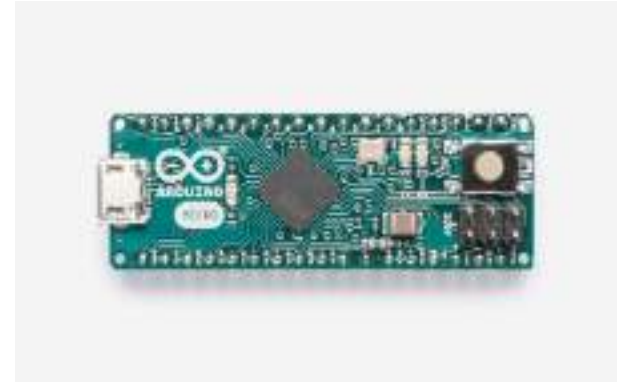


Arduino Environment

- Arduino platform
- Arduino board
- Direct programming
- Arduino schematics
- Arduino IDE
- Compiling code
- Arduino shields libraries
- Arduino basic setup
- Arduino simulators

Arduino platform

Arduino



아두이노 (Arduino)

Physical computing 개발환경

- LED나 모터, 센서와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어낼 수 있다

단일 보드 마이크로컨트롤러 (=PC와 유사)

통합 개발환경을 제공

2005년 이탈리아에서 시작

쉬움, 오픈 하드웨어

방대한 참고자료, 라이브러리



아두이노의 종류와 특징 (1)

- 정식 아두이노 보드

아두이노 UNO 보드 : 표준 보드 (R3)



아두이노 Nano : UNO의 소형화 버전



아두이노 Mega2560 : UNO의 확장 버전



아두이노 Pro/Pro mini : UNO 초소형화 버전
(3.3v/5v 버전이 별도로 존재)



아두이노의 종류와 특징 (2)

- 정식 아두이노 보드

아두이노 Duemilanoves : UNO의 구버전, 현재 단종

아두이노 **Leonardo** : UNO와 유사, 핀 배열과 개수가 약간 틀림

아두이노 MegaADK : 안드로이드 ADK 지원보드 (USB Host 기능)

아두이노 **Micro** : Leonardo 보드의 소형화 버전

아두이노 Pro micro : Nano와 유사하지만 3.3/5v 별도로 존재

아두이노 Zero : ARM core, 고성능 버전

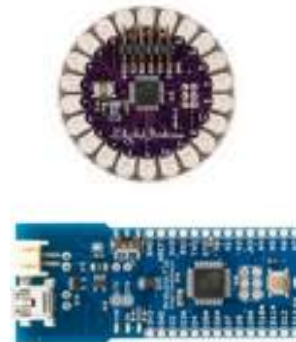
아두이노 Tre, Yun : 아두이노 + 리눅스

- 특수 용도 아두이노 보드

Lily Pad : 웨어러블용(의류용) 특화 보드

아두이노 FIO (Funnel IO) : 무선통신 특화 보드

(Xbee, 배터리 소켓 포함)



아두이노의 종류와 특징 (3)

- 아두이노 호환 보드

3rd Party 에서 제작한 호환성을 갖춘 Custom 보드들

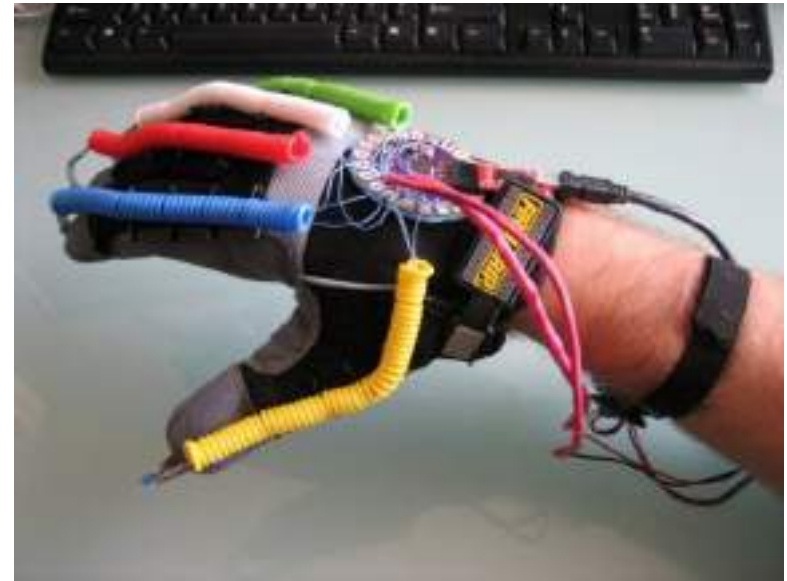


- 아두이노 쉴드

아두이노 UNO 보드 혹은 기타 보드 위에 층층이 쌓아서 사용 가능한 기능 확장용 보드들을 지칭 (예. WiFi 쉴드, Ethernet 쉴드)



Arduino applications



Arduino applications



Arduino applications



Arduino environment

- A development board
 - 8-bit microcontroller
 - Programming hardware
 - USB programming interface
 - I/O pins

Arduino environment

- A software environment
 - Cross-compiler
 - Debugger
 - Simulator
 - Programmer



Arduino environment

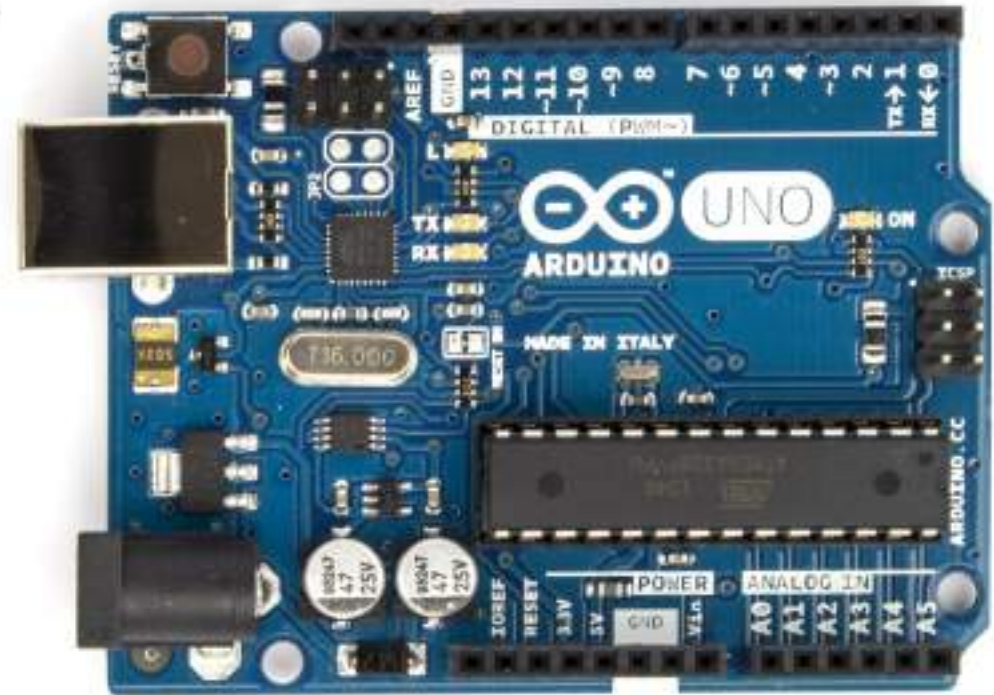
- Special-purpose “shields”
 - Daughter boards
 - Unique functionalities
 - Easy to attach
 - Good libraries provided



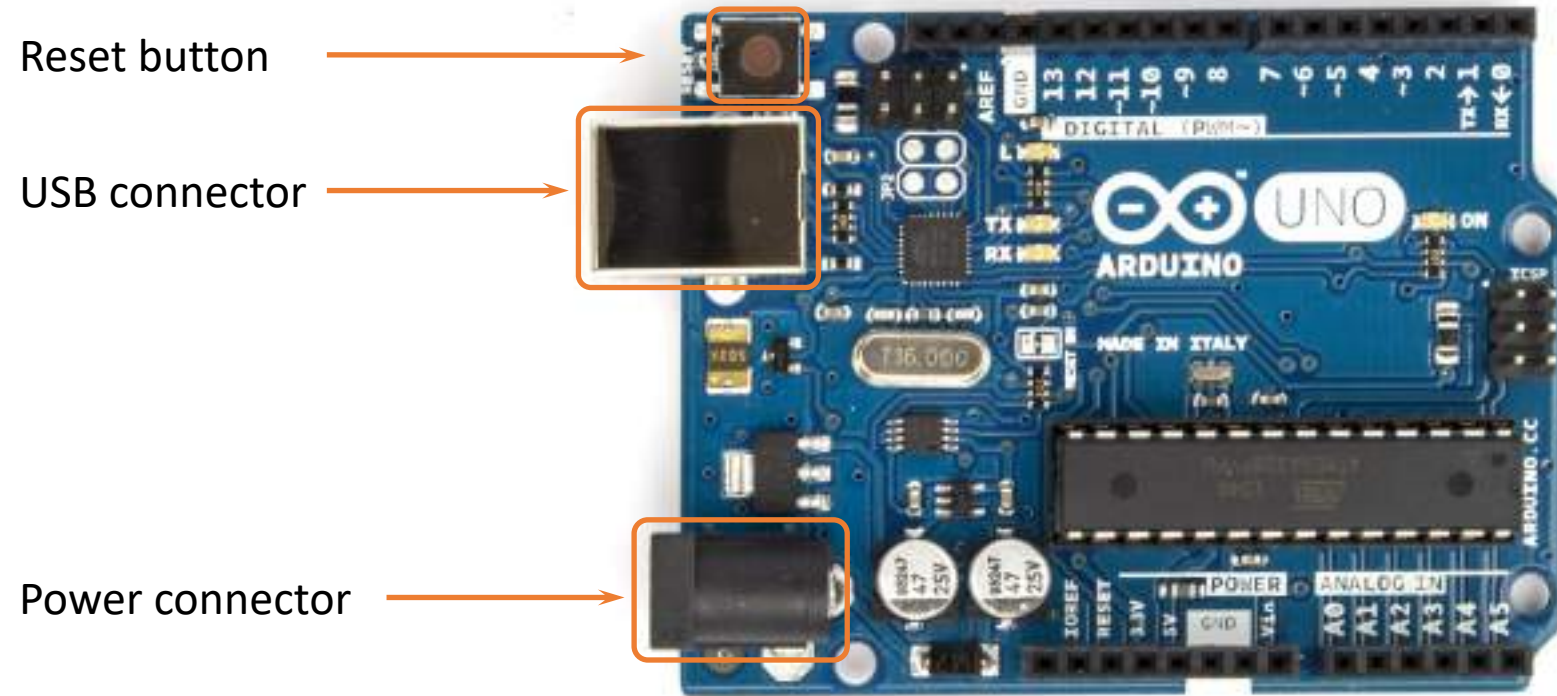
Arduino board

The Arduino development board

- Has a microcontroller and USB interface to a PC
- Large open source community

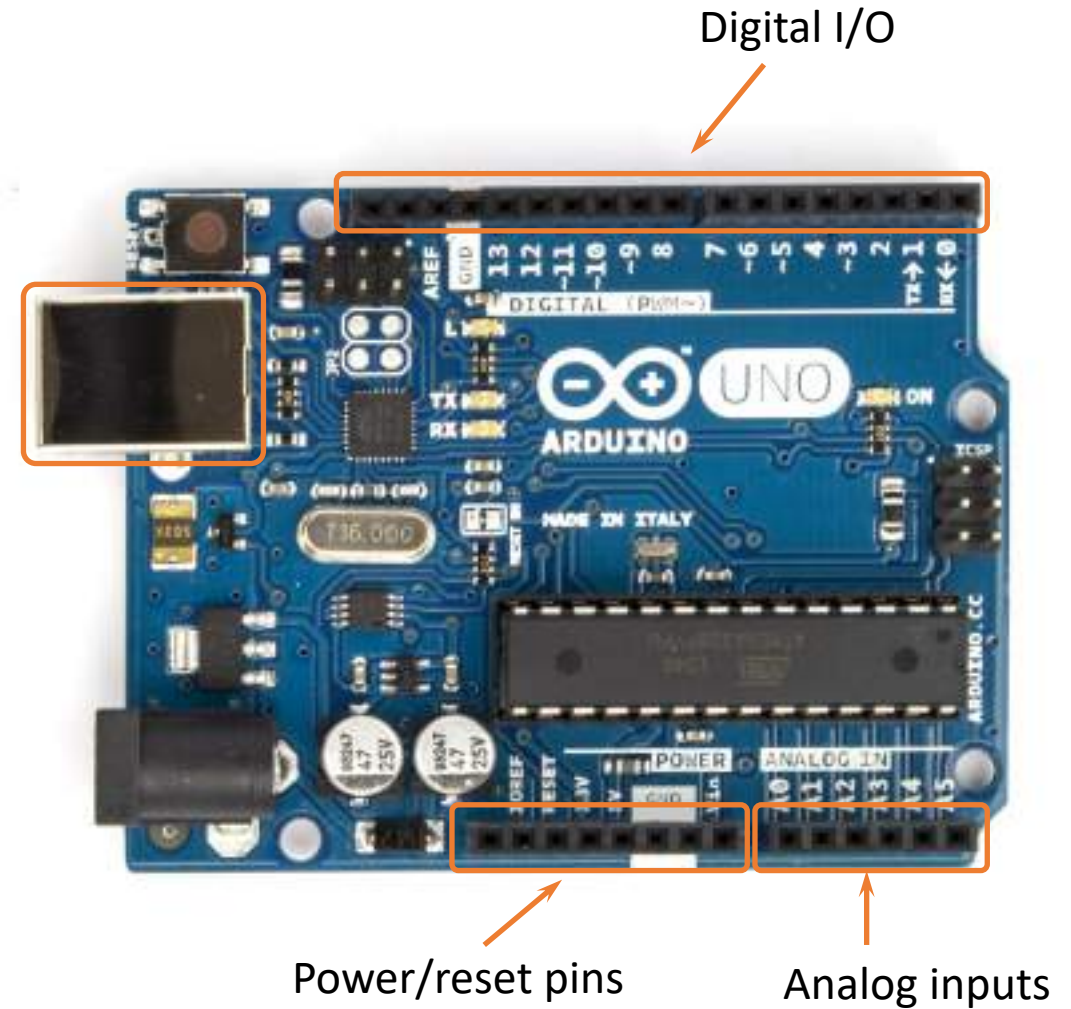


Power/Reset



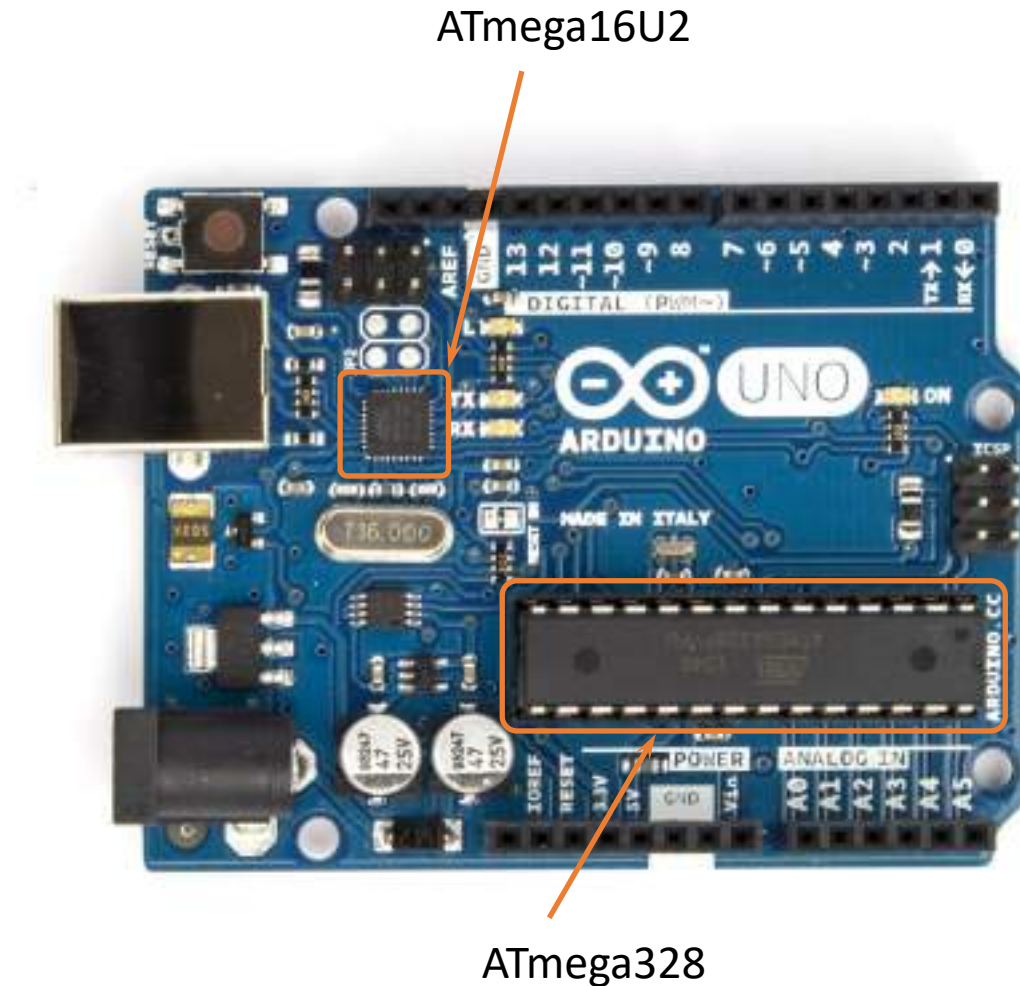
Input/Output pins

- Power/reset pins
- Digital I/O pins
- Analog input pins



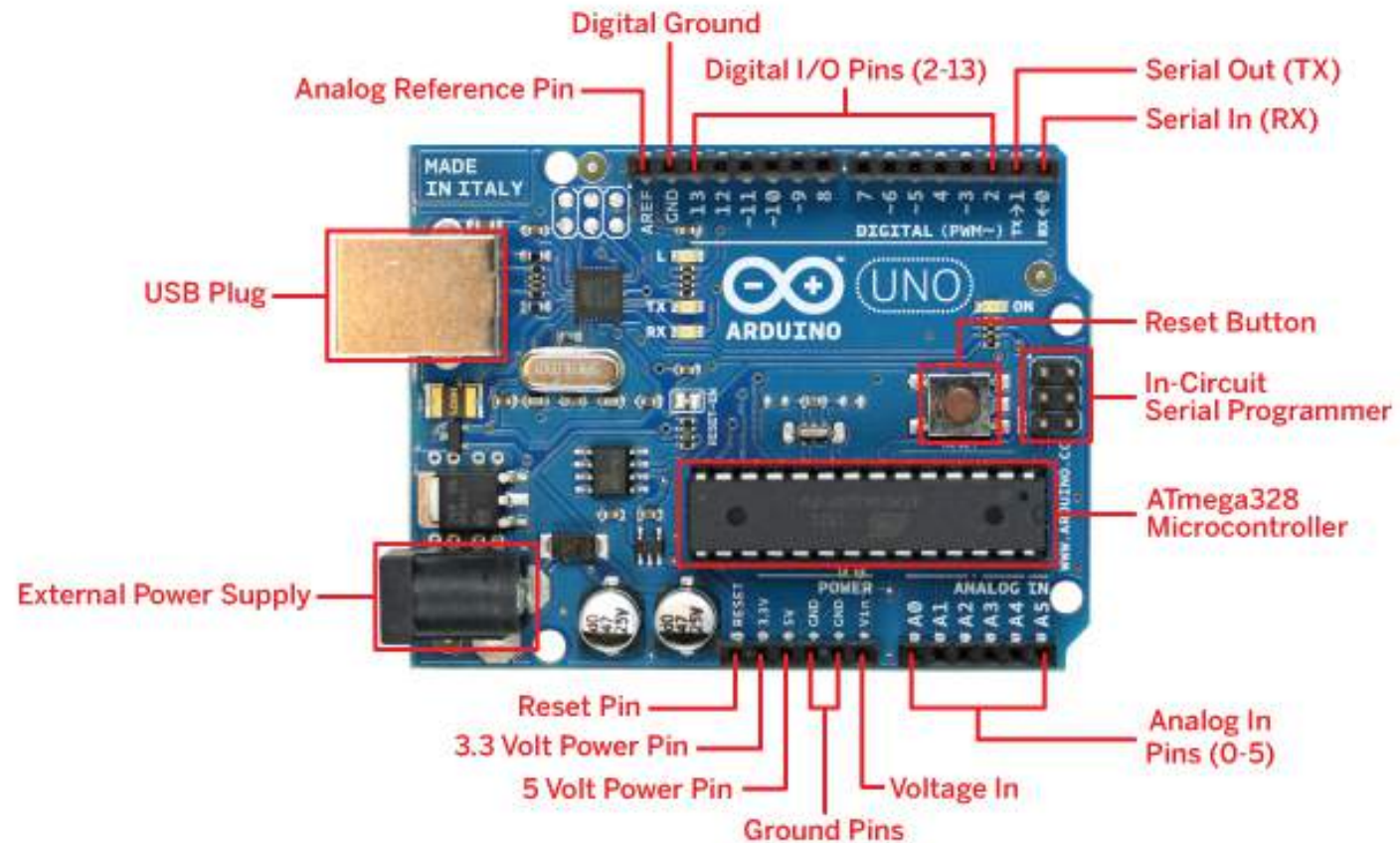
Microcontrollers

- Main processor
 - ATmega328 processor programmed by the user
- USB com. processor
 - ATmega16U2 handles USB communication



아두이노 UNO 보드

- 아두이노 UNO 보드



Microcontroller

Microcontroller	ATmega328
Operating voltage	5V
Input voltage (recommended)	7-12V
Input voltage (limits)	6-20V
Digital I/O pins	14 (of which 6 provide PWM output)
Analog input pins	6
DC current per I/O pin	40mA
DC current for 3.3V pin	50mA
Flash memory	32KB (ATmega328) (of which 0.5KB used by bootloader)
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Clock speed	16MHz

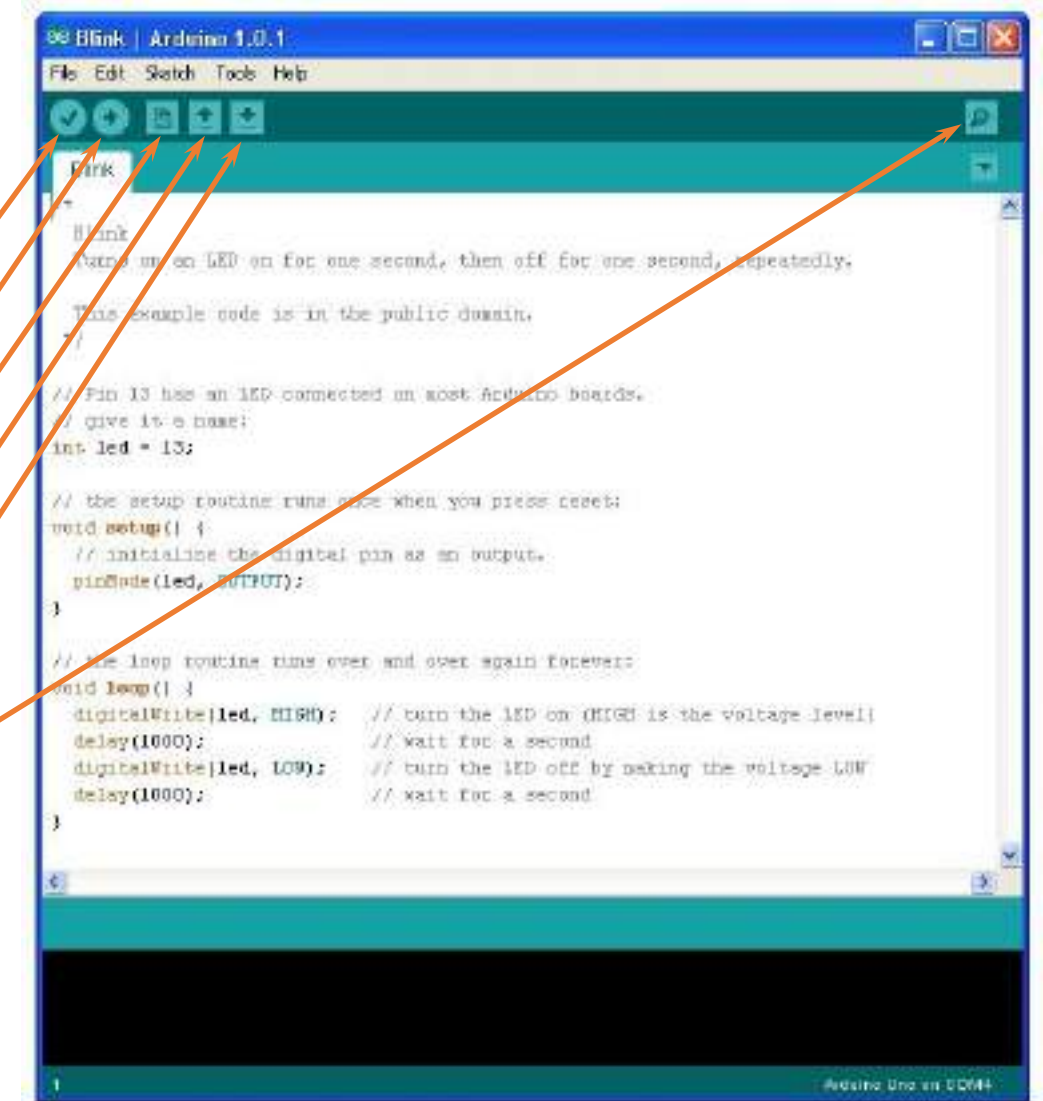
Arduino IDE(통합개발환경)

아두이노 개발환경 설치

- 아두이노 개발환경
 - 아두이노 소스 작성
 - 컴파일, 업로드
 - 디버깅 (Serial monitor)
 - 배포 : <http://arduino.cc/en/Main/Software>
 - 안정화 버전: 1.0.6 (2014. 11. 01)
 - 최신 버전: 1.5.8 BETA (최신 보드 지원, Yun, Due 등)
 - Windows installer, Windows ZIP
 - **중요!!** : 설치폴더 확인 [C:\Program Files\Arduino]

Arduino IDE

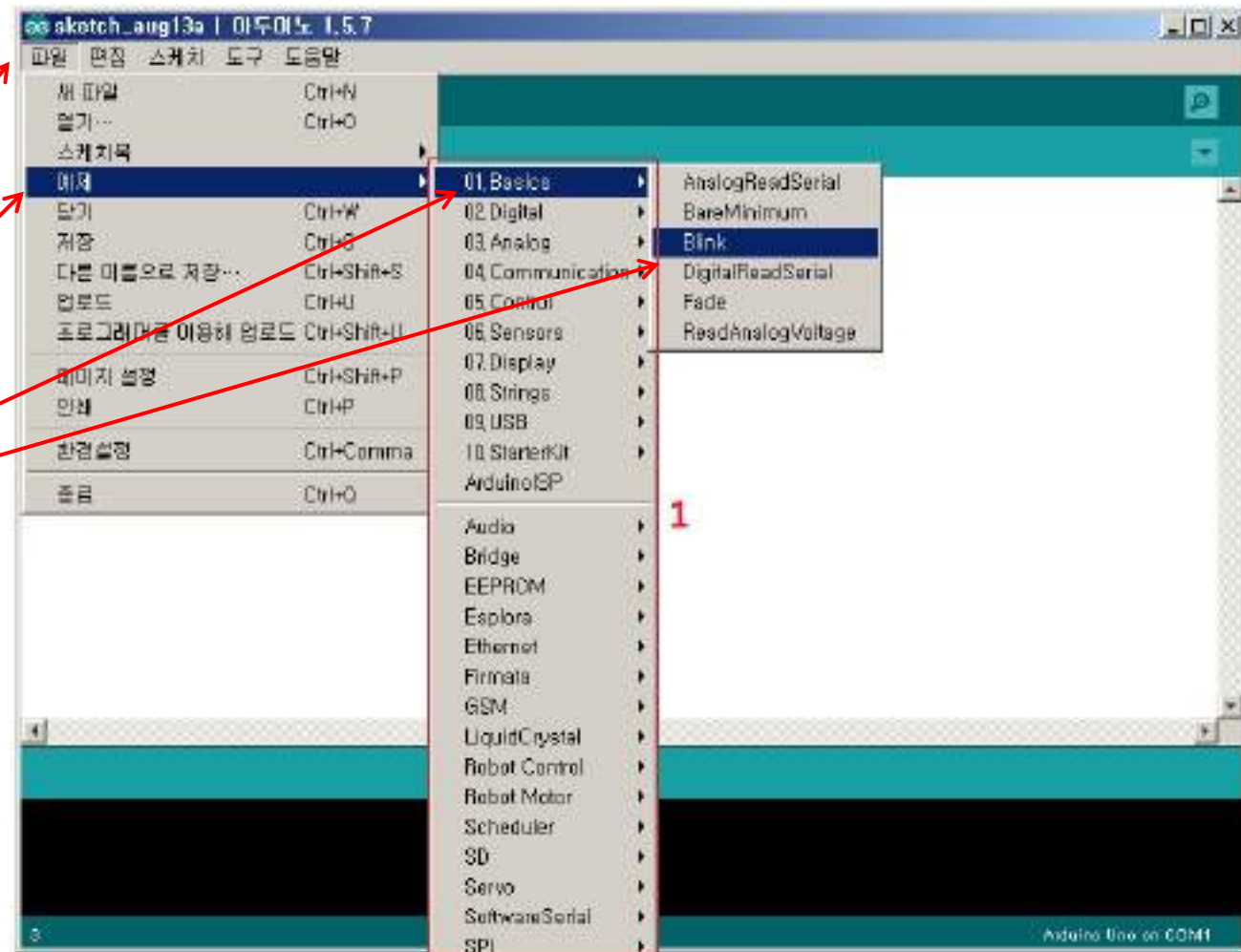
Button	Function
Verify	Compiles code, checks for errors
Upload	Compiles code, checks for errors, uploads to board
New	Creates a new sketch
Open	Opens an existing sketch
Save	Saves your sketch to a file
Serial Monitor	Opens a window to communicate with the board



Blink 예제 테스트

Blink 예제 불러오기

- 메뉴 ->
- 파일 ->
- 예제 ->
- 01. Basics ->
- Blink

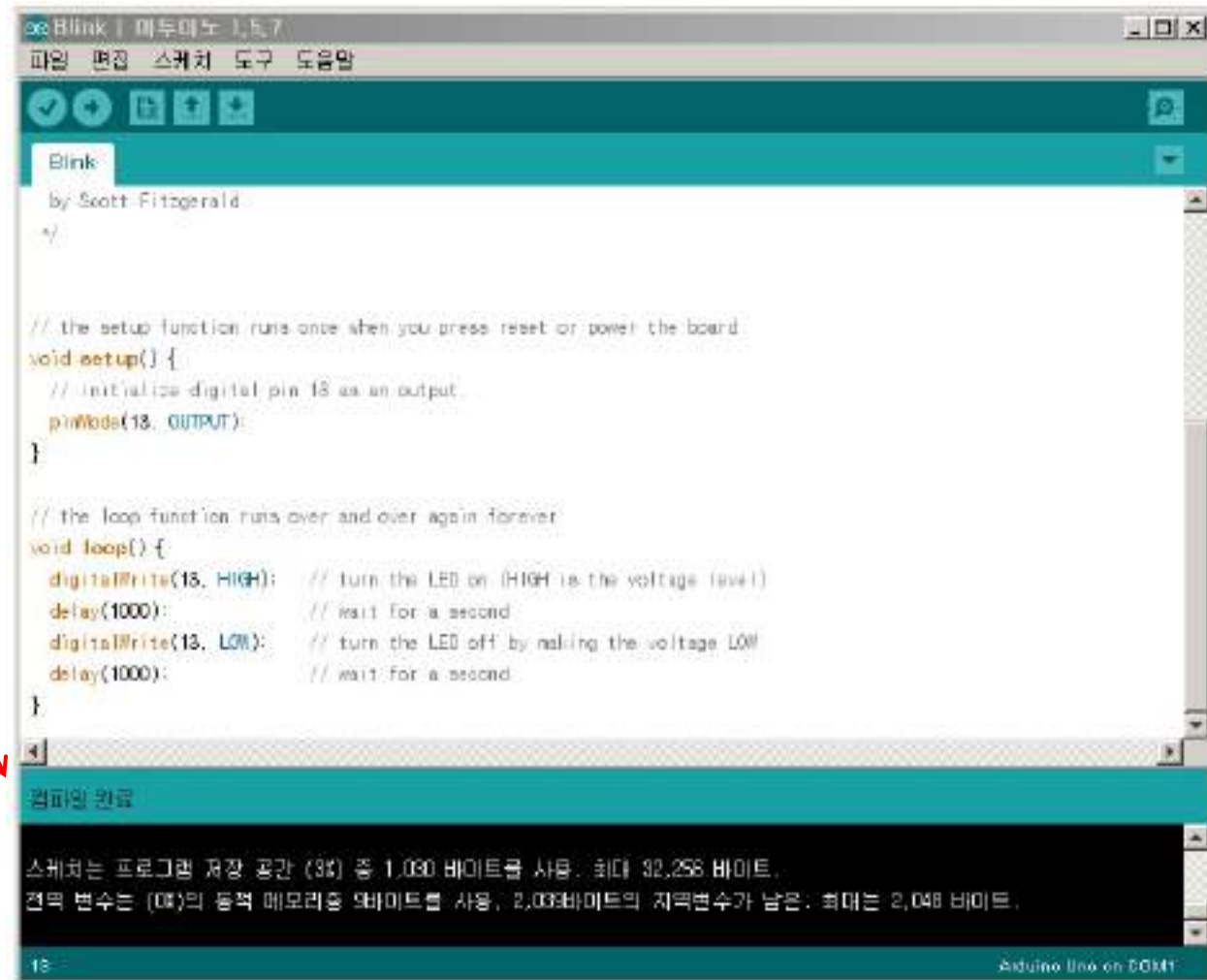


Blink 예제 테스트

컴파일

- 컴파일 버튼

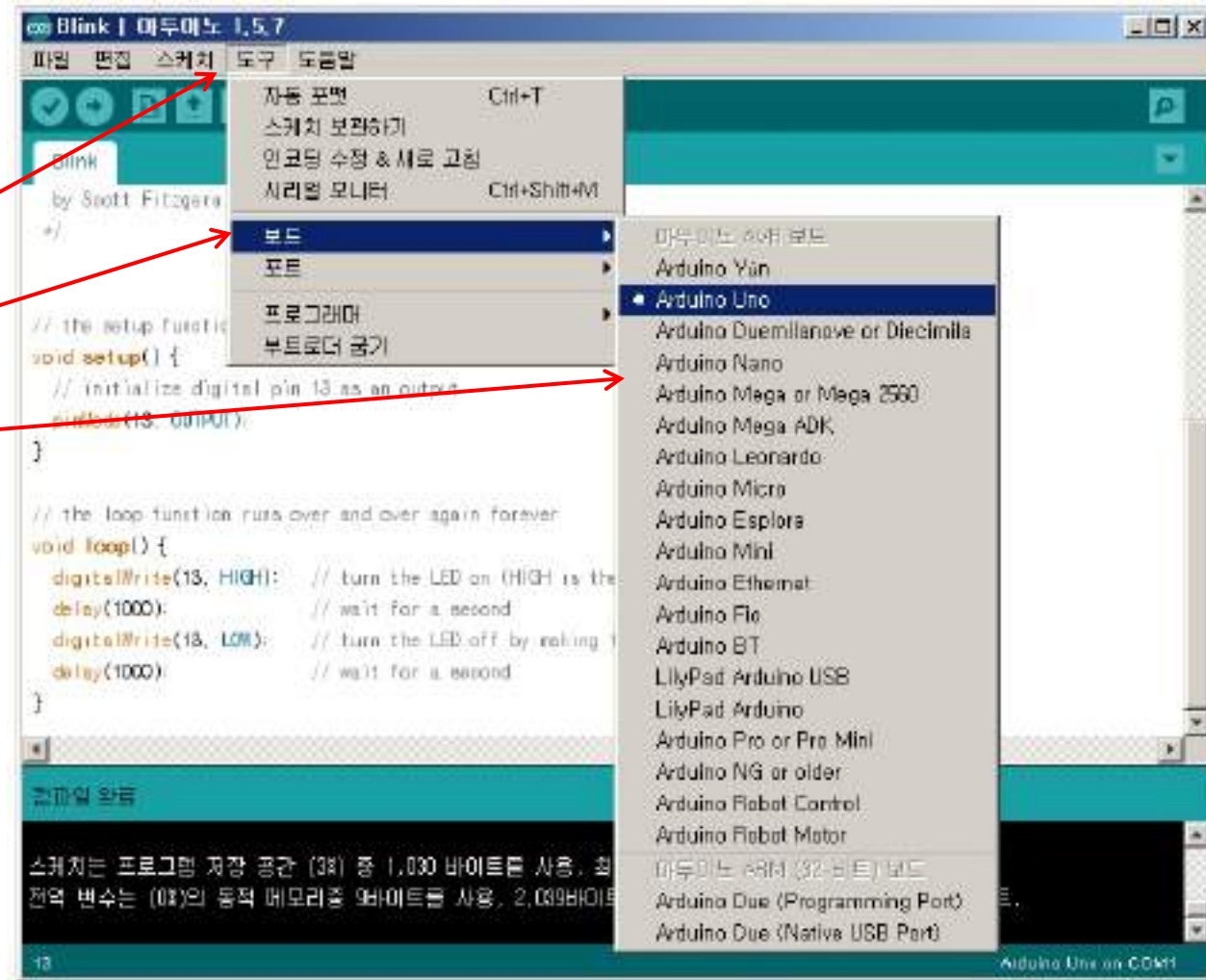
정상 컴파일 되면
아두이노를
USB 케이블로 연결



Blink 예제 테스트

보드 선택

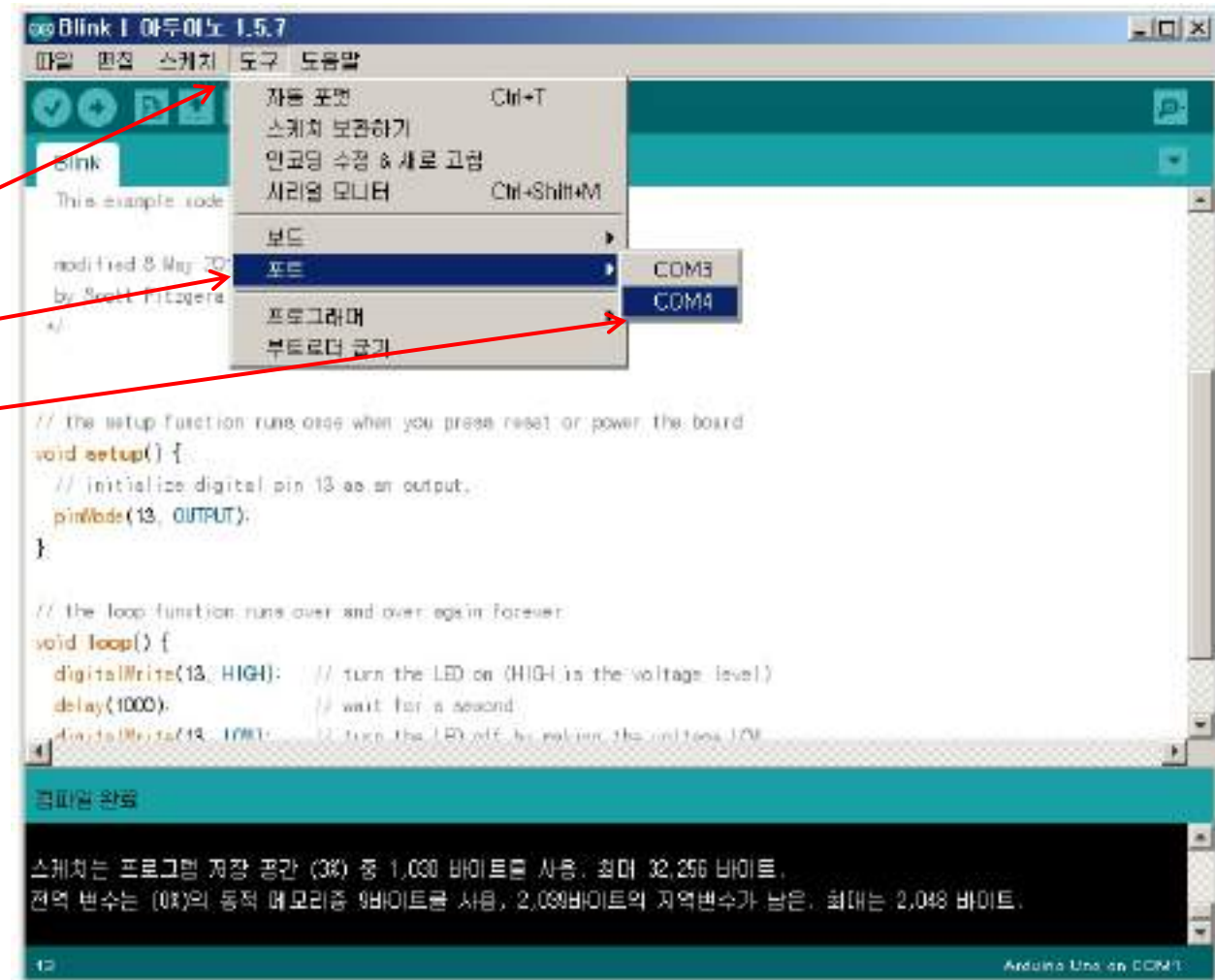
- 도구 ->
- 보드 ->
- 연결된 보드 선택



Blink 예제 테스트

COM 포트 선택

- 도구 ->
- 포트 ->
- 연결된 포트 선택



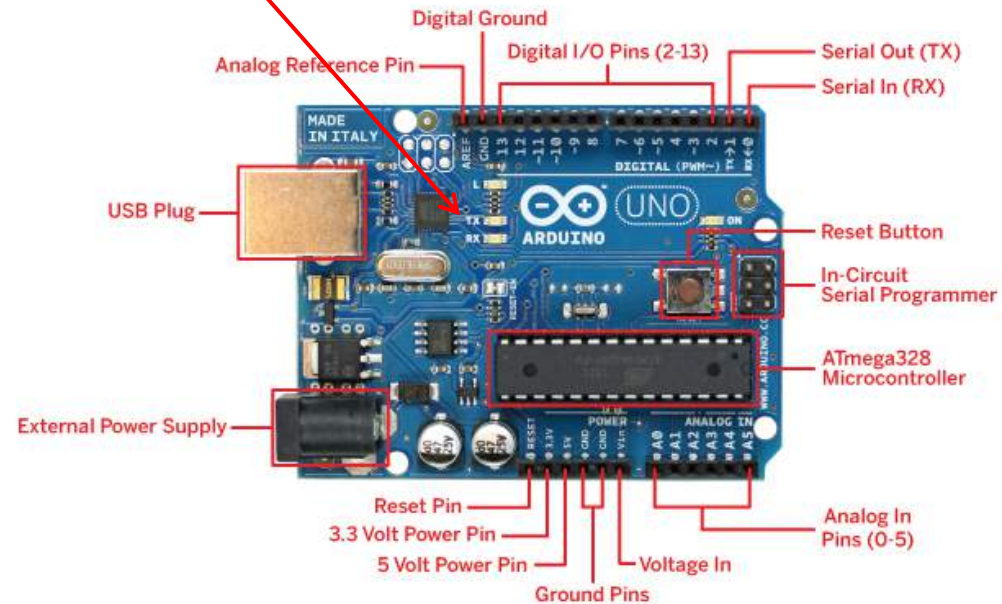
Blink 예제 테스트

- 업로드

업로드 버튼 클릭 후

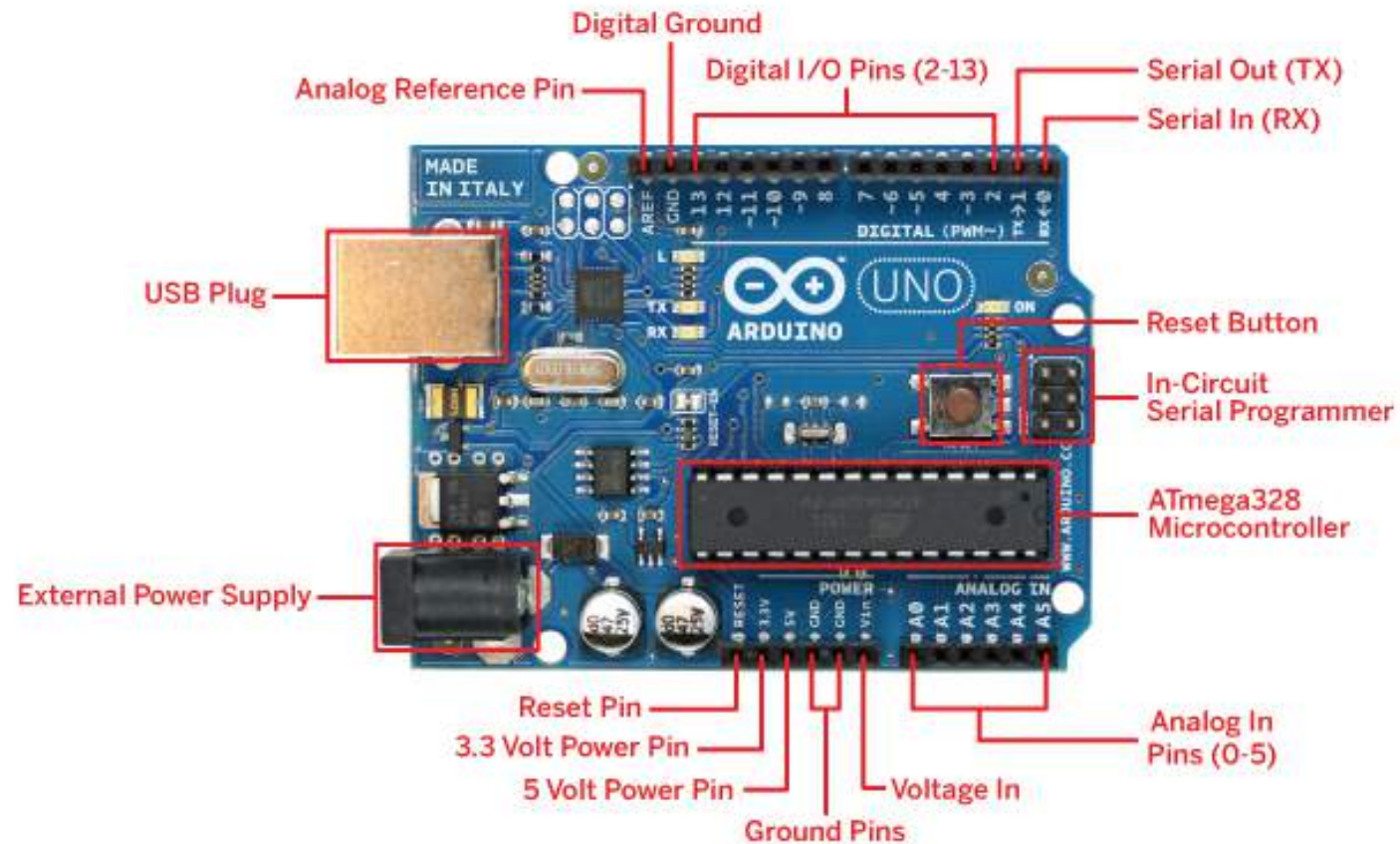
보드 중앙의 TX/RX LED가 깜빡이고

메시지 영역에 '업로드 완료' 메시지가 출력되면 정상



아두이노 UNO 보드

- 아두이노 UNO 보드



아두이노 개발 과정

- 아두이노 개발 과정
 1. 아두이노와 각종 장치들(LED, 모터, 센서...)을 연결
 2. 아두이노 개발환경(IDE)을 실행
 3. 소스코드(스케치, sketch) 작성
 4. 컴파일 및 업로드
 5. 아두이노 동작 확인
 6. 디버깅, 3~5 반복
- 아두이노 테스트를 위한 준비물
 - 아두이노 UNO 보드 (혹은 기타 보드)
 - Starter Kit (LED, 스위치, 각종 센서, 저항, 케이블 등)
 - 아두이노 개발환경 (IDE)
 - 인두기

Compiling code

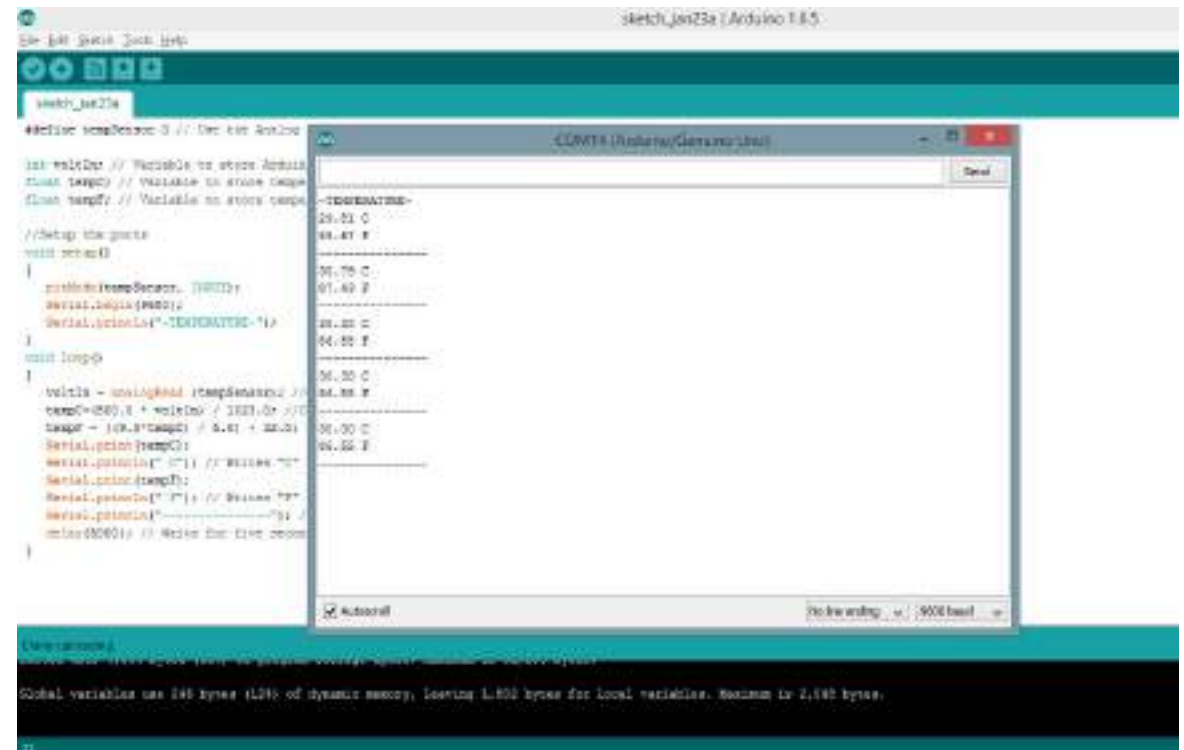
Compiling code

- Verify and Upload both compile
- Messages window will show either completion message or error messages
- Error messages will show line numbers



Serial monitor

- Displays serial data sent from the Arduino
- Allows serial data to be sent to the Arduino from the keyboard
- Library functions in the serial library



The screenshot shows the Arduino IDE interface. The top window displays a sketch named 'sketch_jan23a' with the following code:

```
#define tempSensor 3 // Use the Arduino  
  
int valTemp; // Variable to store Arduino  
float tempC; // Variable to store tempC  
float tempF; // Variable to store tempF  
  
// Setup the pins  
void setup()  
{  
  pinMode(tempSensor, INPUT);  
  Serial.begin(9600);  
  Serial.println("TEMPERATURE");  
}  
  
void loop()  
{  
  //value = analogRead (tempSensor);  
  tempC = (500.0 - valTemp) / 1000.0; //C  
  tempF = (9.0 * tempC) / 5.0 + 32.0;  
  Serial.print(tempC);  
  Serial.println(" C"); // Prints "C"  
  Serial.print(tempF);  
  Serial.println(" F"); // Prints "F"  
  Serial.println("-----");  
  delay(1000); // Wait for five seconds  
}
```

The bottom window, titled 'COM19 (Arduino/GeminiBoard)', shows the serial monitor output. It displays the temperature readings in both Celsius and Fahrenheit, separated by a horizontal line. The output is as follows:

```
TEMPERATURE-  
25.01 C  
69.01 F  
-----  
26.75 C  
80.15 F  
-----  
28.33 C  
83.00 F  
-----  
30.00 C  
86.00 F  
-----  
30.00 C  
86.00 F  
-----
```

At the bottom of the IDE, a status bar indicates: 'Global variables use 240 bytes (23%) of dynamic memory, leaving 1,800 bytes for local variables. Maximum is 2,048 bytes.'

Arduino shields and libraries

Arduino shields

- Add-on boards that interface with another device/IC
- Can be stacked directly on top of the Arduino
- Libraries exist to make interfacing simple
- Open source hardware, but most can be purchased
- Large variety of shields available
- Big advantage of the Arduino platform
- Fairly complete list is at <http://shieldlist.org>



Some Arduino shields



IMU shields:
Measures 6DOF motion
(acceleration and rotation)



Synthesizer shields:
Generates music
Connects to speaker



Ethernet shields:
Interfaces to Ethernet controllers
Builds web servers/clients

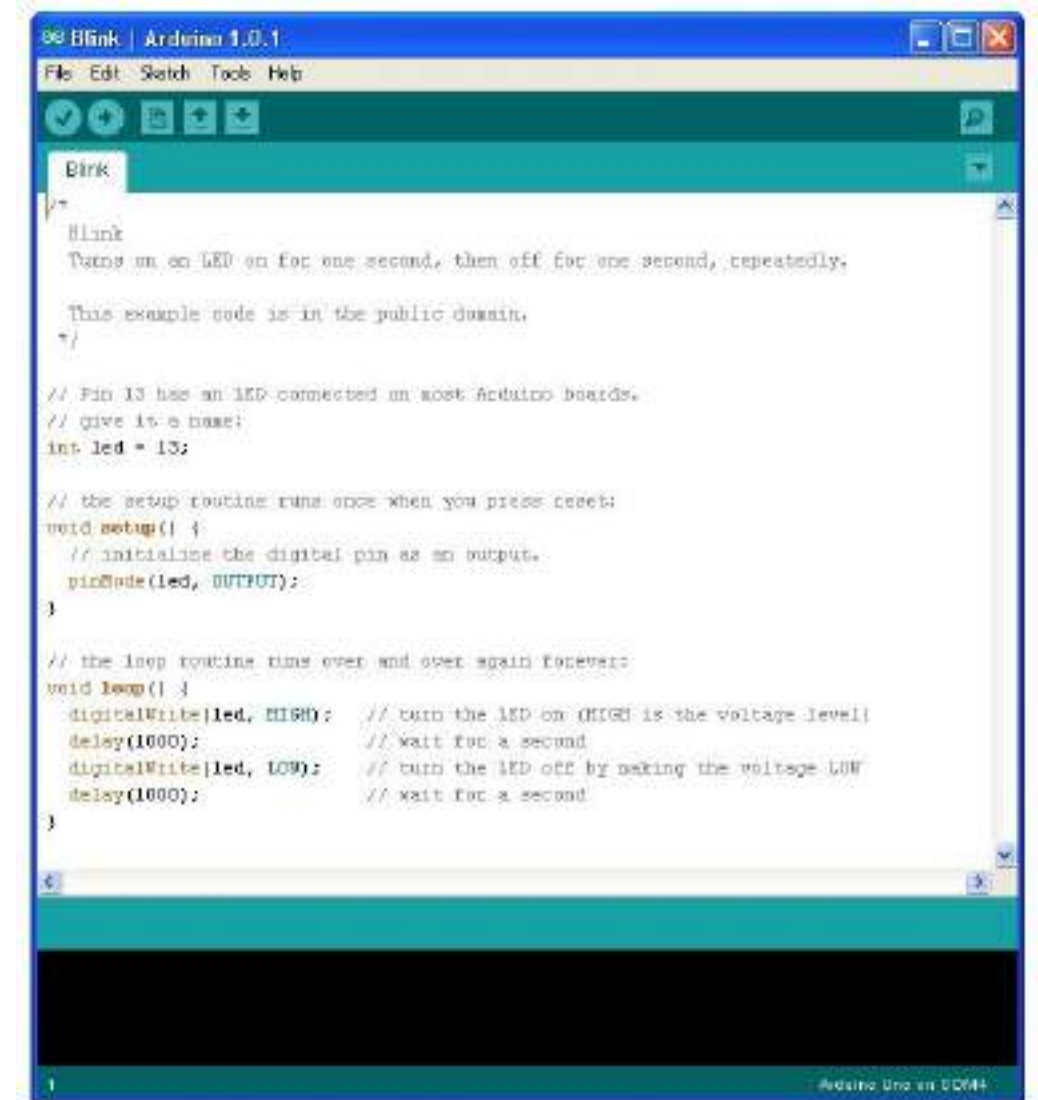
Arduino basic setup

Set up the Arduino IDE

- Download the IDE
 - Easiest to run Windows Installer (<http://www.arduino.cc>)
 - Also installs USB and other drivers
- Connect the board to your computer
 - Use USB cable
- Launch the Arduino application
 - Starts the IDE

Launch the Arduino IDE

- Open the Blink example
 - File → Examples → Basics → Blink



Run a program

- Select your Arduino
 - Tools→Board menu
- Select your serial port
 - Tools→Port menu
 - There should be only one selection (COM3, etc.)
- Upload the program with the upload button
 - This writes the program onto the Flash of the Arduino
- The LED near pin 13 of the Arduino should blink

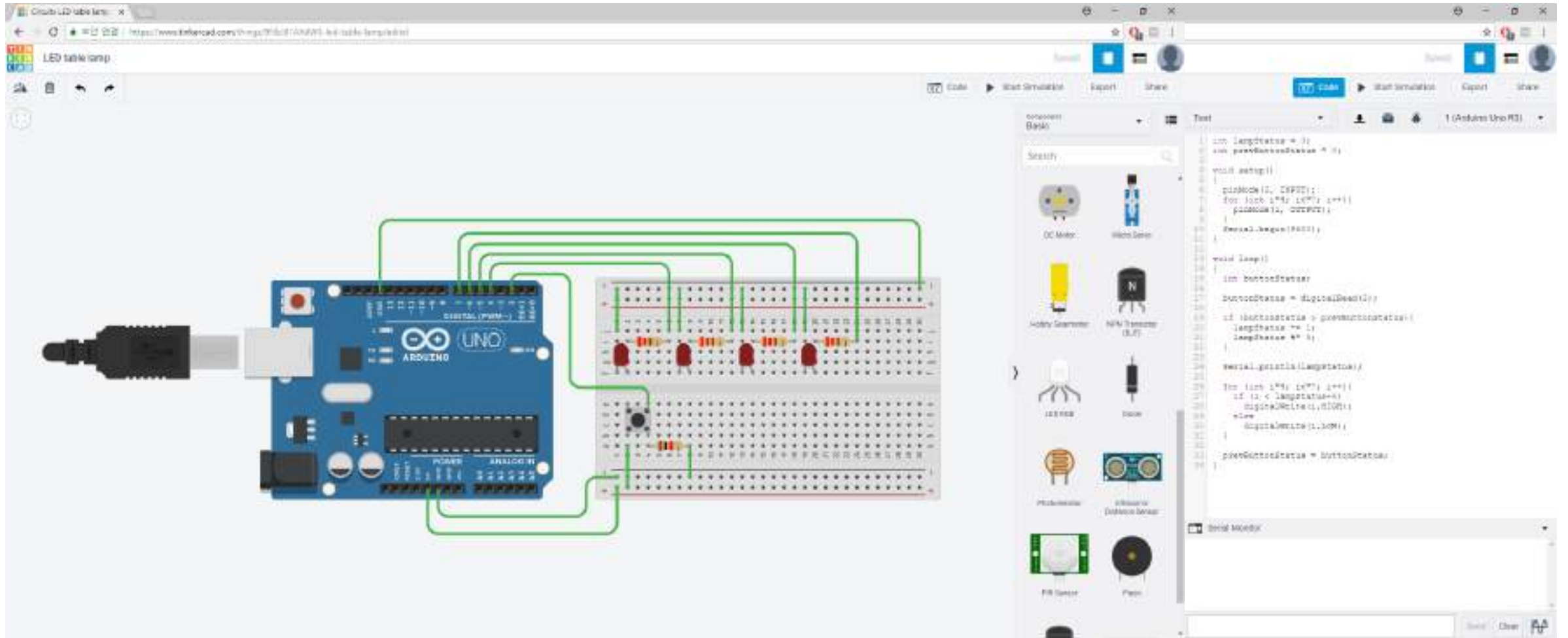
Arduino simulators

Tinkercad: Circuits

- Tinkercad

- <http://tinkercad.com>
- A free online collection of software tools that help people create and make
- Simulates Arduino circuits and programs with a bunch of electronic components, sensors and actuators
- A good way to test your idea before you actually build

Tinkercad: Circuits



Lab

Blink example

- Set up the Arduino development environment
- Load the Blink example.
- Compile, download, and run.
- Modify blink intervals and check if it really works.

Tinkercad

- Go to `http://tinkercad.com`
- Register for your own account.
- Try several Arduino starter examples.
- Modify some codes, and check if it works.