

따라 하면서 배우는

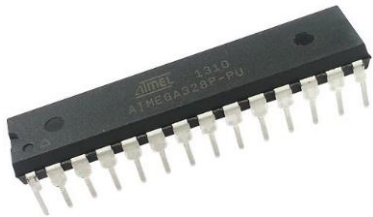
아두이노

아두이노 메가2560으로 마이크로컨트롤러 시작하기

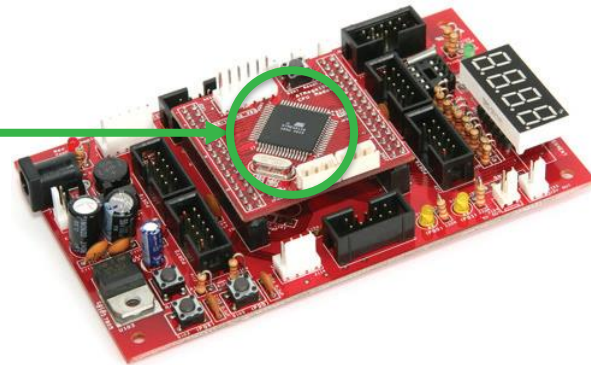
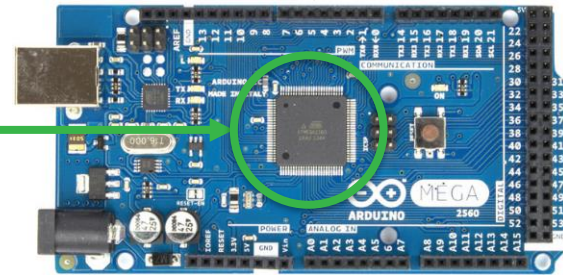
마이크로 컨트롤러

마이크로컨트롤러, 마이크로컨트롤러 보드

- 마이크로컨트롤러 보드 = 마이크로컨트롤러 + α



마이크로컨트롤러

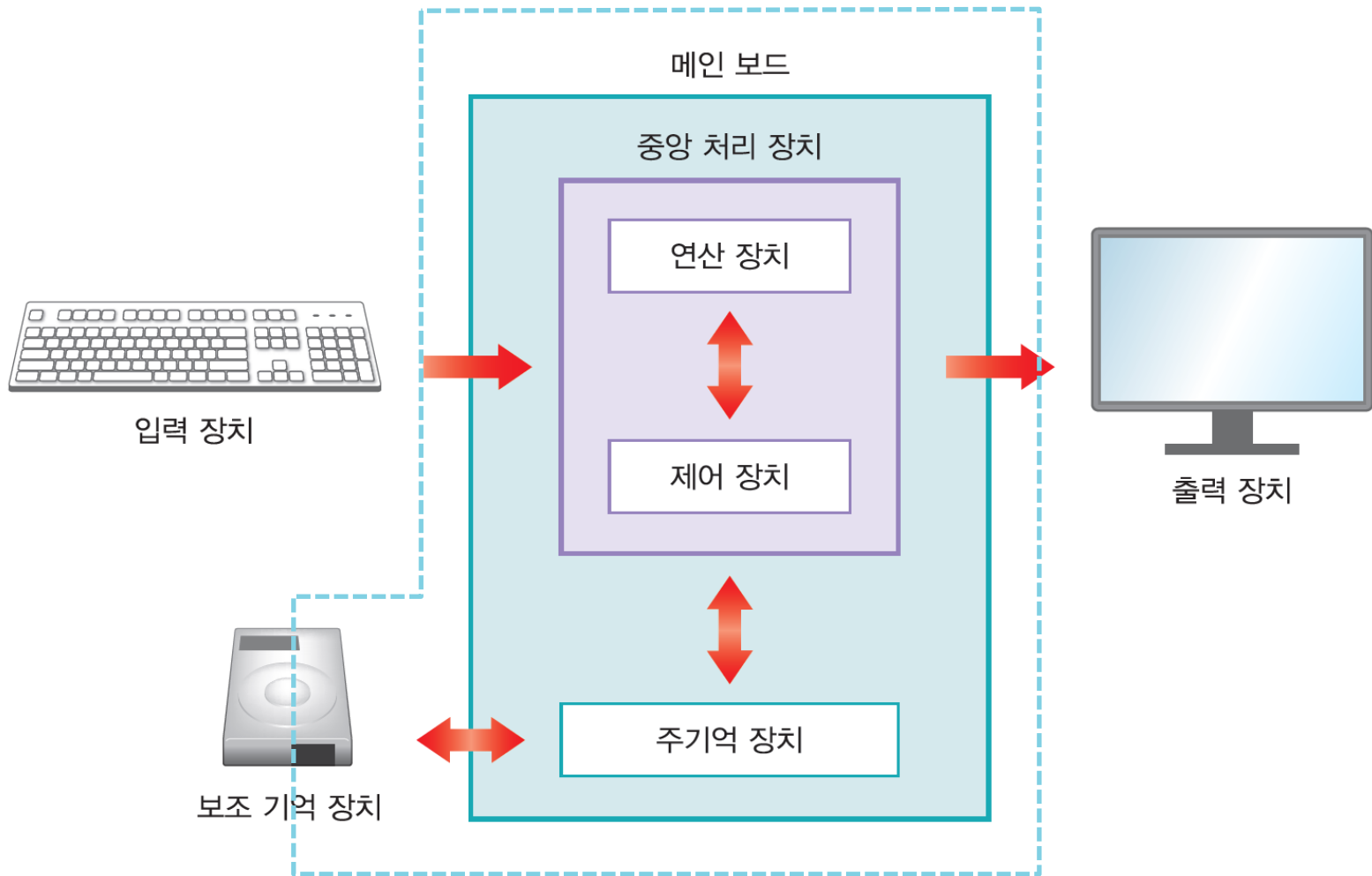


마이크로컨트롤러 보드

■ 칩 위의 컴퓨터

- 하나의 칩으로 구현한 컴퓨터
- 중앙 처리 장치 + 메모리 + HDD + 입출력 인터페이스 + α
- 데스크톱 컴퓨터의 본체와 기본적으로 동일한 구조를 가지며 동일한 동작을 수행함
- 전원만 주어진다면 컴퓨터로 동작할 수 있음
- 입력장치와 출력장치를 연결함으로써 컴퓨터와 동일하게 동작

컴퓨터의 구성 요소



마이크로컨트롤러 = 메인보드 + 일부 HDD 기능

마이크로프로세서, 마이크로컨트롤러

■ 마이크로프로세서 (uP)

- 컴퓨터의 중앙 처리 장치(CPU)를 하나의 칩으로 구현한 반도체 소자
- 최초의 마이크로프로세서

- Intel 사의 4004
- 1969년 발표된 4비트 마이크로프로세서

■ 마이크로컨트롤러 (uC)

- 메인보드 + 일부 HDD 기능
- 싱글 칩 컴퓨터 (Single Chip Computer)
- 작고 가벼운(마이크로, micro) 제어장치(컨트롤러, controller) 구성을 위해 사용되는 특화된 마이크로프로세서의 일종

uP와 uC 사용 시스템 비교

항목	아두이노 메가2560	데스크톱 컴퓨터
CPU	ATmega2560	Intel Core i7
비트	8	64
메모리	256 KByte	8 GByte
클럭	16 MHz (싱글 코어)	3.4 GHz (쿼드 코어)
시스템 가격	약 25,000원 (ATmega2560)	약 1,000,000원

마이크로컨트롤러 사용 예

분야	사용 예
의료	의료기 제어, 자동 심박계
교통	신호등 제어, 주차장 관리
감시	출입자/침입자 감시, 산불 감시
가전	에어컨, 세탁기, 전자레인지
음향	CD 플레이어, 전자 타이머
사무	복사기, 무선 전화기
자동차	엔진 제어, 충돌 방지
기타	게임기, 차고 개폐 장치

uC를 위한 프로그램 개발 과정

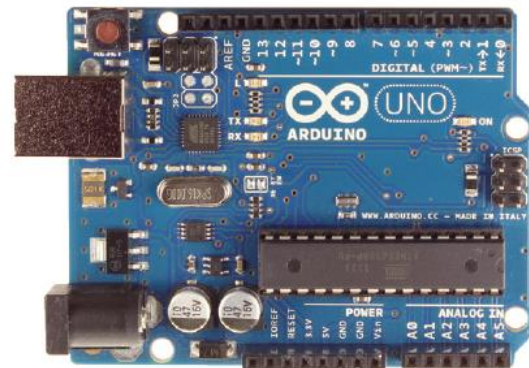
■ 교차 개발 환경

- 프로그램 개발은 **개발 시스템**(데스크톱 컴퓨터)에서 진행
- 개발 시스템은 **목적 시스템**(마이크로컨트롤러)에서 실행되는 기계어 파일을 생성할 수 있는 **교차 컴파일러** 사용
- 개발된 기계어 파일은 전용 장치를 사용하여 목적 시스템으로 업로드



교차 컴파일러가 설치된
개발 시스템

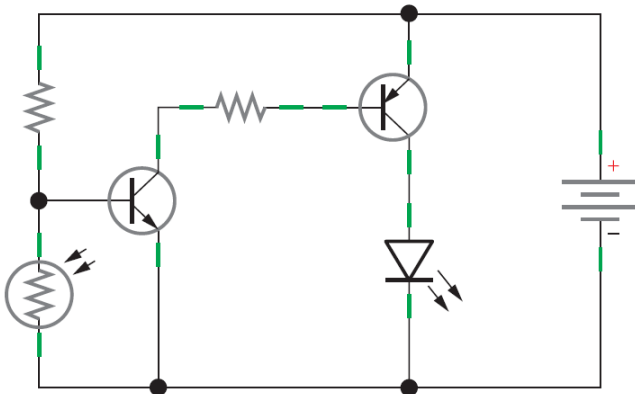
개발된 기계어 파일을
업로드



개발된 프로그램이 실행될
목적 시스템

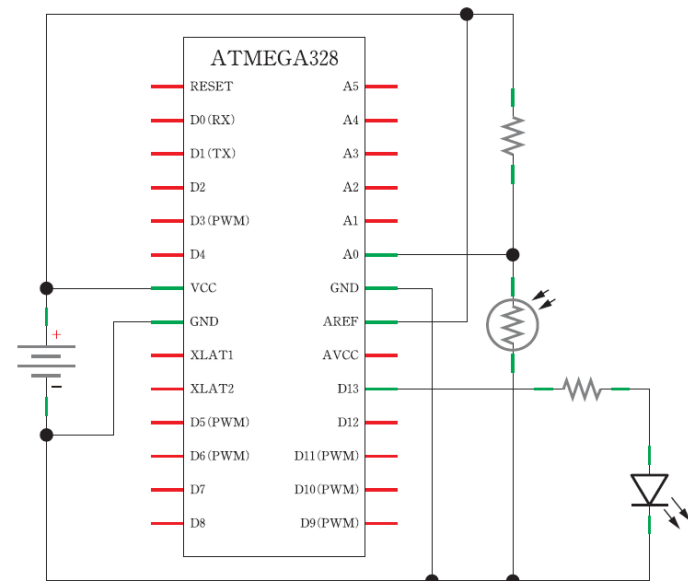
uC는 정말 필요한가? - 자동 점등 회로

uC 없이 '하드웨어'로 구현



uC 사용

'하드웨어 + 소프트웨어'로 구현



■ 단점

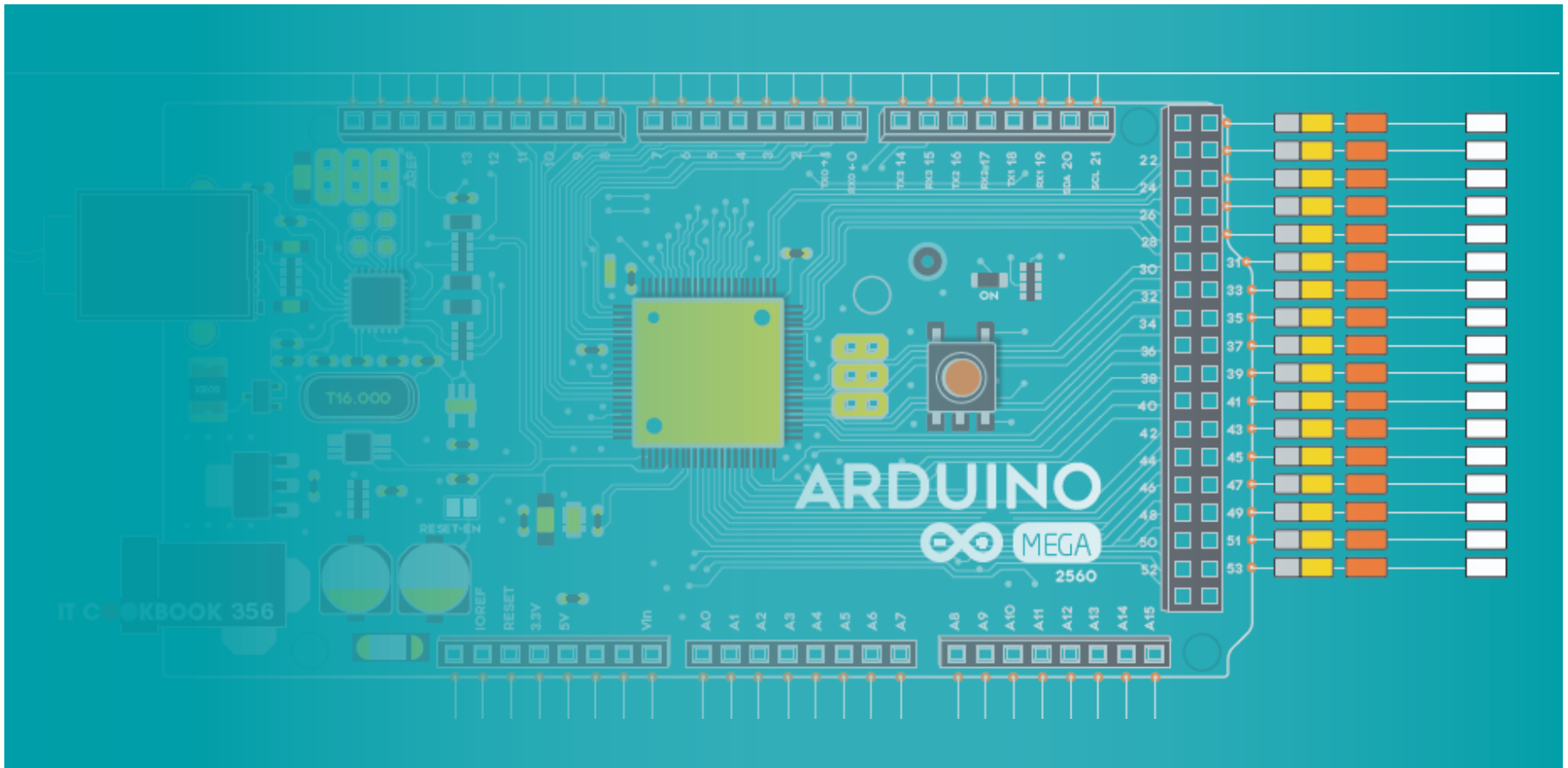
- 별도의 프로그램 기술 필요
- 소프트웨어 오류의 잠재적 가능성
- 별도의 소프트웨어(펌웨어)를 구현해야 함

■ 장점

- 기능을 변경 또는 추가하는 경우, 동일한 입출력을 사용한다면 소프트웨어 변경 만으로 가능
- 마이크로컨트롤러의 가격이 지속적으로 하락하고 있으므로 기능을 100% 사용하지 않는 경우에도 소형화, 경량화, 초기 개발 시간 단축, 유지보수 비용 절감 등의 장점으로 경쟁력 확보 가능

■ 마이크로컨트롤러는

- 작고 단순한 컴퓨터
- 컴퓨터의 본체에 해당하는 기능을 하나의 칩으로 구현 (single chip computer)
- 전원만 주어진다면 동작
 - 단, 컴퓨터와 마찬가지로 프로그램이 설치되어 있어야 함
- 입출력 장치를 연결함으로써 완전한 컴퓨터로 동작 가능
- 간단한 제어 장치를 위해 특화된 컴퓨터
 - 낮은 사양으로 데스크톱 컴퓨터와 비교할 수는 없음



따라 하면서 배우는

아두이노

아두이노 메가2560으로 마이크로컨트롤러 시작하기

아두이노 기본

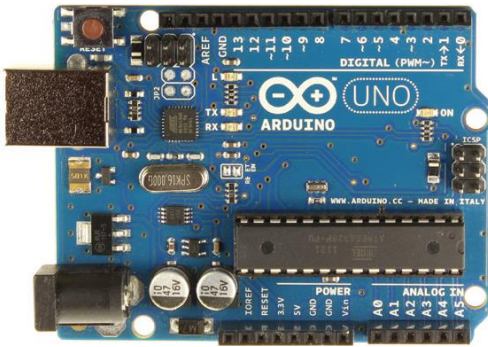
아두이노?

- 2005년 이탈리아 이브레아에서 예술가와 디자이너를 위한 마이크로컨트롤러 프로젝트로 시작
- 아두이노는
 - 하드웨어(아트멜의 AVR 시리즈 마이크로컨트롤러)와
 - 소프트웨어(프로그램 개발을 위한 전용 라이브러리를 포함하는 개발 환경)를 함께 지칭
- 아두이노 하드웨어는 마이크로컨트롤러 보드의 일종임
 - 아두이노 보드라고도 함
 - 마이크로컨트롤러(AVR uC) + 커넥터 + α
- 아두이노의 하드웨어와 소프트웨어는 모두 오픈 소스 정책에 따라 공개되어 있음

아두이노 우노

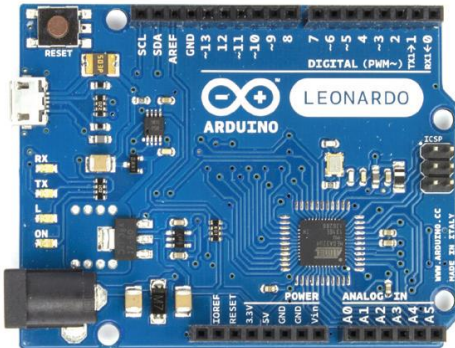


아두이노 보드의 종류 - 핀 헤더 유무



나노

ATmega328



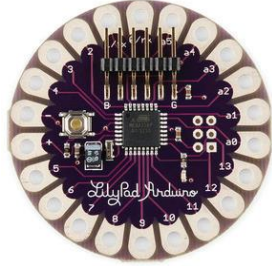
마이크로

ATmega32u4

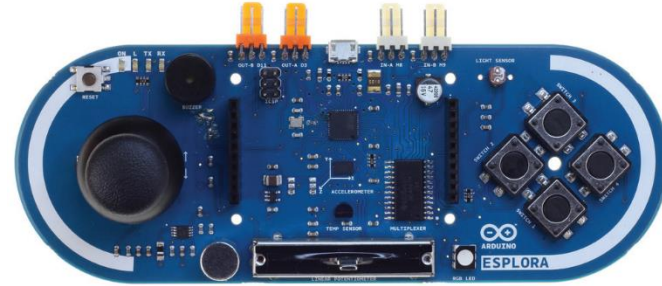
핀 헤더 있음 : 교육&개발

핀 헤더 없음 : 제품 적용

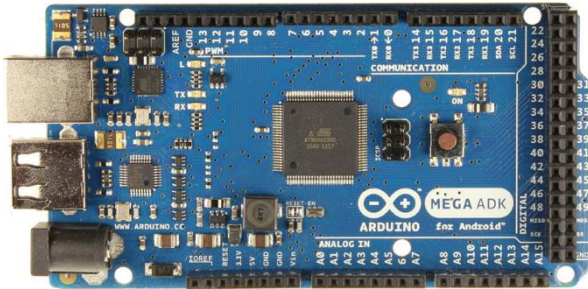
특수 목적용 아두이노 보드



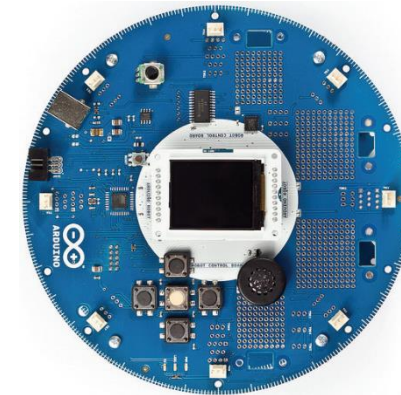
릴리패드 : 웨어러블 제작



에스플로라 : 센서 및 입력 장치 포함



메가ADK : 안드로이드 액세서리 개발



로봇 : 주행장치 개발

아두이노 보드의 종류 - uC의 종류

아두이노	마이크로컨트롤러	아키텍처	CPU 비트 수
우노	ATmega328	AVR	8
레오나르도	ATmega32u4	AVR	8
메가2560	ATmega2560	AVR	8
제로	ATSAMD21G18	ARM Cortex M0+	32
듀에	AT91SAM3X8E	ARM Cortex M3	32

AVR 기반 아두이노 보드

항목	우노	레오나르도	메가2560
uC	ATmega328	ATmega32u4	ATmega2560
클록	16MHz	16MHz	16MHz
핀 수	28	44	100
디지털 입출력 핀 수	14	18(12)	54
아날로그 입력 핀 수	6	6(12)	16
입출력 핀 수	20	24	70
PWM 채널 수	6	7	15
플래시 메모리	32KByte	32KByte	256KByte
SRAM	2KByte	2.5KByte	8KByte
EEPROM	1KByte	1KByte	4KByte

아두이노 우노

- 아두이노 보드 중 가장 기본이 되는 보드

- 8비트 CPU 포함

- 20개의 디지털 입출력 핀 사용 가능

- 0번에서 19번까지 핀 단위의 번호 지정
- 6개의 핀으로 PWM 신호 출력 가능

- 3, 5, 6, 9, 10, 11번 핀

- 6개의 아날로그 입력 핀 사용 가능

- 14번에서 19번까지의 디지털 입력 핀과 동일
- 10비트 ADC 사용 (0~1023의 양자화된 값)

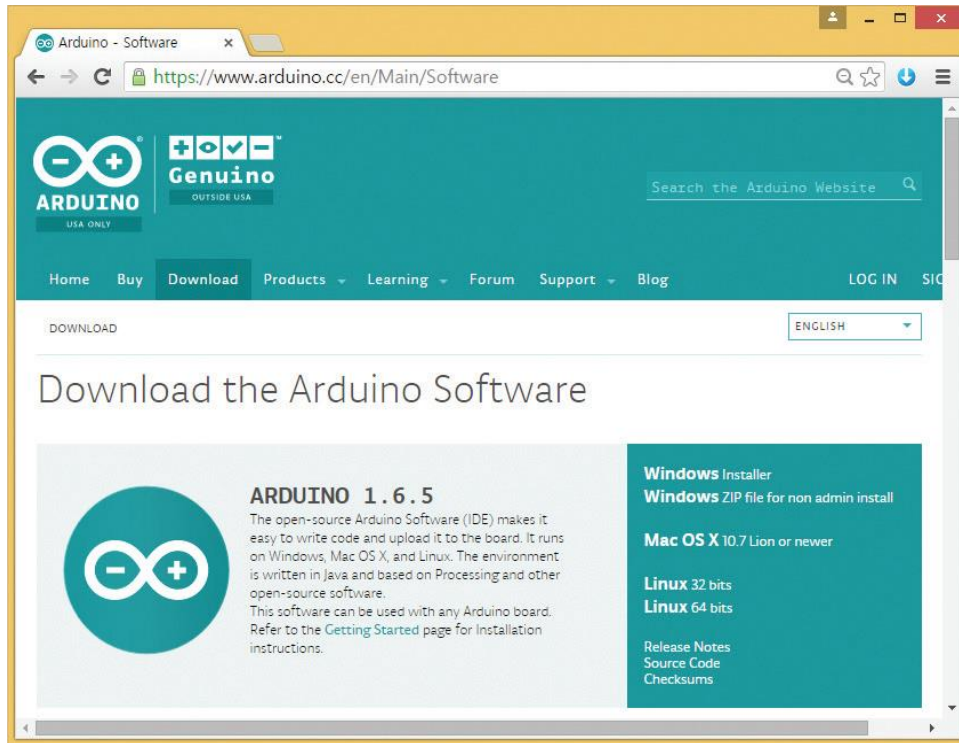
- DAC는 포함되어 있지 않으므로 아날로그 값 출력은 불가능

- 아두이노 메가2560은 디지털/아날로그 핀의 수만 다르고 기본적인 기능은 아두이노 우노와 동일

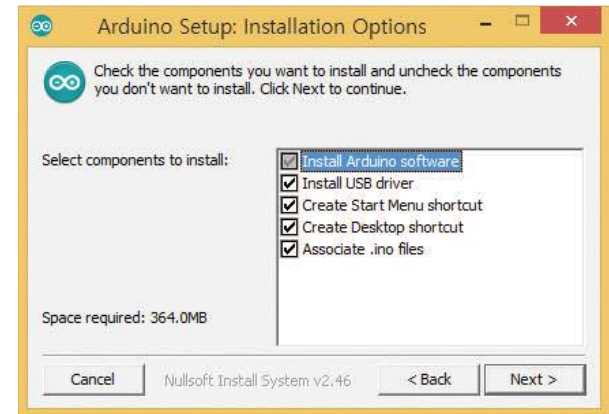
아두이노 우노 사양

항목	내용	비고
마이크로컨트롤러	ATmega328	ATmega328P-PU
동작 전압	5V	-
입력 전압	7~12V	추천 입력 범위
디지털 입출력 핀	14개	6개 PWM 출력 핀 (3, 5, 6, 9, 10, 11번 핀)
아날로그 입력 핀	6개	14번에서 19번까지의 디지털 핀
플래시 메모리	32KB	ATmega328, 부트로더 0.5KB
SRAM	2KB	ATmega328
EEPROM	1KB	ATmega328
클록 주파수	16MHz	-

아두이노 IDE 설치 - 1

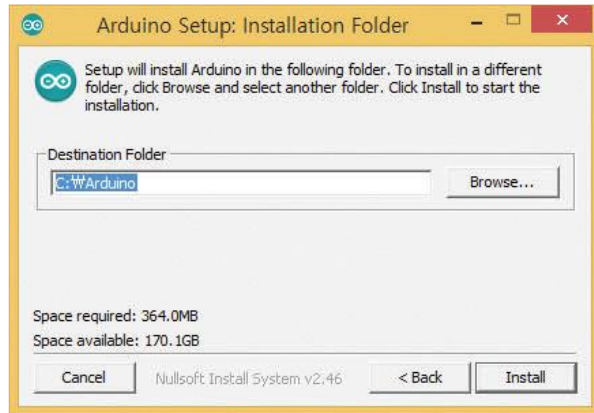


(1) IDE 다운로드 : www.arduino.cc

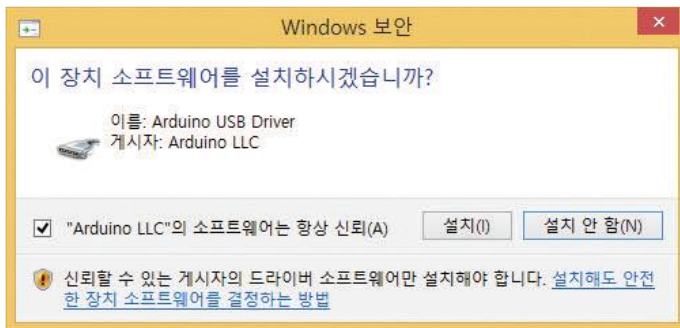


(2) 인스톨러 실행

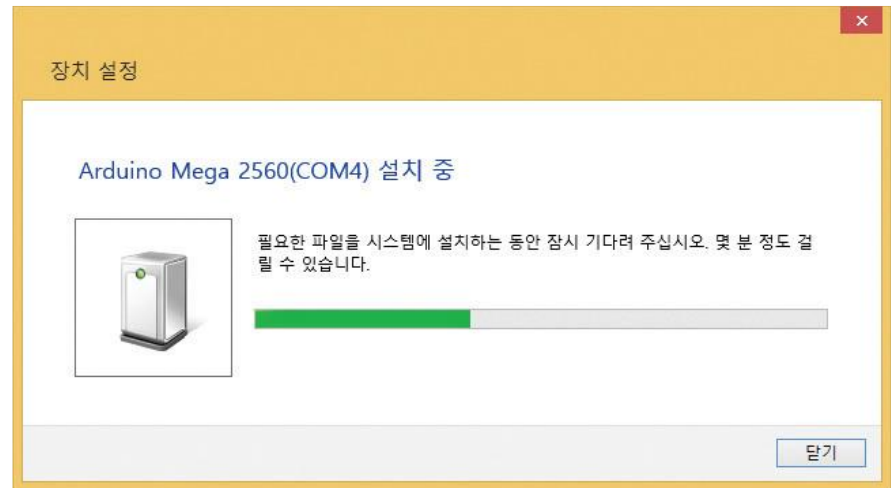
아두이노 IDE 설치 - 2



(3) 설치 디렉터리 지정



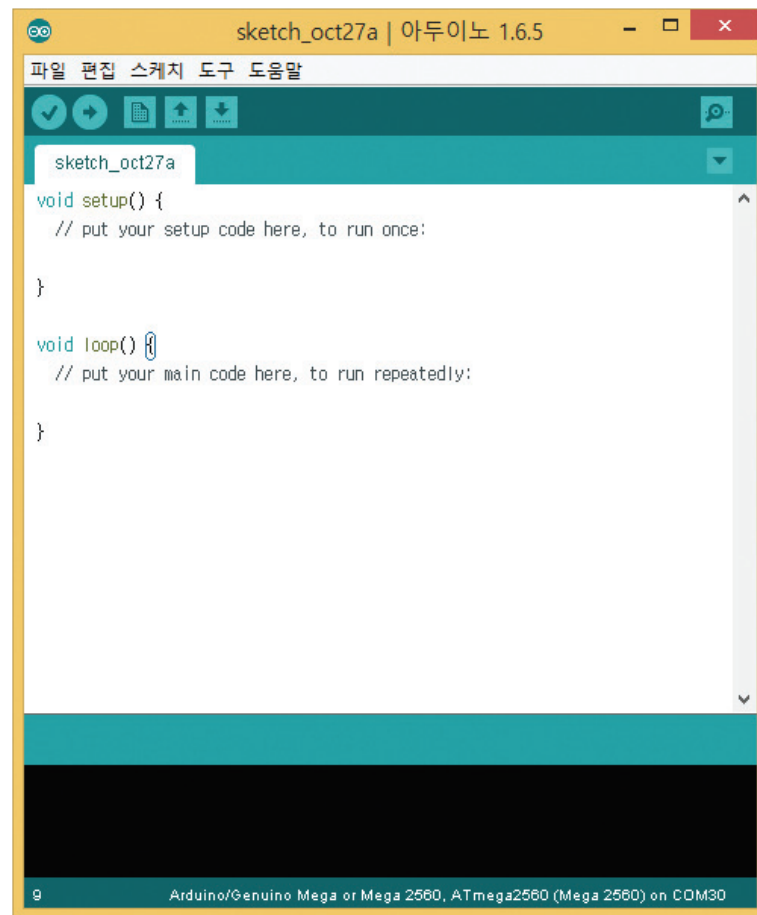
(4) 아두이노 보드 드라이버 자동 설치



(5) 아두이노 보드 연결
: 가상 COM 포트 자동 설정

아두이노 IDE

- 작고 간단한 통합개발환경
- 초보자를 위해 꼭 필요한 기능들만으로 구성
- 한 번의 클릭으로 컴파일에서 업로드까지 진행
- Java로 구현되어 OS 간 이식성이 뛰어남
- 디버깅 기능은 제공하지 않음



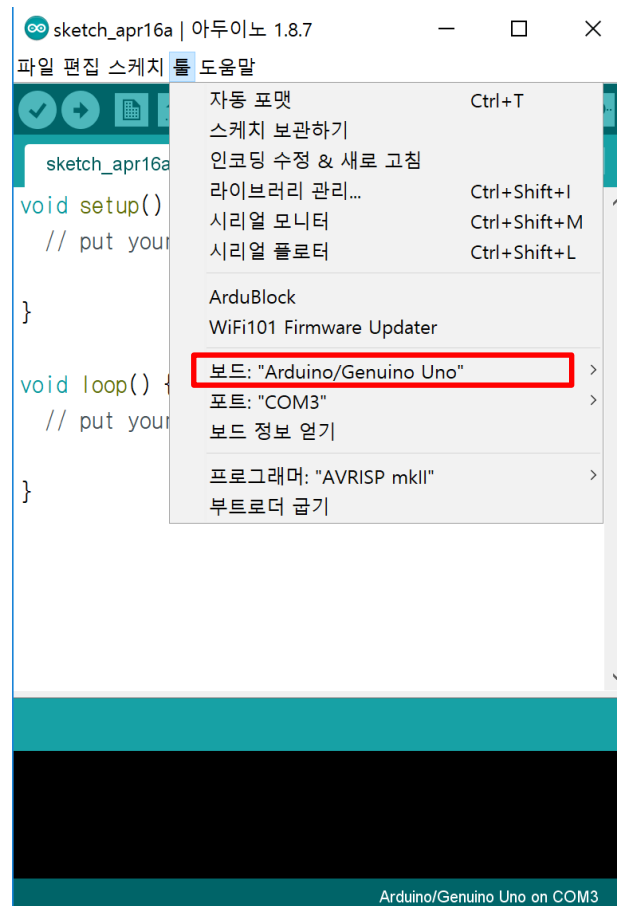
아두이노 SW 장점 – 라이브러리

- 주변장치 또는 uC의 특정 기능을 위한 전용의 C++ 스타일 라이브러리
- 대부분의 아두이노 보드에서 사용 가능
 - AVR 시리즈 uC를 사용한 보드에 비해 ARM Cortex-M 시리즈 uC를 사용한 보드는 호환성이 떨어짐
- 라이브러리별로 디렉터리 관리
 - 라이브러리 디렉터리에는 소스(*.cpp) 및 헤더(*.h) 파일을 포함하는 디렉터리와 예제 디렉터리가 포함
- 라이브러리의 종류
 - 기본 라이브러리 : 아두이노 IDE와 함께 기본적으로 설치
 - 확장 라이브러리 : 개별적으로 설치

프로그램 업로드를 위한 설정 1

■ 아두이노 보드 선택

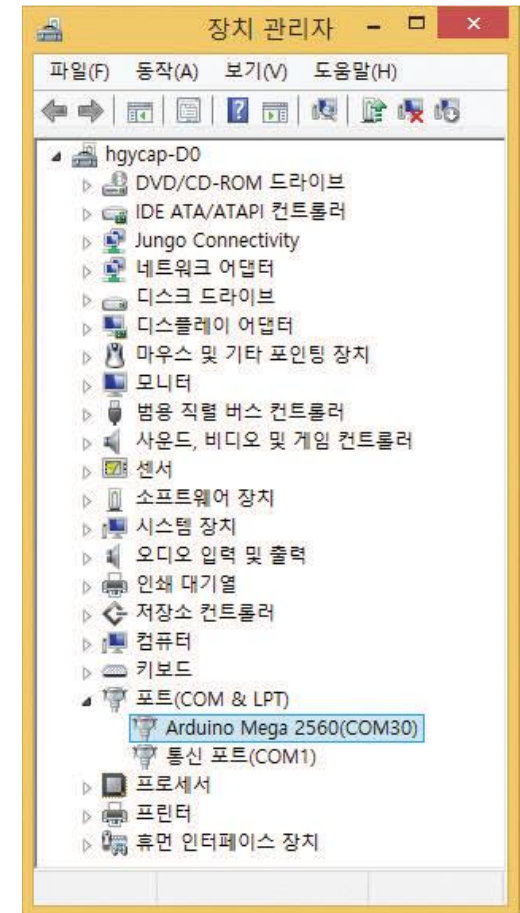
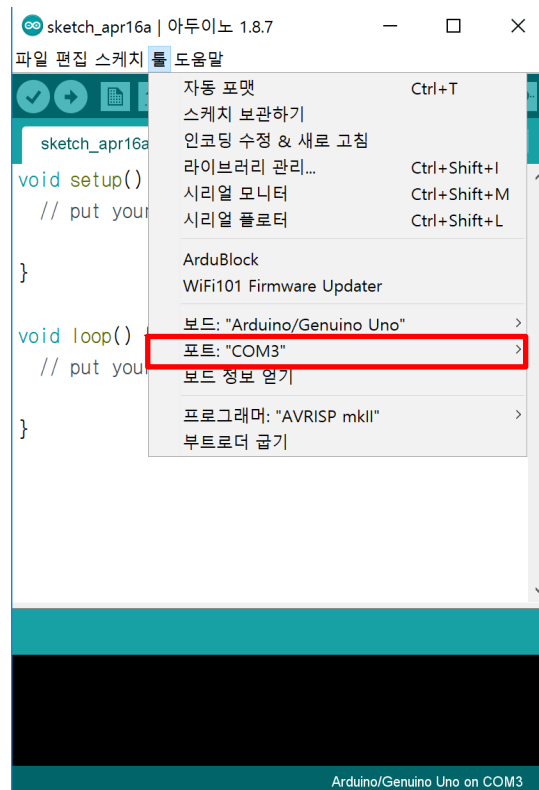
- 사용하고자 하는 보드를 선택
- '도구 → 보드' 메뉴에서 'Arduino/Genuino Uno' 선택



프로그램 업로드를 위한 설정 2

■ 포트 선택

- 아두이노 보드에 할당된 가상 COM 포트 선택
- '도구 → 포트' 메뉴에서 해당 포트 선택
- 아두이노 보드에 할당된 포트는 장치관리자에서 확인



■ 툴바 사용

- 확인 : 스케치 컴파일
- **업로드** : 스케치 컴파일 및 생성된 기계어 파일을 uC로 업로드
→ 한 번의 클릭으로 업로드까지 진행
- 시리얼 모니터 : 컴퓨터와의 데이터 송수신을 위한 프로그램 실행



스케치의 구조

- 스케치는 C/C++을 기반으로 함

- main 함수는 존재하지 않음

- main 함수는 숨겨져 있으므로 신경 쓰지 않아도 됨

- 2개의 기본 함수로 구성

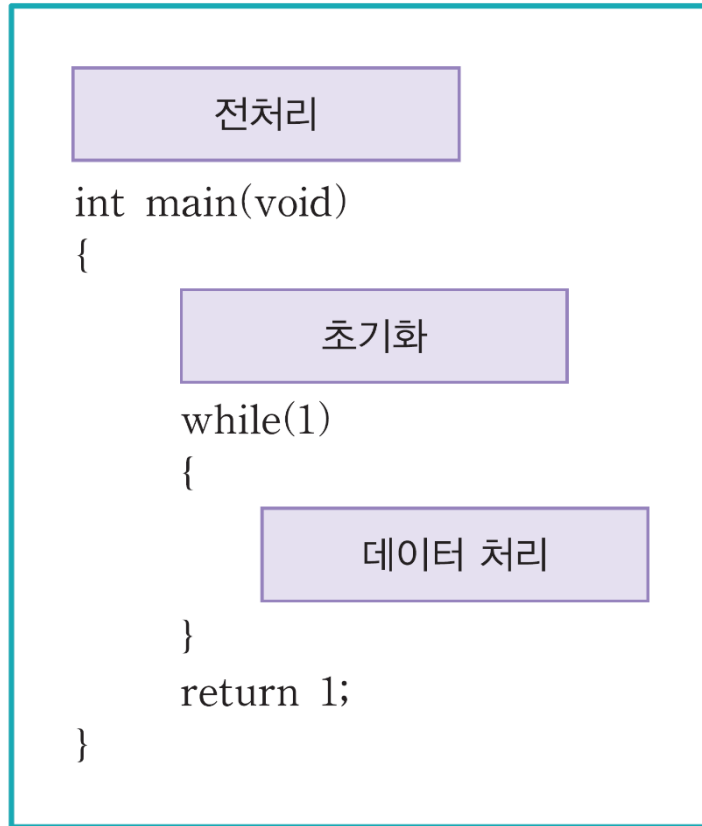
- setup 함수

- 초기화 함수
- 스케치 실행이 시작될 때 한 번만 실행

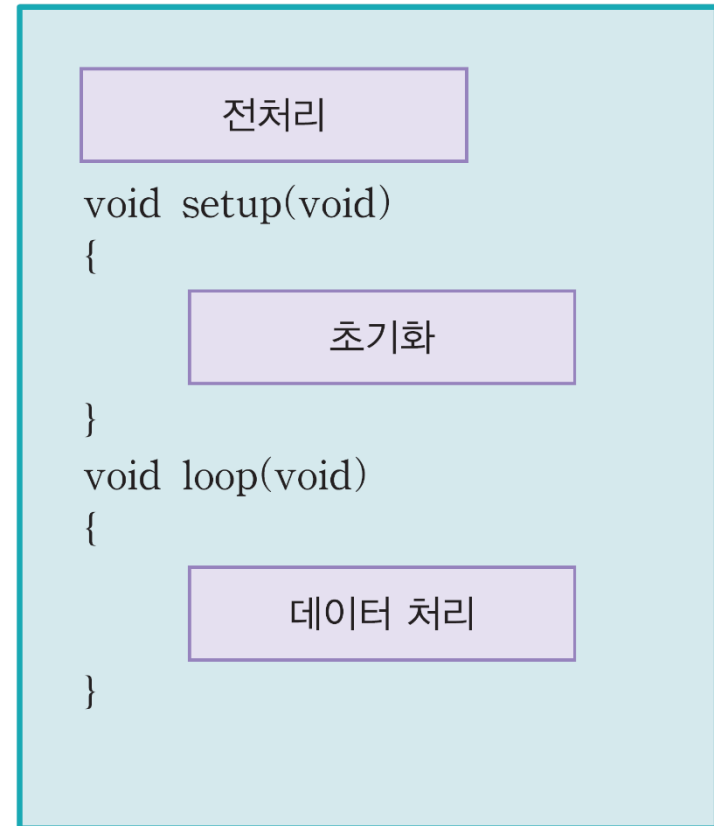
- loop 함수

- 반복 실행 함수
- uC를 위한 프로그램에서 메인/이벤트 루프에 해당

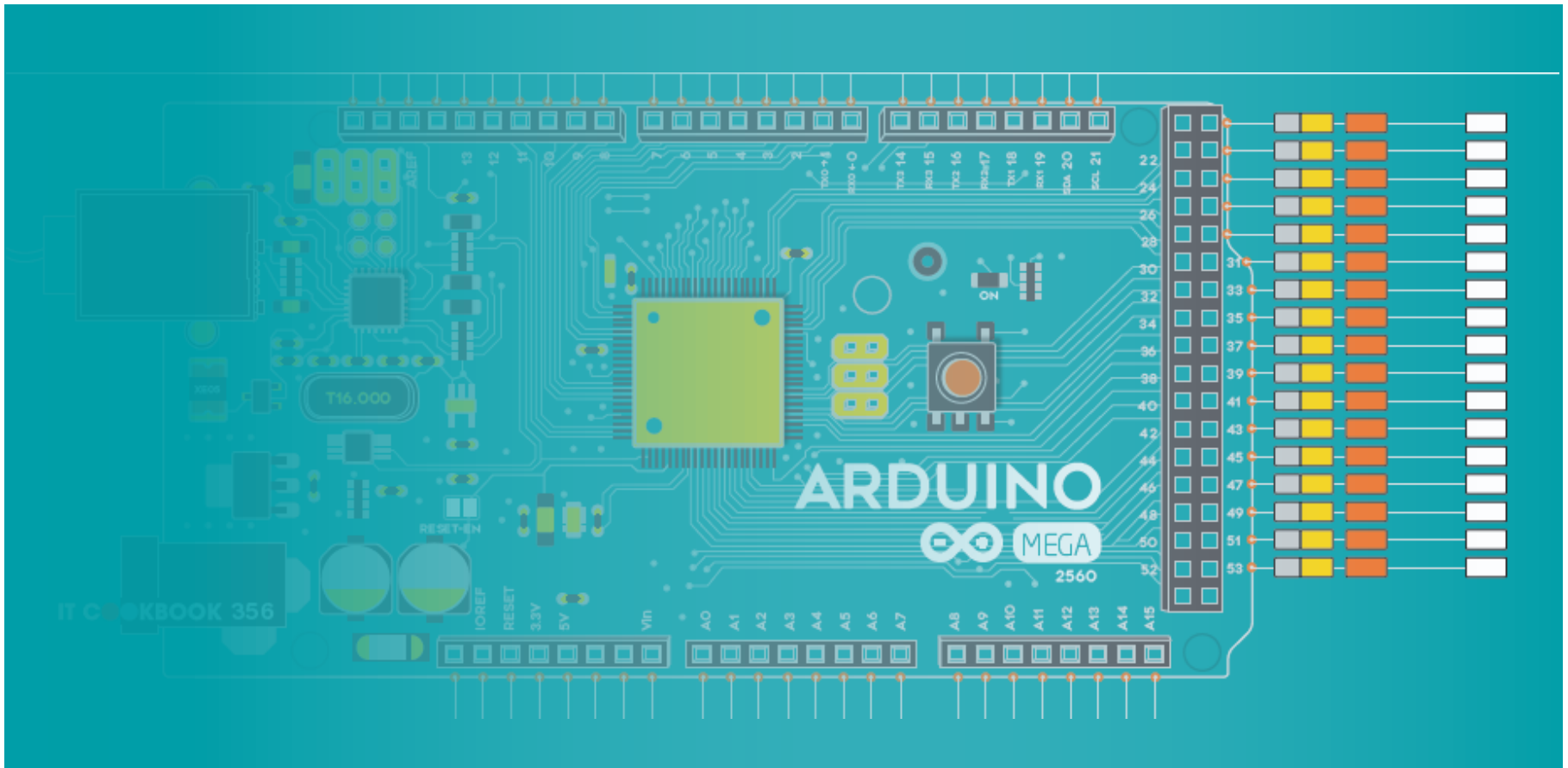
스케치의 구조 비교



uC를 위한 프로그램의 구조



아두이노를 위한 스케치의 구조



따라 하면서 배우는

아두이노

아두이노 메가2560으로 마이크로컨트롤러 시작하기

아두이노 시리얼 통신

아두이노 라이브러리

- AVR 마이크로컨트롤러의 특정 기능이나, 특정 주변장치 제어를 위한 기능을 라이브러리로 구현
- C++ 기반의 객체지향 방식으로 작성
- 종류
 - 기본 라이브러리 : 아두이노에서 제공
 - 아두이노 설치 디렉터리 아래 'libraries' 디렉터리
 - 확장 라이브러리 : 써드 파티에서 제공
 - 스케치북 디렉터리 아래 'libraries' 디렉터리

- 라이브러리와 별도로 아두이노는 2개의 기본 클래스 제공
 - UART 시리얼 통신을 위한 Serial 클래스와 문자열 처리를 위한 String 클래스
 - 별도의 헤더 파일(*.h) 없이 사용 가능

Serial 클래스

- UART 시리얼 통신을 위한 클래스
- 실제 클래스 이름은 Serial_이며, 그 객체가 Serial임
- 우노는 하나의 UART 시리얼 통신 포트만을 제공하는 반면, 메가2560은 4개의 UART 시리얼 통신 포트를 제공
- 메가2560은 4개의 UART 시리얼 통신 포트를 위해 Serial, Serial1, Serial2, Serial3의 전용 객체가 미리 생성되어 있음
- Serial_ 클래스로 객체를 생성하여 사용하는 경우는 없음

채널	연결 핀	해당 객체	설명
0	0(RX), 1(TX)	Serial	우노와 동일
1	19(RX), 18(TX)	Serial1	
2	17(RX), 16(TX)	Serial2	
3	15(RX), 14(TX)	Serial3	

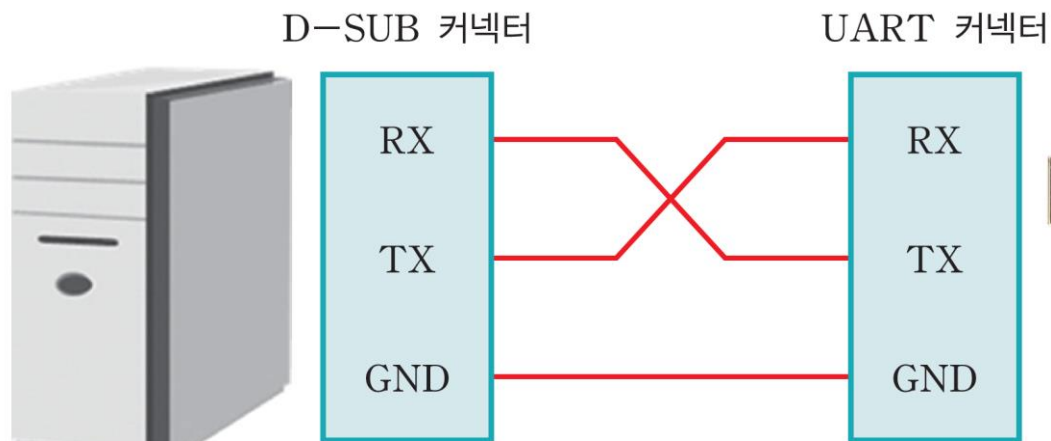
UART 시리얼 통신

- RS-232C에서 정의된 RX, TX, GND 핀을 사용하여 통신
- UART 시리얼 통신은 시리얼 통신의 한 종류이지만, 흔히 시리얼 통신은 UART 시리얼 통신을 가리킴
- RX와 TX는 서로 교차되어 연결되어야 함
- 마이크로컨트롤러의 동작 전압을 기준으로 통신 : TTL 레벨 사용
 - UART 시 아두이노 메가2560 경우 5V 기준 전압을 사용
 - RS-232C를 사용하는 컴퓨터, 3.3V를 사용하는 다른 컨트롤러와의 통신을 위해서는 레벨 변환 장치가 필요
 - 컴퓨터와 USB 통신을 위해서도 변환 장치 필요

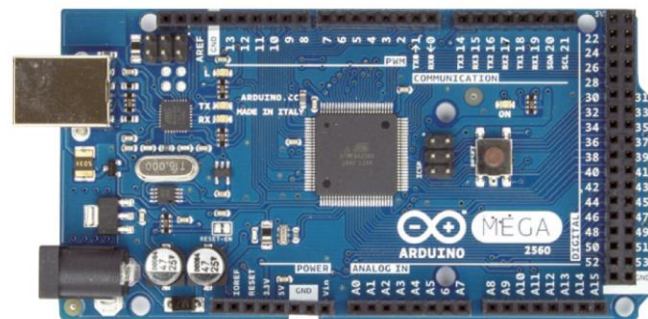
UART 시리얼 통신

- 아두이노 보드와 컴퓨터의 통신은 USB로 이루어짐
 - 아두이노 보드의 ATmega16u2 컨트롤러가 USB와 UART 사이의 변환을 담당
 - Mega2560의 0번 UART 채널 사용
 - 0번 UART 채널에 해당하는 Serial 객체를 통해 USB를 통한 컴퓨터와의 시리얼 통신 수행

컴퓨터-아두이노 시리얼 연결



RS-232C 연결



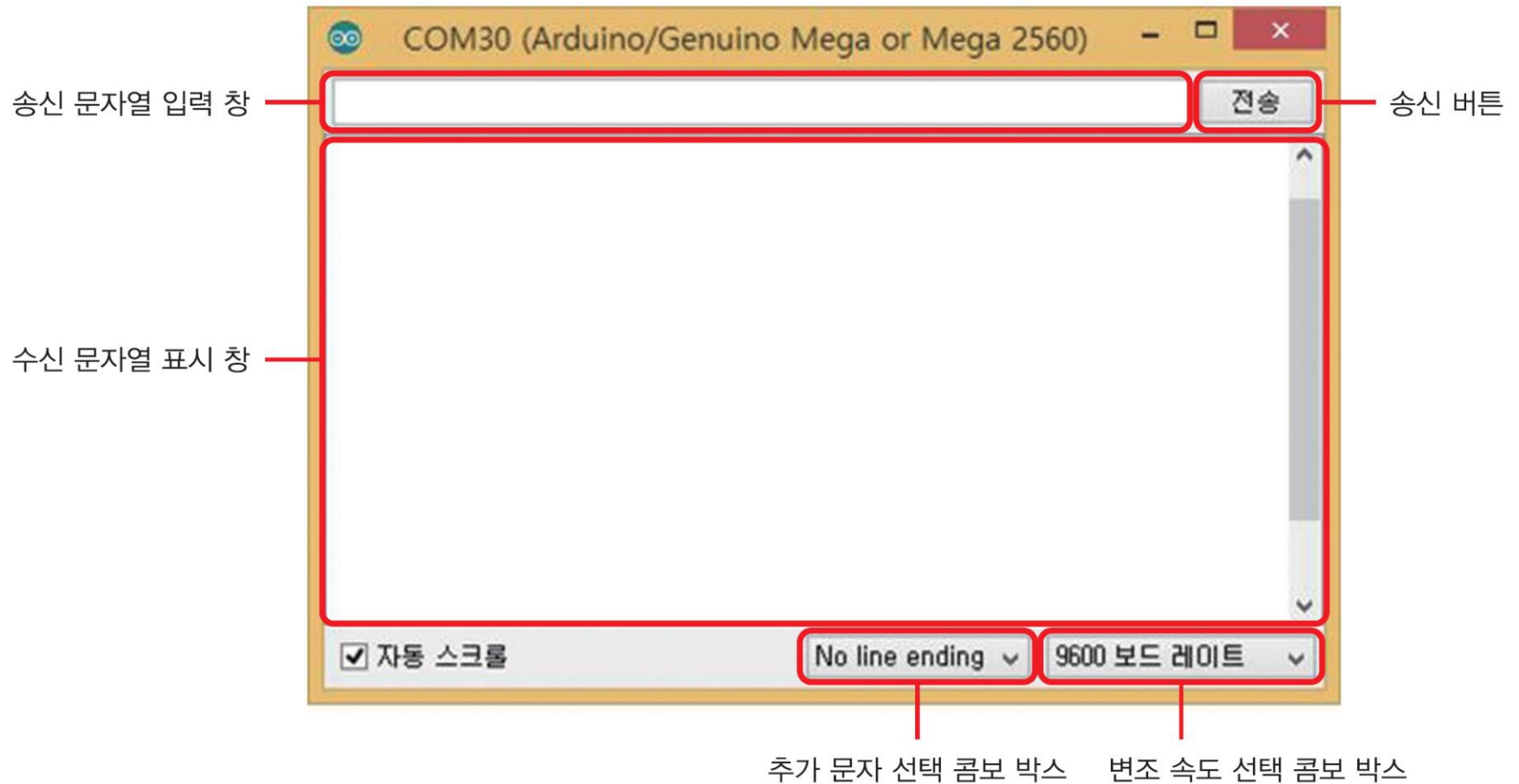
USB 연결

컴퓨터-아두이노 USB 연결

- ① 스케치 업로드를 위해 사용
- ② 시리얼 통신을 위해 사용



시리얼 모니터



Serial 클래스 – 데이터 출력 함수

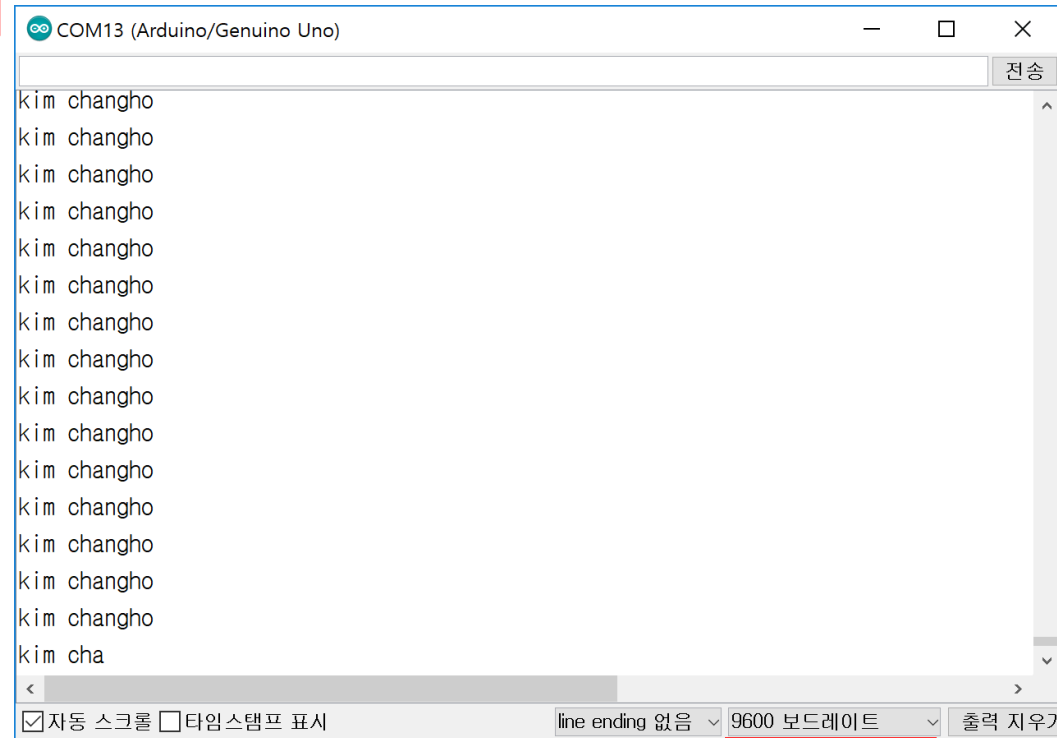
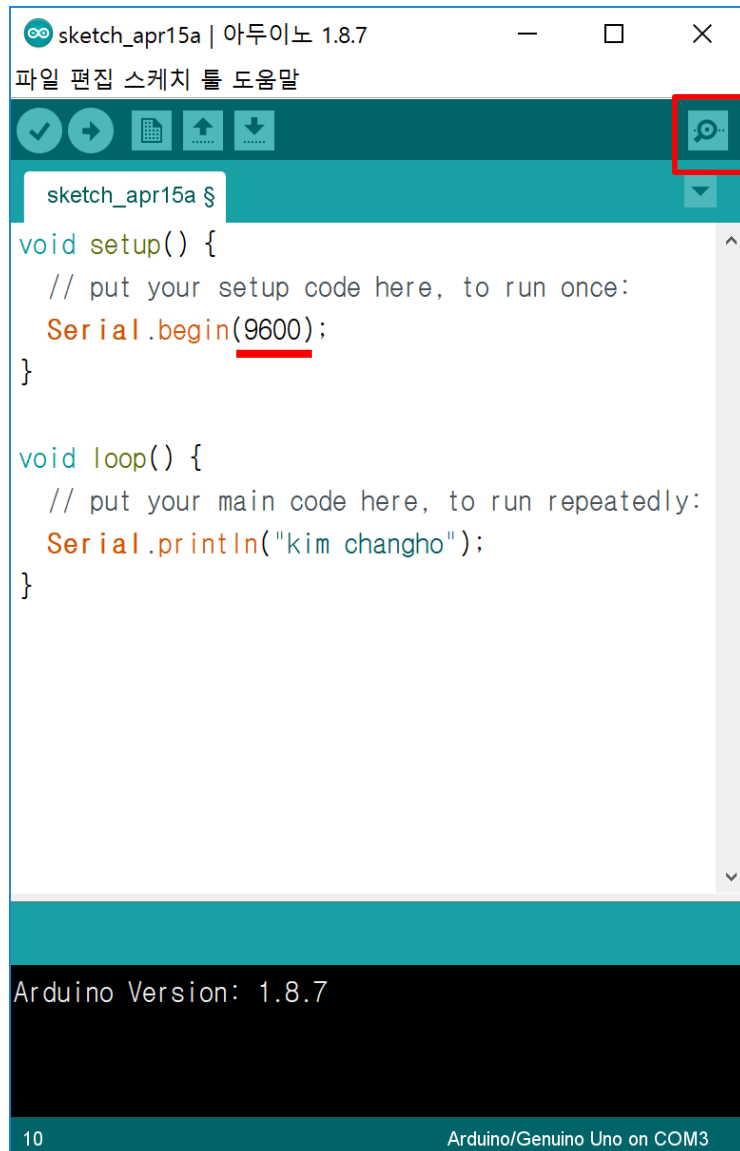
Serial.begin(baudrate)

- 함수설명
 - Serial 통신을 사용하기 위한 선언
- 입력 값
 - baudrate : Serial 통신 속도 (9600, 115200 ...)

Serial.print(value, format)

- 함수설명
 - Serial 통신을 통하여 입력 값을 보냄. (개행문자 미포함)
- 입력 값
 - value : 출력 값 (char, char 배열, String, 정수, 실수 등)
 - format : 출력 형식

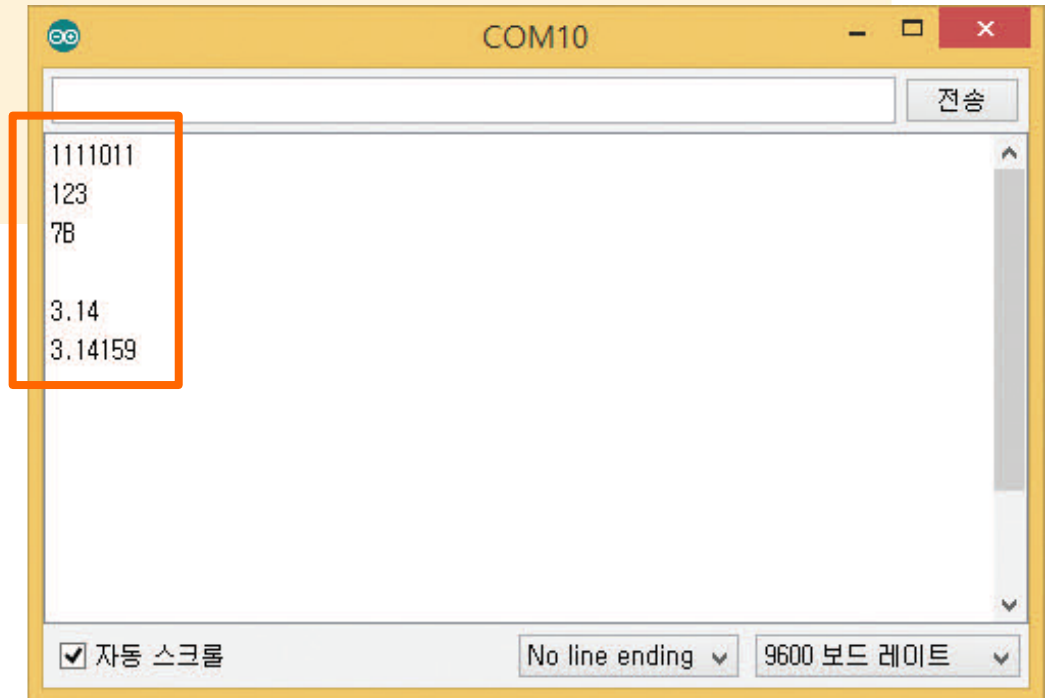
스케치 4-1 : print와 write



스케치 4-2 : 진법과 소수점 이하 자릿수

```
Serial.println(n, BIN);           // 이진수  
Serial.println(n, DEC);           // 십진수, 디폴트값으로 DEC는 생략 가능  
Serial.println(n, HEX);           // 십육진수
```

```
Serial.println();  
Serial.println(f);  
Serial.println(f, 5);
```



Serial 클래스 – 데이터 입력 함수

int available(void)

- 매개변수 : 없음
- 반환값 : 시리얼 통신 수신 버퍼에 저장된 데이터의 바이트 수

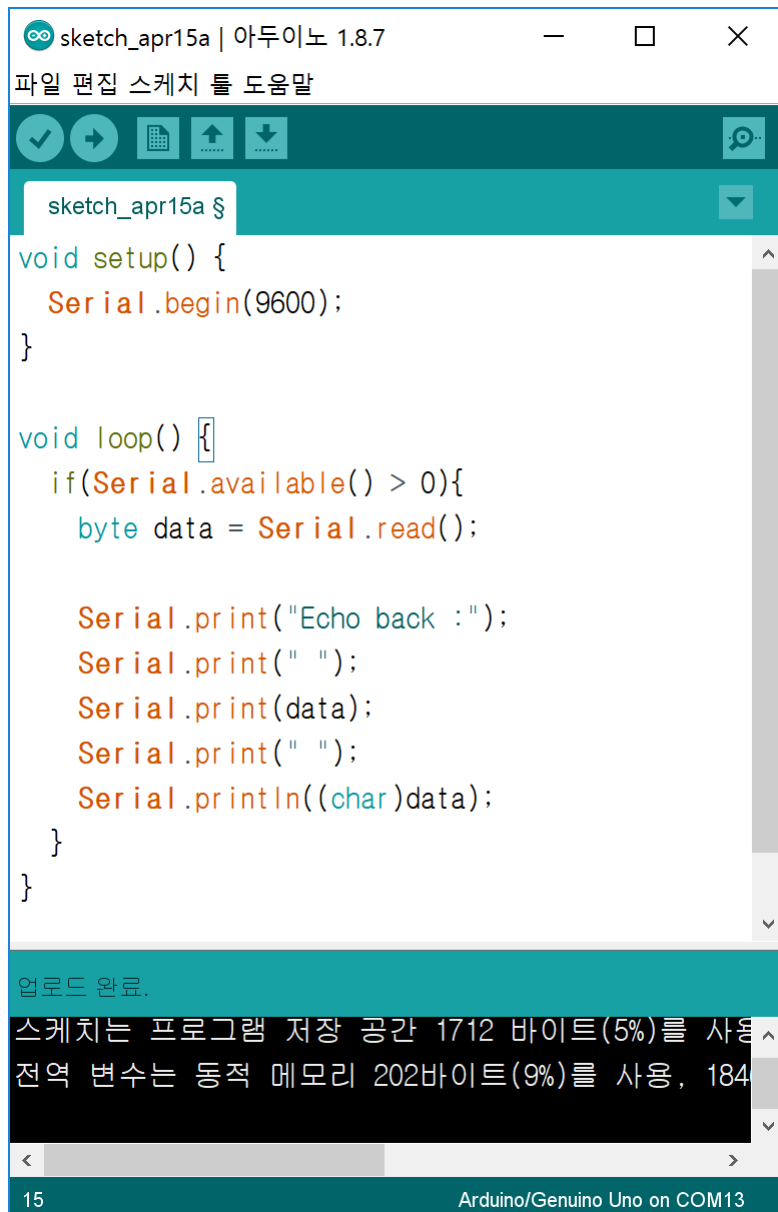
int peek(void)

- 매개변수 : 없음
- 반환값 : 시리얼 통신 수신 버퍼의 첫 번째 바이트 데이터 또는 -1

int read(void)

- 매개변수 : 없음
- 반환값 : 시리얼 통신 수신 버퍼의 첫 번째 문자 데이터 또는 -1

스케치 4-3 : 에코 백



sketch_apr15a | 아두이노 1.8.7

파일 편집 스케치 툴 도움말

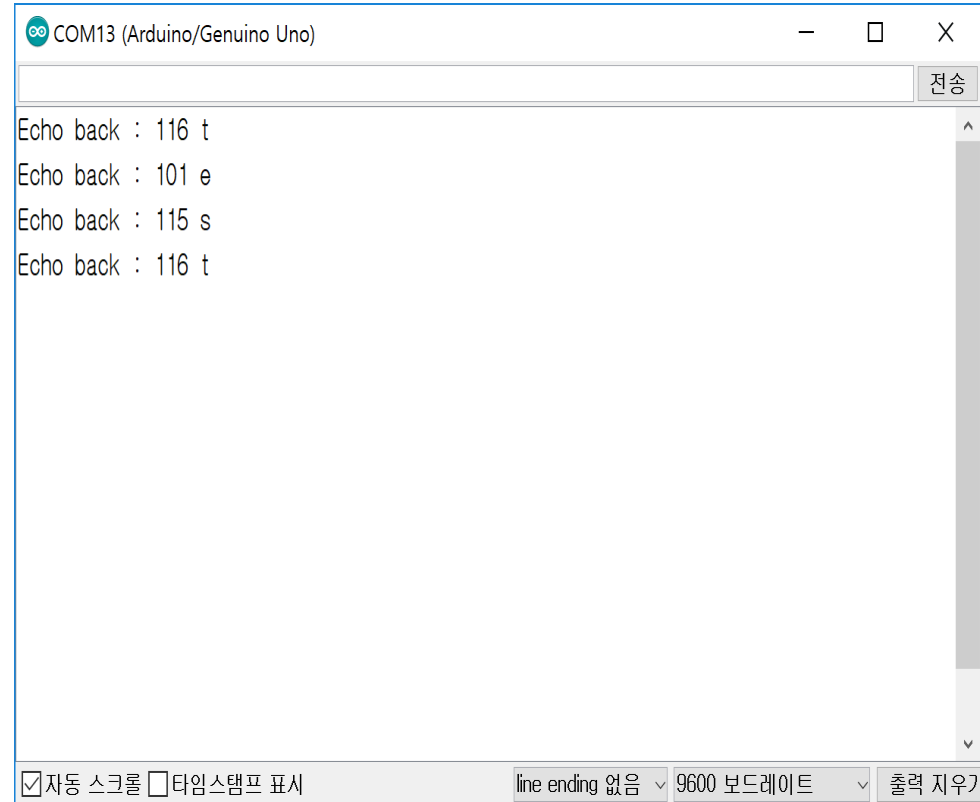
sketch_apr15a §

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(Serial.available() > 0){  
    byte data = Serial.read();  
  
    Serial.print("Echo back :");  
    Serial.print(" ");  
    Serial.print(data);  
    Serial.print(" ");  
    Serial.println((char)data);  
  }  
}
```

업로드 완료.

스케치는 프로그램 저장 공간 1712 바이트(5%)를 사용
전역 변수는 동적 메모리 202바이트(9%)를 사용, 184

15 Arduino/Genuino Uno on COM13



COM13 (Arduino/Genuino Uno)

전송

Echo back : 116 t
Echo back : 101 e
Echo back : 115 s
Echo back : 116 t

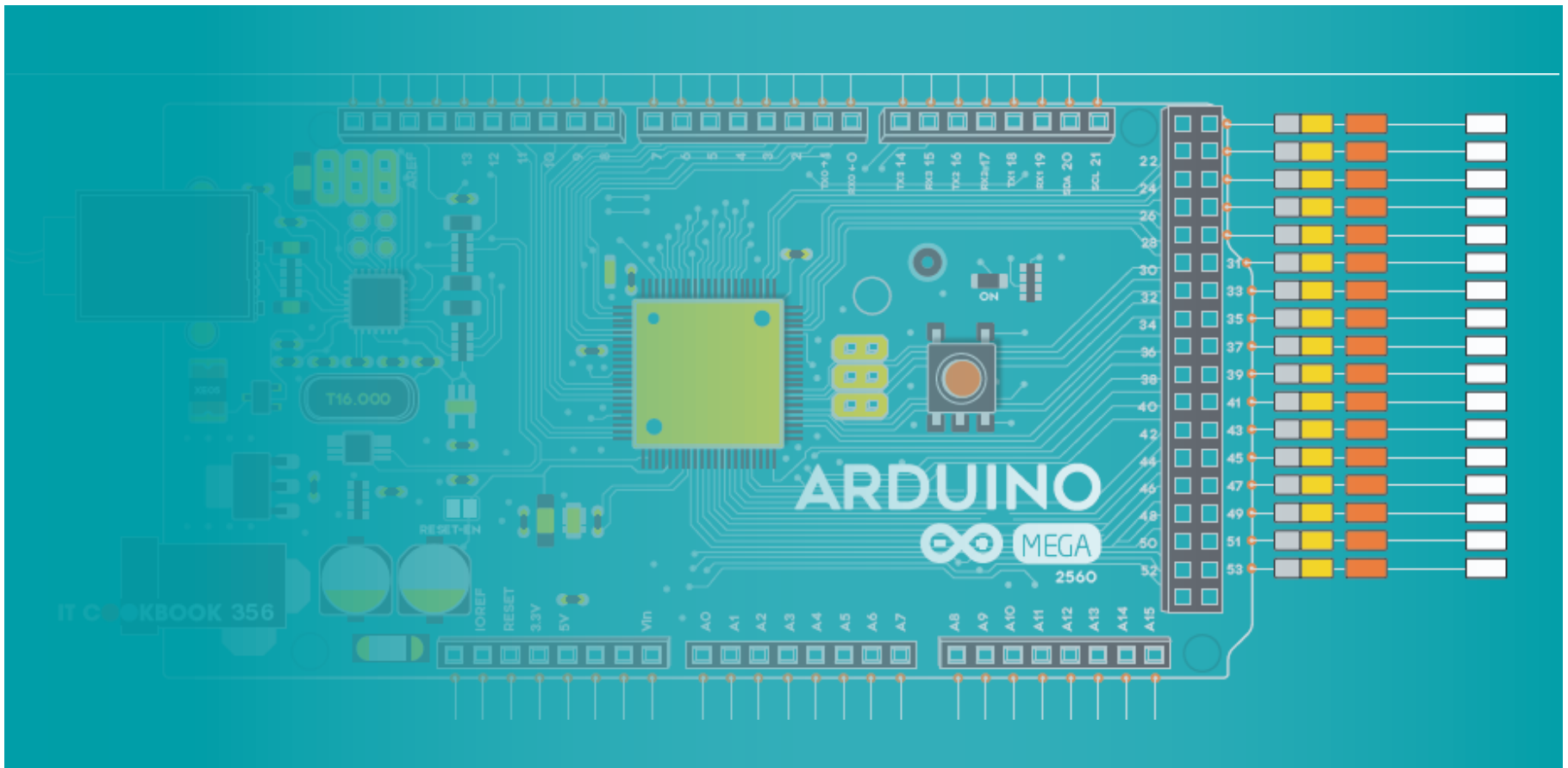
☒ 자동 스크롤 ☐ 타임스탬프 표시 line ending 없음 9600 보드레이트 출력 지우기

■ Serial 클래스

- UART 시리얼 통신을 담당하는 클래스
- 실제 클래스 이름은 Serial_이며 각 채널을 담당하는 전용 객체가 정의되어 있음
- Serial 객체의 경우 스케치 업로드에 사용되는 동일한 채널을 통해 컴퓨터와 UART 시리얼 통신을 수행

■ String 클래스

- 문자열 처리 함수
- 문자열 배열에 비해 손쉬운 String 객체 생성 및 처리 가능
- 문자열 배열에 비해 많은 메모리를 필요로 함



따라 하면서 배우는

아두이노

아두이노 메가2560으로 마이크로컨트롤러 시작하기

디지털 출력

■ 다이오드

- 양극에서 음극으로 순방향으로만 전류가 흐름



(a) 다이오드 기호



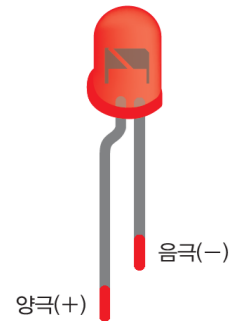
(b) 다이오드 외형

■ LED : Light Emitting Diode, 발광 다이오드

- 순방향 연결에서 빛을 냄
- 화학물질에 따라 다양한 색상의 빛을 냄
- 리모컨의 적외선 LED, 살균 소독용 자외선 LED 등도 존재
- 데이터 핀에 연결하여 비트 단위 데이터 확인

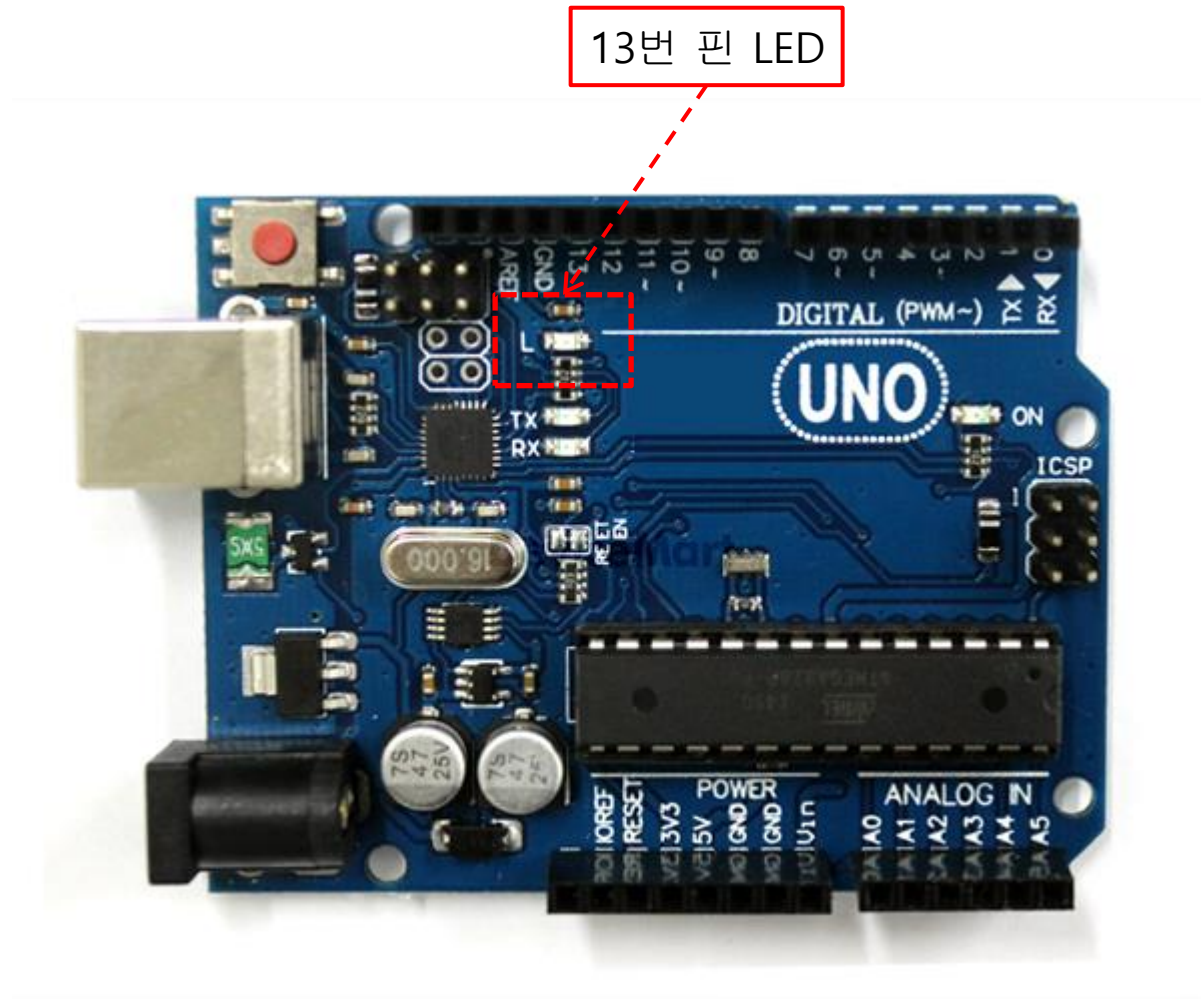


(a) LED 기호



(b) LED 외형

13번 핀에 연결된 내장 LED



블링크 스케치의 함수

void pinMode(uint8_t pin, uint8_t mode)

- 매개변수
 - pin : 설정하고자 하는 핀 번호
 - mode : INPUT, OUTPUT, INPUT_PULLUP 중 하나
- 반환값 : 없음

void digitalWrite(uint8_t pin, uint8_t value)

- 매개변수
 - pin : 핀 번호
 - value : HIGH(1) 또는 LOW(0)
- 반환값 : 없음

void delay(unsigned long ms)

- 매개변수
 - ms : 밀리초 단위의 지연 시간
- 반환값 : 없음

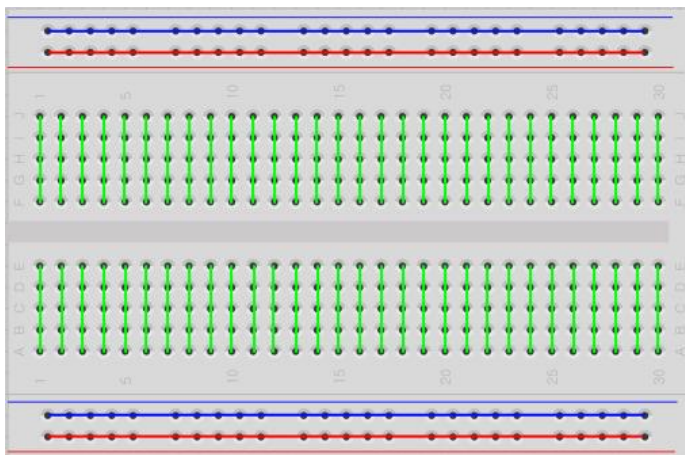
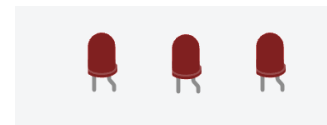
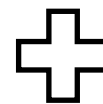
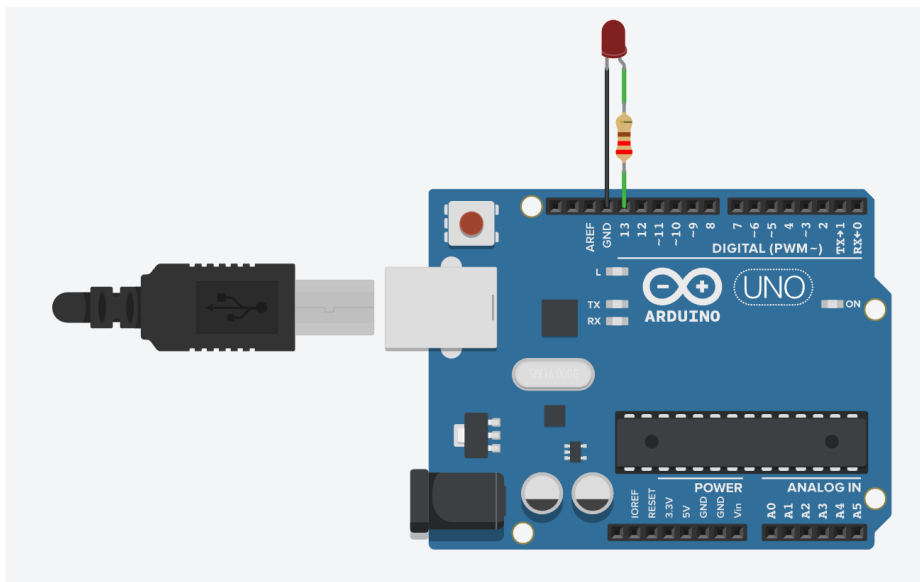
스케치 5-1 : 블링크

```
int led = 13;                                // 디지털 13번 핀에 LED 연결

void setup() {
    pinMode(led, OUTPUT);                    // 디지털 13번 핀을 출력 모드로 설정
}

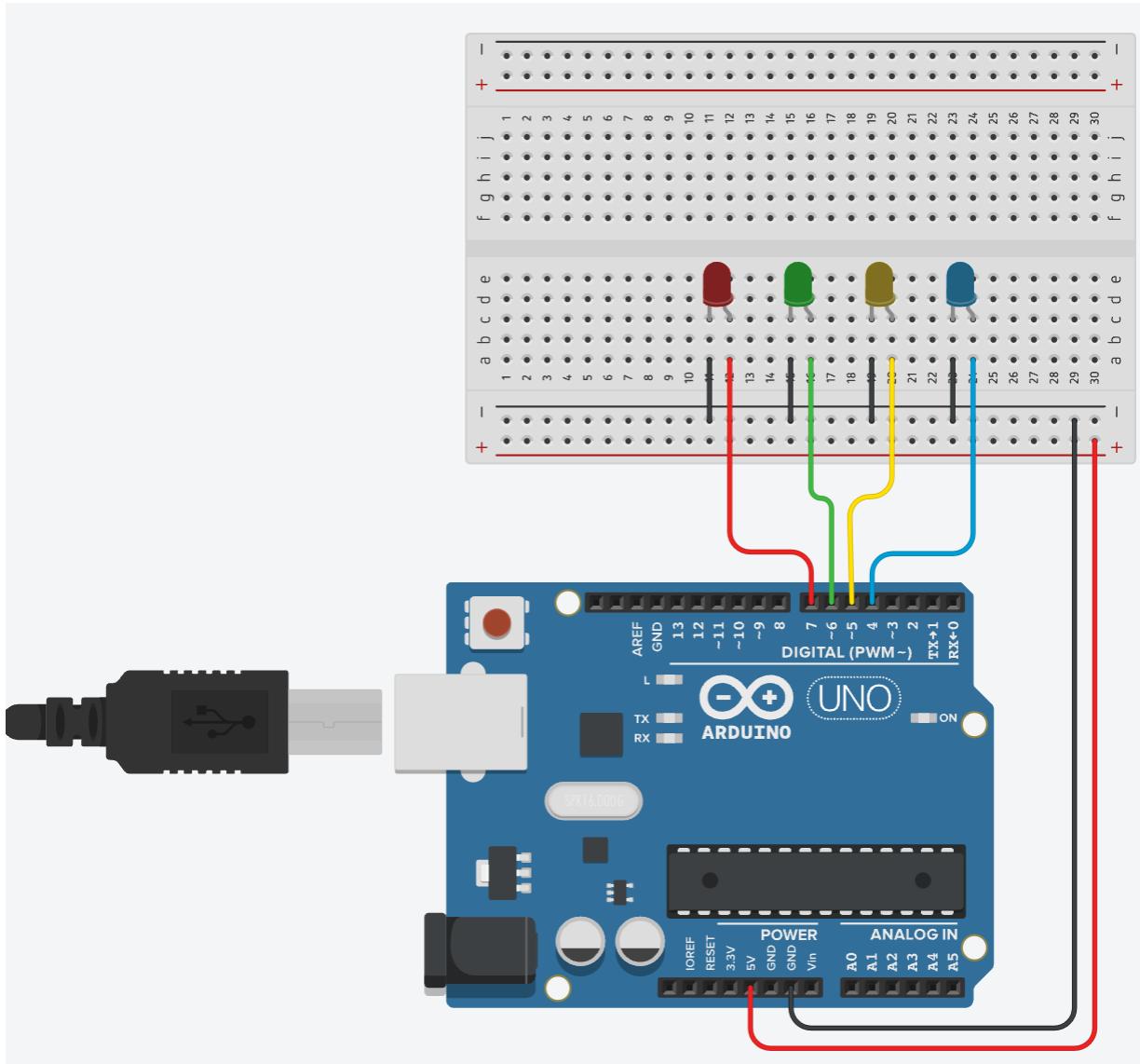
void loop() {
    digitalWrite(led, HIGH);                 // 디지털 13번 핀으로 HIGH(1 또는 ON) 출력
    delay(1000);                             // 1초(1000ms) 대기
    digitalWrite(led, LOW);                  // 디지털 13번 핀으로 LOW(0 또는 OFF) 출력
    delay(1000);                             // 1초(1000ms) 대기
}
```

아두이노와 LED 연결



<브레드보드의 사용법>

디지털 4~7번 핀 LED 연결



스케치 5-2 : 4개의 LED 제어

```
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop()
{
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  delay(1000);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  delay(1000);
}
```

실습 : 시리얼 통신을 이용한 LED 제어

- 시리얼 모니터 입력으로 LED를 제어하는 프로그램 작성
 - ✓ 시리얼 통신으로 1을 보내면 첫번째 LED가 켜짐
 - ✓ 숫자 1~4를 보내서 LED 4개를 제어
 - ✓ 숫자 5를 보내면 4개 다 켜지게 함.

Thank you!!

