Arduino Basic Programming

부산대학교 정보·의생명공학대학
정보컴퓨터공학부

This lecture note provides the english version of Lab. and HW. slides

Arduino H/W

- ✓ CHAPTER 02. 아두이노
  - LESSON 01 아두이노 하드웨어
  - LESSON 02 아두이노 우노
- ✓ CHAPTER 03. 아두이노 메가 2560

Arduino ?

❖ Origin

- 2005년 Interaction Design Institute Ivrea (Italy) 에서 예술가와 디자이너를 위한 µC 프로젝트로 시작

❖ Arduino

- **Hardware** (ATMEL사의 AVR 시리즈 마이크로컨트롤러)와
  - AVR - 1996년 ATMEL 사에서 개발한 Harvard Architecture의 8 bits RISC µC.
    - CPU와 소용량 Flash Memory가 하나의 IC에 집적되어 있음
- **Software** (프로그램 개발을 위한 전용 라이브러리를 포함하는 개발 환경)를 함께 지칭

❖ Arduino H/W는 µC Board의 일종

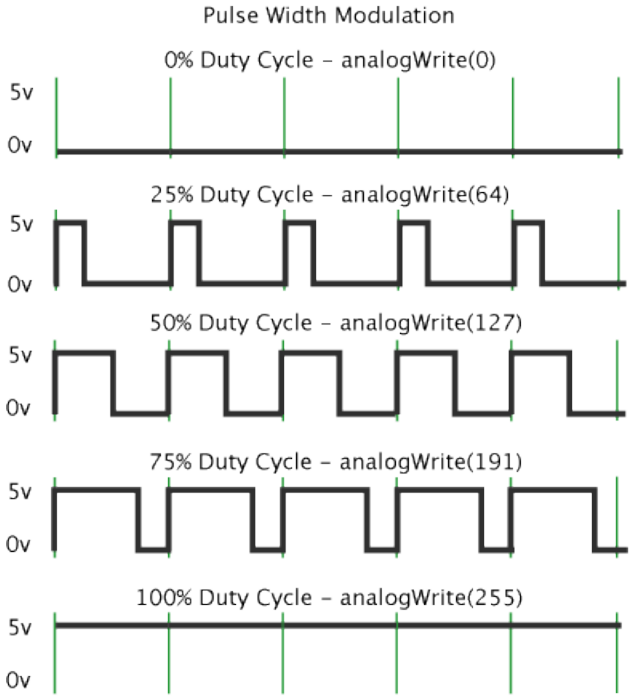
- Arduino Board라고도 함
- 마이크로컨트롤러(AVR µC) + Connector + α
- Arduino의 Hardware와 Software는 모두 오픈 소스 정책에 따라 공개되어 있음
- 마이크로컨트롤러를 사용하여 만들어진 **개발** 보드
  - "개발 보드"와 실제 응용에서 쓰이는 "마이크로컨트롤러"는 구별할 필요가 있음
  - Arduino에는 µC 외에 많은 것들이 추가 되어 있음

Arduino UNO


USB 시리얼 변환 마이크로컨트롤러
리셋 버튼
USB 커넥터
16MHz 클록
외부 전원
디지털 입출력 핀 헤더
ISP 연결 커넥터
ATmega328
아날로그 입력 핀 헤더
DIGITAL (PWM~)
POWER
ANALOG IN
UNO
ARDUINO
WWW.ARDUINO.CC – MADE IN ITALY

Arduino UNO

- ❖ Arduino Board 중 가장 기본이 되는Board
- ❖ 8비트 CPU 포함 : ATmega328
- ❖ 20개의 디지털 입출력 핀 사용 가능
  - 0번에서 19번까지 핀 단위의 번호 지정
  - 6개의 핀으로 PWM(Pulse Width Modulation) 신호 출력 가능
    - 3, 5, 6, 9, 10, 11번 핀
- ❖ 6개의 아날로그 입력 핀 사용 가능
  - 14번에서 19번까지의 디지털 입력 핀과 동일
  - 10비트 ADC (Analog to Digital Converter) 사용 (0~1023의 양자화된 값)
  - DAC(Digital to Analog Converter)는 포함되어 있지 않으므로 **아날로그 값 출력은 불가능**



Arduino UNO 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	-

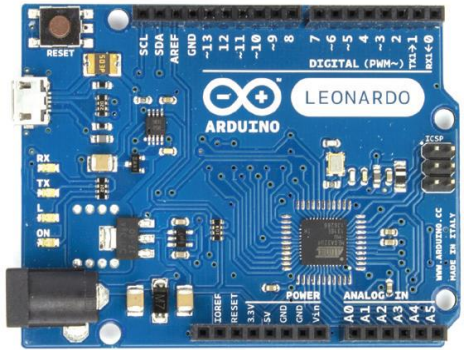
Arduino Board의 종류 – 핀 헤더 (Pin Header) 유무

Arduino
UNO

Arduino
Nano

ATmega328

Arduino
LEONARDO



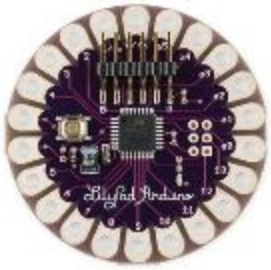
Arduino
Micro

ATmega32u4

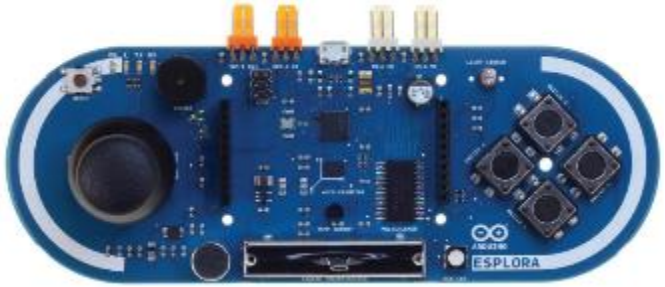
핀 헤더 있음 : 교육&개발

핀 헤더 없음 : 제품 적용

특수 목적용 Arduino Board



LILYPAD : Wearable



Esplora : Integrated sensors and actuators

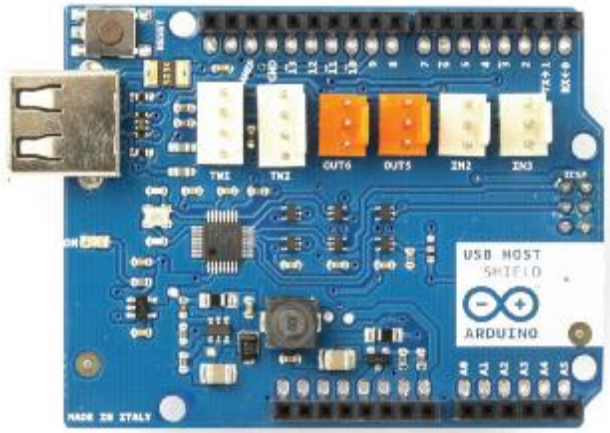
Mega ADK :
For Android Accessory Development

Arduino ROBOT (Retired)

Shield

- ❖ Arduino Board의 기능 확장을 위한 하드웨어
- ❖ 다양한 Arduino 공식 Shield 및 3rd party 업체의 호환 Shield 존재
  - 호환 Shield는 Arduino의 오픈 하드웨어 정책에 따라 가능

WiFi Shield



USB Host Shield

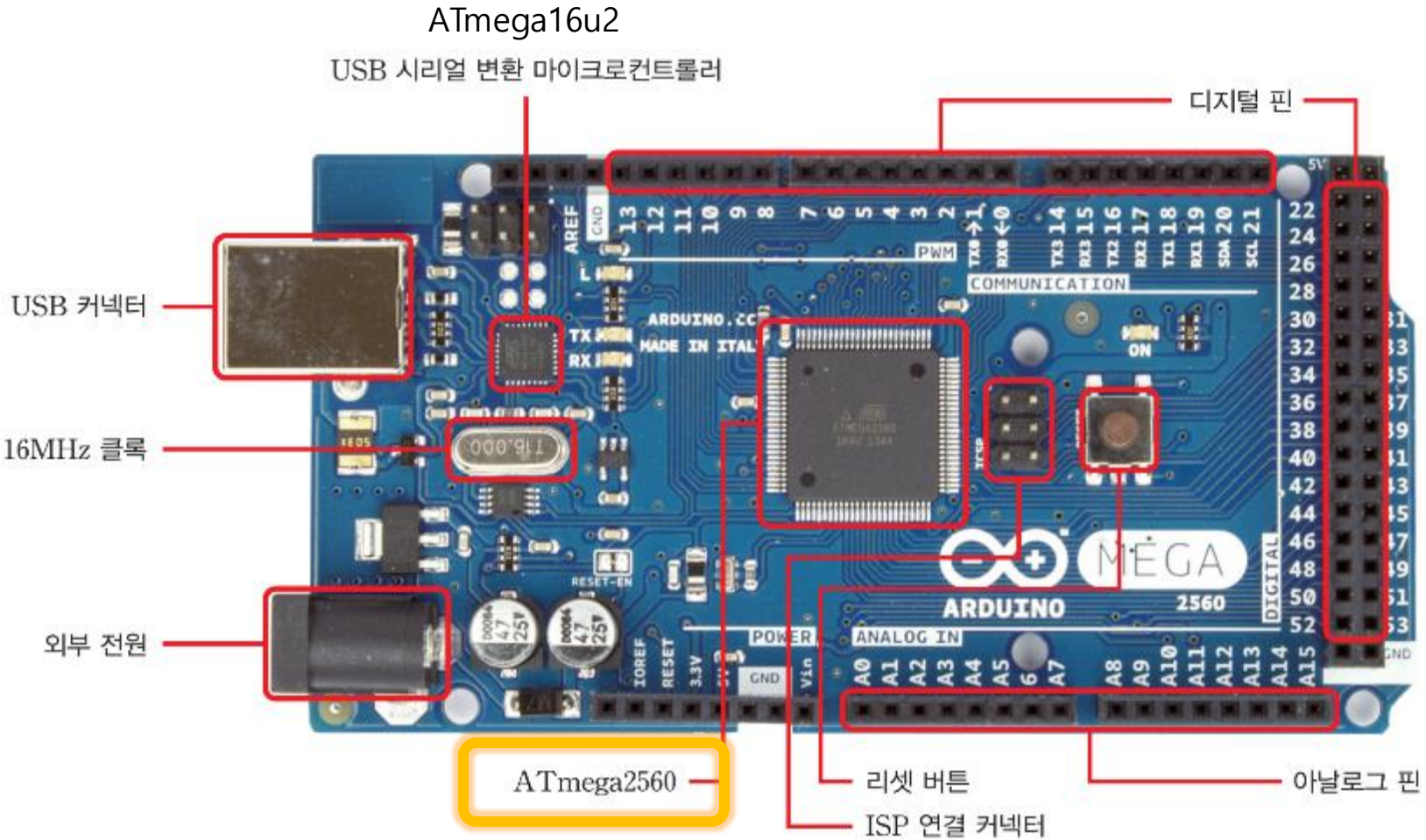
Arduino Board 종류 –μC의 종류

Arduino	Microcontroller	Architecture	CPU bits
Uno	ATmega328	AVR	8
Leonardo	ATmega32u4	AVR	8
Mega2560	ATmega2560	AVR	8
Zero	ATSAMD21G18	ARM Cortex M0+	32
Due	AT91SAM3X8E	ARM Cortex M3	32

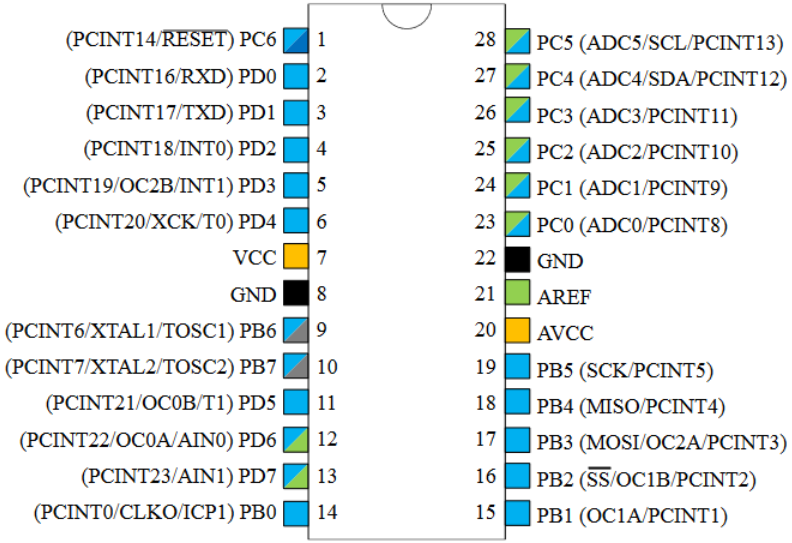
AVR 기반 Arduino Board

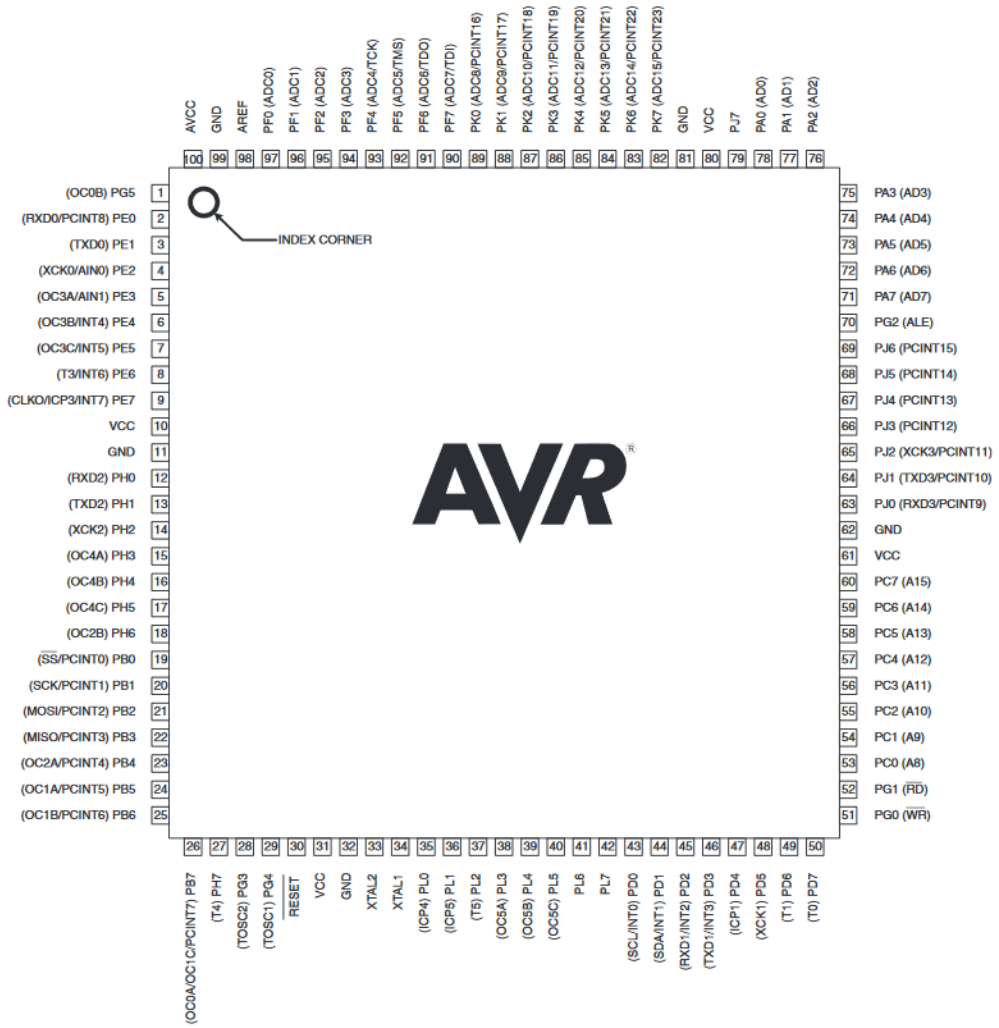
항목	UNO	Leonardo	Mega2560
uC	ATmega328	ATmega32u4	**ATmega2560**
클록	16MHz	16MHz	**16MHz**
핀 수	28	44	**100**
디지털 입출력 핀 수	14	18(12)	**54**
아날로그 입력 핀 수	6	6(12)	**16**
입출력 핀 수	20	24	**70**
PWM 채널 수	6	7	**15**
플래시 메모리	32KByte	32KByte	**256KByte**
SRAM	2KByte	2.5KByte	**8KByte**
EEPROM	1KByte	1KByte	**4KByte**

Arduino Mega2560

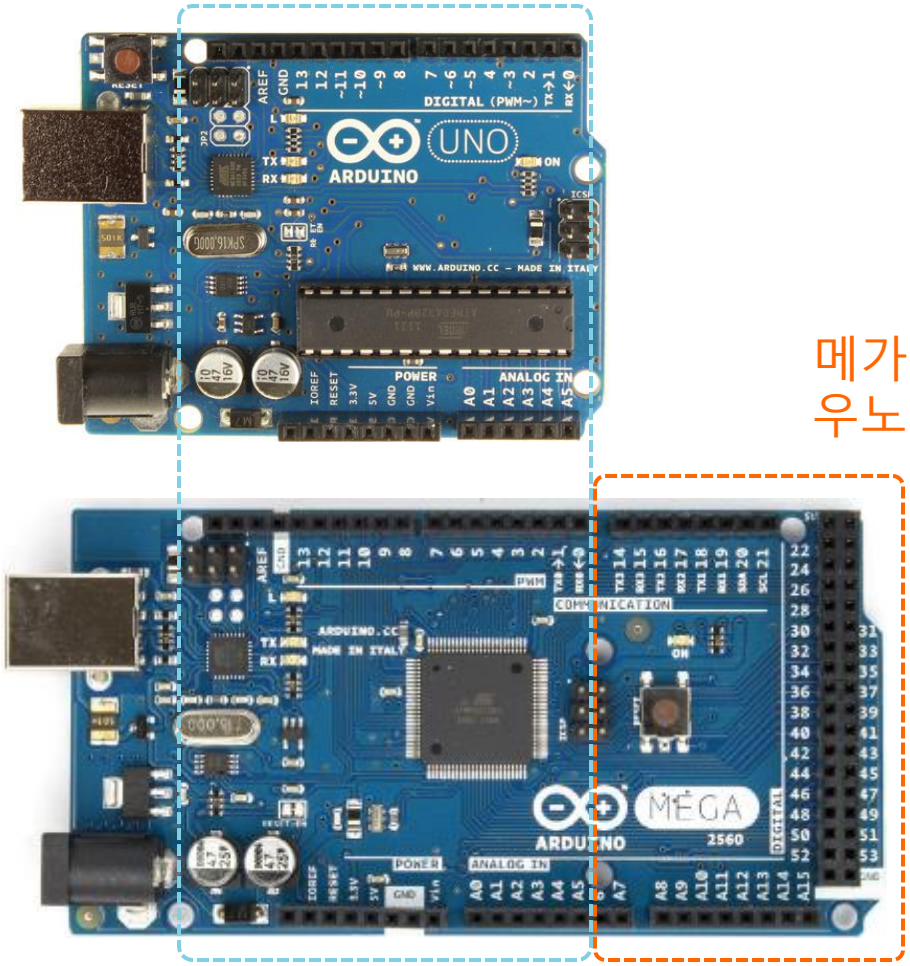


ATmega328 vs ATmega2560





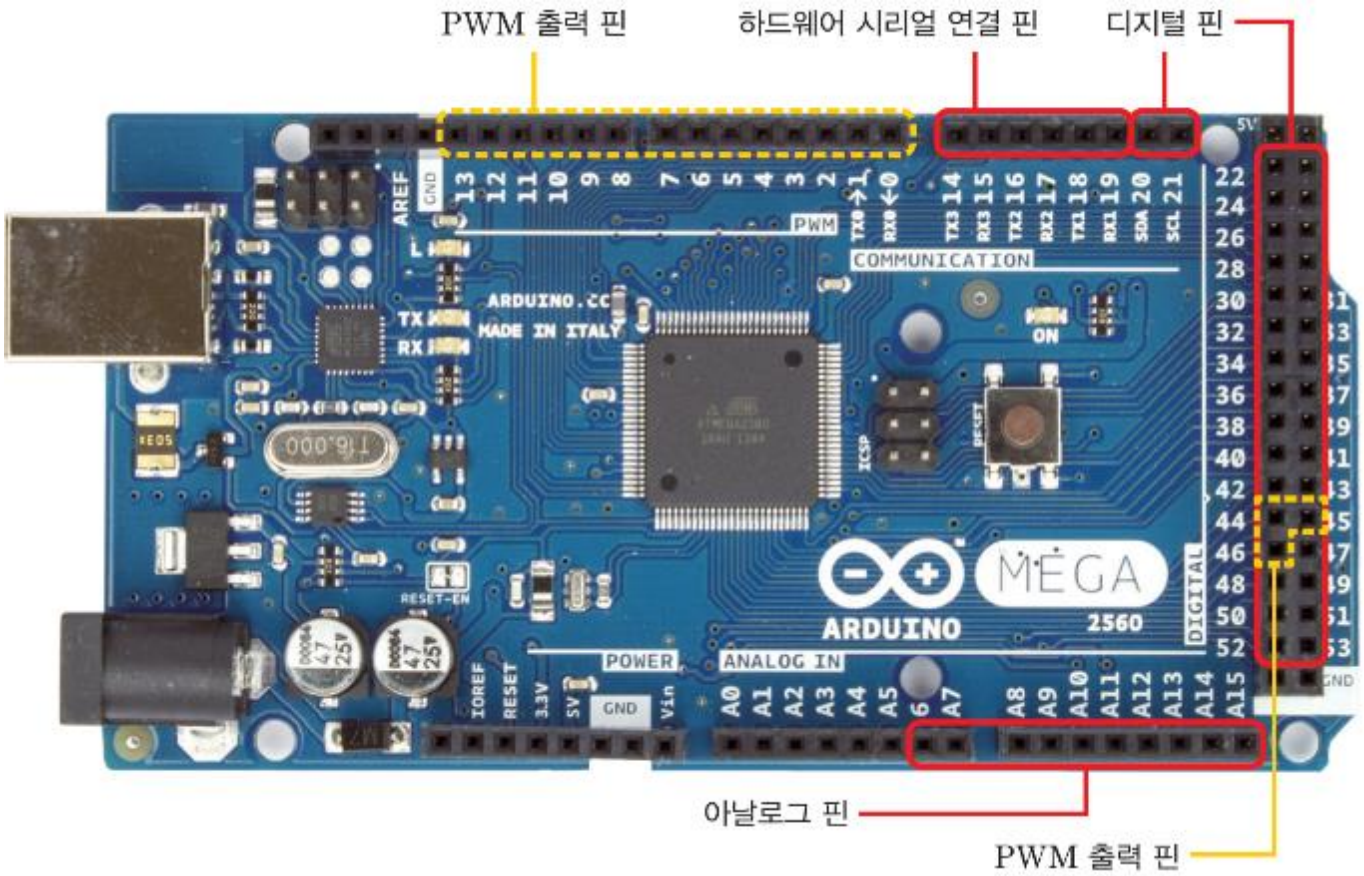
Arduino UNO vs. Mega2560



메가2560의 확장 핀 배열 :
우노에 비해 추가 기능 제공

동일한 핀 배열 : UNO, Mega2560 등에 공통으로 사용할 수 있는 "Shield" 제작 가능

Mega2560의 추가/변경 기능



* Arduino UNO 기준, Mega2560의 기능 추가 또는 변경

Arduino Mega2560

- **Arduino UNO와 비교했을 때 더 많은 입출력 핀 사용**
  - 70개 핀(UNO는 20개 핀)을 디지털 입출력으로 사용 가능
  - 16개 핀(UNO는 6개 핀)을 아날로그 입력으로 사용 가능
  - 15개 핀(UNO는 6개 핀)을 PWM 출력으로 사용 가능
  - 4개(UNO는 1개)의 UART 시리얼 통신 포트 사용 가능
- **Arduino UNO와 비교했을 때 더 큰 메모리 사용 가능**
  - 256KB (UNO는 32KB) 크기의 플래시 메모리 사용 가능
  - 8KB (UNO는 2KB) 크기의 SRAM 사용 가능
  - 4KB (UNO는 1KB) 크기의 EEPROM 사용 가능
- **입출력 핀 수와 메모리 크기를 제외하면 UNO와 Mega2560은 기본적으로 동일한 특성을 가짐**

맺는말

❖ Arduino UNO

- Arduino의 기본 보드
- 20개의 입출력 핀으로 연결할 수 있는 주변장치에 한계가 있음
- 32KB의 플래시 메모리로 복잡한 스케치(아두이노 프로그램) 작성에 한계가 있음

❖ Arduino Mega2560의 장점

- 70개의 많은 입출력 핀 : 다양한 주변장치를 한꺼번에 연결할 수 있음
- 256KB의 큰 플래시 메모리 : 크고 복잡한 스케치를 작성할 수 있음
- 따라서 다양한 주변장치를 연결하여 학습용으로 사용하기에 적합

Arduino S/W & Programming

- ✓ CHAPTER 02. 아두이노
  - LESSON 03 아두이노 소프트웨어
  - LESSON 04 아두이노 프로그래밍
  - LESSON 05 아두이노 스케치 구조

Lab.2-1

Arduino IDE Installation and Practice

❖ Experiment Sequence

① Download the Arduino IDE (Integrated Development Environment) : www.arduino.cc

② Run Installer

③ Specify a Destination Folder

④ Install Arduino Board Driver (Automatically proceeded by the Installer)

⑤ Connect Arduino Board to PC and Run IDE

⑥ Open Example Codes (Blink, ASCIITable)

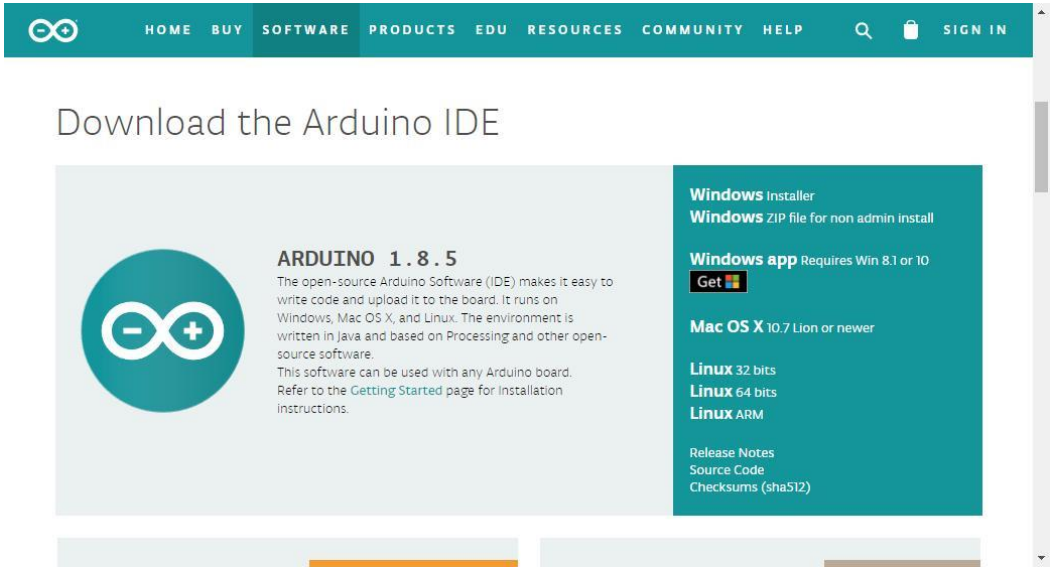
⑦ IDE Preference Setup

⑧ Cross Compile & Upload

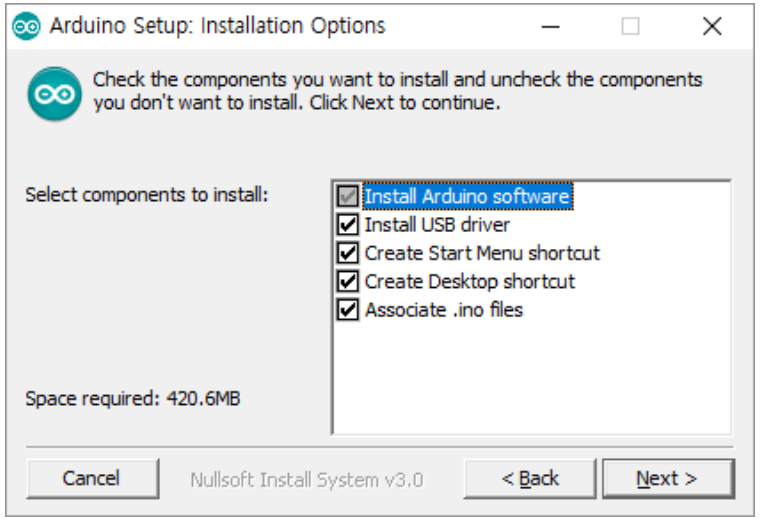
⑨ Check the Result

Lab.2-1

①,② Download and Install Arduino IDE

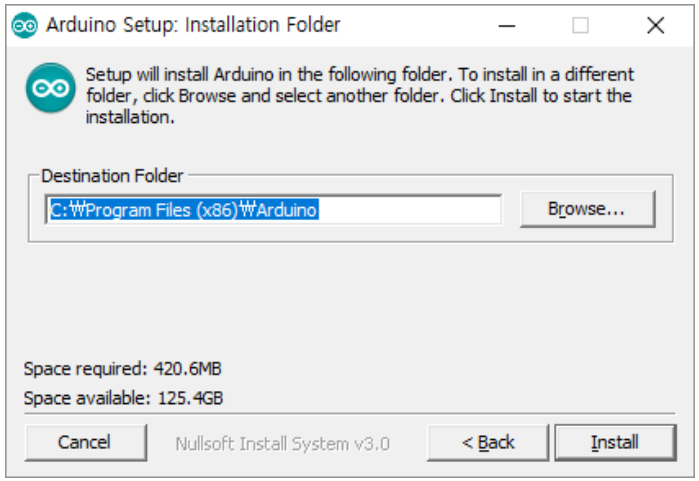


① Download IDE : www.arduino.cc



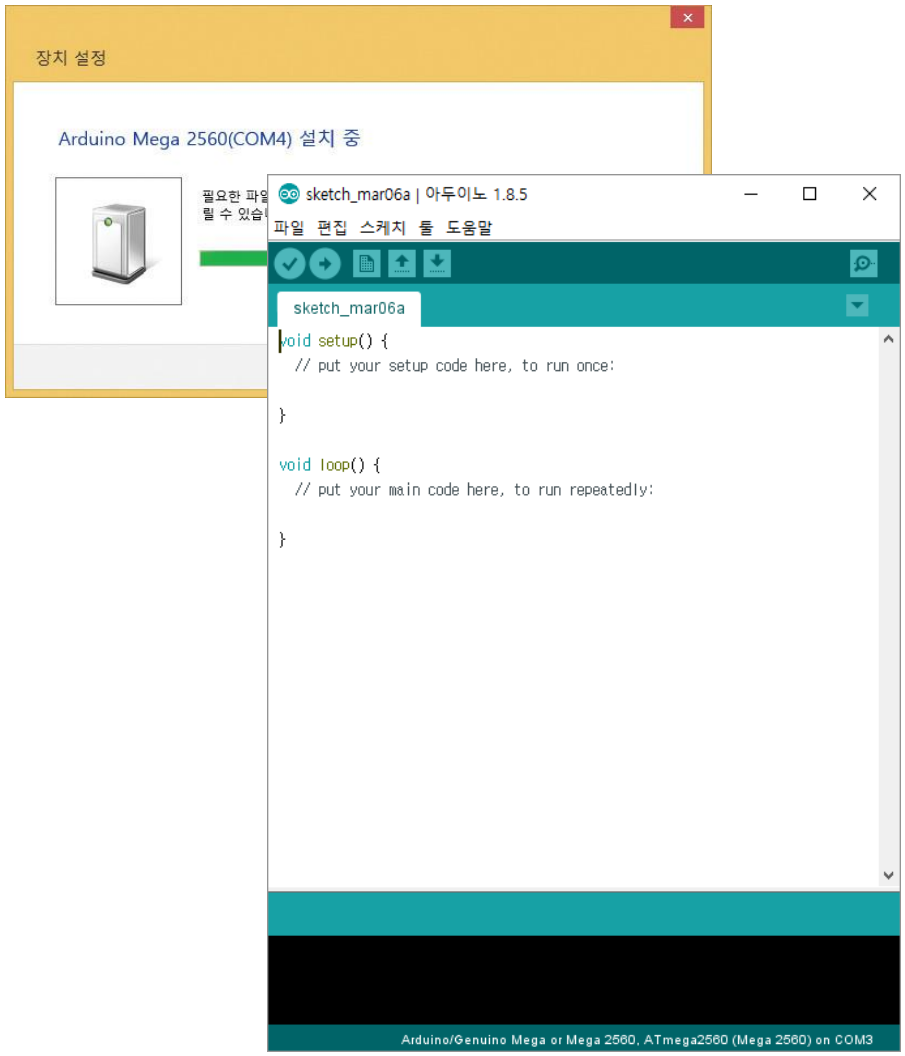
② Run Installer

③~⑤ Arduino IDE Installation and Execution



③ Specify a destination folder

④ Install an Arduino driver

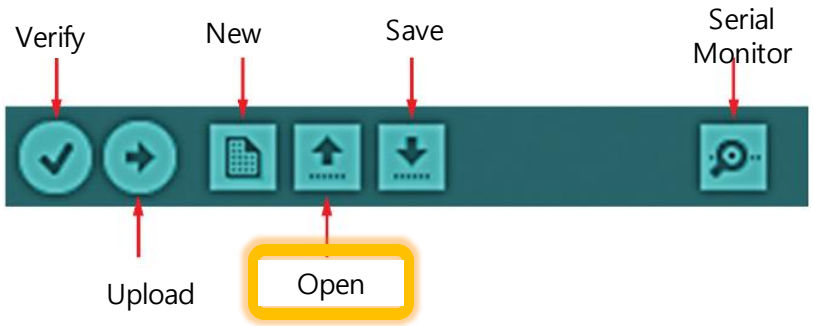


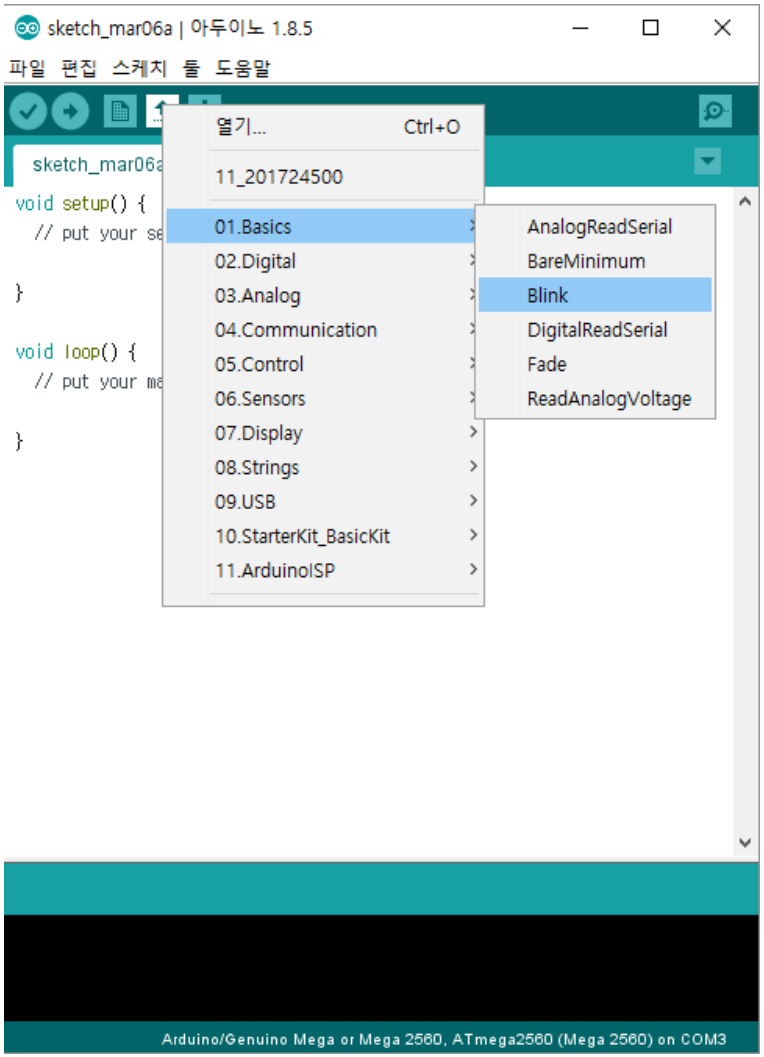
⑤ Connect Arduino board and run IDE
: setup COM port

⑥ Open Example Codes (Blink)

❖ Click "Open" in IDE Toolbar

- or click "File > Exmaples" in menu bar





❖ Examples

- 01.Basics -> Blink
  - Example code for the first practice
- 04.Communications -> ASCIITable
  - Example code for the second practice

Lab.2-1

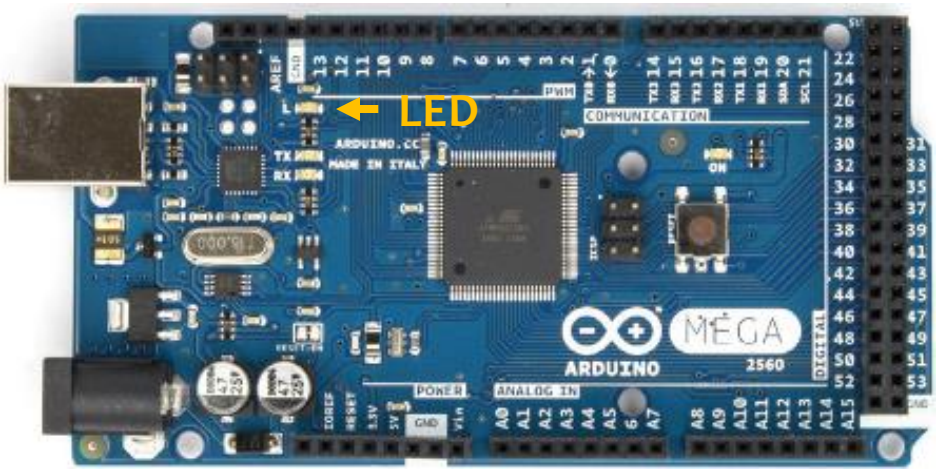
Blink Sketch

❖ Sketch

- A Sketch is the name of an Arduino program becaus it is very easy to write just like drawing

❖ Blink Sketch

- This sketch blinks an LED connected to digital pin 13 with 1000ms (=1 sec) interval.
- 'Hello World' of Arduino programming



Blink | 아두이노 1.8.5

파일 편집 스케치 툴 도움말

Blink

```
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

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

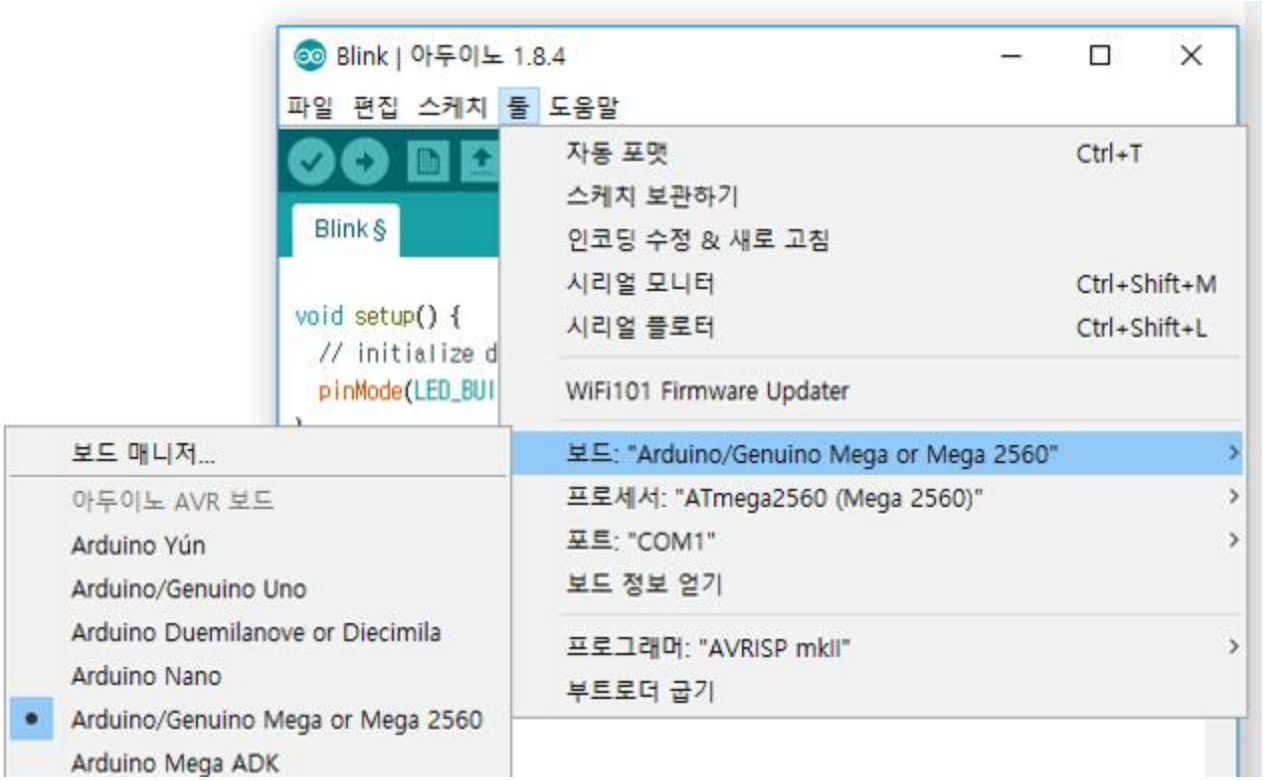
자동 포맷을 위해 변경이 필요없습니다.

24 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

⑦ IDE Preference Setup - Board

❖ Board Selection

- Setup a type of board which you are using
- Select 'Arduino/Genuino Mega or Mega 2560' in 'Tools → Board' menu



⑦ IDE Preference Setup - Processor

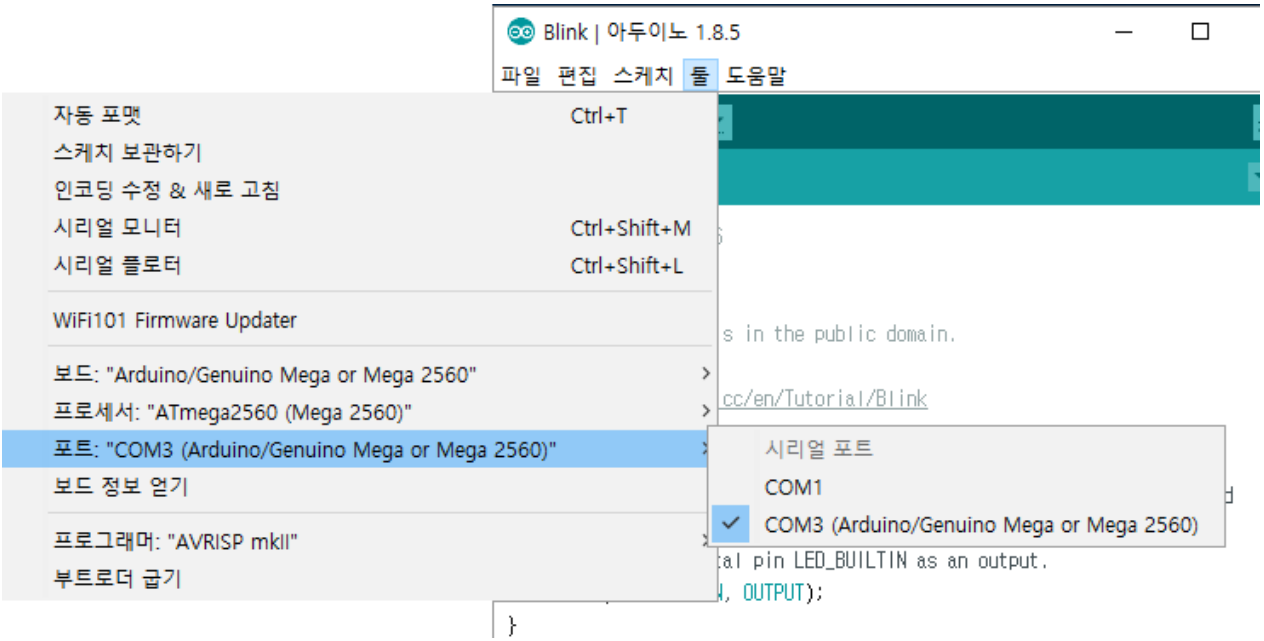
❖ Processor Selection

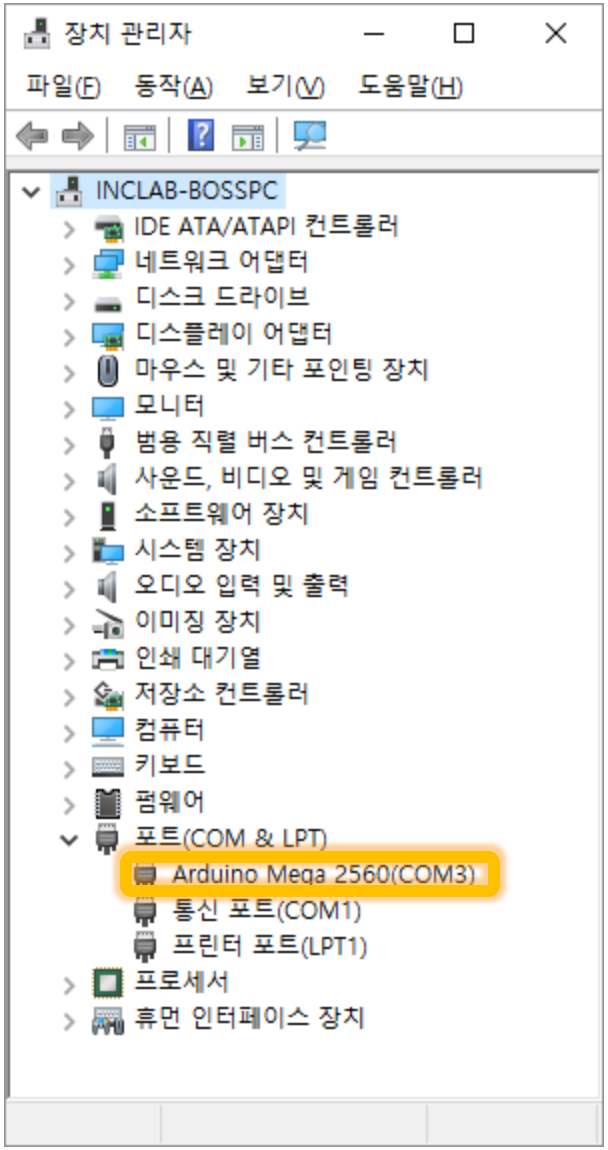
- Select the type of an µC which is used by the current Arduino board
- Select 'ATmega2560' in 'Tools → Processor' menu

⑦ IDE Preference Setup - Port

❖ Port Selection

- Setup the virtual COM port assigned to the Arduino
- Select the port in ‘ Tools → Port’ menu
- Verify which port is assigned to Arduino board in “Device Manager “

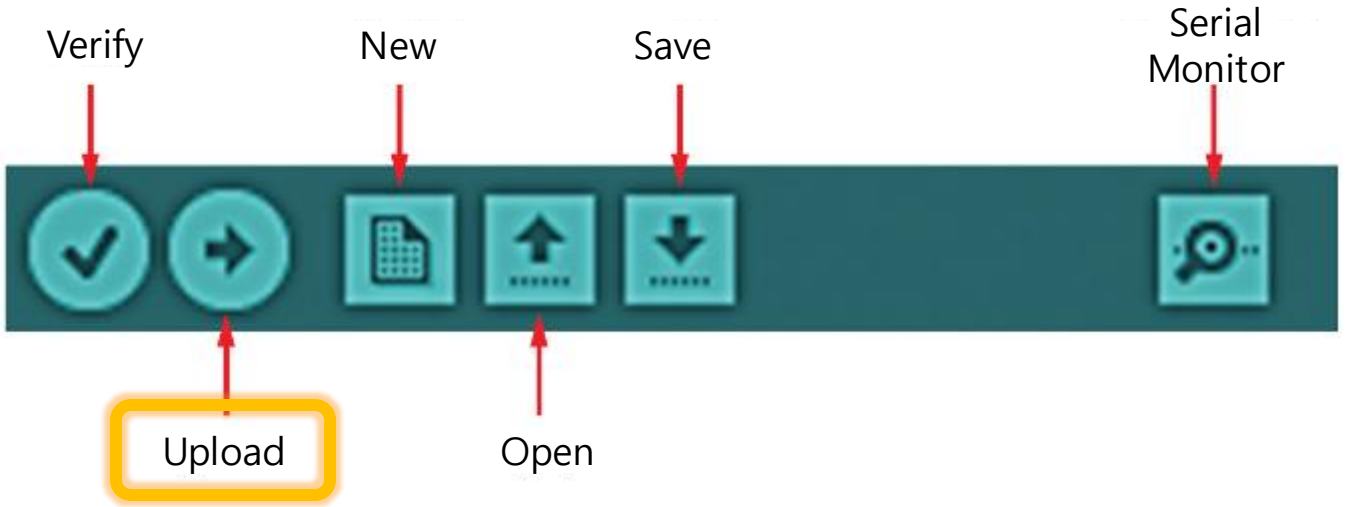




⑧ Cross Compile & Upload

❖ Toolbar

- Verify : Sketch (Cross) Compile
- Upload : Compile a Sketch (Cross) and upload a produced execution code
  → Just one click for both compile and upload
- Serial Monitor : Executing the program for serial communication between Arduino and PC



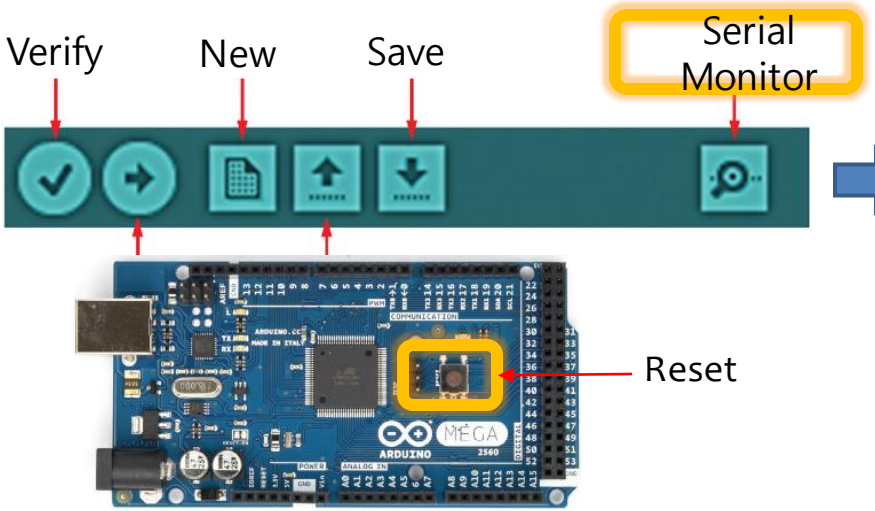
⑨ Check the Result

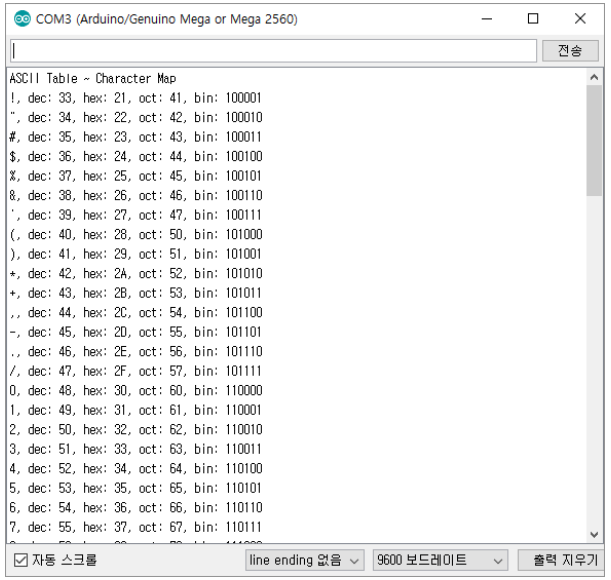
❖ **Ex. 1 Check the LED blinking**

❖ **Ex. 2 Compile and Upload ASCIITable example**

- ASCIITable Sketch – Print out ASCII characters to serial port
  - Check the output of the Sketch on the Serial Monitor
- Serial Port : A communication channel between Arduino and PC
  - For program uploading : PC → Arduino
  - Serial port can be used for communication from Arduino to PC and vice a versa
  - The information sent by Arduino is printed out on 'Serial Monitor'
- Ref. – Re-execution ASCIITable Sketch
  - Push the Reset botton on the board

After the completion of the experiments, you have to get confirmation by TA and submit the score result on PLATO LAB 2-1





Arduino IDE

- 작고 간단한 통합개발환경/IDE
- 초보자를 위해 꼭 필요한 기능들만으로 구성
- 한 번의 클릭으로 Compile에서 Upload까지 진행
- Java로 구현되어 OS 간 이식성(Portability)이 뛰어남
- Debugging 기능은 제공하지 않음

Arduino SW 장점 – 추상화된 함수

❖ 서로 다른 µC를 사용한 Arduino 보드를
동일한 코드로 동일한 동작을 구현할 수 있도록 해 줌

ATmega2560의 Blink

```
DDRB  |= 0x80;
PORTB |= 0x80;
```

ATmega328의 Blink

```
DDRB  |= 0x20;
PORTB |= 0x20;
```

AVR 스타일 : µC에 따라 약간씩 차이가 있음

Arduino Mega2560과 Arduino UNO의 Blink

```
pinMode(13, OUTPUT);
digitalWrite(13, HIGH);
```

Arduino Style : 모든 Arduino Board에서 동일함

Sketch의 구조

❖ μC의 프로그램

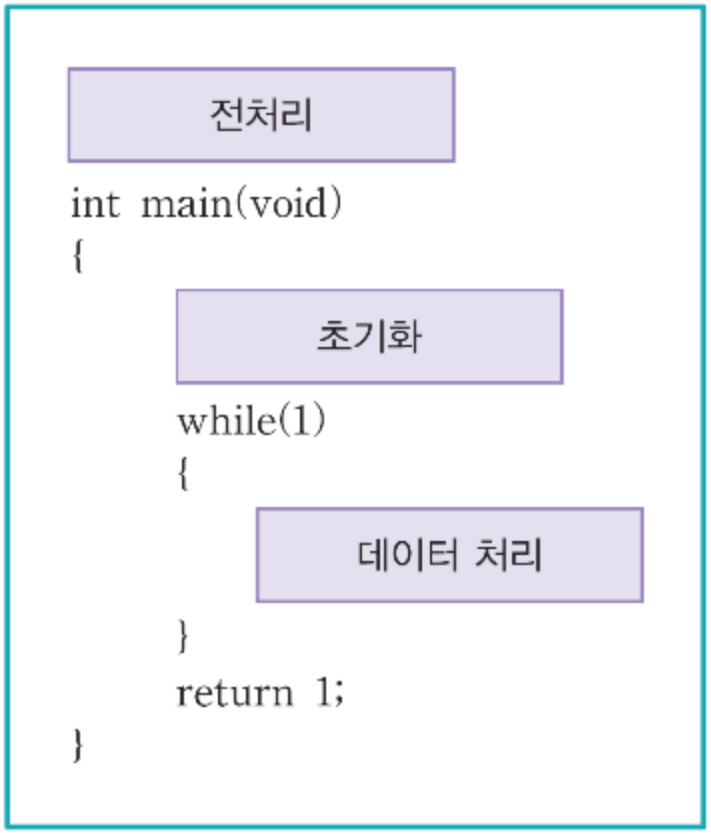
- 일반 컴퓨터와 달리 **오직 하나의 프로그램**만 설치
- 설치된 프로그램은 **전원이 주어지는 동안 끝나지 않는 무한 루프**를 통해 동작

❖ Arduino Sketch Programming Language : C/C++ 기반

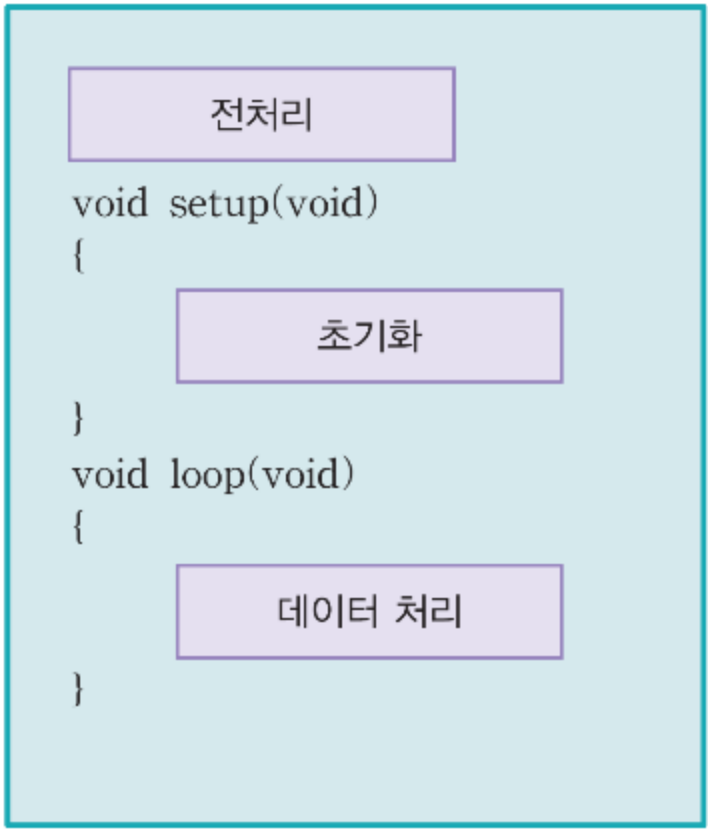
❖ Sketch 구조 : 2개의 기본 함수로 구성

- 직관적 프로그래밍을 위해 main 함수 없이 setup(), loop() 두 함수로 구성
  - 실제 main 함수는 숨겨져 있으며 Sketch 개발자는 신경 쓰지 않아도 됨
- **Setup 함수**
  - 초기화 함수
  - 스케치 실행이 시작될 때 한 번만 실행
- **loop 함수**
  - 반복 실행 함수
  - μC를 위한 프로그램에서 메인/이벤트 루프에 해당

Sketch의 구조 비교

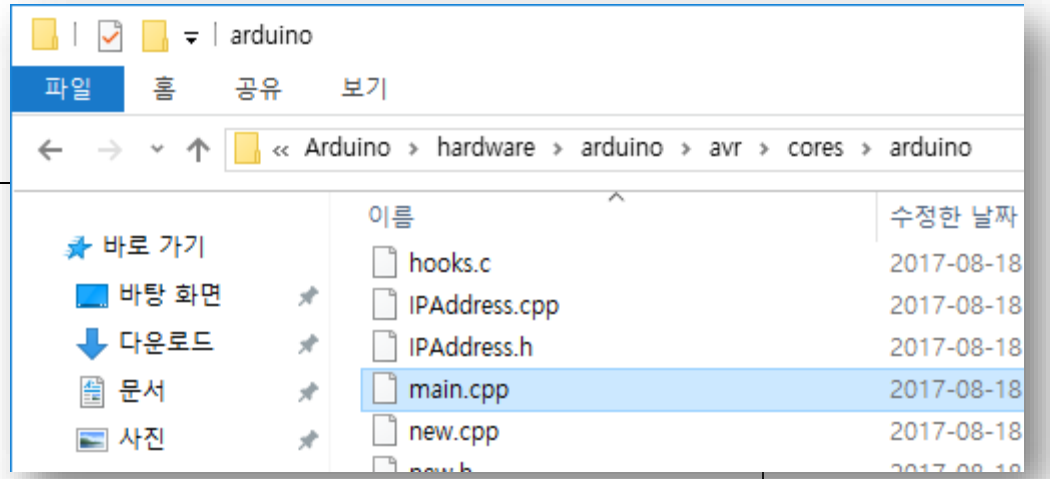


μC를 위한 프로그램의 구조



Arduino를 위한 Sketch의 구조

Main.cpp




```
#include <Arduino.h>

int main(void)
{
        init();

        initVariant();

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

        setup();

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

        return 0;
}
```

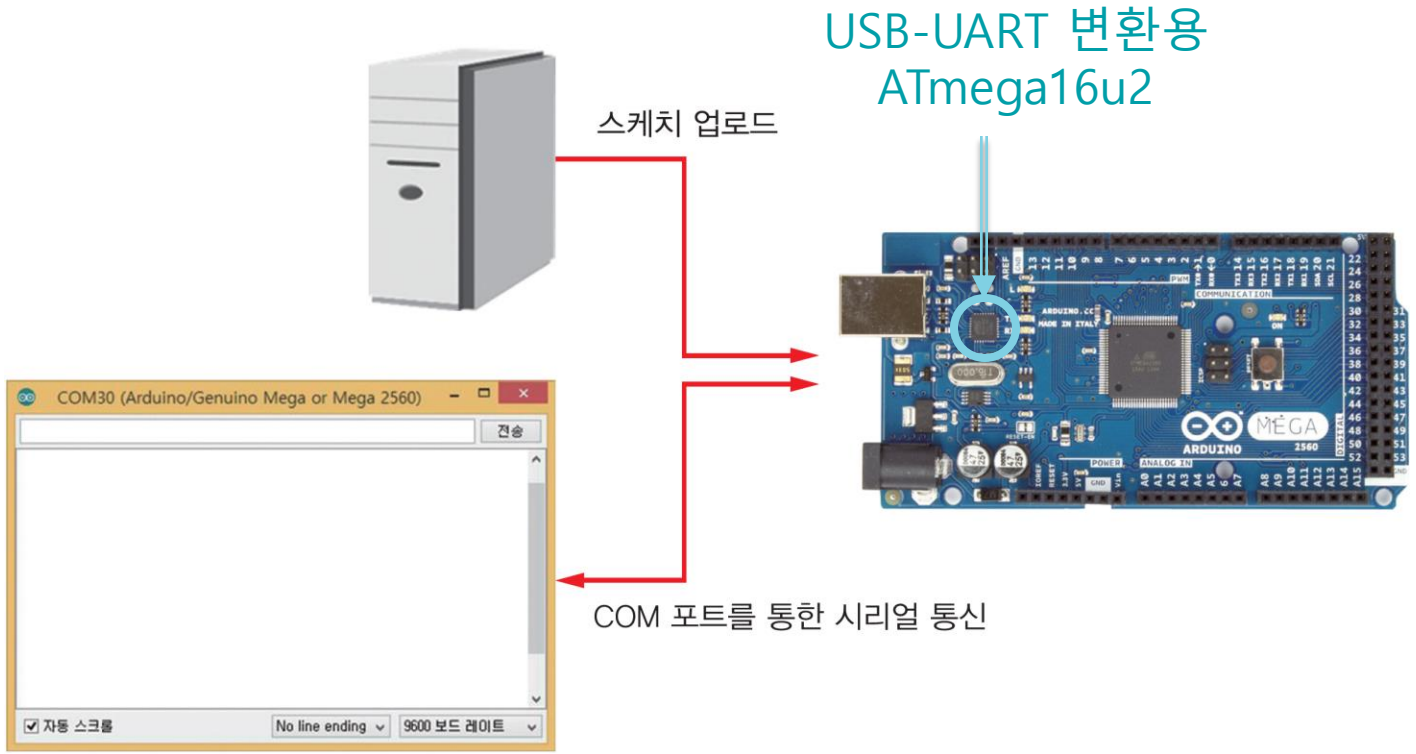
Arduino Serial Class

- ✓ CHAPTER 04. 아두이노 기본 클래스
  - LESSON 01 아두이노 라이브러리
  - LESSON 02 시리얼 통신
  - LESSON 03 Serial 클래스

Arduino와 컴퓨터의 연결 – UART Serial 통신

① 스케치 업로드를 위해 사용

② (양방향) Serial 통신을 위해 사용



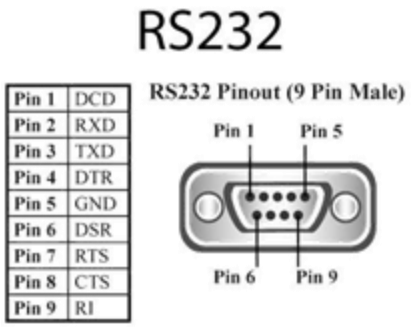
Arduino UART Serial 통신

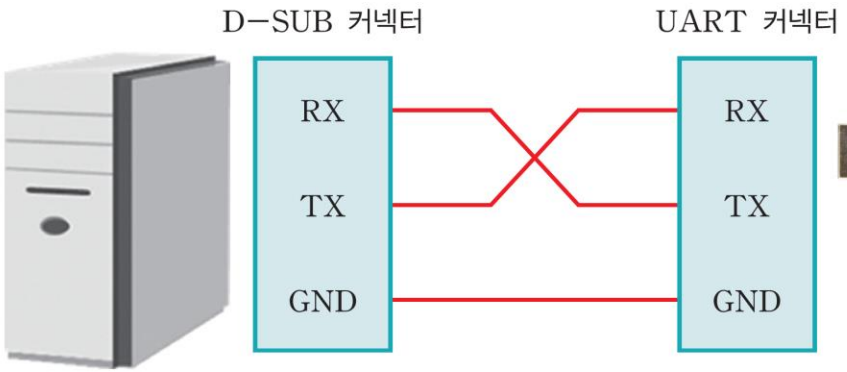
❖ Universal Asynchronous Receiver/Transmitter

- Universal : RS-232C, RS-422, RS-485 등 다양한 통신 프로토콜에 쓰이는 범용 통신 기술
- Asynchronous : 비동기 → 동기화를 위한 별도의 Clock을 사용하지 않음
- 하드웨어 수준에서 지원하는 저수준 통신 기술

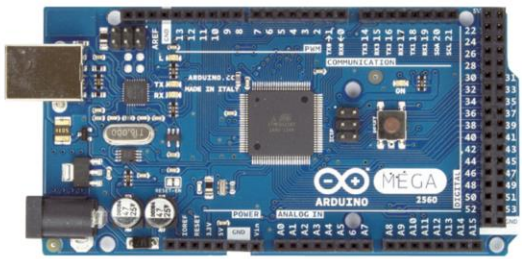
❖ Arduino UART Serial 통신

- RS-232C프로토콜 중 RX, TX, GND의 세 개의 Pin 만을 사용
- TX와 RX 교차 연결





RS-232C 연결

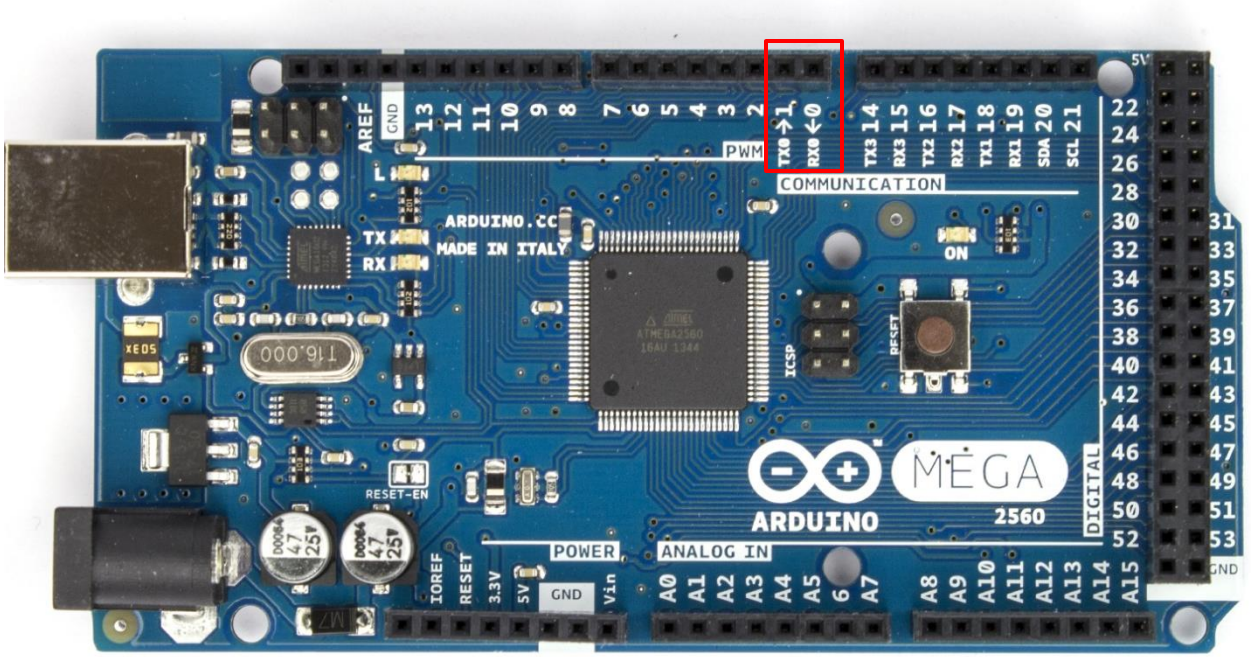


USB 연결

Arduino UART Serial 통신

❖ Arduino 보드와 컴퓨터의 통신은 USB로 이루어짐

- Arduino 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



Arduino Serial Class

❖ UART Serial Class

- Sketch 작성에서 일반적으로 쓰이는 UART Serial 통신을 위한 Class
- 별도의 헤더 파일 (*.h) include 없이 바로 사용할 수 있는 Arduino 기본 Class

❖ 실제 Class 이름은 Serial_이며, 그 객체가 Serial임

- UNO는 하나의 UART Serial 통신 포트만을 제공하는 반면, Mega2560은 4개의 UART Serial 통신 포트를 제공
- Mega2560은 4개의 UART Serial 통신 포트를 위해
  Serial, Serial1, Serial2, Serial3의 전용 객체가 미리 정의되어 생성되어 있음
  - Serial_ Class에서 직접 객체를 생성하여 사용하는 경우는 없음

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

OOP & C++ Class 사용법

❖ 객체지향프로그래밍(Object Oriented Programming : OOP)

- 프로그램을 데이터와 처리 방법으로 나누는 것이 아니라
  수 많은 "객체(Object)"로 나누고 객체간의 상호 작용으로 프로그램을 기술하는 방법
- 객체는 **속성(Attribute/member variable)**과 **함수(Method/member function)**를 가짐

❖ C++ Class Example

- Class와 (class/object) instance : Ex) Class – 사람, Instance – 김영미
- object instance에 .을 덧붙여 method를 호출하거나 attributes에 접근

```
class Rectangle {
    int width;
    int height;

    void draw(void);
    int area (void);
}
```

Class Name

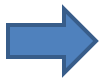
Attributes
Member Variable

Method
Member function

```
Rectangle rect;

rect.width = 5;
rect.height = 4;

rect.draw();
```

Arduino Serial Class

- **Arduino Reference → https://www.arduino.cc/reference/en/**
- **Arduino Serial Class Reference**
  - https://www.arduino.cc/reference/en/language/functions/communication/serial/
  - Serial Class가 제공하는 전체 Methods / Member Functions



If (Serial)
available()
availableForWrite()
begin()
end()
find()
findUntil()
flush()
parseFloat()
parseInt()
peek()
print()
println()
read()
readBytes()
readBytesUntil()
setTimeout()
write()
serialEvent()

  - begin() 함수는 무슨 역할을 할까?
  - 데이터 출력을 위해 print(), println(), write()
  - 데이터 입력을 위해 available(), peek(), read() 함수를
  - 실습을 통해 사용해 보자

Lab.2-2

Input/Output by using Serial Class

❖ Experiment Sequence

① Write Sketch code using Serial print/write function and execute it

② Write Serial print Sketch with specifying the format and execute it

③ Write Serial Monitor Echo Back Sketch and execute it

④ Check the Result

① Serial print/write Sketch

1. **Create new file (Ctrl+N) named as follows**

   - LAB2_2_1_{ID} : e.g. If ID is 201724500, file name is LAB2_2_1_201724500

2. **Refer to the definition of print() and write() at Arduino Serial Class Reference**

- Serial.print(val) / Serial.print(val, format)
  - Prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is.
  - Serial.print(78) gives "78"
  - Serial.print(1.23456) gives "1.23"
  - Serial.print('N') gives "N"
  - Serial.print("Hello world.") gives "Hello world."
- Serial.write(val) / Serial.write(str)
  - Writes binary data to the serial port. This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.

① Serial print/write Sketch

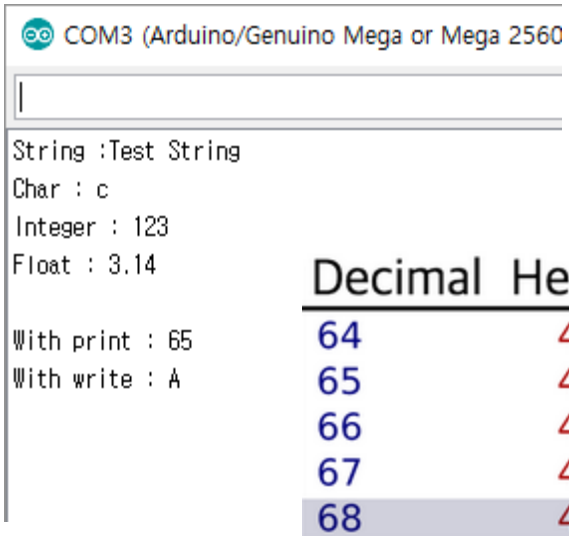
3. Input Sketch 4-1 code in page 68 of the textbook, then compile & upload

LAB2_2_1_201724500

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("String :");
  Serial.println("Test String");
  Serial.print("Char : ");
  Serial.println('c');
  Serial.print("Integer : ");
  Serial.println(123);
  Serial.print("Float : ");
  Serial.println(3.14);

  byte data = 65;
  Serial.println();
  Serial.print("With print : ");
  Serial.println(data);

  Serial.print("With write : ");
  Serial.write(data);

  while (true);
}
```

4. Check the ouput on Serial Monitor and compare them with Sketch code

- What's the difference between print() and println()?
- What's the difference between print() and write()?
- What does byte mean in 'byte data = 65;'
  - Which data types are supported in Arduino ?



Decimal	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D

Lab.2-2

② Serial Print Sketch with Specifying the Format

1. Create new file (Ctrl+N) named as follows

- LAB2_2_2_{ID}

2. Refer the meaning of 2nd parameter of print() at Reference

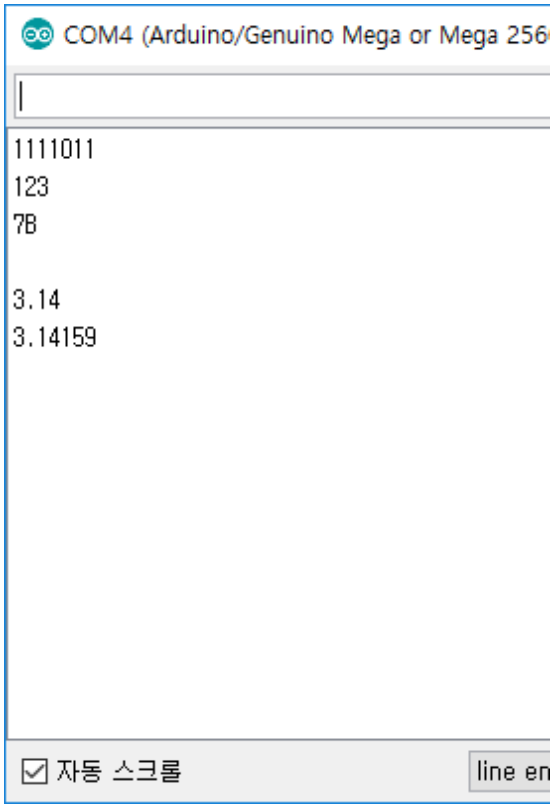
- An optional second parameter specifies the base (format) to use; permitted values are BIN(binary, or base 2), OCT(octal, or base 8), DEC(decimal, or base 10), HEX(hexadecimal, or base 16). For floating point numbers, this parameter specifies the number of decimal places to use. For example-
  - Serial.print(78, BIN) gives "1001110"
  - Serial.print(78, OCT) gives "116"
  - Serial.print(78, DEC) gives "78"
  - Serial.print(78, HEX) gives "4E"
  - Serial.println(1.23456, 0) gives "1"
  - Serial.println(1.23456, 2) gives "1.23"
  - Serial.println(1.23456, 4) gives "1.2346"

Lab.2-2

② Serial Print Sketch with Specifying the Format

3. Input the Sketch 4-2 code on page 70 of the Textbook, then compile & upload

```
void loop() {
  int n = 123;
  float f = 3.1415927;

  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);

  while (true);
}
```



③ Serial Monitor Echo Back Sketch

1. Create new file (Ctrl+N) named as follows

- LAB2_2_3_{ID}

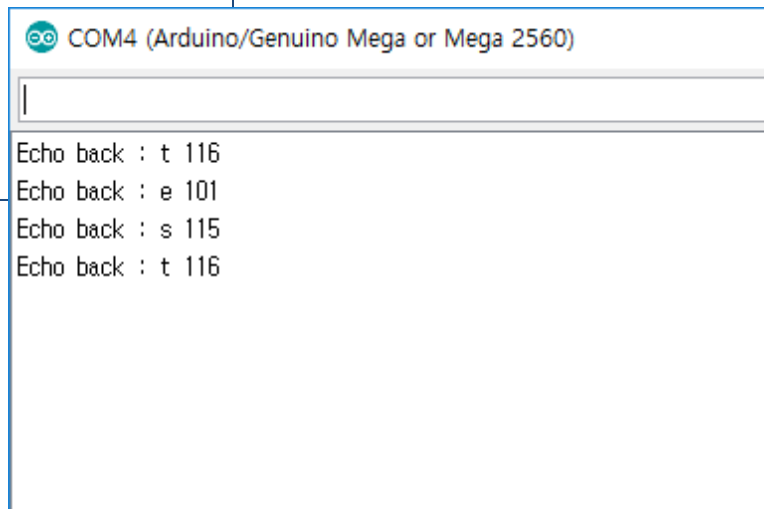
2. Refer the definition of available(), peek(), read() at Reference

- Serial.available(void)
  - **Get the number of bytes (characters) available for reading from the serial por**t. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes). available() inherits from the Stream utility class
- Serial.read(void)
  - Reads incoming serial data. read() inherits from the Stream utility class.
  - **Removing data from the internal serial buffer** after reading.
- Serial.peek(void)
  - Returns the next byte (character) of incoming serial data **without removing it from the internal serial buffer**. That is, successive calls to peek() will return the same character, as will the next call to read(). peek() inherits from the Stream utility class.

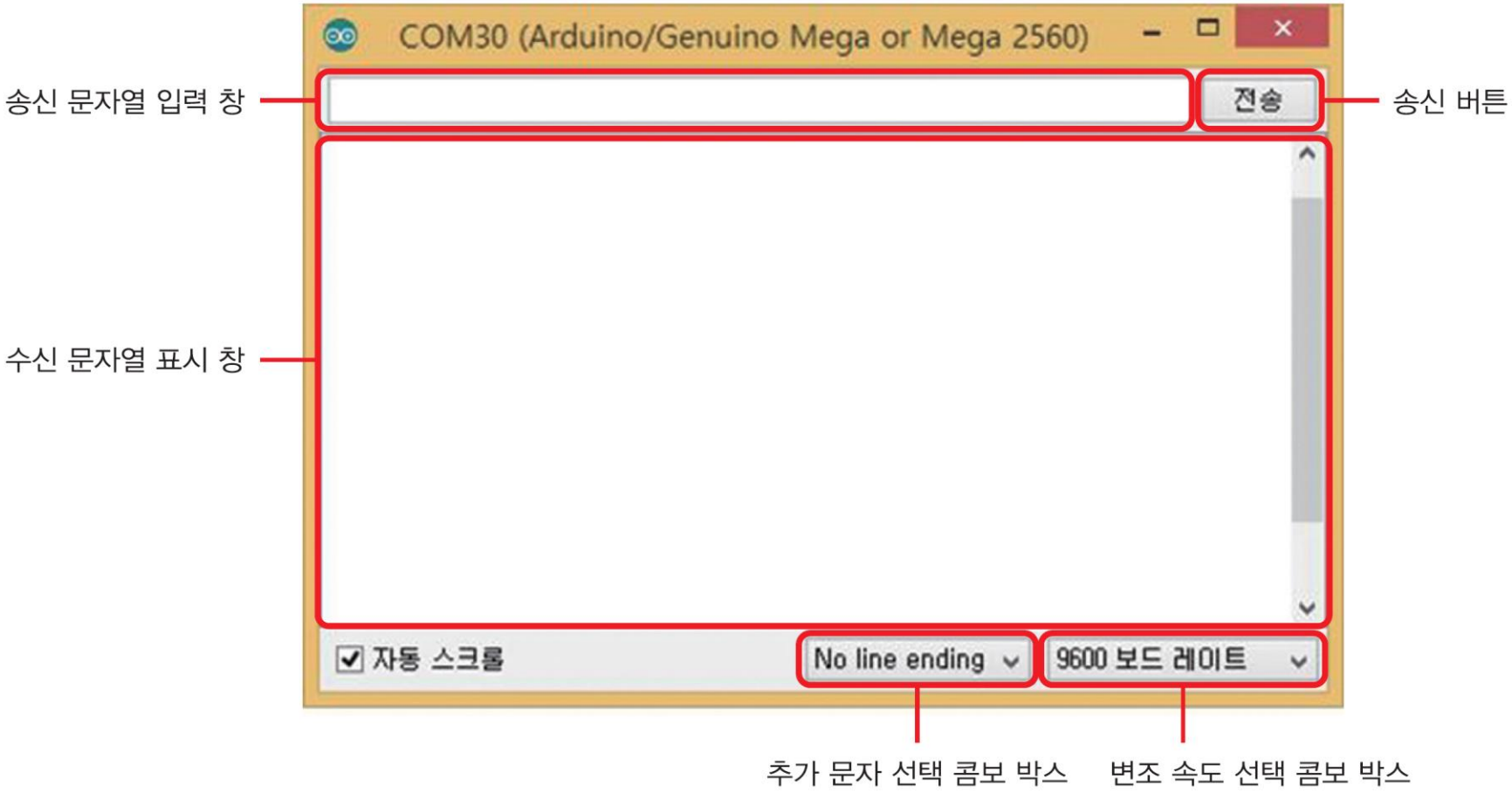
③ Serial Monitor Echo Back Sketch

3. Input the Sketch 4-3 code in page 72 of the Textbook, then Compile & Upload

```
void loop() {
  if (Serial.available() > 0) { // 수신된 데이터 존재 여부 확인
    byte data = Serial.read(); // 바이트 단위로 읽기

    Serial.print("Echo back : ");
    Serial.write(data); // 문자 출력
    Serial.print(" ");
    Serial.println(data); // ASCII 값 출력
  }
}
```

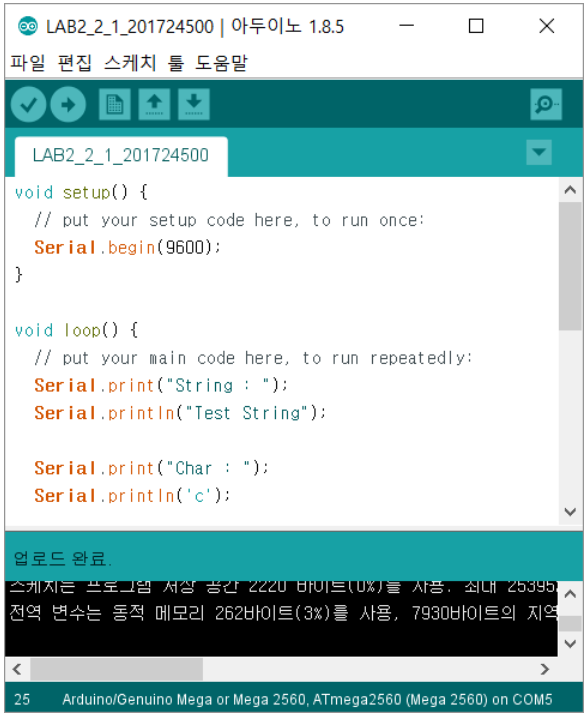


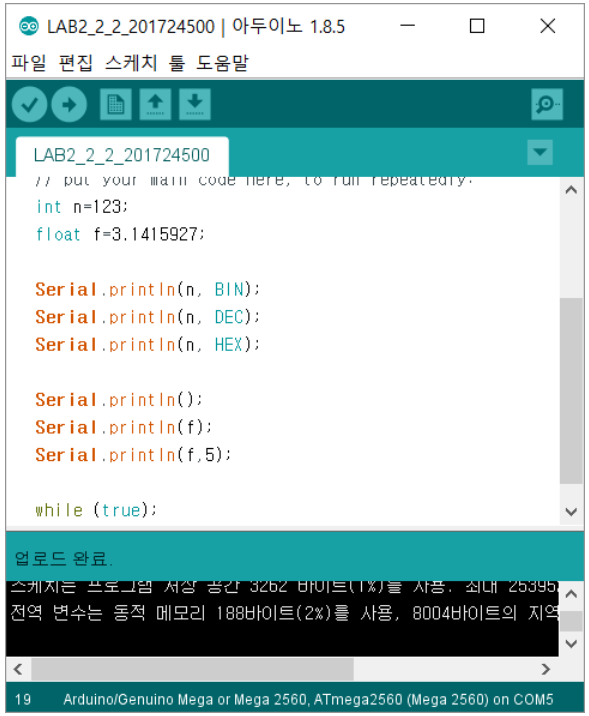
Serial Monitor



④ Check the Result (by TA)

1. Show the three Sketch code you have written
2. Show the result of LAB 2-2-3 Sketch on Serial Monitor
3. Answer TA's questions about Sketch code





Lab.2-3

How to Upload the Video Clips of Experiments

❖ Practice for uploading the Video Clips of Experiments

- In this class, usually you have to make video clips showing the experiment results and submit them
- This practice shows how to submit your video clips of experiment results by using Google Photo
  - This practice just considers unfamiliar students to smartphone or computer. If you don't want to use Google Photo, You can use any other ways to submit your handin.

❖ Experiment Sequence

① Create Google account, Install Google Photo APP

② Make a Video of Blink Experiment by using a Smartphone

③ Google Photo Upload by using a Smartphone

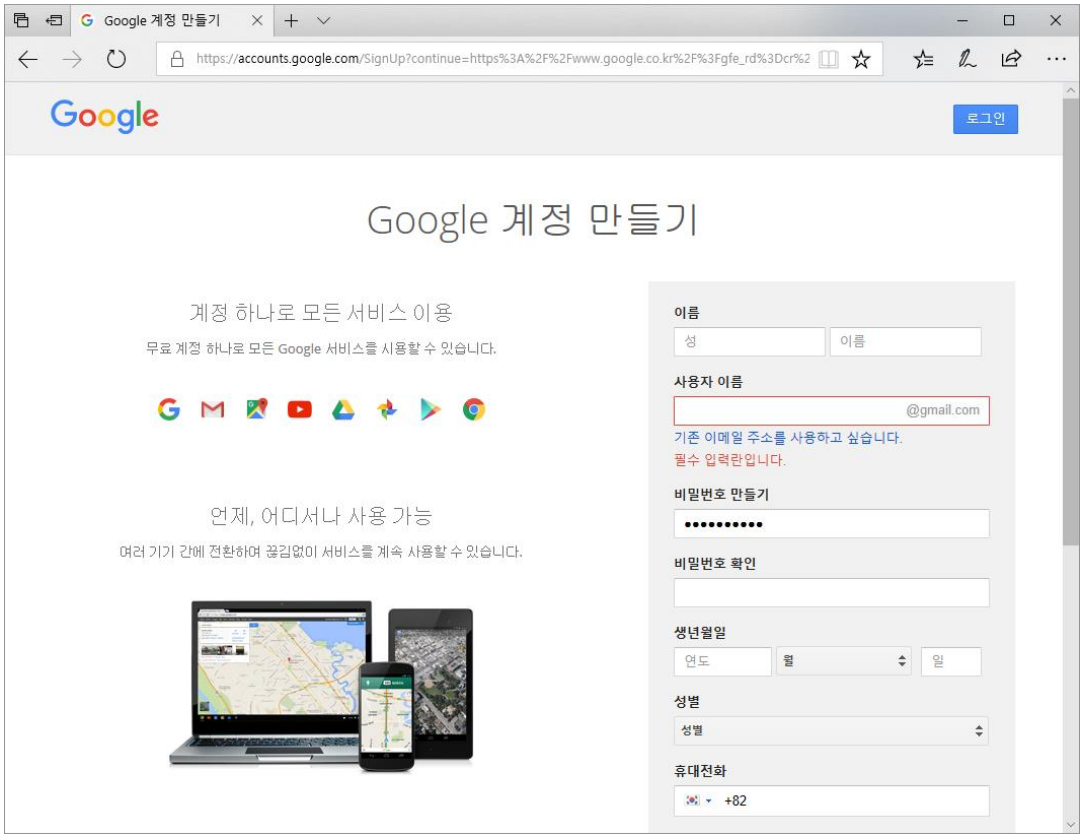
④ Sharing a Video Clip Link in Google Photo Site

⑤ Create Hyperlink of Video Clip in PLATO

Lab.2-3

① Create Google account, Install Google Photo APP

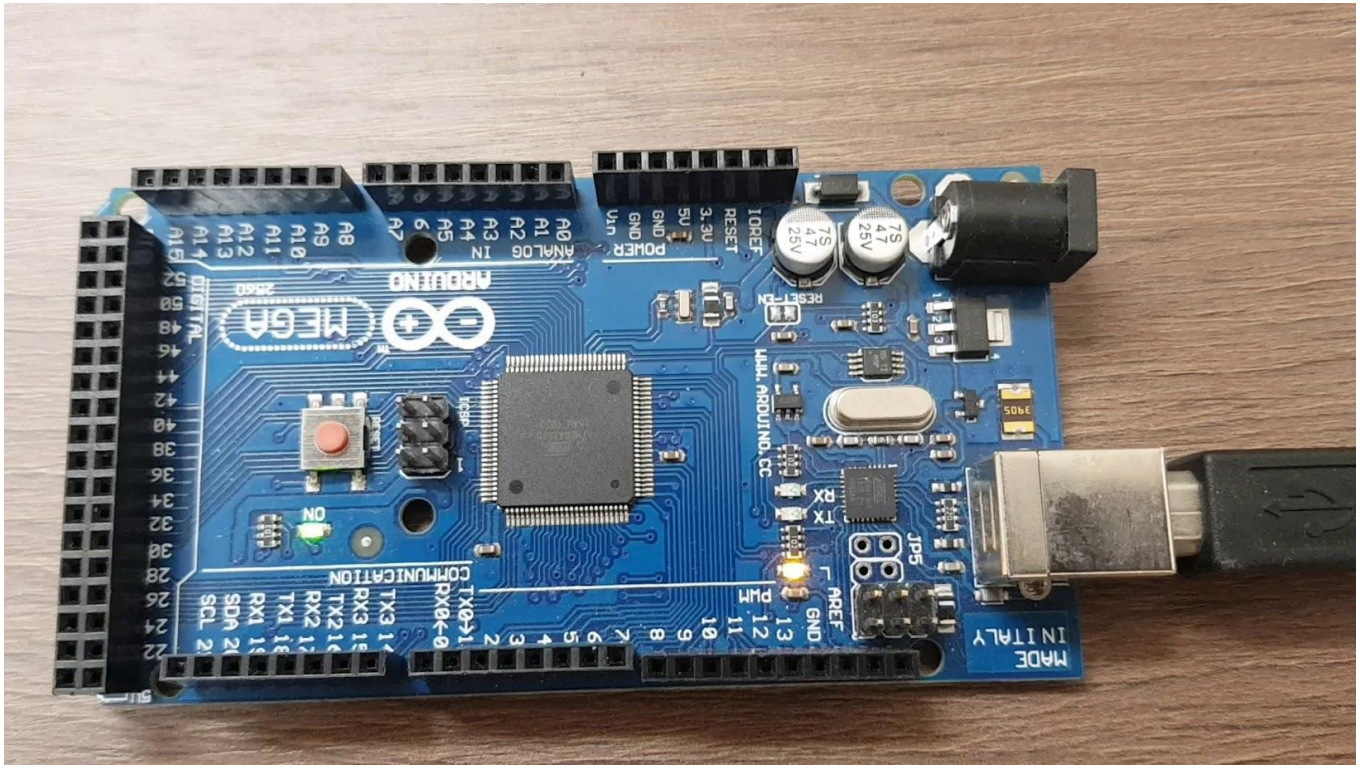
- **If you don't have a Google account, create a new account**
  - or if you have, just use it
- **Install Google Photo APP on your own smartphone**



Google Photos

② Make a Video of Blink Expt. by using Smartphone

1. Upload Blink Sketch of LAB 2-1 to Arduino
2. Make a video showing LED blinking by using a smartphone



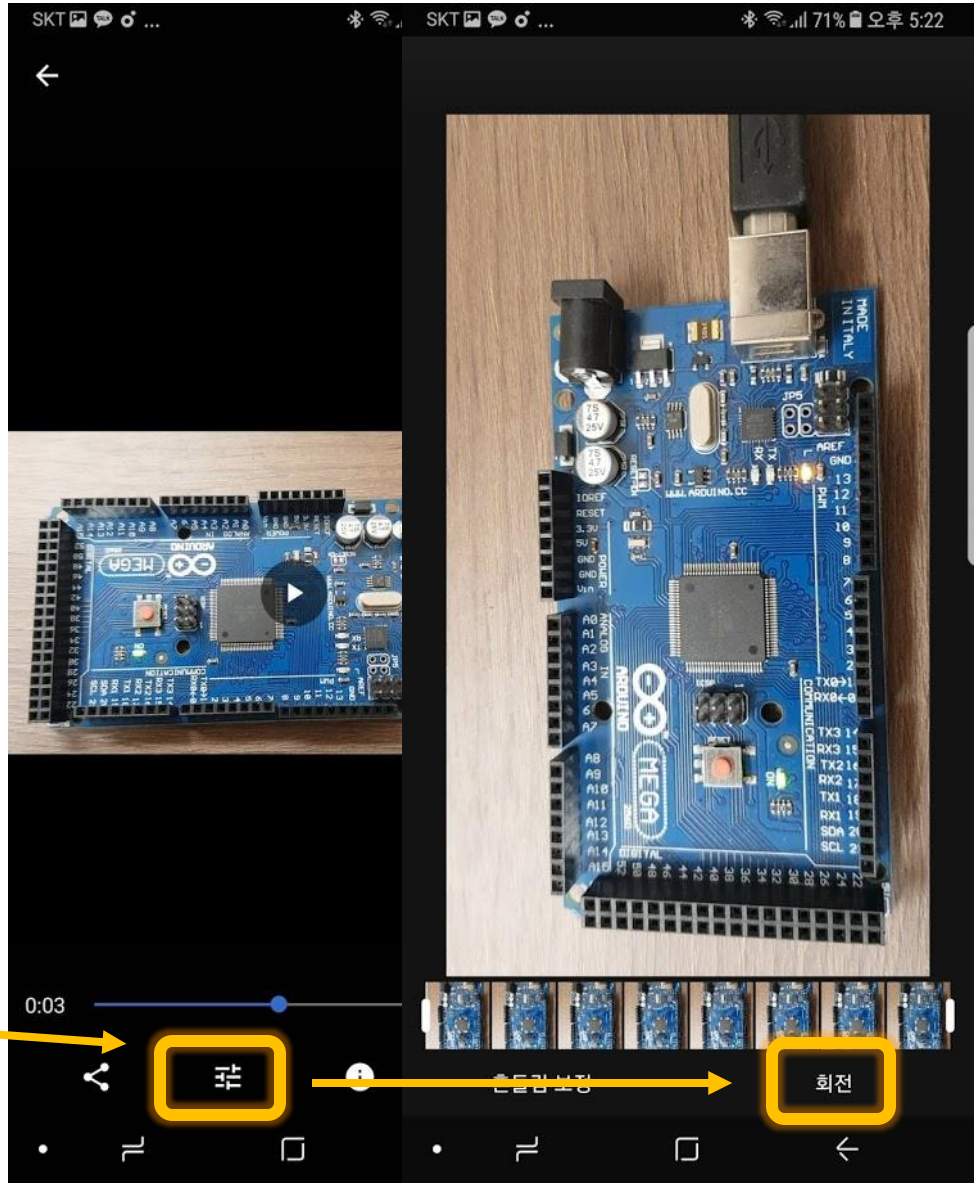
③ Google Photo Upload by using Smartphone

1. Check the video clip in Google Photo App

- Video clips are automatically Uploaded / Backuped to the Cloud. But it could require a Wi-Fi connection due to configuration. In this case, you can backup a video clip after selecting it explicitly.

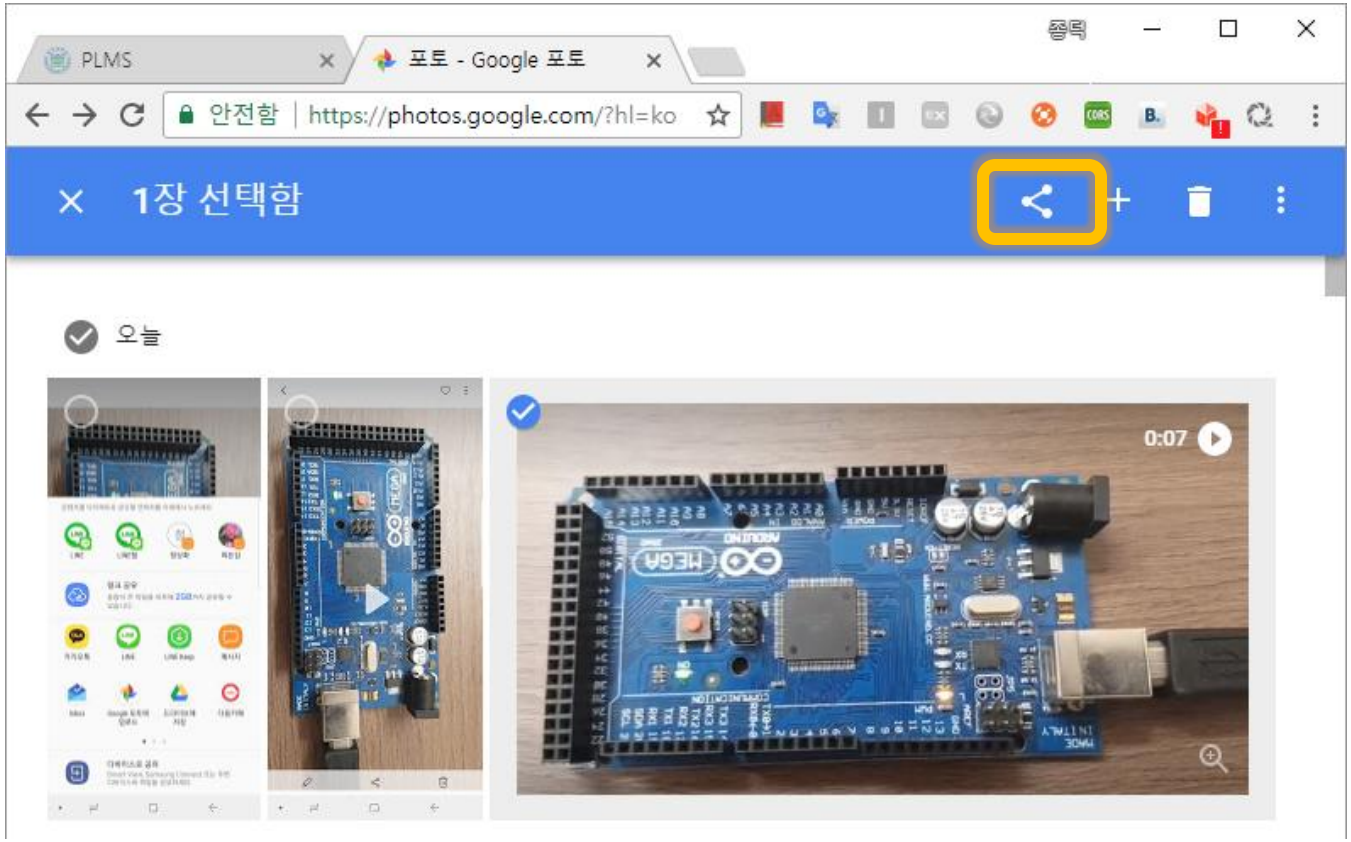
2. If necessary, rotate a video before uploading

- Sometimes, a video clip is saved in vertical mode even if you make a video horizontally. In this case, you can rotate the video clip by using editing mode of Google Photo.
- Pause → Edit → Rotate



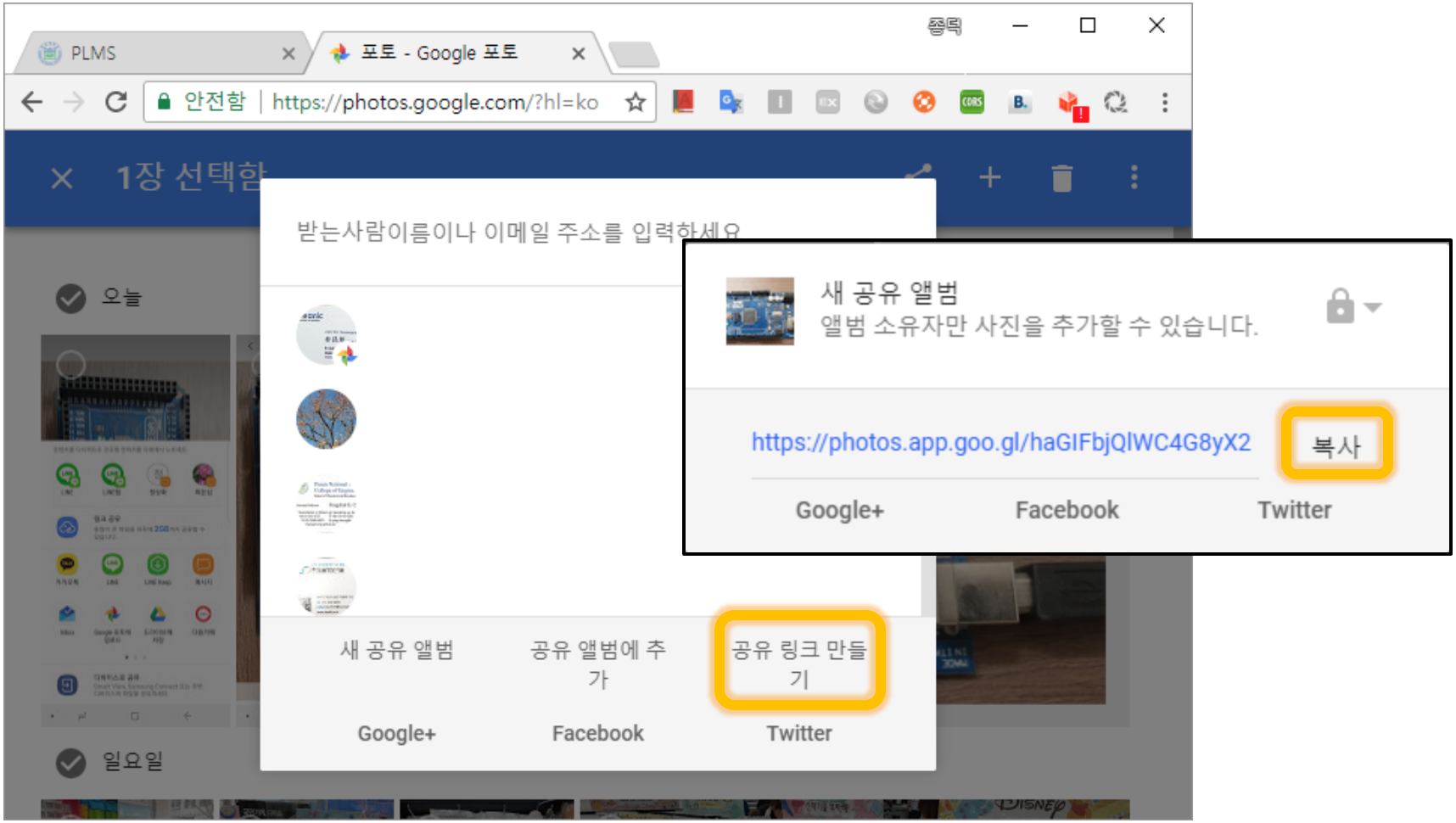
④ Sharing a Video Clip Link in Google Photo Site

- **Check the video you have uploaded at https://photos.google.com/**
  - If you can't, redo previous step ③
- **Select the video clip for submitting, then click the sharing button**



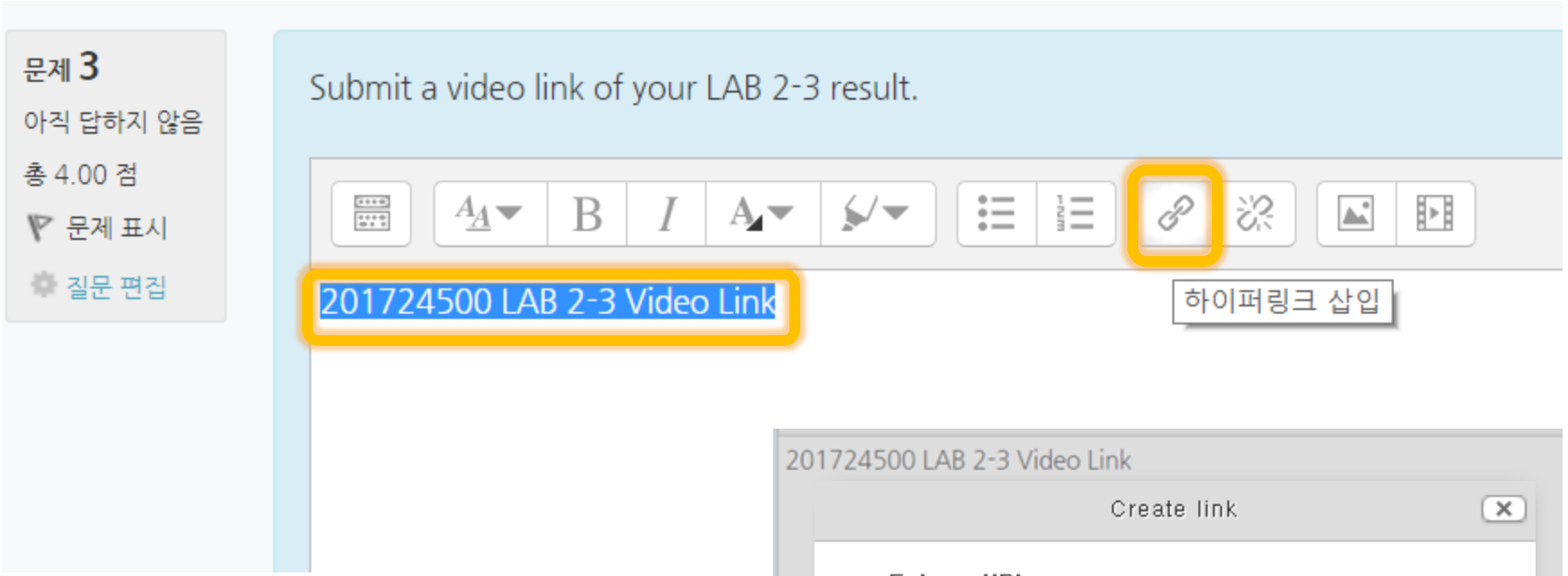
④ Sharing a Video Clip Link in Google Photo Site

❖ Click '공유 링크 만들기', then copy the Link URL



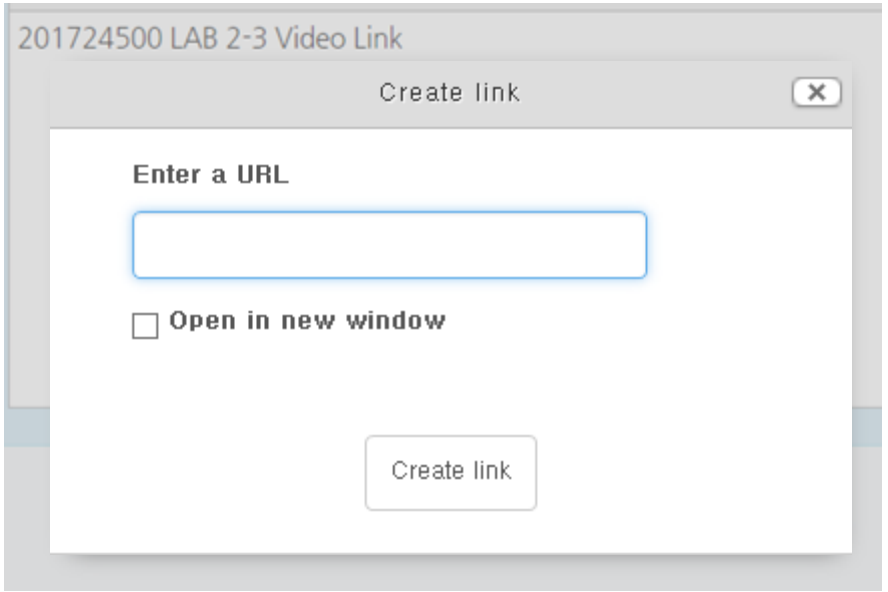
⑤Create a Hyperlink of Video Clip in PLATO

❖ **Insert "{ID} LAB 2-3 Video Link" in PLATO LAB 2-3. Select the string you have inserted, then click the 'Create Link' button.**



❖ **Paste Link URL you have copied in previous step ④ with Ctrl-C.**

- Check 'Open in new window'



Lab.2-3

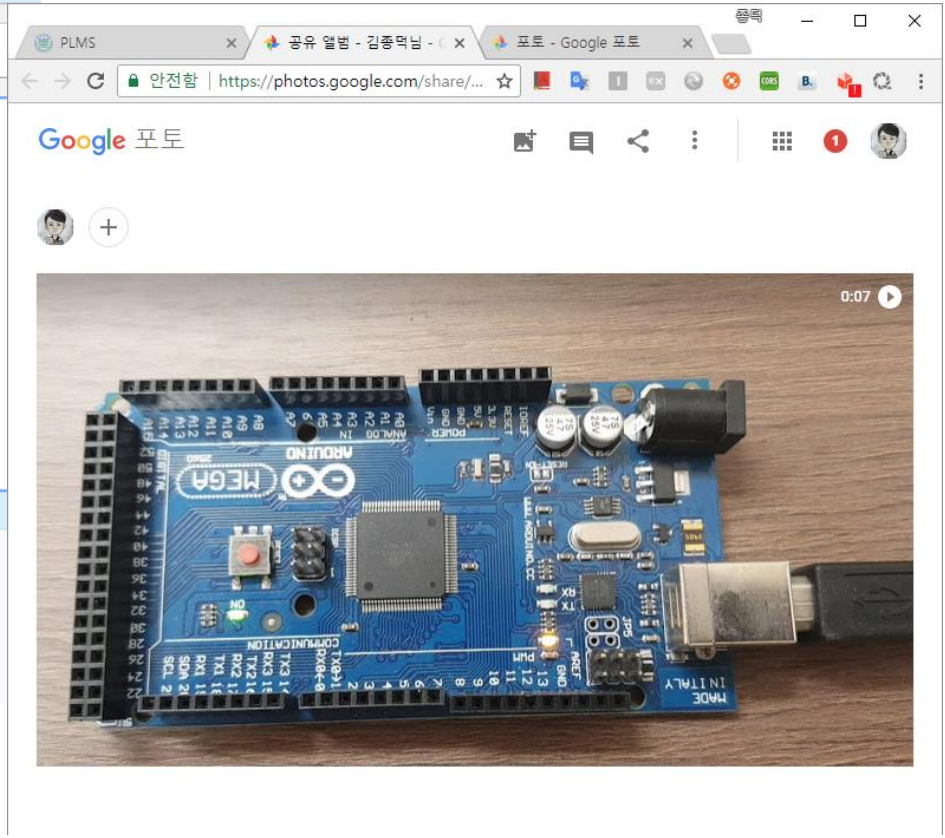
⑤Create a Hyperlink of Video Clip in PLATO

❖ **Befor final submission, please check if the video link is created correctly by right click on the link.**

Submit a video link of your LAB 2-3 result.

201724500 LAB 2-3 Vi

새 탭에서 링크 열기(T)
새 창에서 링크 열기(W)
시크릿 창에서 링크 열기(G)
다른 이름으로 링크 저장(K)...
링크 주소 복사(E)
잘라내기(T) Ctrl+X
복사(C) Ctrl+C
붙여넣기(P) Ctrl+V
일반 텍스트로 붙여넣기 Ctrl+Shift+V
전체 선택(A) Ctrl+A
맞춤법 검사(S)
쓰기 방향



HW.2

Homework

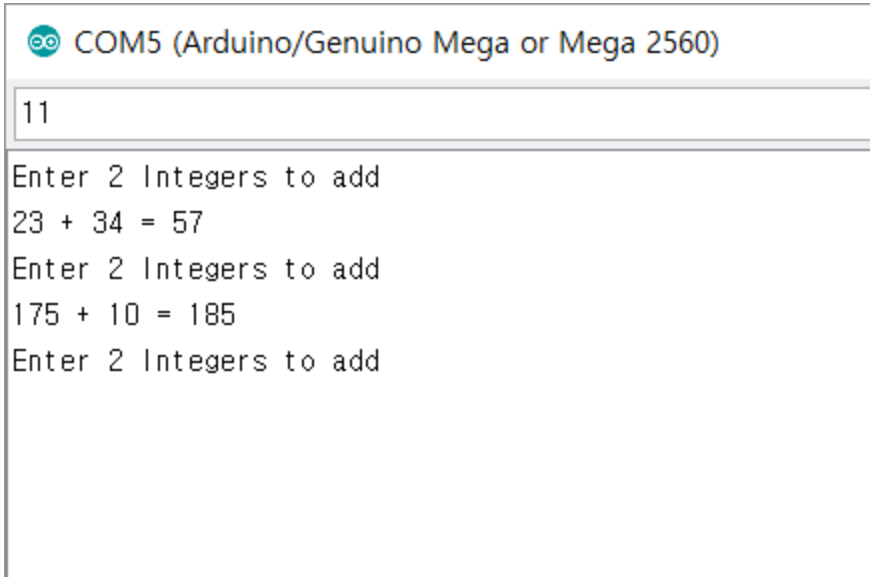
❖ Write a Sketch code that reads two integer inputs from Serial Monitor, then print out the sum of them to Serial Monitor.

- Refer ParseInt() of Serial Class for integer input.
- Before entering integer input, print out the message that requires entering integers.
  - e.g.) "Enter 2 Integers to add"
- How to enter interger inputs from Serial Monitor.
  - Your Sketch has to support two different way for entering inputs. The first way is pushing the 'Send' button for every single integer input. Second is pushing the button only once after entering two integer inputs together. (Each input should be splitted by space.)
- Your Sketch has to perform the operation described above repeatedly.
  - If the Sketch code performs only once, you will get some penalty.

❖ Submission

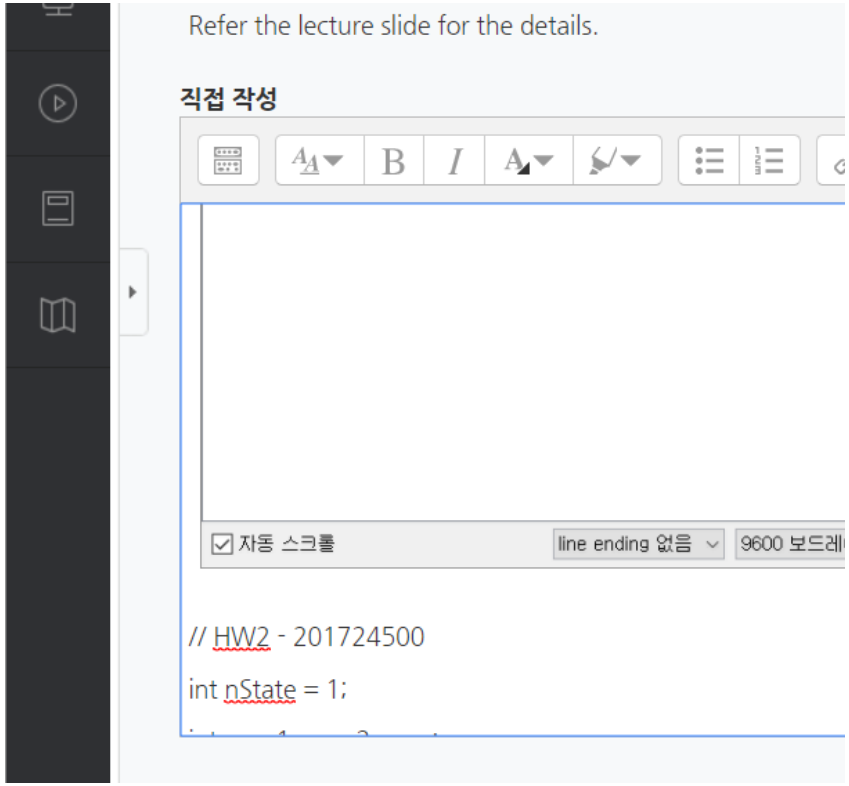
- Submit your handin through HW2 in PLATO.
  1. Insert your Sketch code.
  2. Upload an image capturing the Serial Monitor

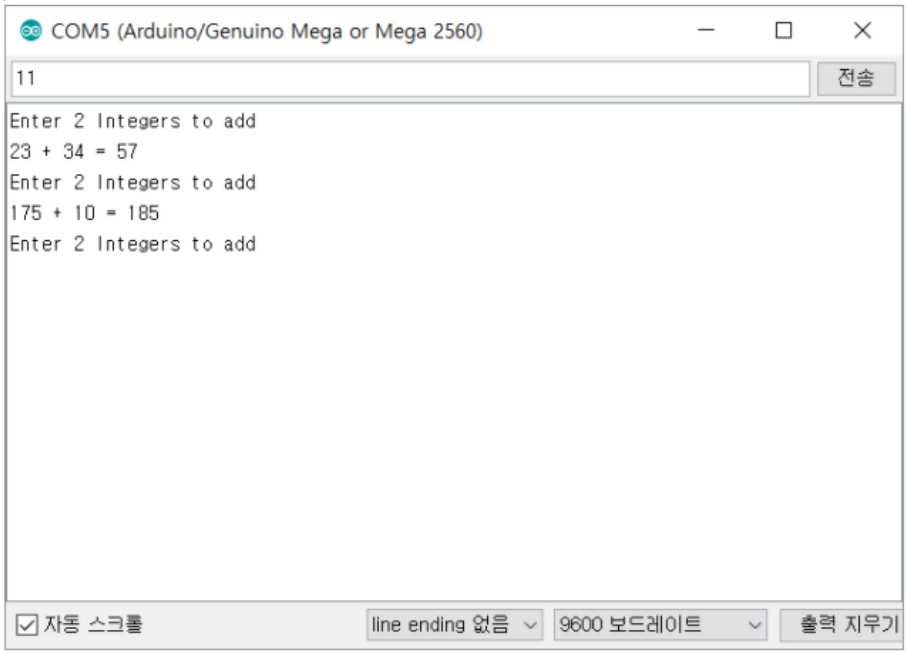
❖ e.g.)



HW.2 Submission Example

- As you can see the following figures, Please submit both your code and capture images






```
// HW2 - 201724500
int nState = 1;
int num1, num2, sum;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}
```

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]