ 의 줄임

Ch4. Analog OUTPUT

- 1) Principle of PWM
- 2) PWM 1 LED
- 3) PWM 3 LED

Ch5. Library 활용

- 1) 초음파 센서
- 2) 온도/습도 센서

3) PWM 3 LED

프로젝트 :

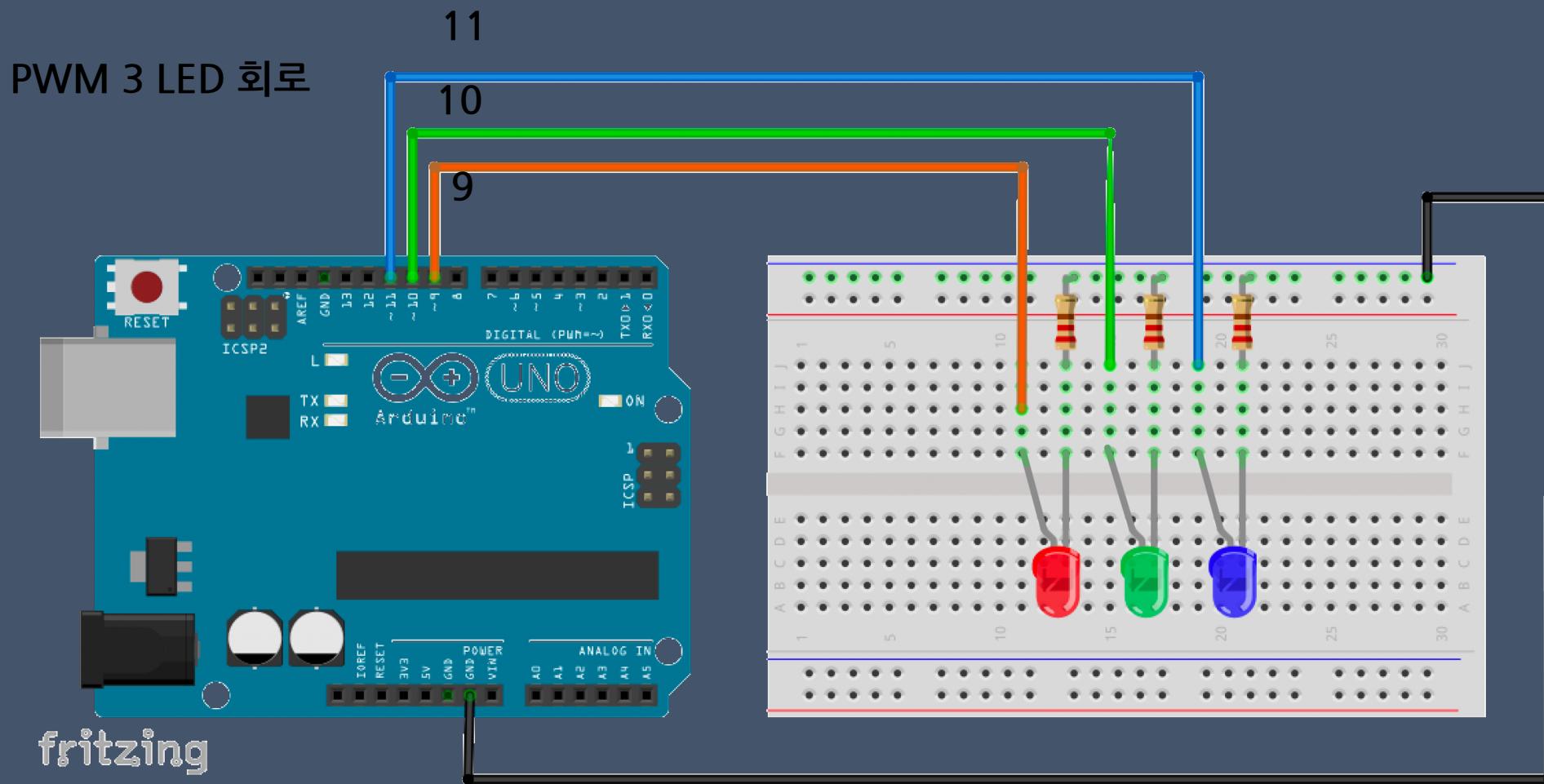
3 개의 LED 를 순차적으로 밝고 어둡게 한다.

코딩에서

function 개념 소개,

New Tap 소개

3) PWM 3 LED



3) PWM 3 LED

PWM 3 LED 스케치
with function

The screenshot shows the Arduino IDE interface with a sketch titled "fuction_Exam_ino". The code uses functions to control three LEDs via PWM on pins 9, 10, and 11.

```
// function
void setup() {
pinMode(9, OUTPUT) ; // LED1
pinMode(10, OUTPUT) ; // LED2
pinMode(11, OUTPUT) ; // LED3
}

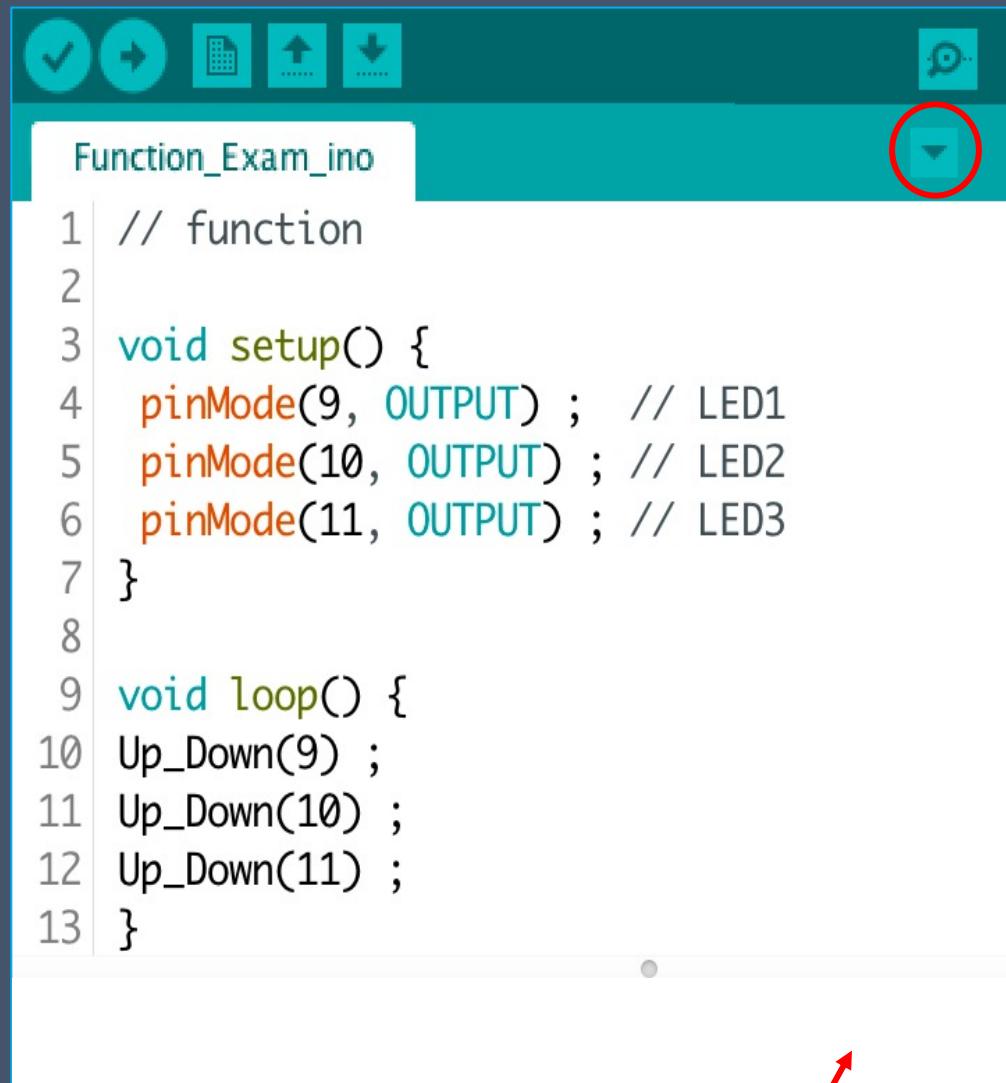
void loop() {
Up_Down(9) ;
Up_Down(10) ;
Up_Down(11) ;
}

//----- 만든 함수 -----
void Up_Down(int pin) {
for (int i=0; i<=255; i++) {
analogWrite(pin,i);
delay(5);
}
for (int i=255; i>=0; i--) {
analogWrite(pin,i);
delay(5) ;
}
}
```

The code includes a `Up_Down` function that alternates between increasing and decreasing the analog output on a specified pin from 0 to 255, with a 5ms delay between steps. The `loop` function calls this function for three pins (9, 10, and 11).

3) PWM 3 LED

New Tap icon



```
Function_Exam.ino
1 // function
2
3 void setup() {
4   pinMode(9, OUTPUT) ; // LED1
5   pinMode(10, OUTPUT) ; // LED2
6   pinMode(11, OUTPUT) ; // LED3
7 }
8
9 void loop() {
10 Up_Down(9) ;
11 Up_Down(10) ;
12 Up_Down(11) ;
13 }
```



```
Function_Exam.ino Up_Down
1 // ----- 만든 함수 -----
2 void Up_Down(int pin) {
3   for (int i=0; i<=255; i++) {
4     analogWrite(pin, i) ;
5     delay(5) ;
6   }
7   for (int i=255; i>0; i--) {
8     analogWrite(pin, i) ;
9     delay(5) ;
10 }
11 }
```

컴파일 완료.