

Chapter01 아두이노 시작

01 아두이노란?

02 아두이노 하드웨어(H/W)

03 아두이노 IDE 준비하기

04 아두이노 스케치의 기본 구성

05 아두이노 핀 살펴보기

01 아두이노란?

◆ 제어시스템(MCU 이용)

● 하드웨어(Hardware)

- 주문형 하드웨어 - 전문지식 요구, 대량생산에 적합
- 범용 하드웨어 - 안정성/표준화 문제, 필요한 주변 부품 설계

● 펌웨어(Firmware)

- MCU 메모리에 저장되어 주변 장치를 제어하는 프로그램

01 아두이노란?

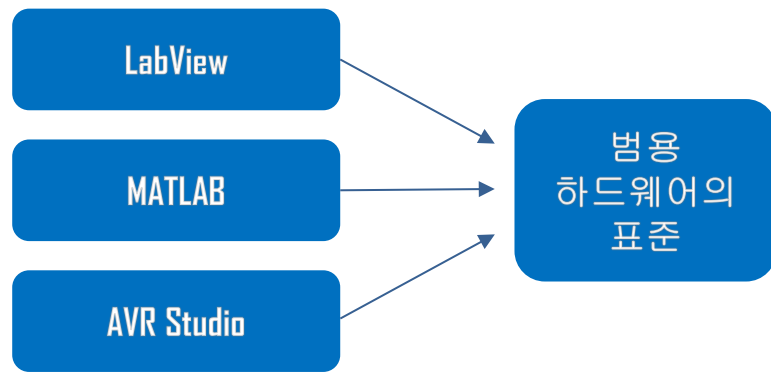
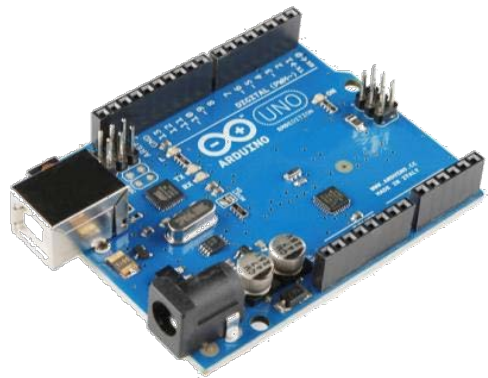
◆ 아두이노

- 2005년 Italy의 Massimo Banzi & David Cuatielles에 의해 개발
- 오픈소스 하드웨어(Open Source Hardware)
 - 제품의 모든 것이 공개된 하드웨어(회로도, PCB 등)
- 아두이노 기반 개발
 - 단순한 로봇, 온습도계, 동작감지기, 스마트홈, 로봇교육 프로그램 등

01 아두이노란?

◆ 아두이노 탄생 목적

- 전문 지식이 없는 초보자들이 쉽게 원하는 제어 장치 제작
- 전자공학을 쉽게 이해하기



01 아두이노란?

◆ 아두이노 입력

- 각종 센서, VR, Cds, Joystick, 버튼, 터치 등

◆ 아두이노 출력

- LED, LCD, 모터, 부저 등

02 아두이노 하드웨어(H/W)

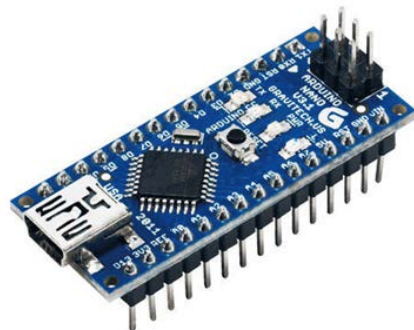
◆ Arduino Uno

- 가장 많이 사용되는 기본적인 아두이노 보드
- 8bit ATmega328p 마이컴 사용
- 보드의 핀 배열이 표준과 같이 사용



◆ Arduino NANO

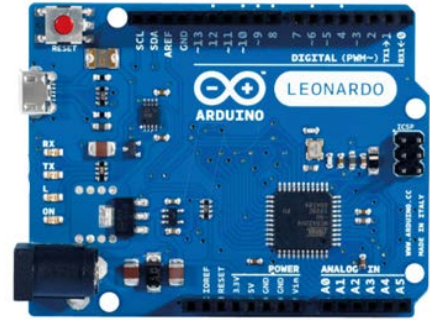
- 아두이노 우노 보드와 거의 동일한 구성
- USB 2.0 미니 B타입 케이블 사용



02 아두이노 하드웨어(H/W)

◆ Arduino Leonardo

- USB 기능 내장
- 8bit ATmega32u4 마이컴 사용
- 2개의 하드웨어 시리얼 포트 사용 가능



◆ Arduino MEGA

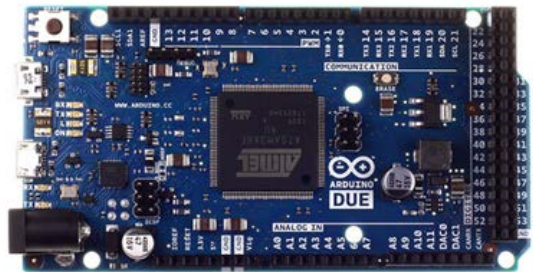
- 8bit ATmega2560 마이컴 사용
- 우노 보드보다 기능과 핀 수가 많음



02 아두이노 하드웨어(H/W)

◆ Arduino Due

- 32bit Cortex-M3 마이컴 사용
- 기능과 성능이 높음
- 핀 수가 매우 많음
- 전문적인 제품개발과 연구 목적으로 사용



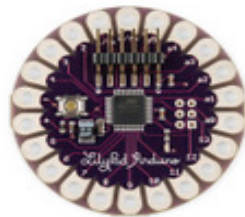
02 아두이노 하드웨어(H/W)

◆ 아두이노 쉴드

- 아두이노 UNO 보드 또는 기타 보드 위에 층층이 쌓아서 사용
- 기능 확장용 보드들을 지칭 (WiFi 쉴드, Ethernet 쉴드)

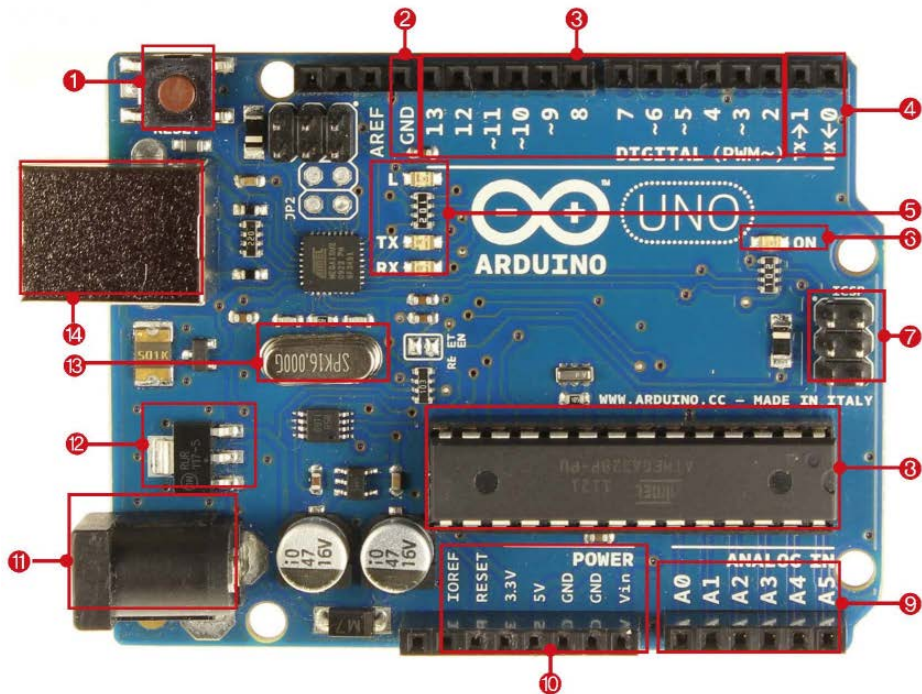
◆ 특수 용도 아두이노 보드

- Lily Pad(웨어러블용 특화 보드)
- 아두이노 FIO(Funnel IO 무선통신 특화 보드)



02 아두이노 하드웨어(H/W)

◆ 아두이노 보드(Arduino UNO) 살펴보기



- ❶ 리셋 버튼
- ❷ 전원(GND)
- ❸ 디지털 입출력 × 13
- ❹ Serial out(TX), Serial in(RX)
- ❺ TX, RX LED
- ❻ 전원 LED
- ❼ ICSP 핀
- ❽ 마이크로컨트롤러
- ❾ 아날로그 핀 × 6
- ❿ 전원(리셋, 3.3v, 5V, GND, Vin)
- ⓫ 외부전원소켓
- ⓬ 전압 레귤레이터
- ⓭ 수정발진기
- ⓮ USB 포트

03 아두이노 IDE 준비하기

◆ 아두이노 IDE 설치하기

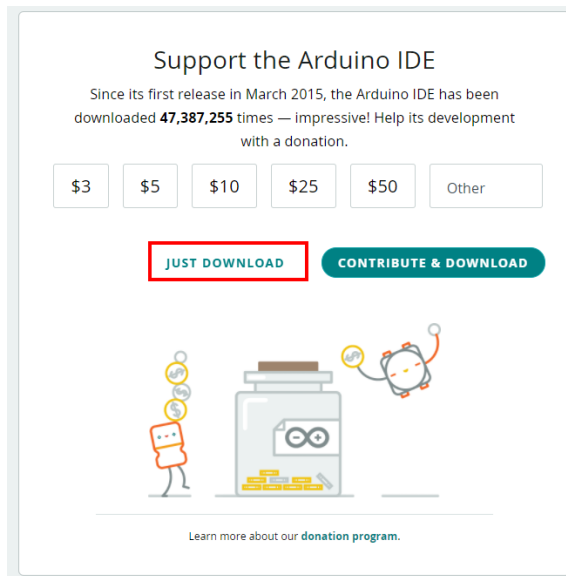
- 다운로드 - 크롬(<http://arduino.cc/en/software>)

The image shows a screenshot of the Arduino website. The top navigation bar is teal with the Arduino logo on the left and menu items: HARDWARE, SOFTWARE (highlighted with a red box), DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. Below the navigation bar, there are two main sections. The first section is titled 'Arduino Web Editor' and includes a description: 'Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.' It features two buttons: 'CODE ONLINE' and 'GETTING STARTED'. To the right of this section is a small image of an Arduino board and a laptop. The second section is titled 'Downloads' and features a large teal box for 'Arduino IDE 1.8.13'. This box contains the Arduino logo, the version number, a description of the IDE, and a 'GETTING STARTED' link. To the right of this box is a 'DOWNLOAD OPTIONS' section with a red box around the 'Windows Win 7 and newer' option. Other options include 'Windows app Win 8.1 or 10', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM 32 bits', 'Linux ARM 64 bits', and 'Mac OS X 10.10 or newer'. There is also a 'Release Notes' link and a 'Checksums (sha512)' link.

03 아두이노 IDE 준비하기

◆ 아두이노 IDE 설치하기

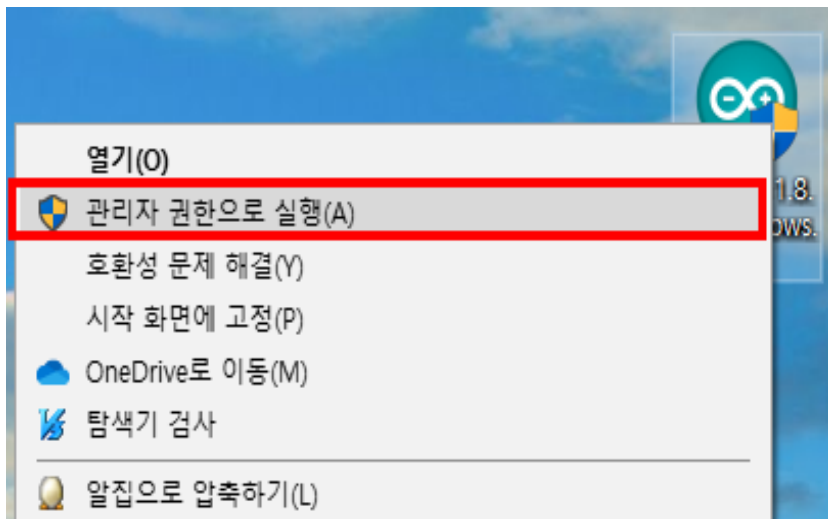
- 『JUST DOWNLOAD』 클릭(다운로드)



03 아두이노 IDE 준비하기

◆ 아두이노 IDE 설치하기

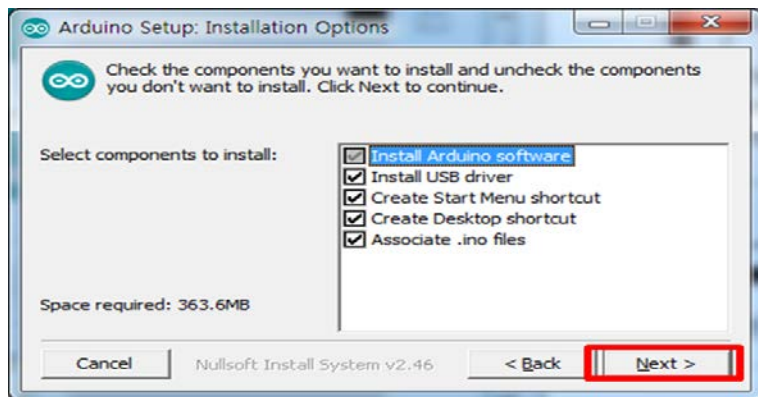
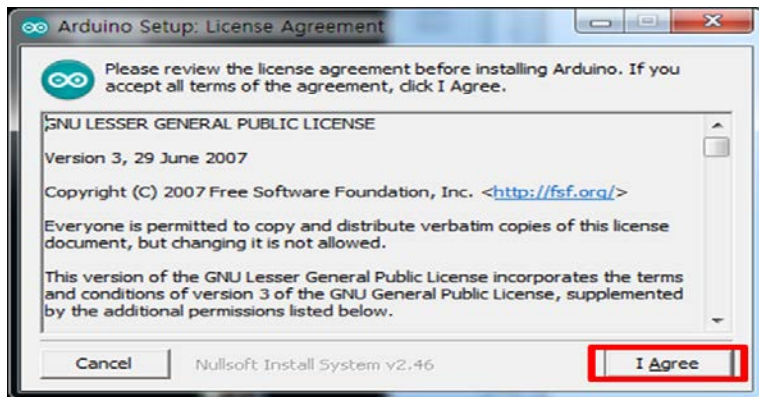
- 관리자 권한으로 실행(설치) – 아두이노 설치 파일 우클릭



03 아두이노 IDE 준비하기

◆ 아두이노 IDE 설치하기

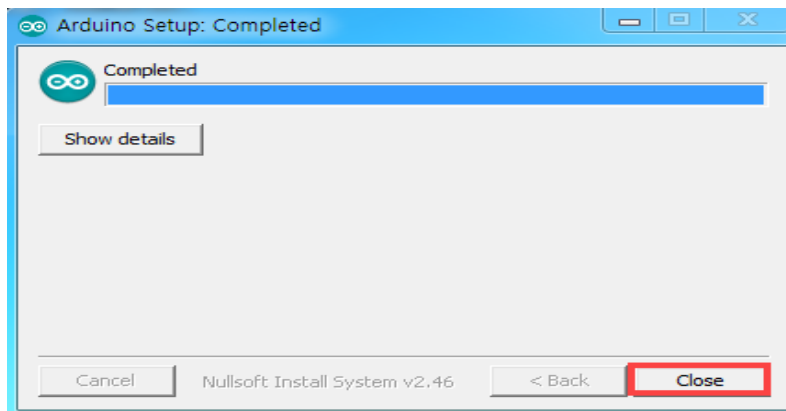
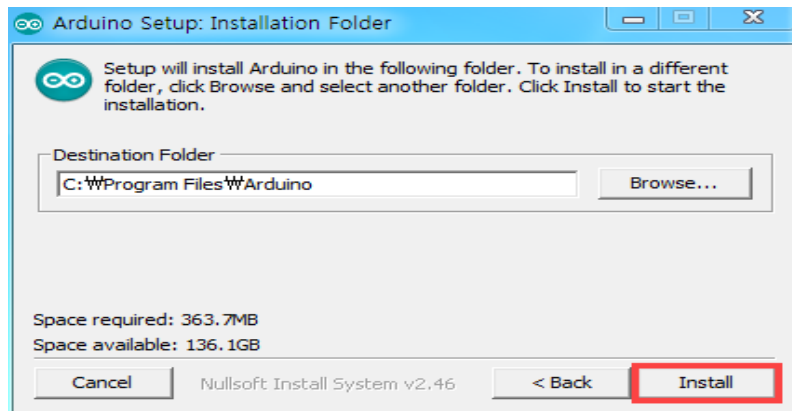
- 라이선스 동의 → [I Agree] 클릭
- 모든항목 선택 → [Next] 클릭



03 아두이노 IDE 준비하기

◆ 아두이노 IDE 설치하기

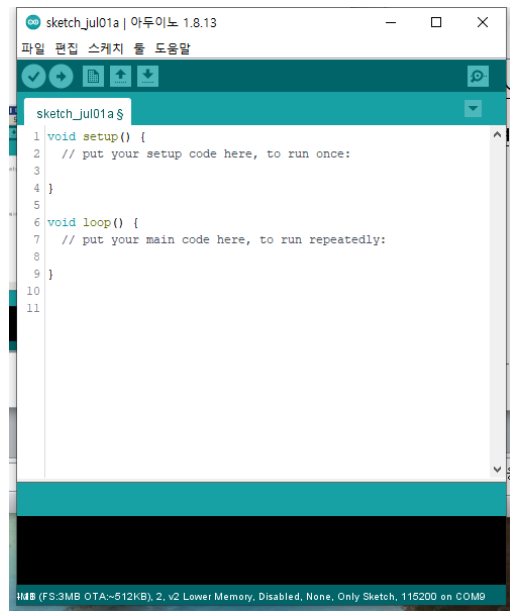
- 아두이노 셋업 → [Install] 클릭(장치 소프트웨어 설치)
- Completed → [Close] 클릭(설치 완료)



03 아두이노 IDE 준비하기

◆ 아두이노 IDE 설치하기

- 아두이노 아이콘 더블 클릭 실행(바탕화면)

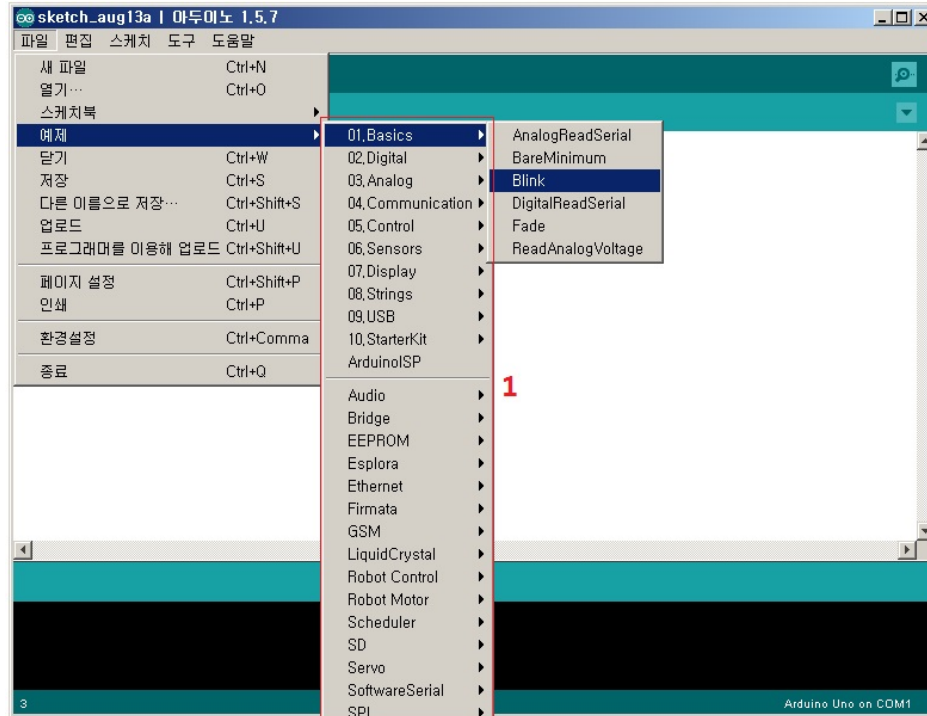
A screenshot of the Arduino IDE interface. The window title is "sketch_jul01a | 아두이노 1.8.13". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar contains icons for file operations and execution. The main editor area shows a sketch with the following code:

```
sketch_jul01a $
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
11
```

The status bar at the bottom indicates "1MB (FS:3MB OTA~512KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM9".

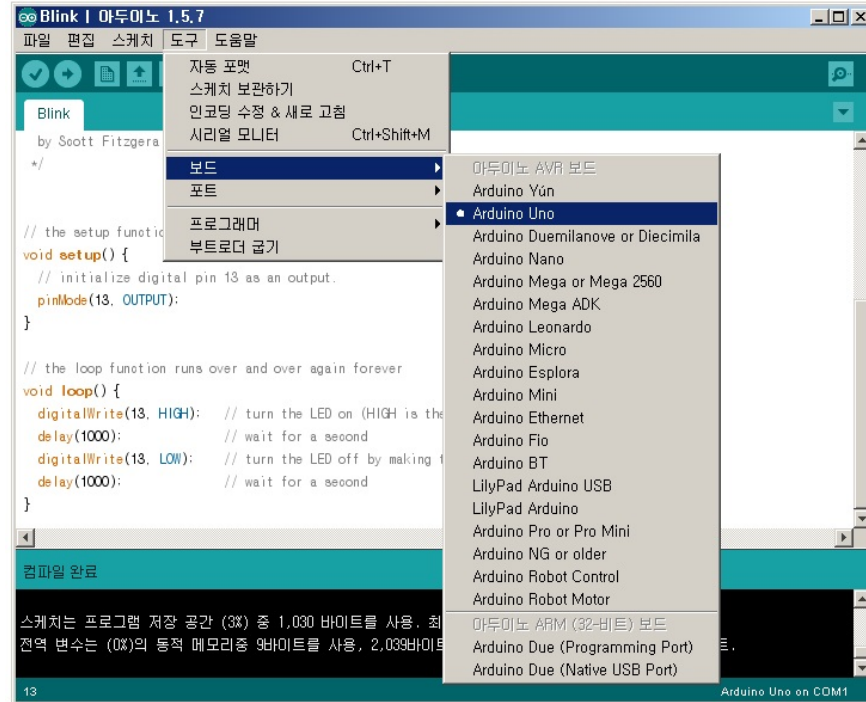
03 아두이노 IDE 준비하기

◆ Blink 예제 불러오기



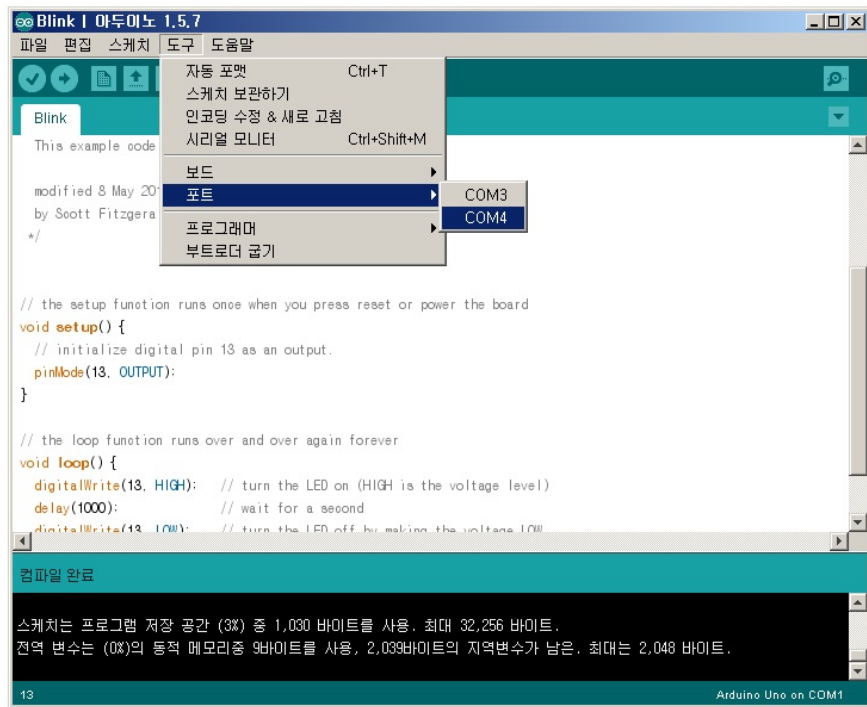
03 아두이노 IDE 준비하기

◆ Blink 예제 보드 세팅(아두이노 우노 보드 PC와 시리얼 연결)



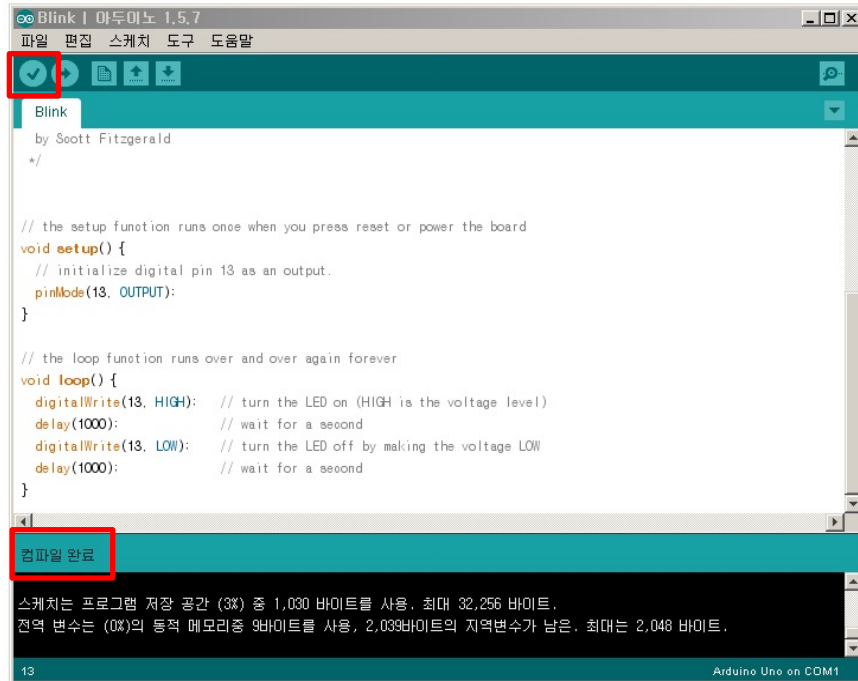
03 아두이노 IDE 준비하기

◆ Blink 예제 포트 세팅(아두이노 우노 보드 PC와 시리얼 연결)



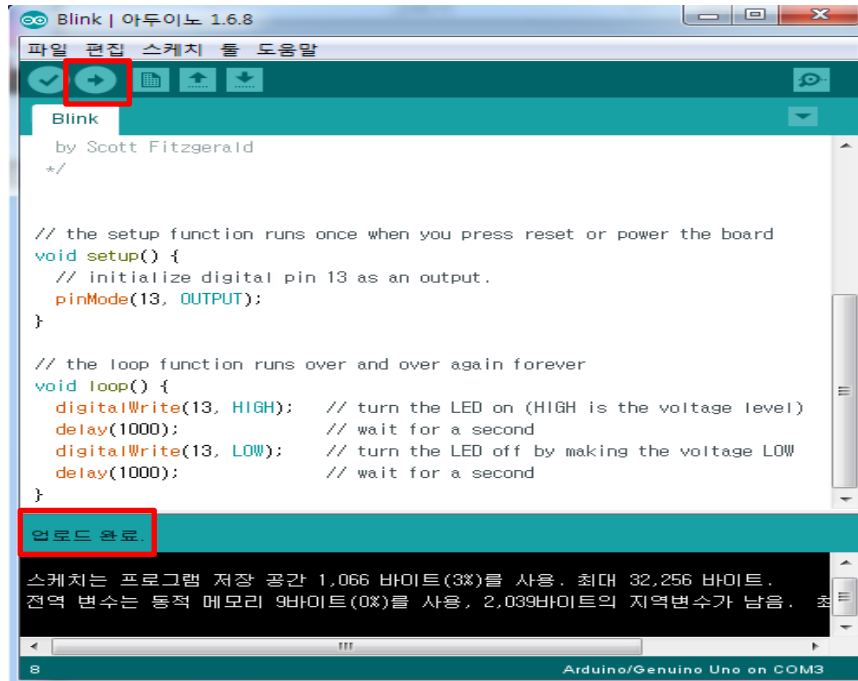
03 아두이노 IDE 준비하기

◆ Blink 예제 컴파일



03 아두이노 IDE 준비하기

◆ Blink 예제 업로드(보드 중앙의 TX/RX LED 깜빡)



04 아두이노 스케치의 기본 구성

◆ 스케치의 기본 구성

```

Blink
-----
//
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/
//
int led = 13;

void setup() {
  // initialize digital pin 13 as an output.
  pinMode(led, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

주석

- 스케치의 이름, 목적, 갱신 내용 등의 정보
- 항상 정확히 기재하는 습관

변수, 상수 라이브러리 등의 정의

- 스케치에서 사용할 전역 변수, 상수, 라이브러리 등 사전 설정이 필요한 것들 정의

설정 루틴(setup)

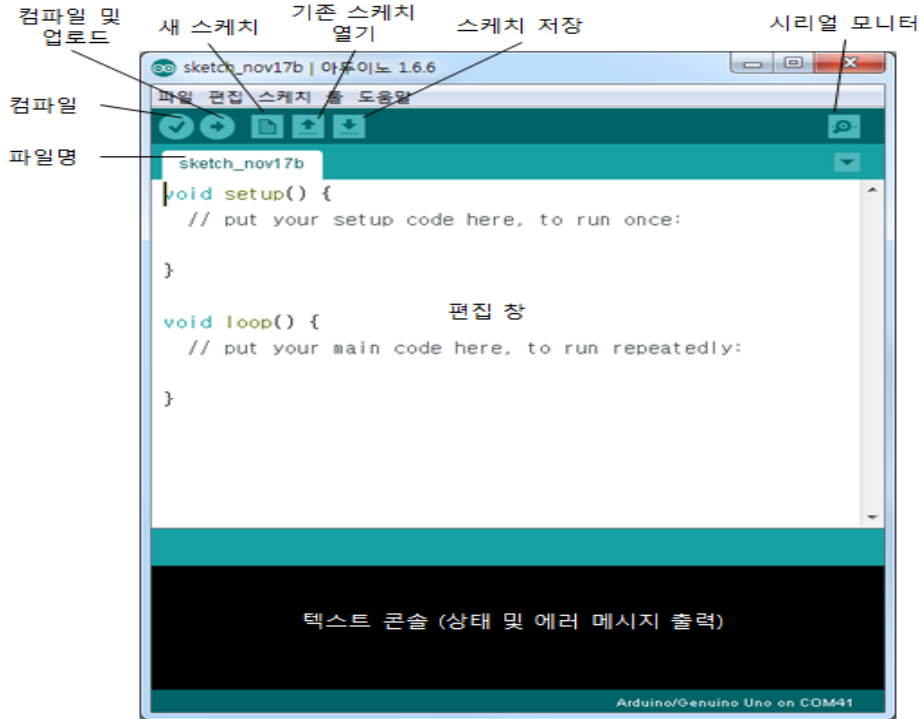
- 설정루틴에서는 각 핀의 인풋과 아웃풋 및 시리얼 통신 등을 설정
- Arduino가 동작할 때 최초 한 번만 실행

반복 루틴(loop)

- 무한 반복되는 루틴으로 C언어의 'main()' 함수와 동일한 기능
- 동작에 관한 스케치는 이 부분에서 작성

04 아두이노 스케치의 기본 구성

◆ 아두이노 IDE 화면 구성



파일명

- 새 파일을 생성시키면 우선 오늘 날짜로 표시

컴파일

- 작성한 스케치를 Arduino가 인식할 수 있는 언어로 변경

컴파일 및 업로드

- 컴파일과 업로드를 연속적으로 수행(에러가 없으면 자동 업로드)

새 스케치

- 새로운 스케치 작성

기존 스케치 열기

- 저장되어 있던 스케치 파일 열기

스케치 저장

- 현재의 스케치 저장

시리얼 모니터

- 시리얼 통신 상태를 볼 수 있는 창(Windows의 하이퍼터미널)

편집 창

- 스케치를 작성하고 편집

텍스트 콘솔

- Arduino의 상태 및 에러 메시지 출력

04 아두이노 스케치의 기본 구성

◆ 아두이노 라이브러리

- 다양한 센서, 모터, 모듈을 쉽게 사용할 수 있도록 지원
- 일반적으로 압축 파일(.ZIP)로 배포
- 헤더 파일 include 한 뒤 초기화 해서 사용

04 아두이노 스케치의 기본 구성

◆ 라이브러리 폴더

- 외부 라이브러리

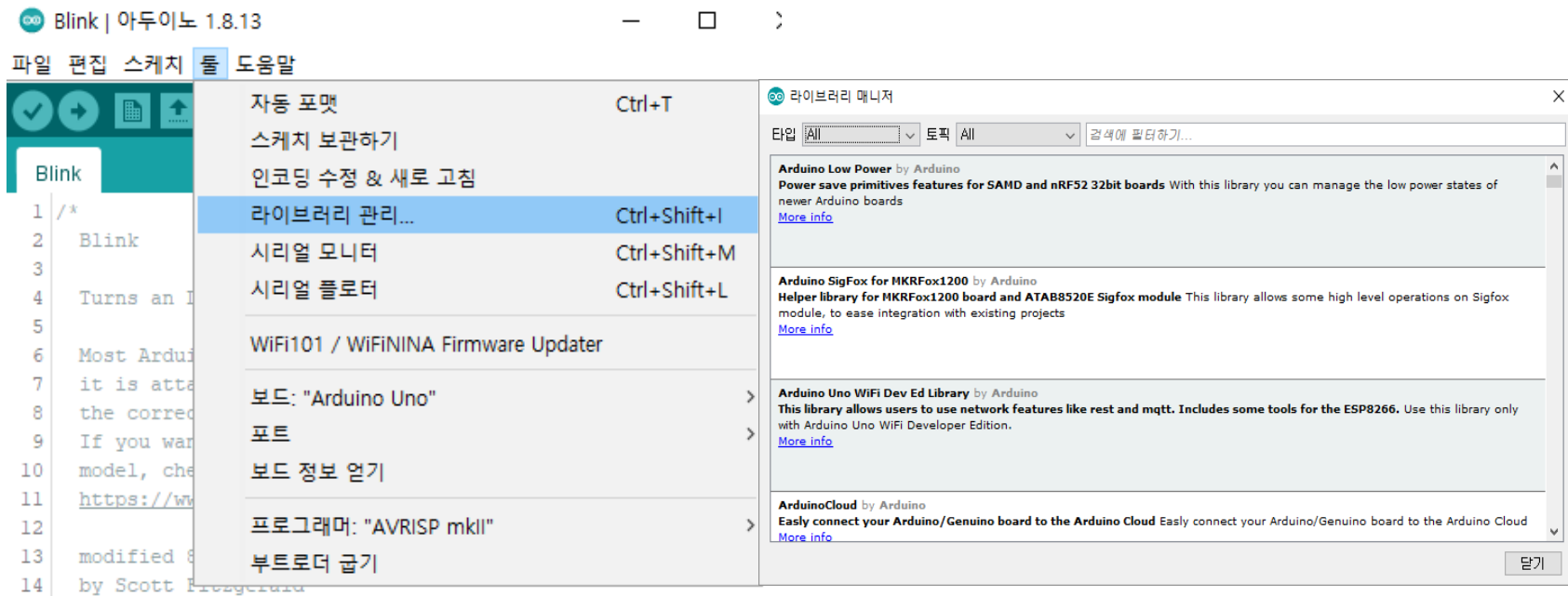
- C:\Program Files(x86)\Arduino\libraries

- 내부 라이브러리(기본 라이브러리)

- C:\Program Files(x86)\Arduino\hardware\arduino\avr\libraries

04 아두이노 스케치의 기본 구성

◆ 라이브러리 매니저(라이브러리 추가)



04 아두이노 스케치의 기본 구성

◆ 숨겨진 main 함수

- setup, loop 함수는 main 함수에서 호출

```
int main(void)
{
    init();

    initVariant();

#if defined(USBCON)
    USBDevice.attach();
#endif

    setup();

    for (;;) {
        loop();
        if (serialEventRun) serialEventRun();
    }

    return 0;
}
```

The diagram illustrates the flow of control from the `main` function to the `setup` and `loop` functions. Red arrows point from the `setup();` call in the `main` function to the `void setup() {` definition, and from the `loop();` call inside the `for` loop to the `void loop() {` definition.

04 아두이노 스케치의 기본 구성

◆ 숨겨진 main 함수

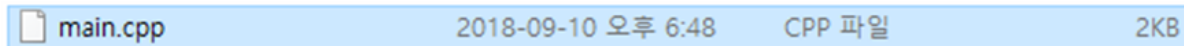
● main.cpp

*** 아두이노는 C/C++ 언어로 작성합니다. C/C++ 언어로 작성하는 프로그램의 시작은 main 함수로 시작합니다. 아두이노 스케치에서 main 함수는 숨겨져 있습니다.

main 함수의 내용을 확인하기 위해 다음 폴더로 이동합니다.



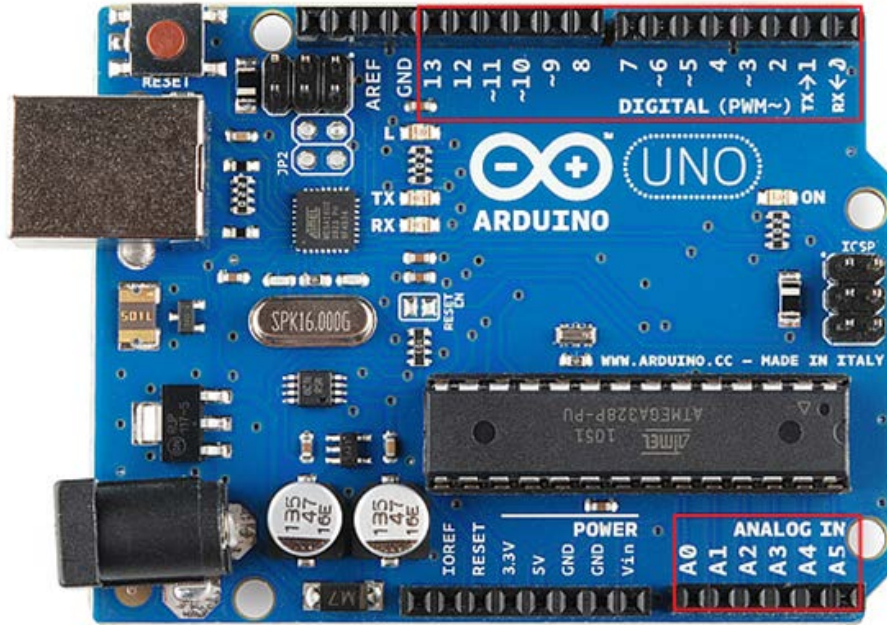
main.cpp 파일을 열어 main 함수의 내용을 확인합니다.



*** main.cpp 파일은 아두이노 스케치를 컴파일하는 과정에서 같이 포함됩니다.

05 아두이노 핀 살펴보기

◆ 아두이노 우도 보드 핀 맵



05 아두이노 핀 살펴보기

◆ 디지털 출력과 입력

- 디지털 출력

- `digitalWrite(LED, 1)` `//int LED = 3;`

- 디지털 입력

- `int buttonValue = digitalRead(BUTTON)` `//int BUTTON = 2;`

05 아두이노 핀 살펴보기

◆ 아날로그 출력(PWM 출력)

- 아날로그 출력(PWM 출력) - ~핀번호

- `analogWrite(LED, 255)` `//int LED = 3;`

◆ 아날로그 입력

- 아날로그 입력

- `int sensorValue = analogRead(SENSOR)` `//int SENSOR = 0;(A0~A5)`

06 추가 내용

◆ 전압(V)

- 전위가 높은 쪽과 낮은 쪽의 차이
- 1쿨롱(coulomb: 전하의 단위)의 전하 갖고 있는 에너지
- Arduino에서는 직류 3.3[V]와 5[V] 지원

06 추가 내용

◆ 전류(A)

- 1초당 1쿨롱의 전하가 단위 면적을 통과했을 때를 1[A]로 정의
- Arduino에서는 1/1000[A] 단위(mA 사용)

◆ 저항(Ω)

- 전류의 흐름을 방해하는 정도(색 띠나 숫자로 값 표시)

06 추가 내용

◆ 브레드 보드(Bread-board)

- 시제품 제작이나 실험용 와이어를 보드에 꽂아 사용

