

MakeUp
아두이노

#MakeUp아두이노 #01

아두이노 기초

아두이노 기초

아두이노

‘아두이노(Arduino)’는 마이크로컨트롤러(micro-controller)를 장착한 보드이다.
사용자가 작성한 프로그램을 아두이노의 마이크로컨트롤러에 심어 외부에 연결한 I/O 장치를 구동한다.

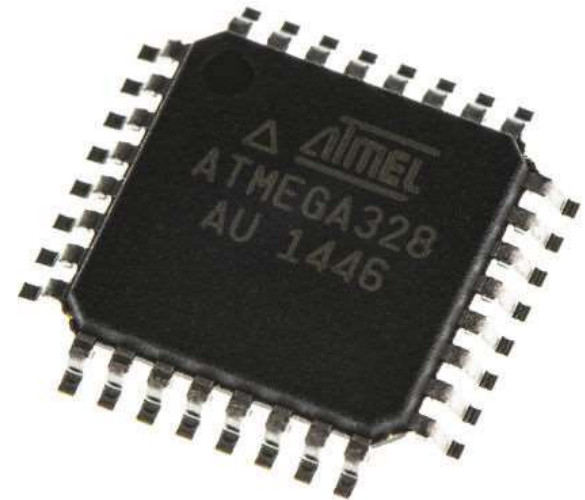
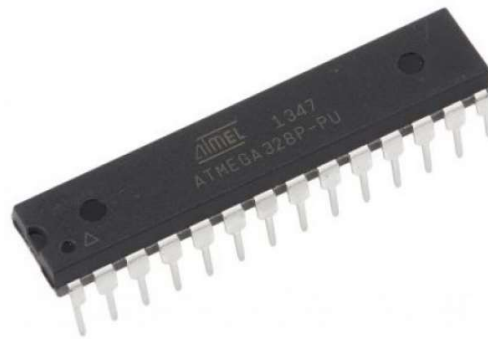


아두이노 기초

마이크로컨트롤러?

마이크로컨트롤러(microcontroller) 또는 MCU(microcontroller unit)는 마이크로프로세서와 입출력 모듈을 하나의 칩으로 만들어 정해진 기능을 수행하는 컴퓨터를 말한다.
CPU 코어, 메모리 그리고 프로그램 가능한 입/출력을 가지고 있다.

Atmega328			
(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)



아두이노 기초

아두이노 우노(UNO)

아두이노는 성능과 크기, 모양 등에 따라 다양한 기종이 존재하는데 표준 보드로 사용하는 것이 아두이노 우노이다. 여기에는 ATMEGA328P가 장착되어 있다.

[Spec]

- 16MHz의 클럭
- 디지털 I/O포트 14개
- 아날로그 입력포트 6개
- 프로그램 저장용 flash 메모리 32kB
- 데이터 저장용 EEPROM 1kB
- SRAM 2kB

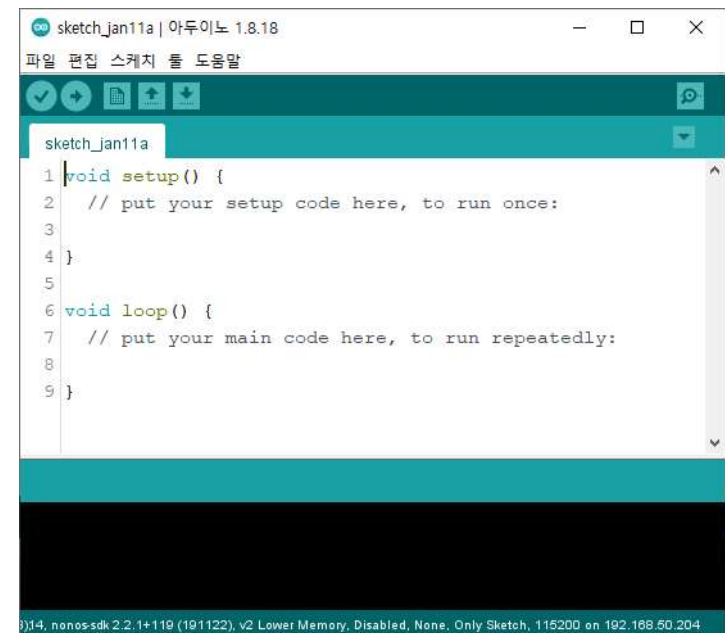


아두이노 기초

아두이노 IDE(통합 개발환경)

아두이노 보드에 프로그래밍을 하기 위해서는 텍스트 에디터 및 아두이노용 코드로 변환해 주는 컴파일러가 있어야 한다. 이 역할을 하는 것이 아두이노용 IDE (Integrated Development Environment; 통합개발환경)이다.

다운로드: <https://www.arduino.cc>

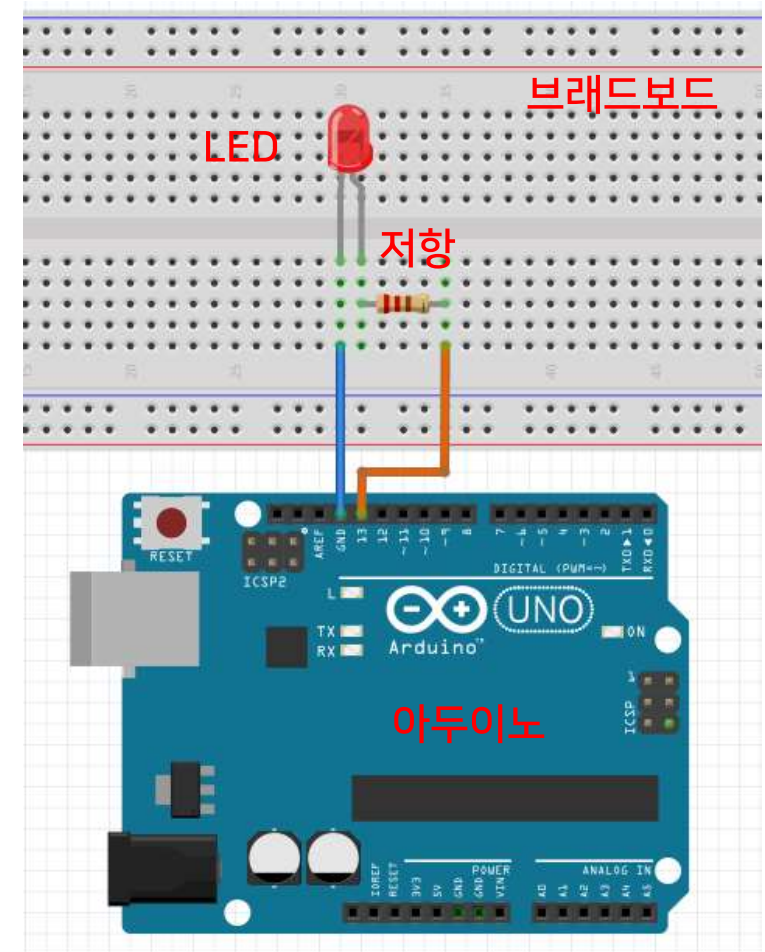
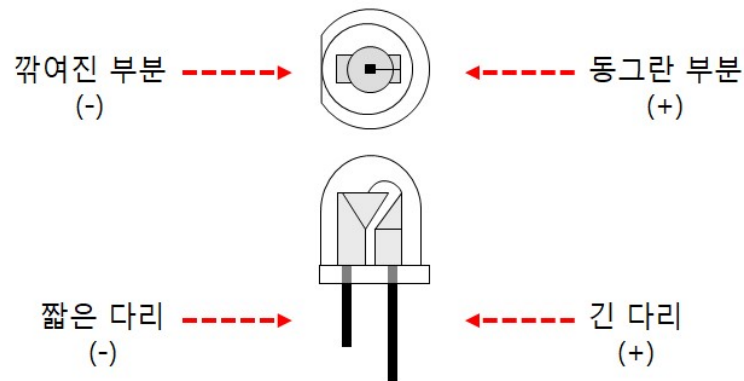


아두이노 기초

출력 회로 구성하기

아두이노 우노의 디지털 13번 핀에 LED를 연결

- LED는 바로 연결하지 않고 저항을 통해 연결한다.
- LED는 +- 극성이 있다.
- (D13핀)-(저항)-(LED+)/(LED-)-(GND핀) 과 같이 연결

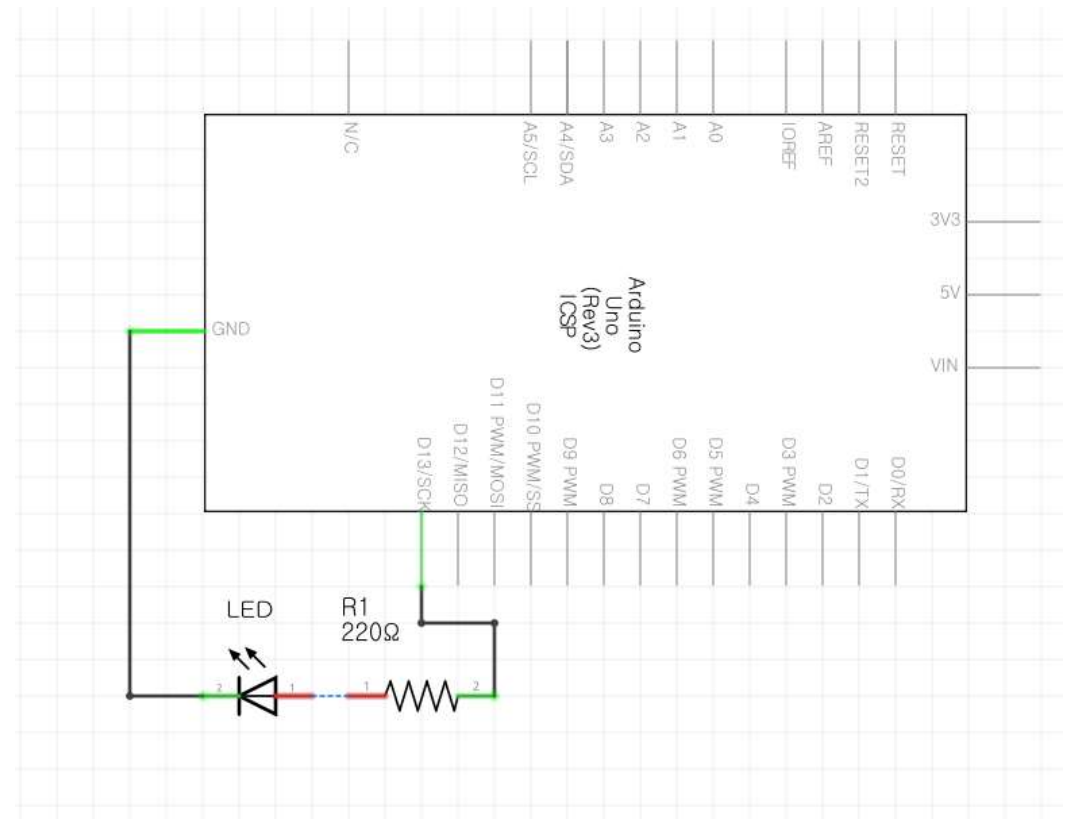


아두이노 기초

출력 회로 살펴보기

아두이노 LED 출력 회로도

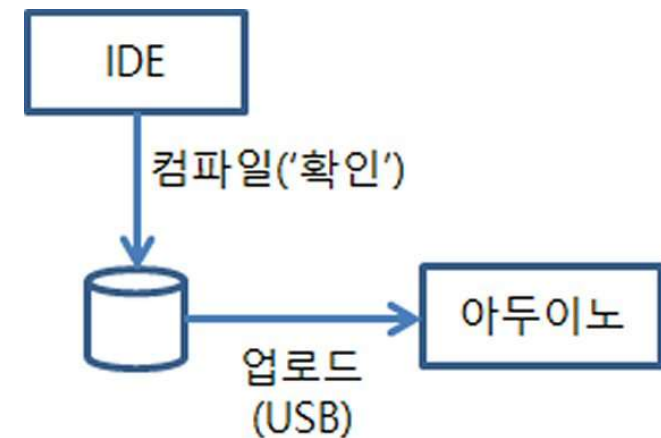
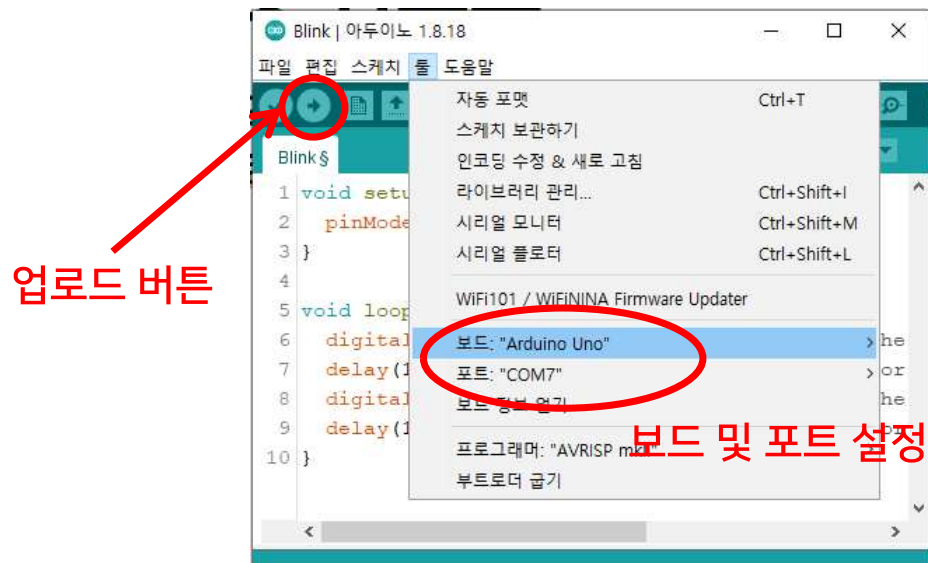
- GND : 전류의 (-)극 부분 (접지 = 0V)
- 5V : 아두이노는 USB포트로 부터 DC 5V 전원을 공급 받는다. (+)극
- D13: 아두이노 프로그램을 통해 디지털 입출력이 제어된다. (0V / 5V)



아두이노 기초

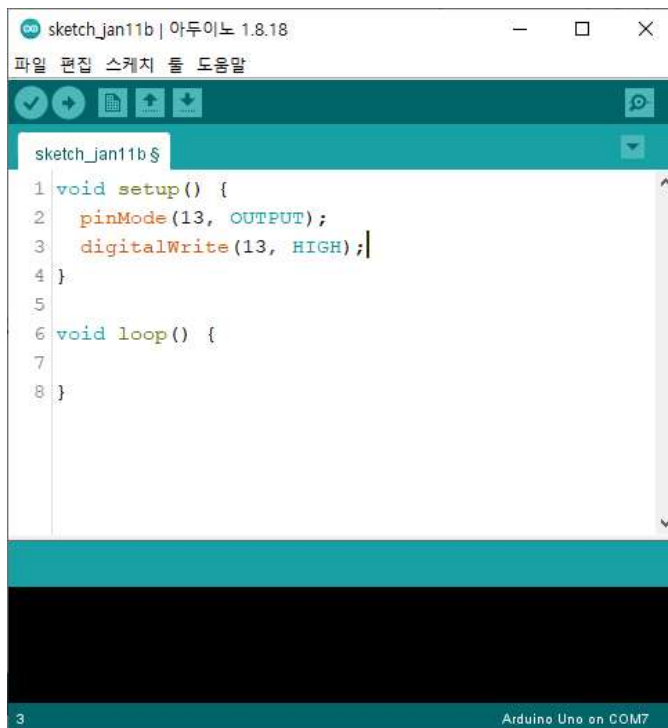
프로그래밍 및 업로드

아두이노와 PC를 USB로 연결하고 보드 및 포트를 설정한다. (아두이노 종류에 맞게 설정)
아두이노 IDE에서 코드 작성 후 “업로드”버튼을 누르면 작성한 코드가 기계어로 변환되어 USB 시리얼 인터페이스를 통해 아두이노의 마이크로컨트롤러 칩에 업로드 된다.



아두이노 기초

프로그래밍



The screenshot shows the Arduino IDE interface with a sketch named 'sketch_jan11b'. The code is as follows:

```
1 void setup() {  
2   pinMode(13, OUTPUT);  
3   digitalWrite(13, HIGH);  
4 }  
5  
6 void loop() {  
7  
8 }
```

The status bar at the bottom indicates '3' lines of code and 'Arduino Uno on COM7'.

예제1



The screenshot shows the Arduino IDE interface with a sketch named 'Blink'. The code is as follows:

```
1 void setup() {  
2   pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(13, HIGH);  
7   delay(1000);  
8   digitalWrite(13, LOW);  
9   delay(1000);  
10 }
```

The status bar at the bottom indicates '9' lines of code and 'Arduino Uno on COM7'.

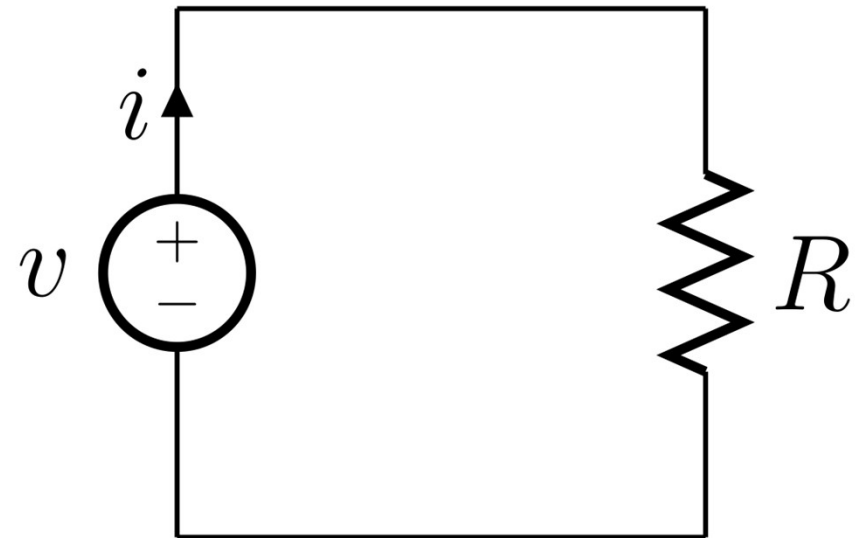
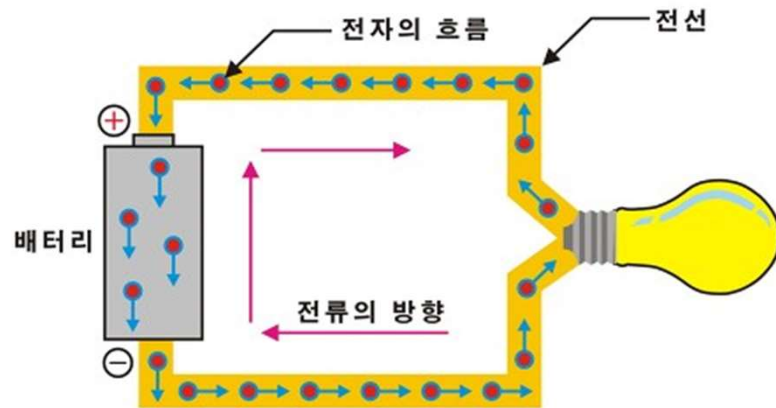
예제2

아두이노 기초

전류의 흐름

전류는 +극에서 -극으로 흐른다.

전류가 흐르기 위해서는 극성 간 전압(전자를 밀어주는 힘)이 필요.



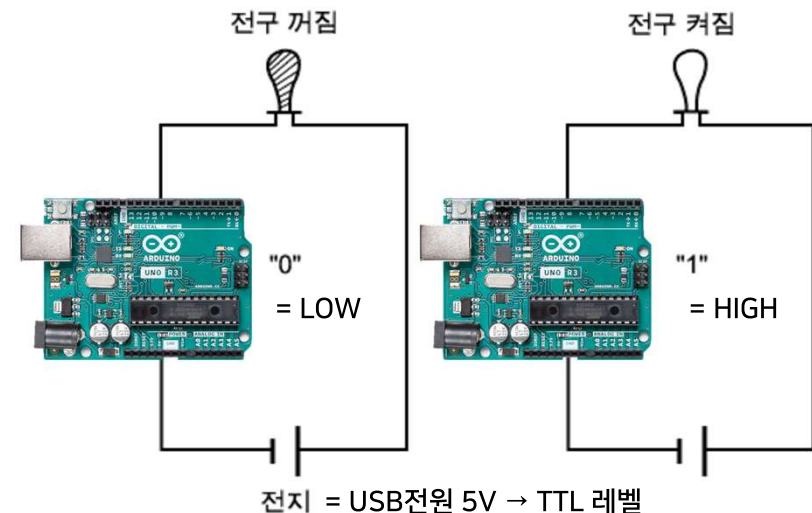
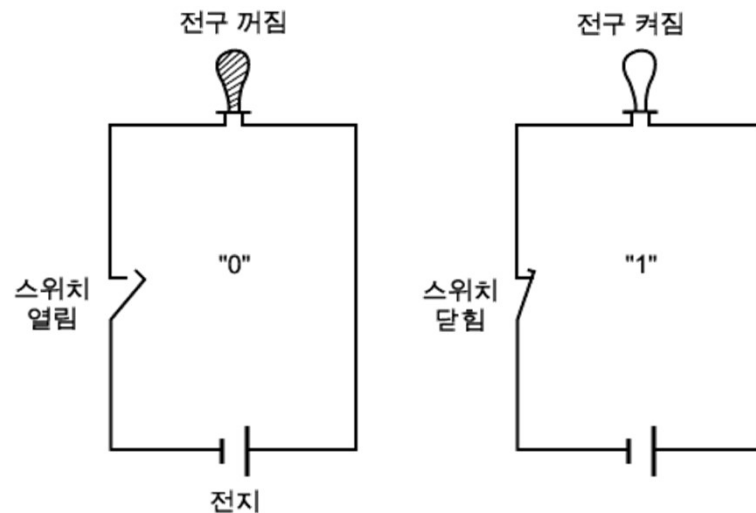
아두이노 기초

디지털에서의 0과 1

디지털은 **0** 또는 **1**의 두 가지 상태로만 표현된다.

회로적으로 표현하면 **0=전류가 흐르지 않는 상태(0V)** / **1=전류가 흐름(5V)** 과 같이 표현할 수 있다.

아두이노 코드에서는 **LOW = 0** / **HIGH = 1** 와 같이 표현한다.

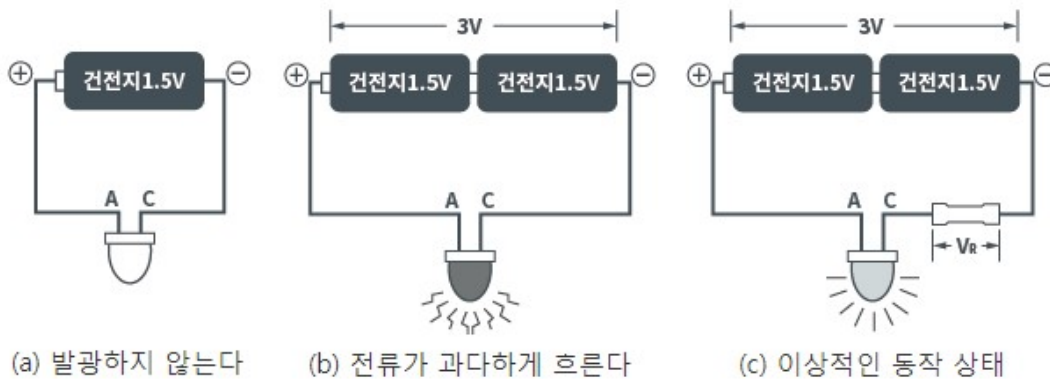


아두이노 기초

저항

LED(발광 다이오드)가 빛나기 위해서는 LED양단에 2V이상 10~20mA정도의 전류가 흐를 때 적정하게 빛난다. 전압이 부족하면 빛이 나지 않고, 전압이 과할 경우 과다한 전류로 인해 열이 발생하여 수명이 짧아지고 소손될 수 있다.

이러한 LED의 손상을 방지하기 위해 저항을 사용하여 전류를 적절하게 제한한다.



저항의 양 끝에 가해지는 전압 V_R 은 LED가 정상적으로 동작하고 있을 때 약 2V가 되므로, 건전지의 전압 3V에서 LED의 양끝 전압 2V를 빼면 :

$$V_R = 3(V) - 2(V) = 1(V)$$

LED에 흐르는 전류 I_{LED} 을 15mA($1mA=1/1000A$)라고 하면, 옴의 법칙으로 인해 저항치 R 은 :

$$R = \frac{V_R}{I_{LED}} = \frac{1(V)}{0.015(A)} = 67(\Omega)$$

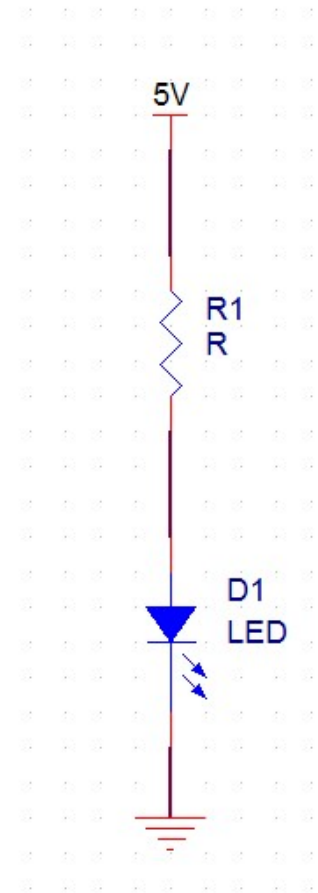
아두이노 기초

저항

LED를 5V에서 사용하기 위해서는 몇 옴의 저항이 적절할까?

$$V_{R1} = 5V - 2V = 3V$$

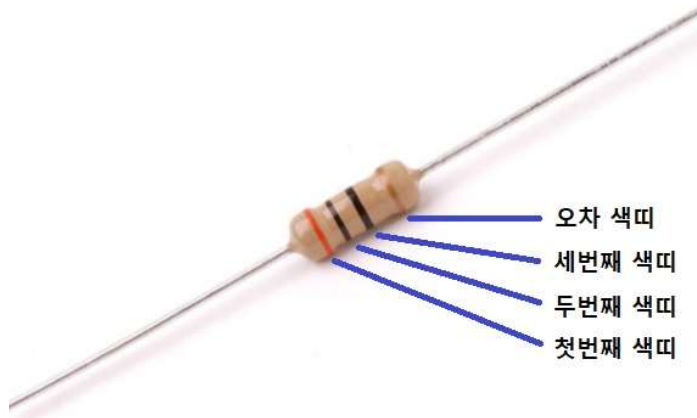
$$R1 = 3V / 0.015A = \mathbf{200\Omega}$$



아두이노 기초

저항

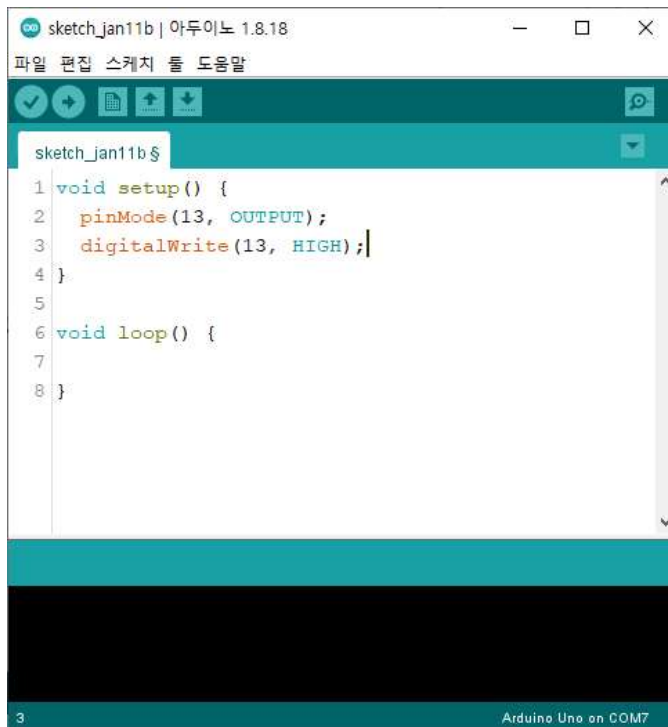
저항 부품은 색상으로 그 값을 표현한다.
오른쪽 표와 같이 읽는다.



500Ω ±5%			
Black	0	X 1	±1%
Brown	1	X 10	±2%
Red	2	X 10 ²	±5%
Orange	3	X 10 ³	±10%
Yellow	4	X 10 ⁴	
Green	5	X 10 ⁵	
Blue	6	X 10 ⁶	
Purple	7	÷10	
Grey	8	÷100	
White	9		
Gold			
Silver			

아두이노 기초

코드 살펴보기

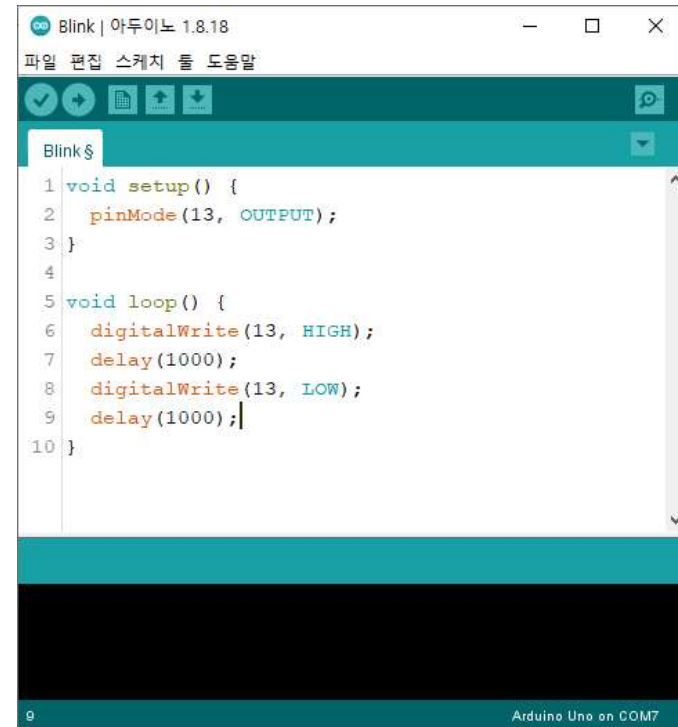


The screenshot shows the Arduino IDE interface with a sketch named 'sketch_jan11b'. The code is as follows:

```
1 void setup() {  
2   pinMode(13, OUTPUT);  
3   digitalWrite(13, HIGH);  
4 }  
5  
6 void loop() {  
7  
8 }
```

The status bar at the bottom indicates '3' and 'Arduino Uno on COM7'.

예제1



The screenshot shows the Arduino IDE interface with a sketch named 'Blink'. The code is as follows:

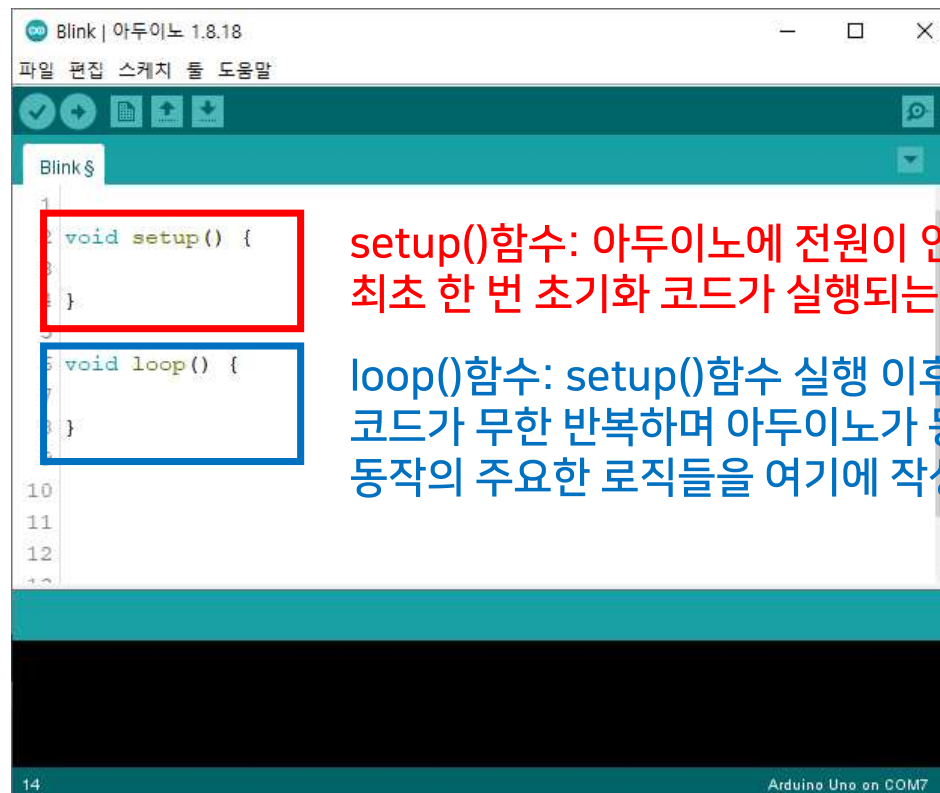
```
1 void setup() {  
2   pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(13, HIGH);  
7   delay(1000);  
8   digitalWrite(13, LOW);  
9   delay(1000);  
10 }
```

The status bar at the bottom indicates '9' and 'Arduino Uno on COM7'.

예제2

아두이노 기초

코드 살펴보기 - 아두이노 스케치의 기본 구조

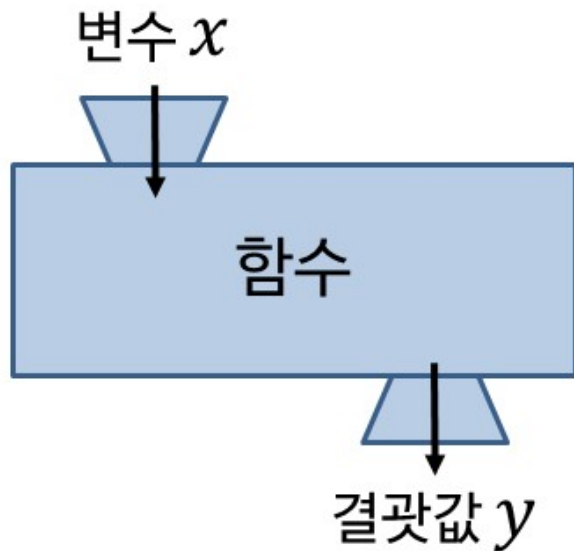


setup()함수: 아두이노에 전원이 인가되면
최초 한 번 초기화 코드가 실행되는 부분

loop()함수: setup()함수 실행 이후 loop()함수에 작성한
코드가 무한 반복하며 아두이노가 동작한다. (무한루프)
동작의 주요한 로직들을 여기에 작성한다.

아두이노 기초

코드 살펴보기 - 함수



수학에서의 함수

```
int increase(int x) {  
    return x + 1;  
}  
  
int result = plus(1); // result = 2;  
  
int plus(int a, int b) {  
    return a + b;  
}  
  
int result = plus(1, 2); // result = 3;  
  
void led_on() {  
    digitalWrite(13, HIGH);  
}  
  
led_on(); // LED를 켜는 동작이 실행됨
```

코딩에서의 함수 → 특정한 연산이나 동작을 정의

아두이노 기초

코드 살펴보기 - 함수

반환(리턴) 자료형

함수 이름

매개변수(파라미터)

```
int increase(int x) {  
    return x + 1;  
}
```

```
int result = plus(1); // result = 2;
```

매개변수와 리턴이 없는 함수

```
void led_on() {  
    digitalWrite(13, HIGH);  
}
```

```
led_on(); // LED를 켜는 동작이 실행됨
```

아두이노 기초

코드 살펴보기 - 변수, 상수

변수: 프로그램 상에서 임의의 데이터(값)들을 일시적으로 저장하는 공간.

데이터의 수정 및 변경이 가능하다.

상수: 프로그램 상에서 변경되지 않는 데이터 (고정된 값)

```
const int pin_mode = 0; // const 키워드 : 변수처럼 이름을 붙여 상수를 선언할 수 있다.
```

```
int pin = 13; // pin이라는 int형 변수에 13 값을 넣는다. (저장한다.)
```

변수 자료형 변수 이름 대입 연산자 상수

아두이노 기초

코드 살펴보기 - 변수, 상수

아두이노 스케치에서 사용할 수 있는 기본 데이터 타입 (C++)

자료형	크기(Byte)		비고	값의 범위	
	8bit 보드	32bit 보드		8bit 보드	32bit 보드
char	1		문자형	-128~127	
byte	1		부호 없음	0~255	
int	2	4		-32,768~32,767	(-2^31) ~ (2^31 - 1)
word	2	4	부호 없음		
short	2			-32,768~32,767	
long	4			-2,147,483,648~2,147,483,647	
unsigned char	1		부호 없음	0~255	
unsigned int	2	4		0~65,535	0~(2^32 - 1)
unsigned short	2			0~65,535	
unsigned long	4			0~4,294,967,295	

아두이노 기초

코드 살펴보기 - 주석

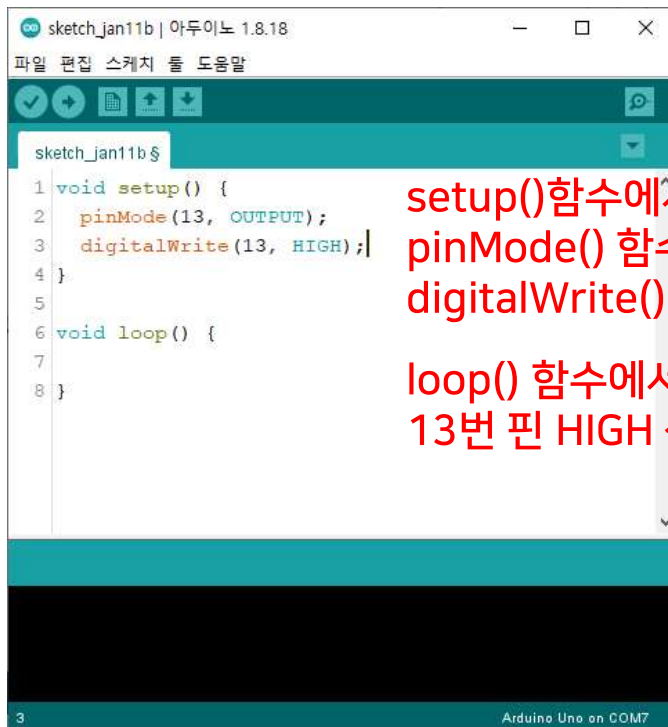
주석은 코드에 삽입되는 코멘트로, 프로그램 로직과 실행 과정에 영향을 미치지 않는다.

- 코드에 대한 간략한 설명이나 메시지를 남기는데 사용
- 한 줄 주석(//), 여러 줄 주석(/* */)이 있다.

```
1  /*  
2   Blink  
3   LED가 1초 간격으로 깜빡이는 코드  
4   아두이노 기본 예제에 포함되어 있다.  
5  */  
6  
7  // the setup function runs once when you press reset or power the board  
8  void setup() {  
9      pinMode(LED_BUILTIN, OUTPUT); // 내장 LED핀을 출력 모드로 설정한다.  
10 }  
11  
12 // the loop function runs over and over again forever  
13 void loop() {  
14     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
15     delay(1000); // wait for a second  
16     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
17     delay(1000); // wait for a second  
18 }
```

아두이노 기초

코드 살펴보기 - 예제 해석



```
sketch_jan11b $
1 void setup() {
2   pinMode(13, OUTPUT);
3   digitalWrite(13, HIGH);
4 }
5
6 void loop() {
7
8 }
```

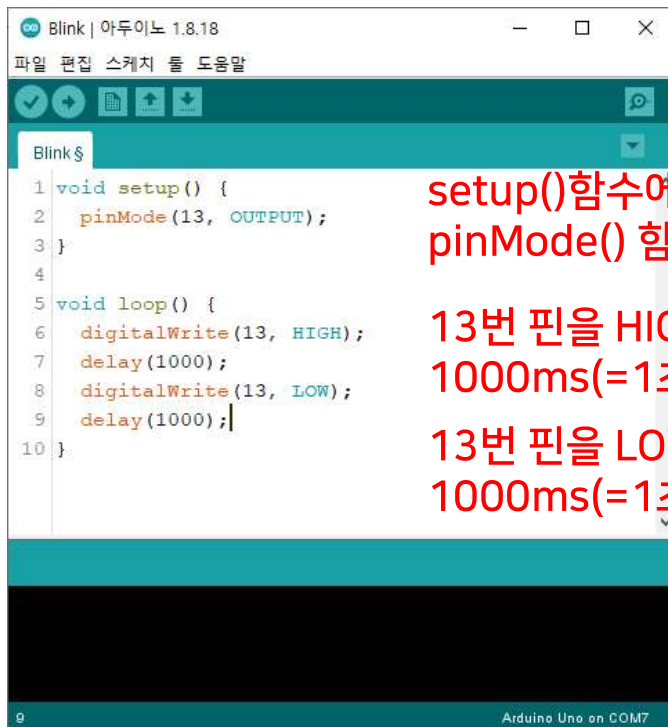
setup()함수에서 최초 한 번
pinMode() 함수를 호출하여 13번 핀을 출력으로 설정한다.
digitalWrite() 함수를 호출하여 13번 핀을 HIGH상태로 설정한다.

loop() 함수에서 아무 것도 하지 않으므로
13번 핀 HIGH 상태가 계속 유지된다.

예제 1

아두이노 기초

코드 살펴보기 - 예제 해석



```
1 void setup() {  
2   pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(13, HIGH);  
7   delay(1000);  
8   digitalWrite(13, LOW);  
9   delay(1000);  
10 }
```

setup()함수에서 최초 한 번
pinMode() 함수로 13번 핀을 출력 모드로 설정한다.

13번 핀을 HIGH 상태로 만든다.
1000ms(=1초)간 멈춤
13번 핀을 LOW 상태로 만든다.
1000ms(=1초)간 멈춤

} Loop() 함수 안 = 무한 반복

예제2

아두이노 기초

응용해보기 1

- 3개의 LED 1, 2, 3을 D1, D2, D3 포트에 출력할 수 있도록 연결해보시오.
- 위 구성한 LED 1,2,3을 1초에 한번씩 순차적으로 켜지고 꺼지도록 프로그램(스케치)을 작성하시오.
- setup() 부분에 D1, D2, D3 각 포트를 출력 포트에 설정한다.
- loop() 부분에 아래 조건들을 코드로 작성
 - LED1을 켜고 1초간 지연
 - LED1을 끄고 LED2를 켜고 1초간 지연
 - LED2를 끄고 LED3를 켜고 1초간 지연
 - LED3을 끈다.

아두이노 기초

응용해보기2

- 3개의 LED 1, 2, 3을 D1, D2, D3 포트에 출력할 수 있도록 연결해보시오.
- 위 구성한 LED 1,2,3을 1초에 한번씩 순차적으로 누적하여 켜지고, 다시 순차적으로 꺼지도록 프로그램(스케치)을 작성하시오.

정리

아두이노

마이크로컨트롤러

전압, 전류, 저항, 디지털 회로

스케치 (setup(), loop())

함수, 변수, 상수, 주석
