Future Vehicle Education Workshop

Subject: Arduino

Automation Lab.

Introduction

■ Arduino mega 2560

→ Ultra-compact computer to control multiple devices and sensors

Introduction

■ Arduino mega Pin Description

- Power Pins: Pins used to power devices connected to Arduinos
  - (+) Power: 5V or 3.3V, VIN
  - (-) Power: GND

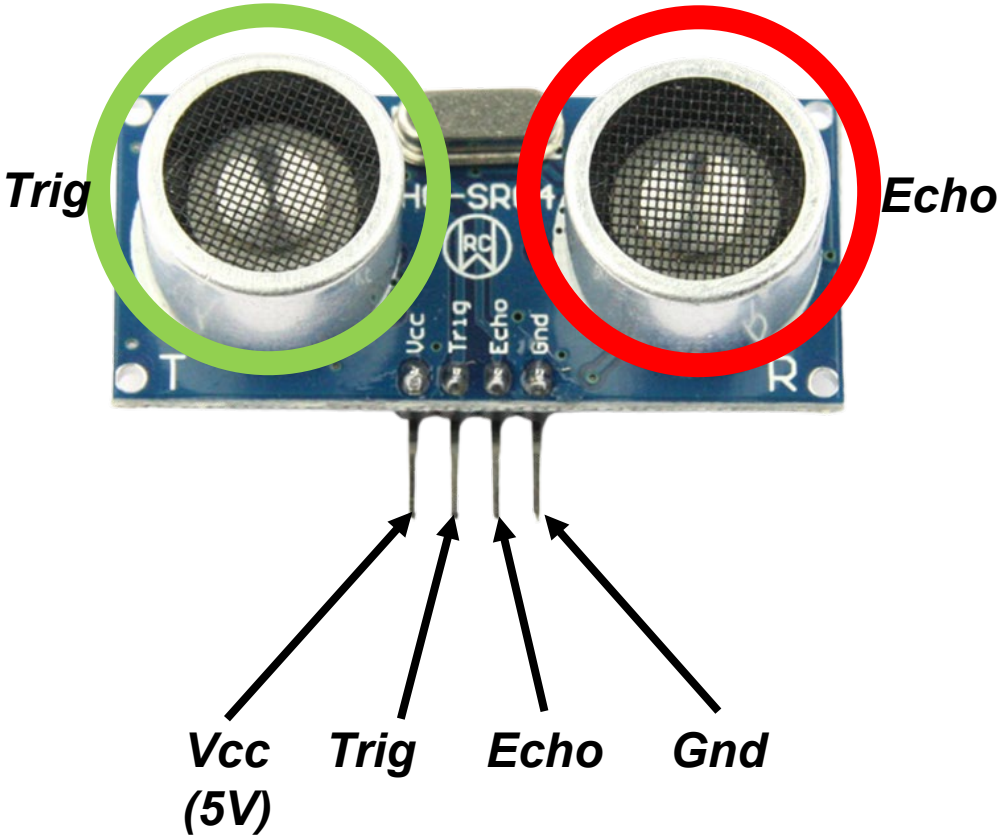
- Digital I/O Pins: Pins that read or send digital signals of 0(LOW) or 1(HIGH)

- Analog input pin: Pin that reads analog signal with an integer value between 0~1023
- Digital Input/Output (PWM) Pins: Convert digital signals like analog signals, 0~255 values
- USB connector: connector used to communicate with the PC, upload the Arduino program

Introduction

■ Ultrasonic Sensors

→ Sensors that use ultrasound to recognize the distance to an object located in front of it

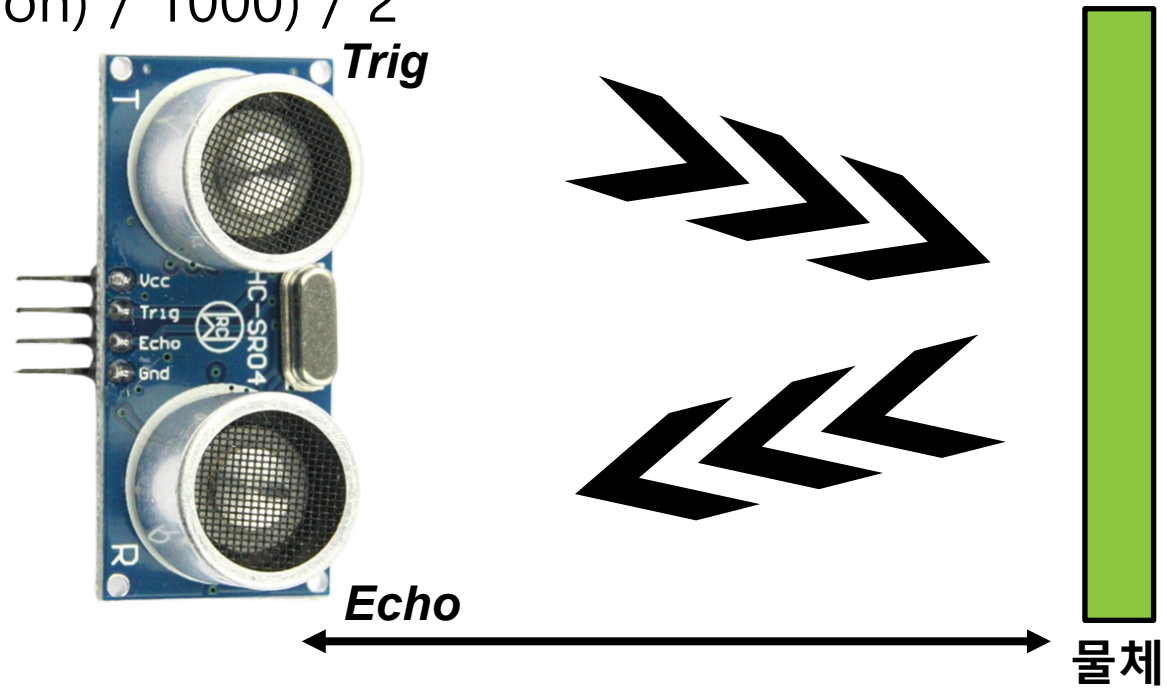


Introduction

■ Ultrasonic Sensor Description

- The ultrasound wave is sent from the transmitter (Trig) and then reflected and returned is recognized by the receiver (Echo)
- Calculate the distance by measuring the time it took from transmission to reception of ultrasound

  ((340 * duration) / 1000) / 2

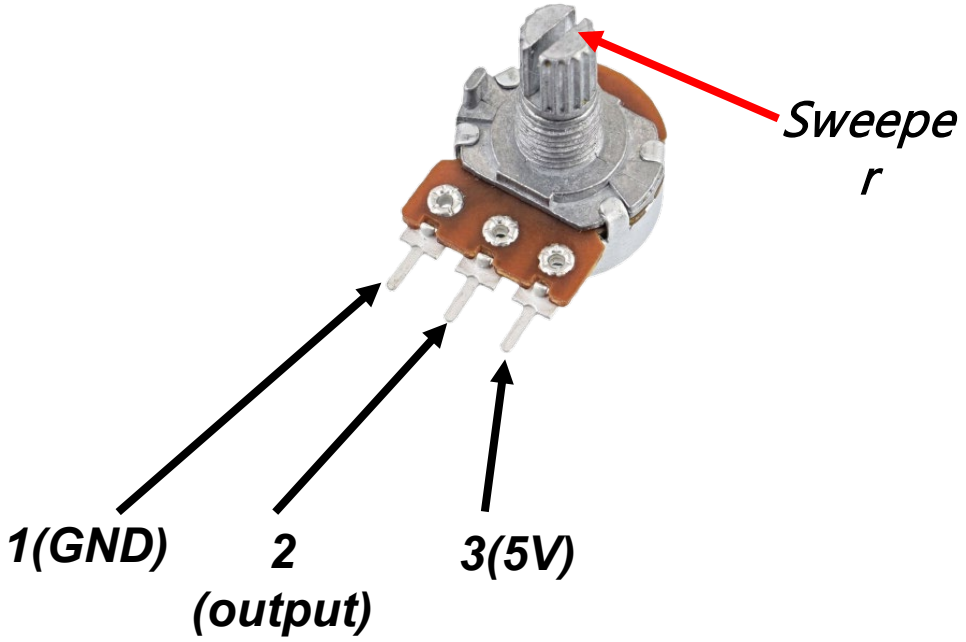


Introduction

■ Variable Resistance

→ Resistance that can change the value of the current flowing through the circuit

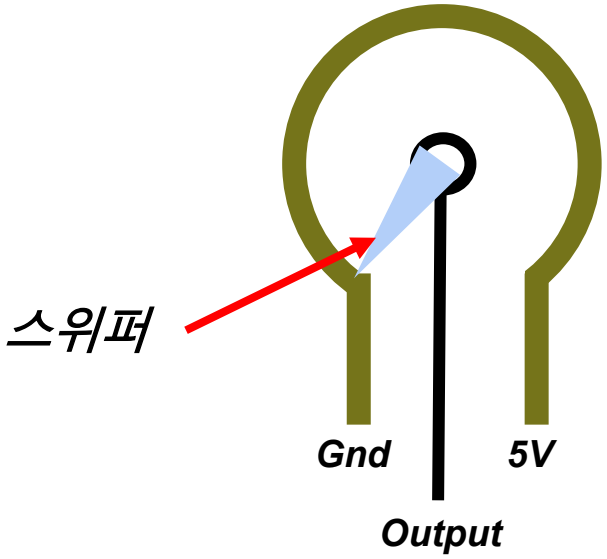
→ Output the strength of the resistor as an analog value via the Output pin



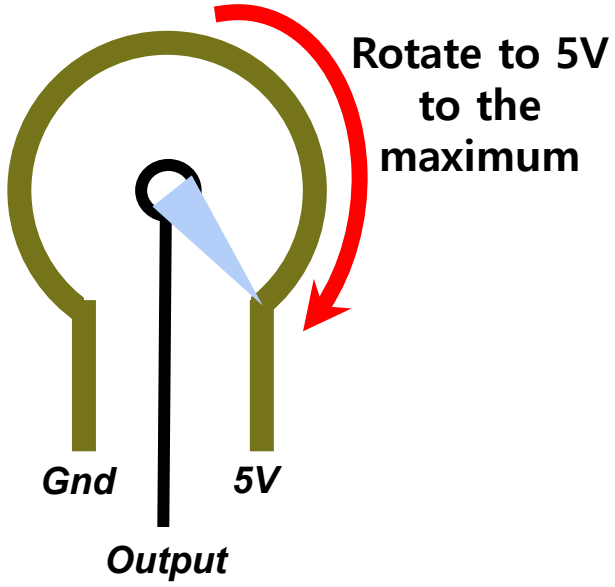
Introduction

■ Variable Resistance Description

→ Move the sweeper to determine the resistance strength of the variable resistance
→ Turn the sweeper to the side connected to 5V to output the maximum value, and turn the sweeper to the side connected to GND to output the minimum value



<Output minimum value (0)>



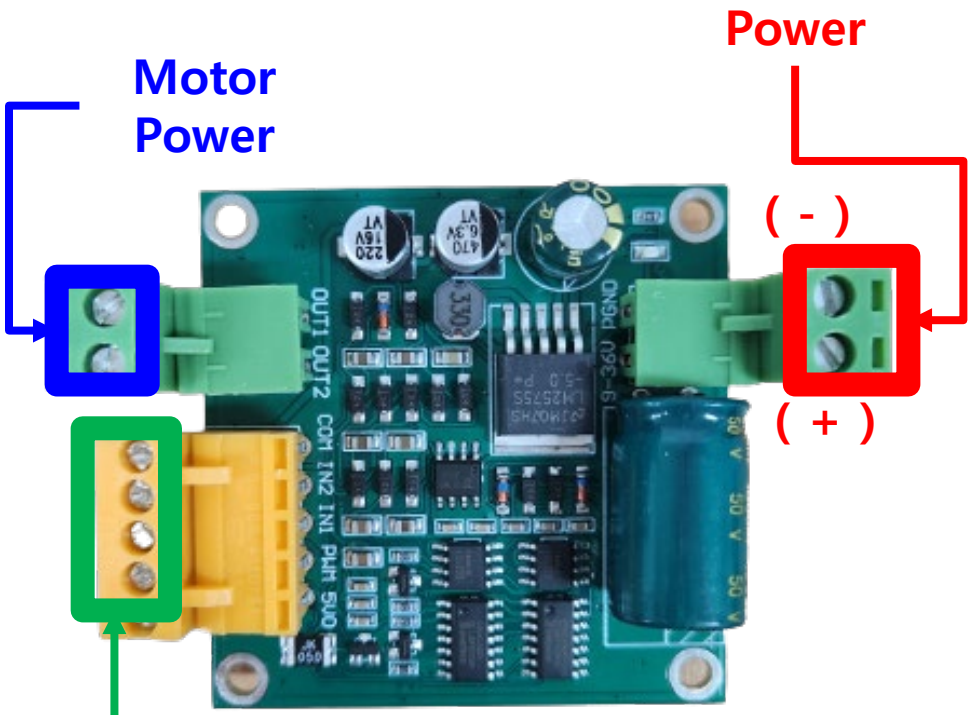
<Output maximum value (1023)>

Introduction

■ Gear Motor & Motor Driver

→ Motors designed to generate high torque
→ Motor driver for motor power supply

Gear Motor(12V)



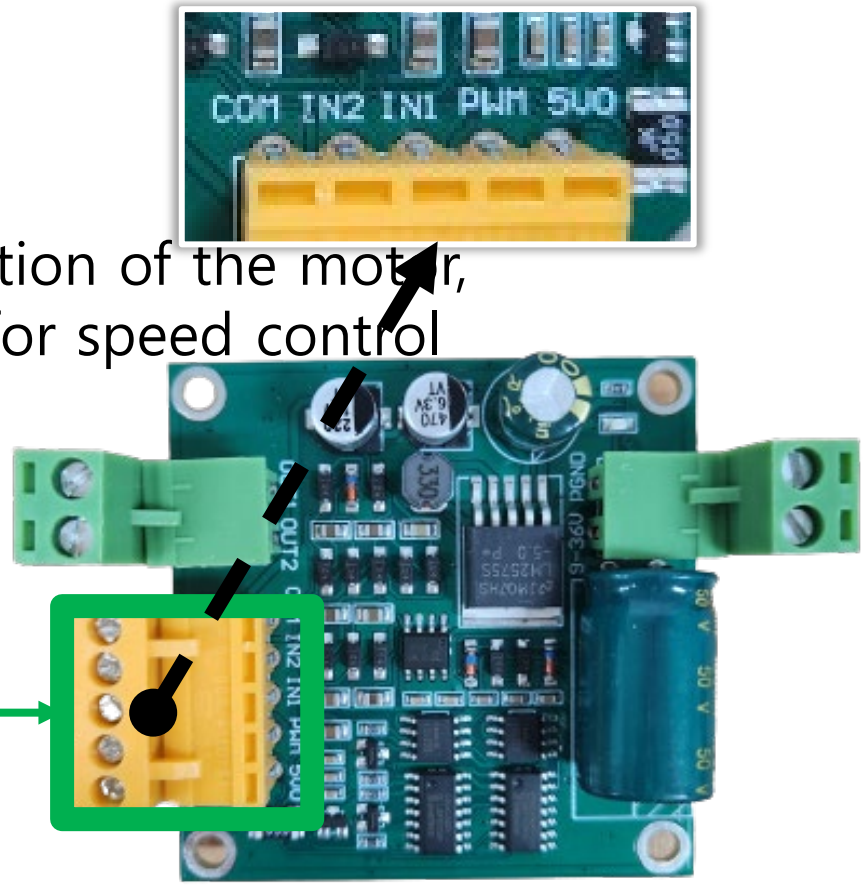
Motor Driver

Introduction

■ Motor Driver Control Description

- COM Pin: Connecting with Arduino's GND
- PWM Pin: Connects with Arduino's 5V
- IN1, IN2 pins: Determining the direction of rotation of the motor, connecting to the digital pins Connects to PWM pins for speed control

IN1	IN2	output
0	0	stop
1	0	Forward rotation
0	1	Reverse rotation



Motor Control

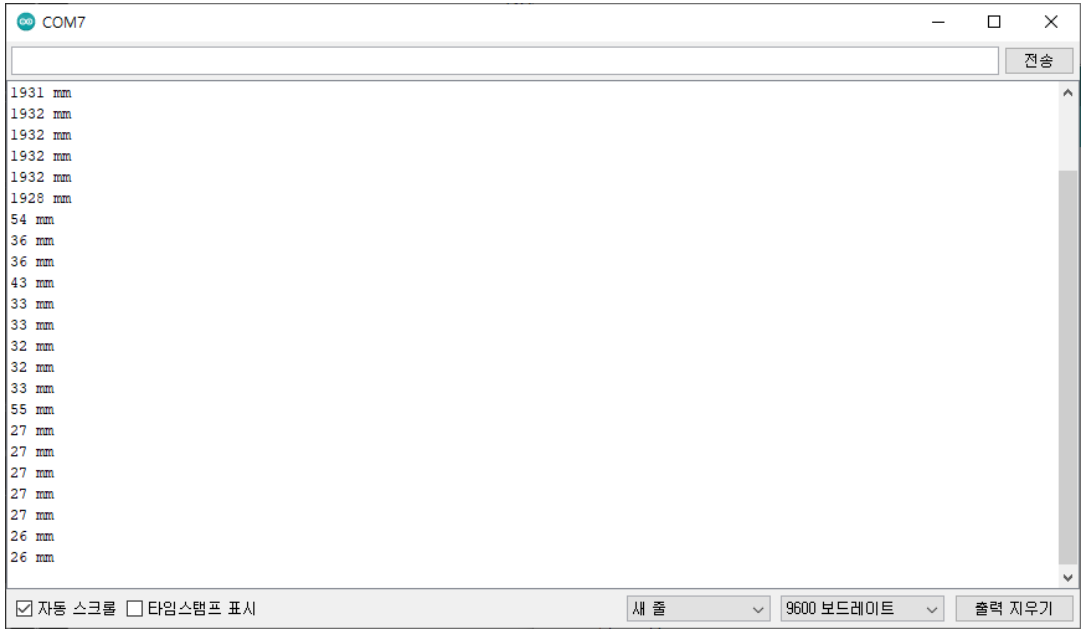
Motor Driver

Introduction

■ Serial Communication

→ How Arduino communicates with your PC
→ It is possible to transmit or receive data with Arduino.

USB connection to PC (laptop)



<Serial Monitor Screen>

Exercise

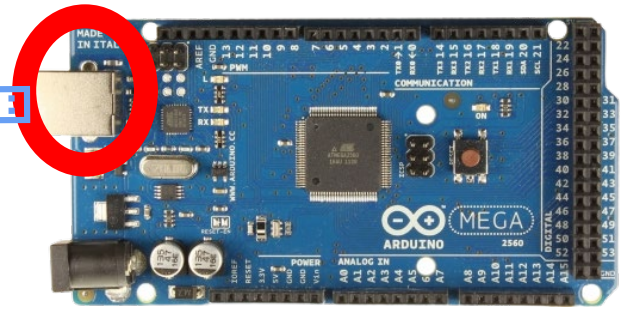
Embedded System Lab.

1. Hardware Setting

■ Arduino Hardware Setting

→ Must be preceded by software setup
→ Connect the USB cable directly to the USB port on your laptop (PC)

Notebook(PC)



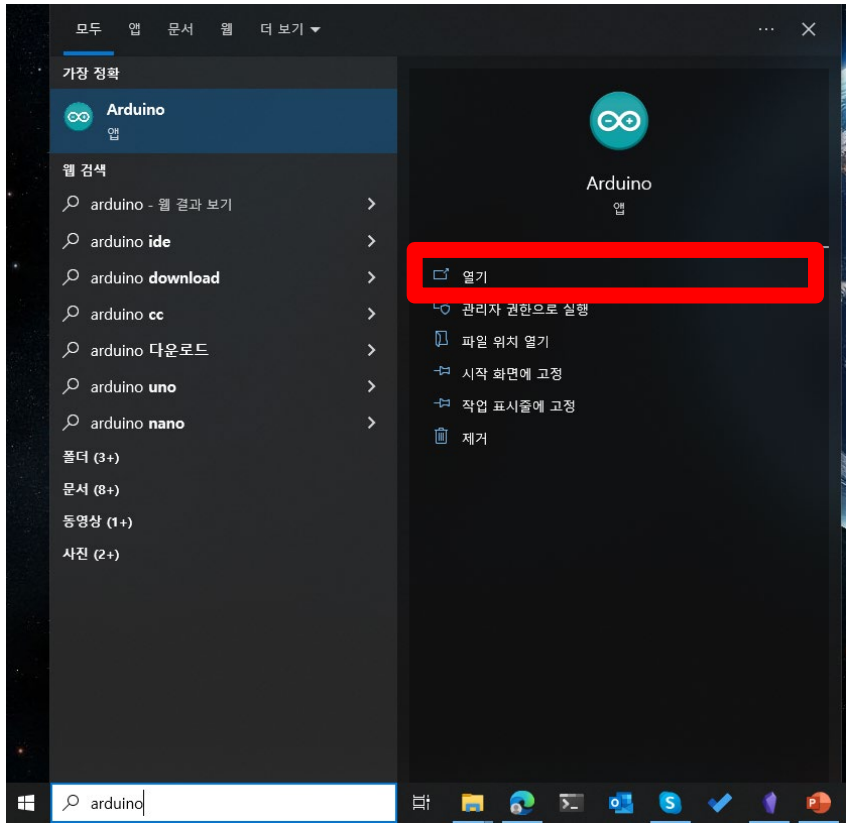
Arduino board

2. Software Setting

■ Launch the Arduino app

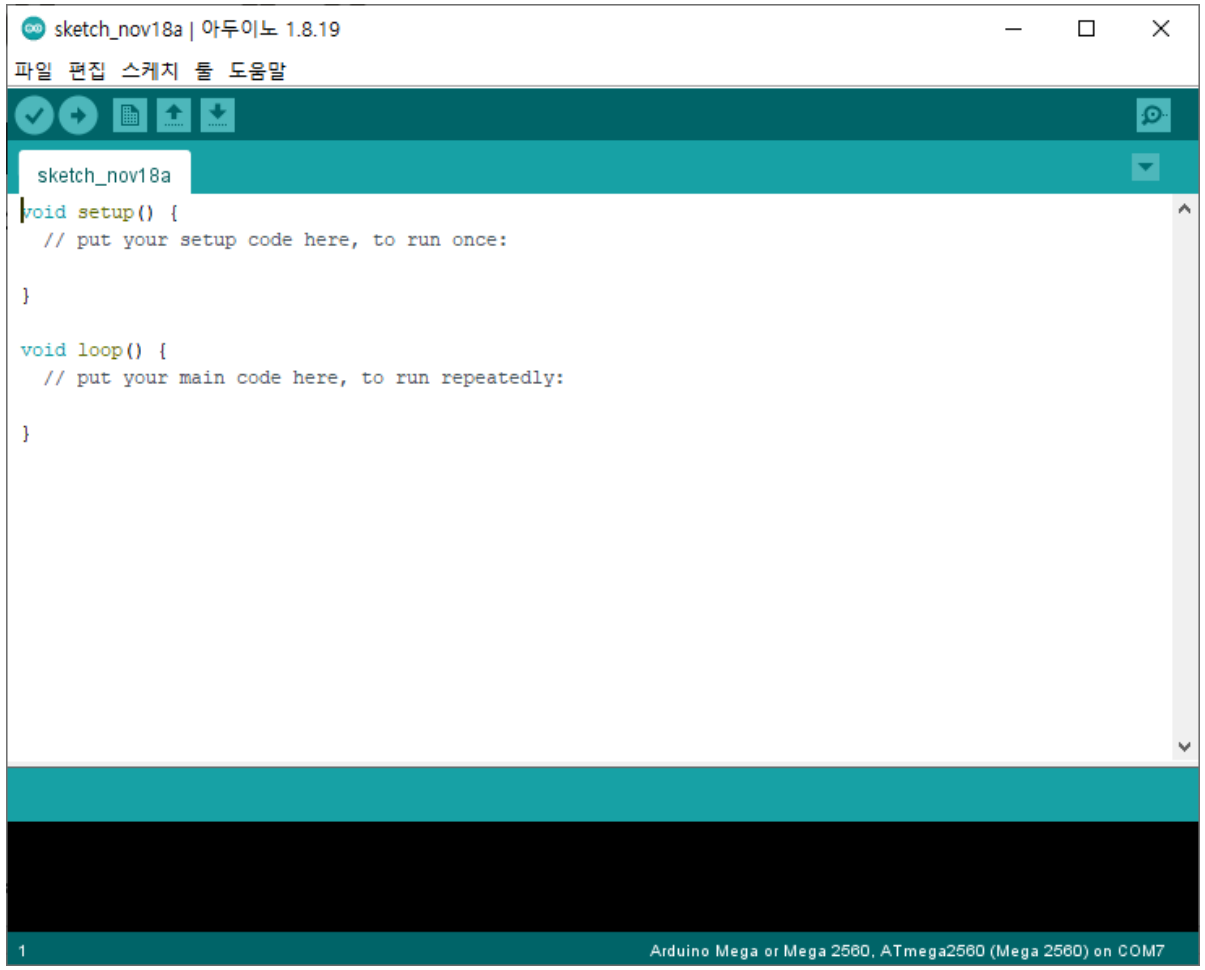
→ Type "Win key+q" on your keyboard, then type "arduino" and click "Open"

→ To install the app, refer to the “Environment Setting" material!



2. Software Setting

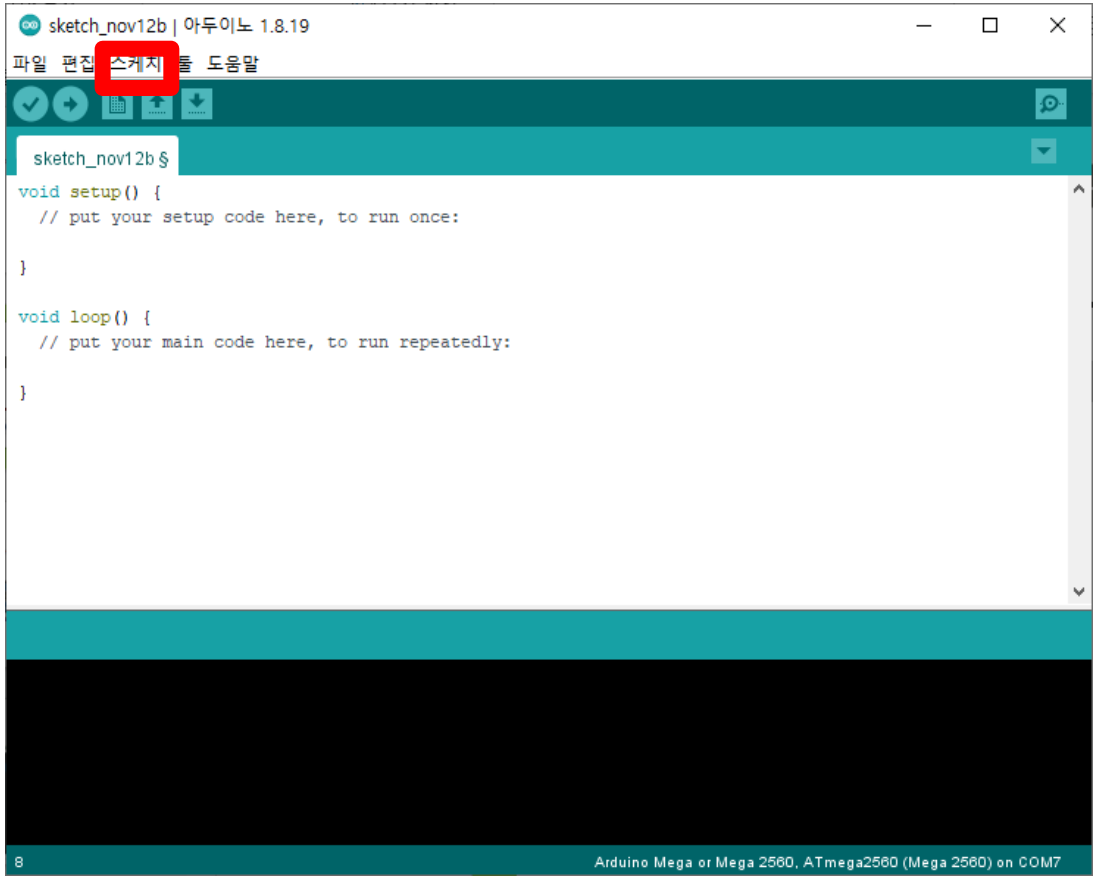
■ Arduino app initial screen



2. Software Setting

■ Add a library

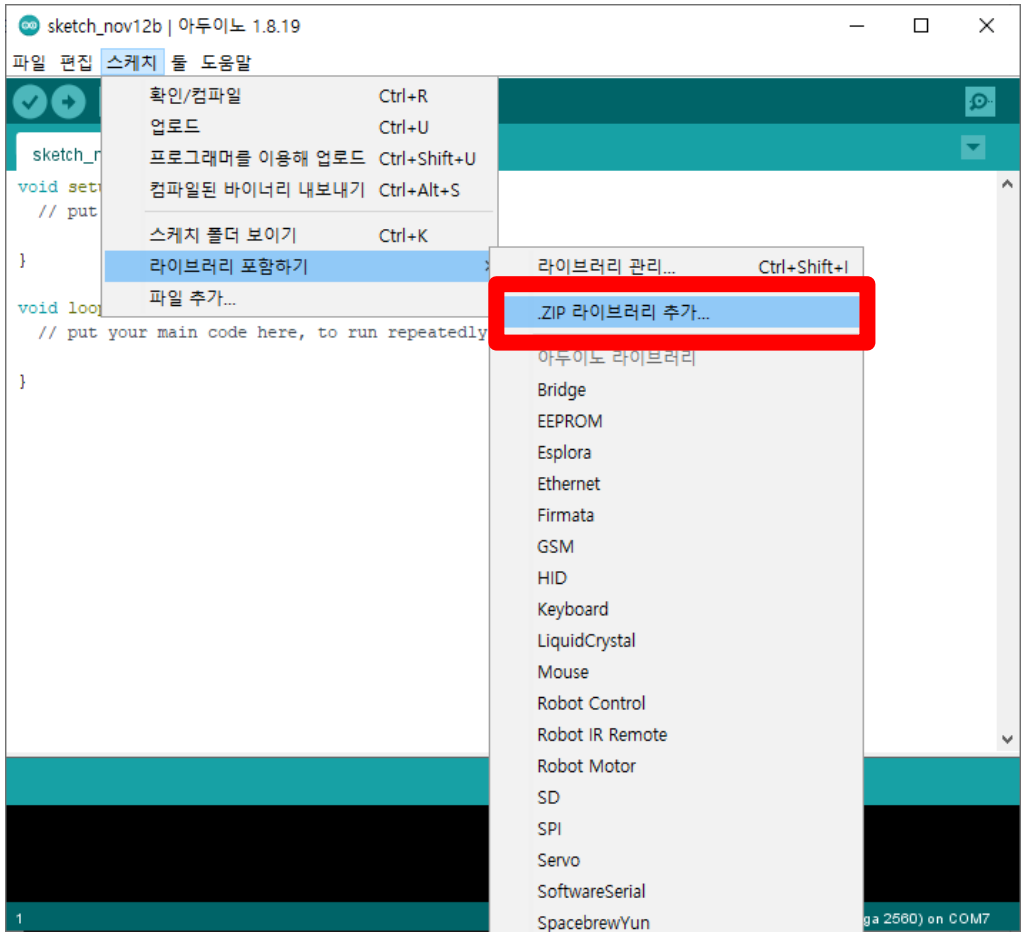
→ Click "Sketch"



2. Software Setting

■ Add a library

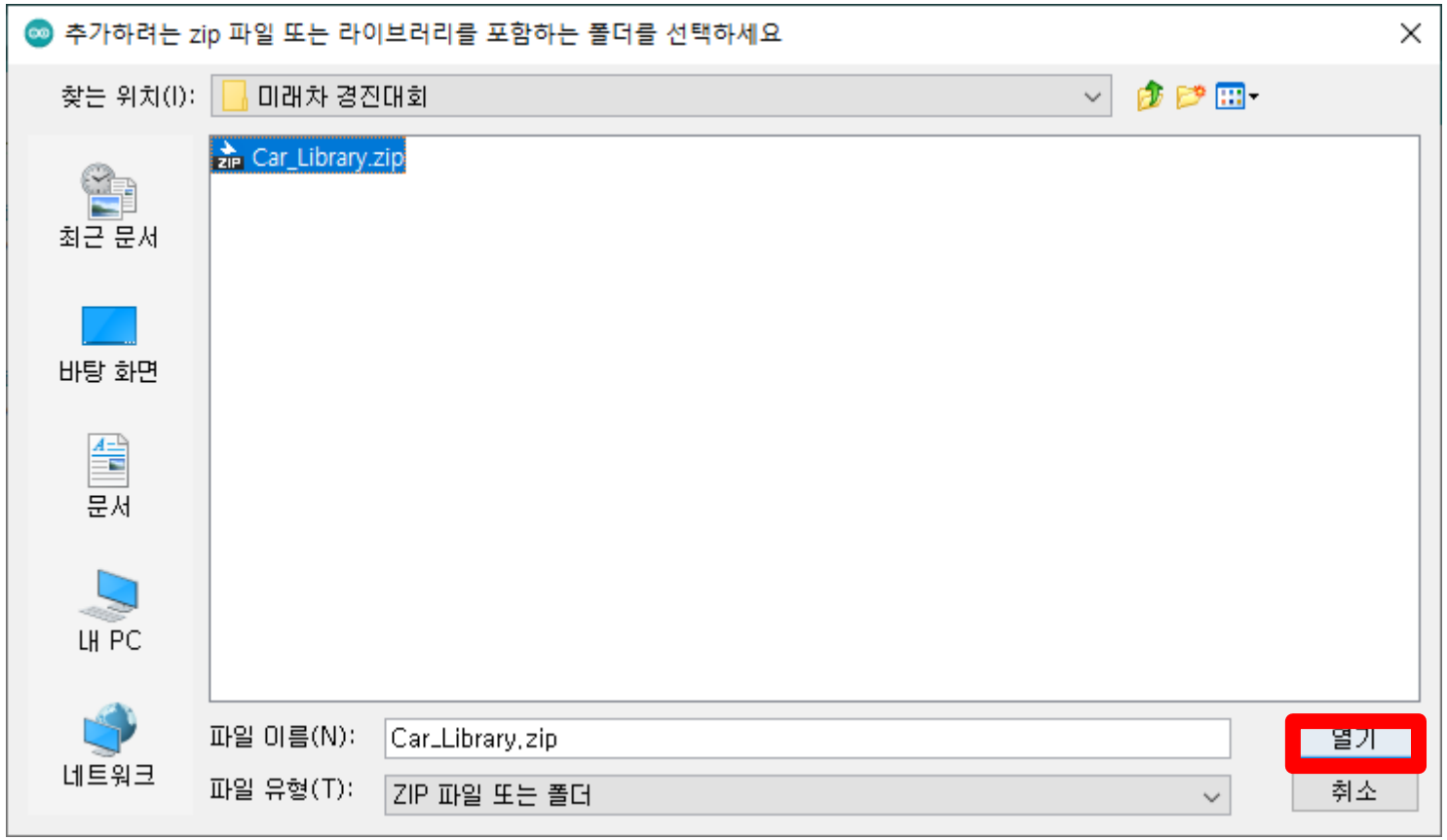
→ "Embed a library"-> Add .ZIP libraries... Click



2. Software Setting

■ Add Library

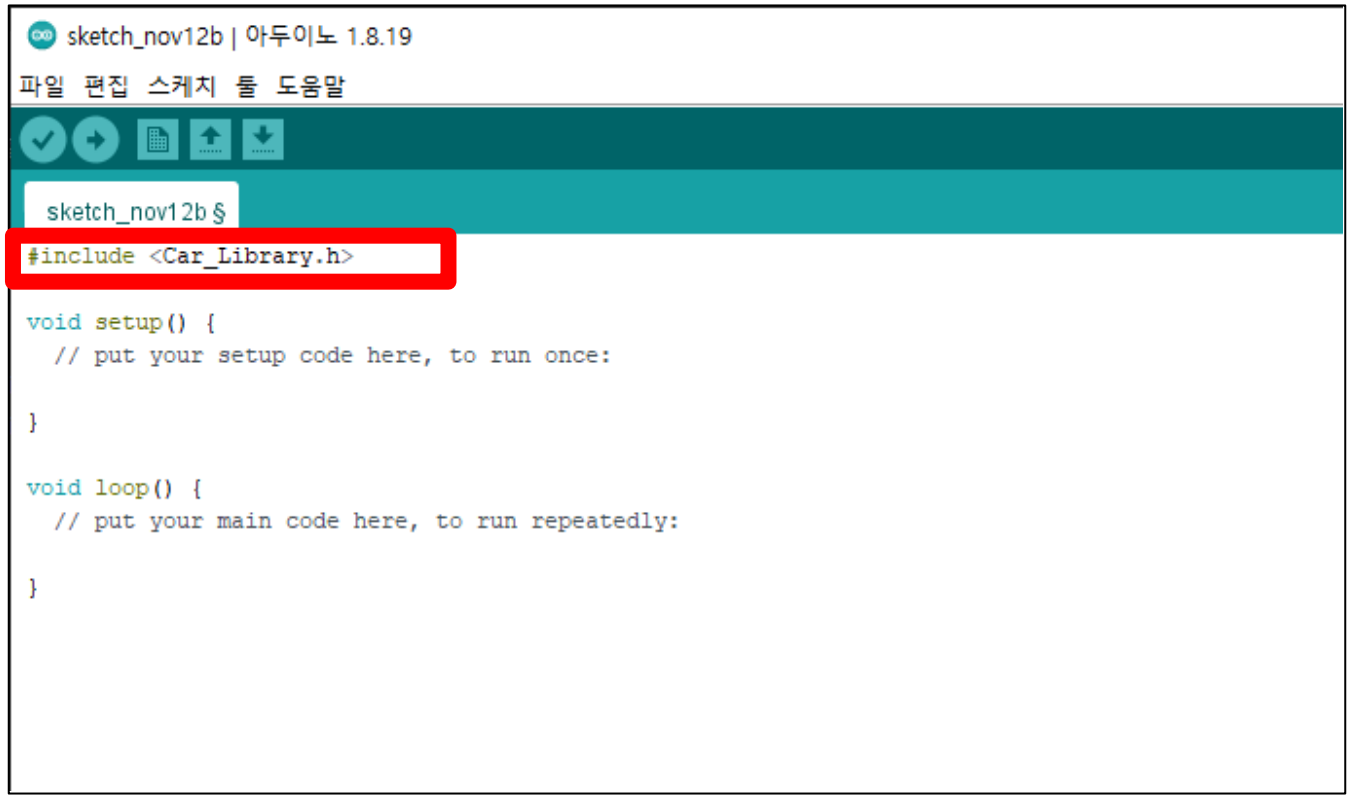
→ When that window appears, select the "Car_Library" compressed file downloaded from Github and click "Open"



2. Software Setting

■ Add a library

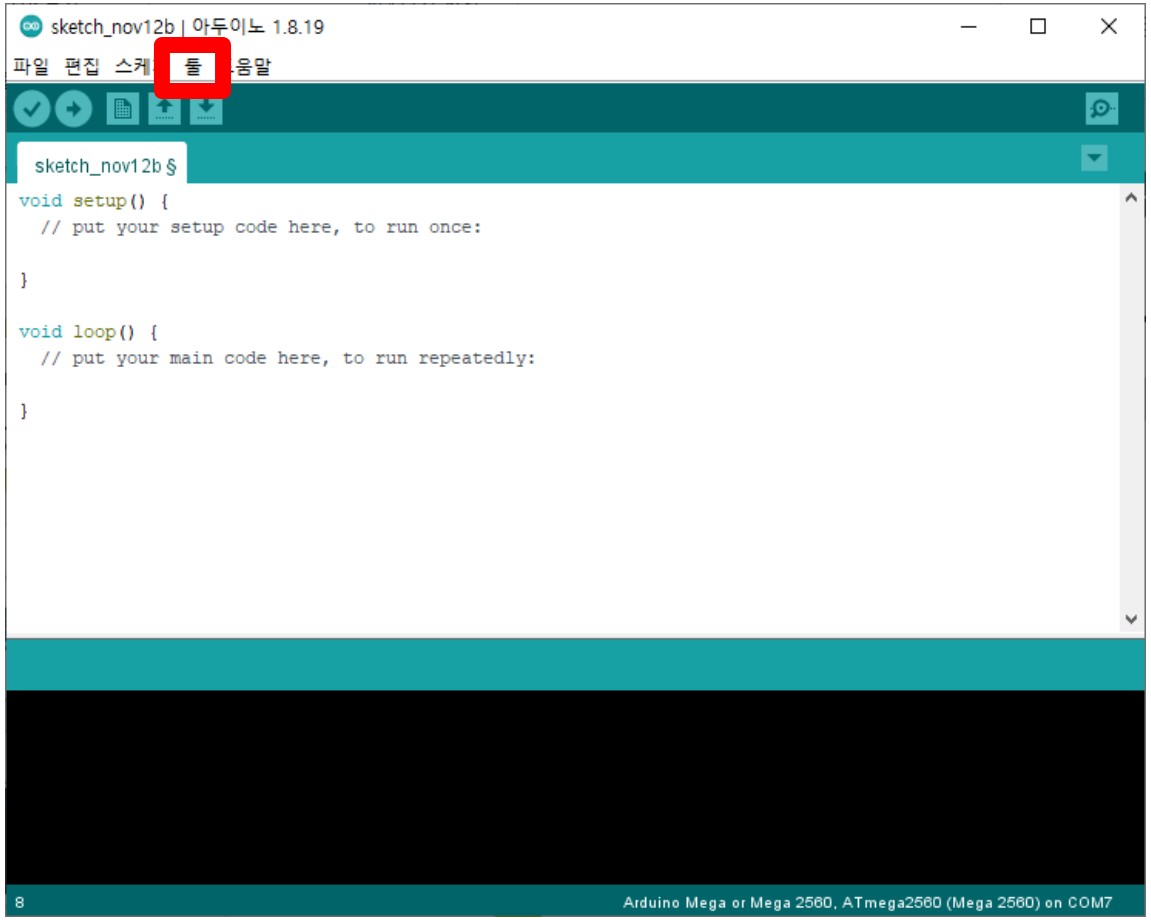
→ Enter "#include <Car_Library.h>" at the top of the source code



2. Software Setting

■ Board Settings

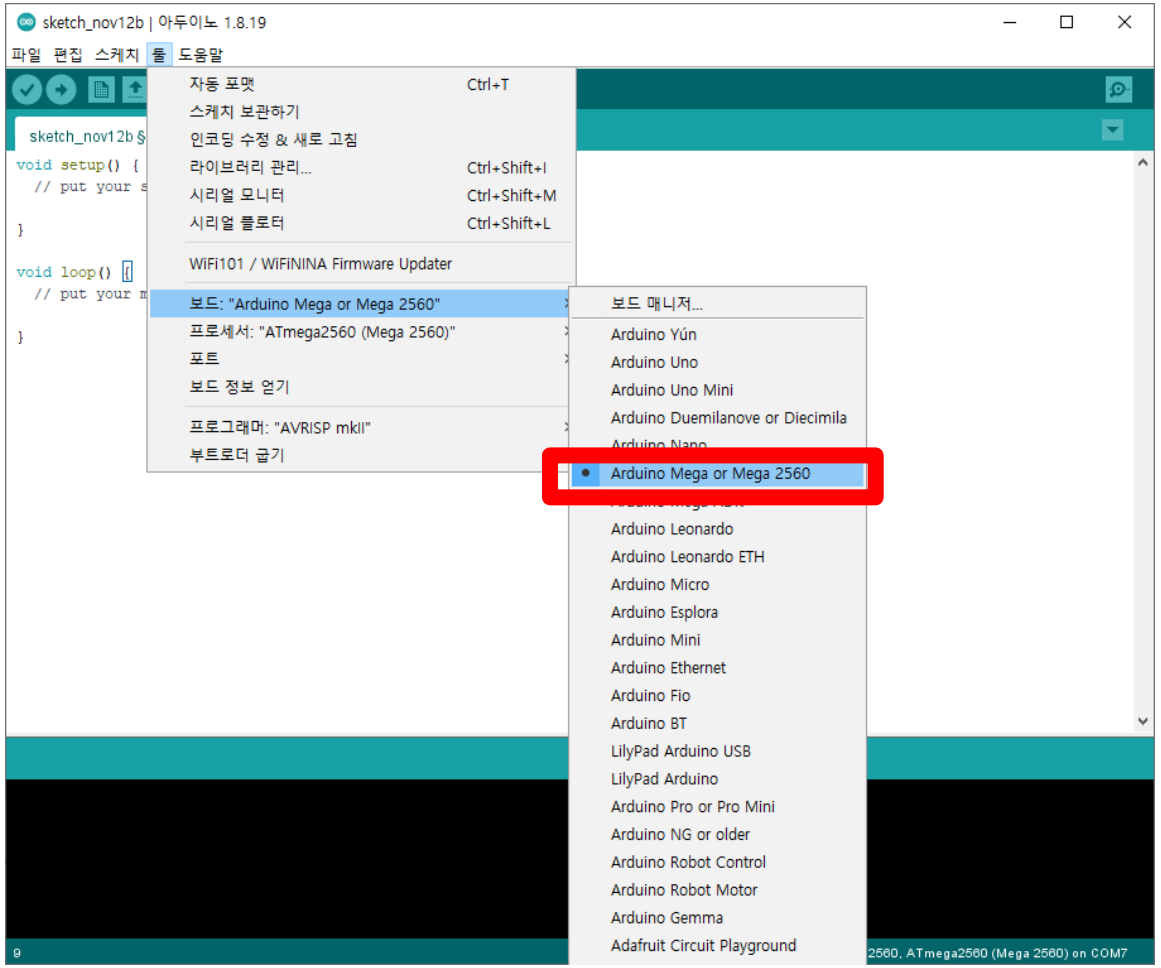
→ After launching the Arduino app, click "Tools"



2. Software Setting

■ Board Settings

→ Tool -> Board -> Select “Arduino Mega or Mega 2560”

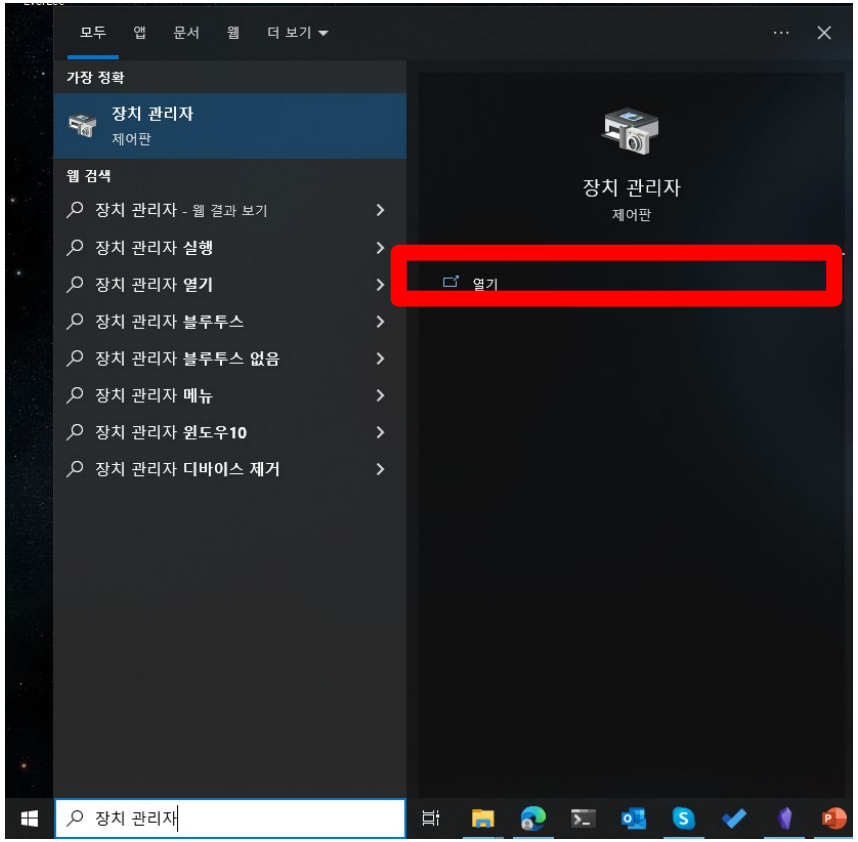


2. Software Setting

■ Serial Connection

→ Type "Win key+q" on your keyboard, then type "Device Manager" and click "Open"

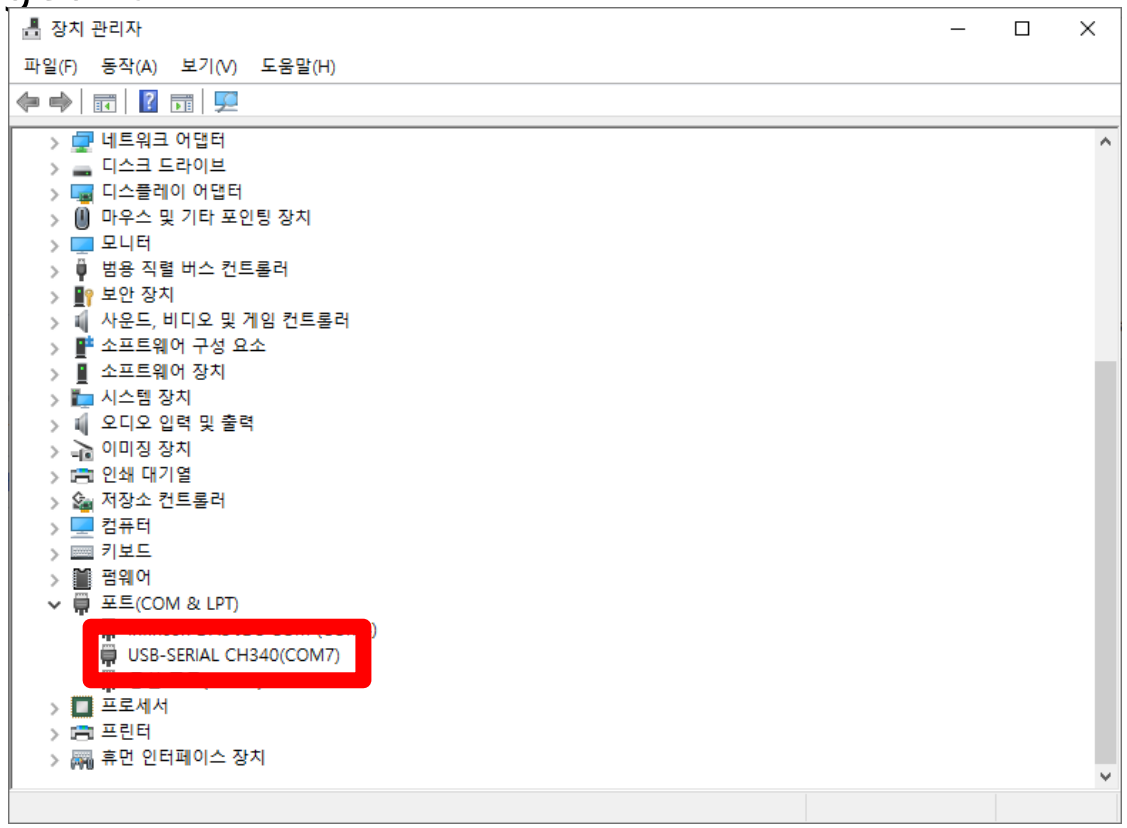
→ Arduino and PC must be connected



2. Software Setting

■ Check the serial port number

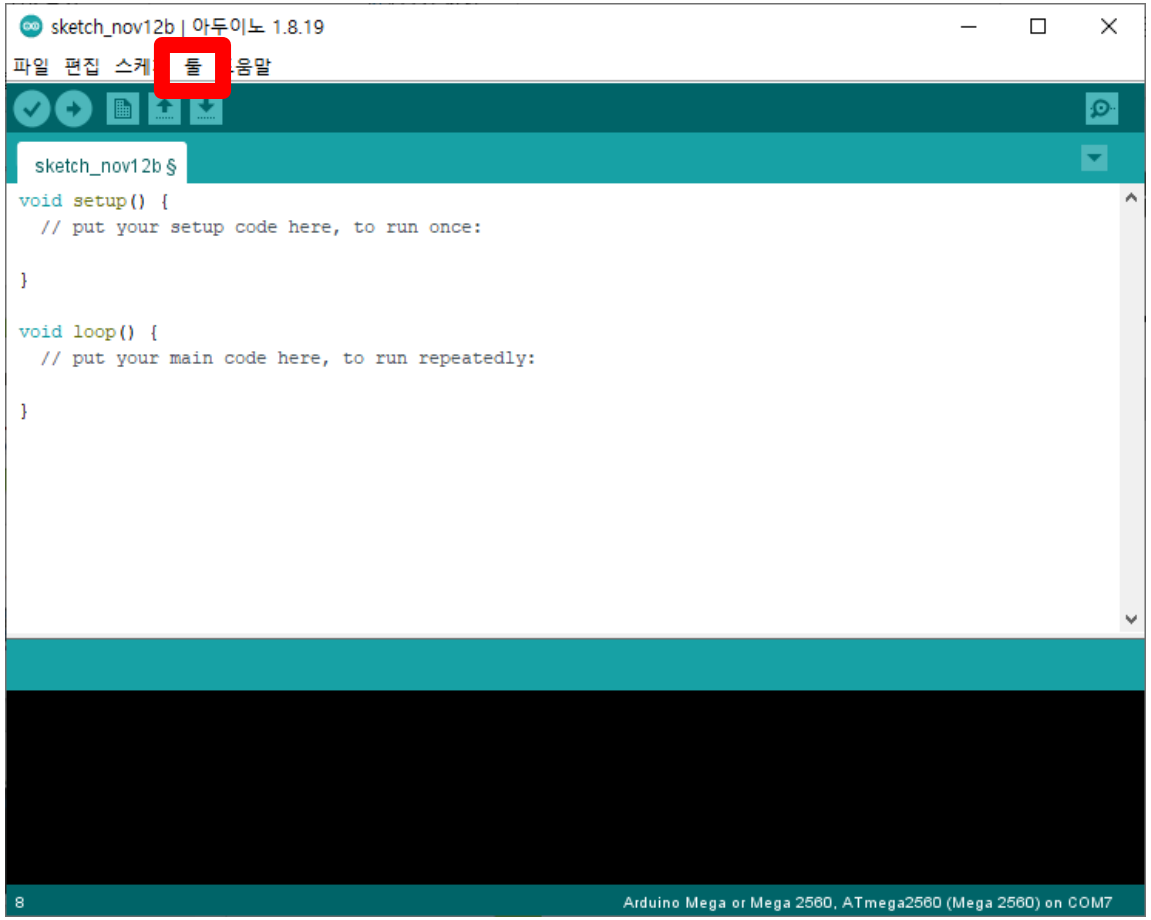
→ Device Manager - > Ports (COM & LPT) - > Check USB-SERIAL Port Number (COM*)
→ The number may change if you plug the USB cable into a port other than the one you originally plugged it in



2. Software Setting

■ Serial communication connection

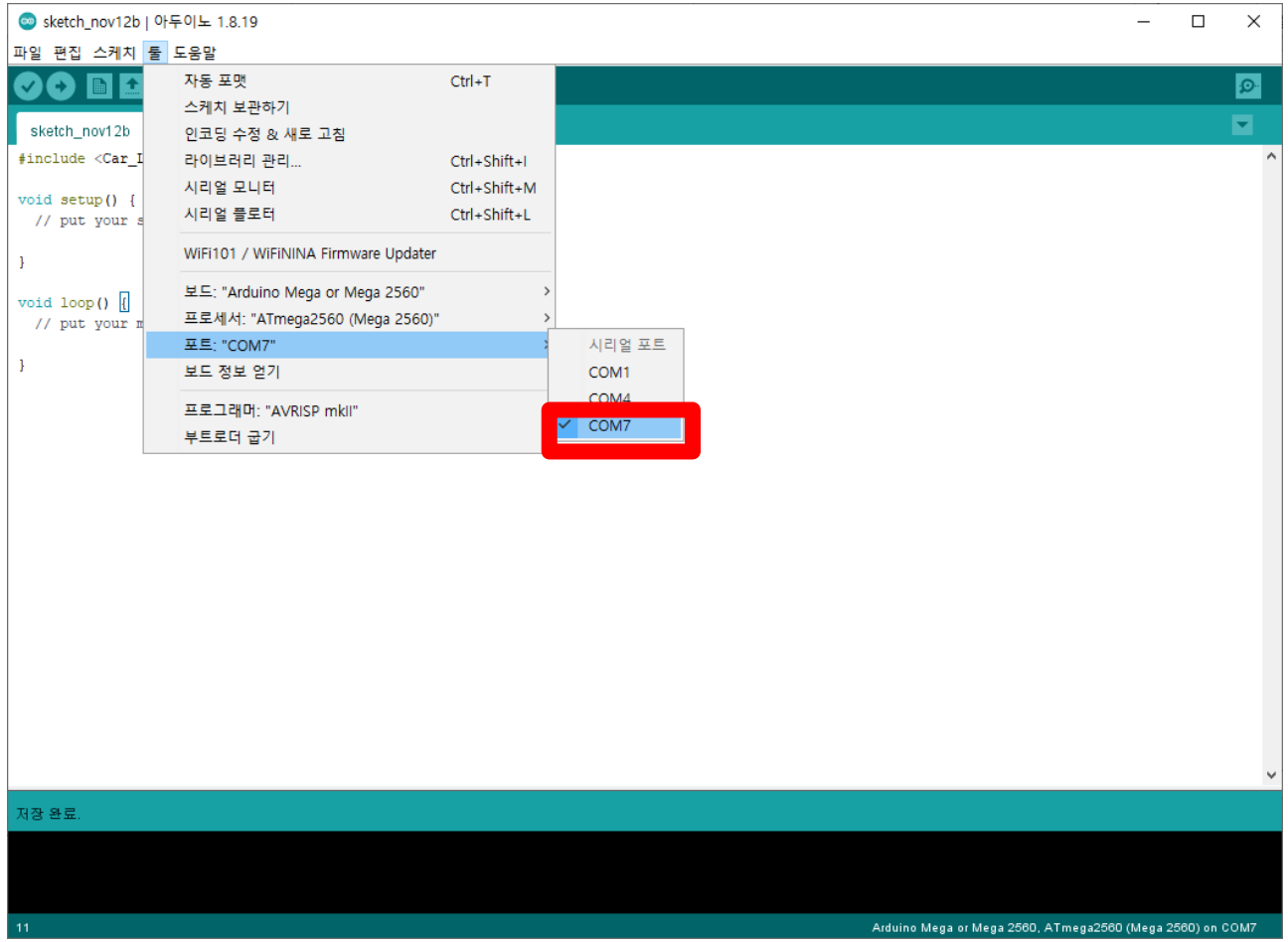
→ After launching the Arduino app, click "Tools"



2. Software Setting

■ Serial Port Settings

→ Tools -> Port -> Select the port number that you check before



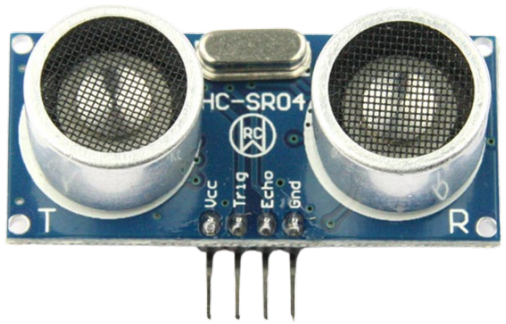
Exercise 1

Exercise 1

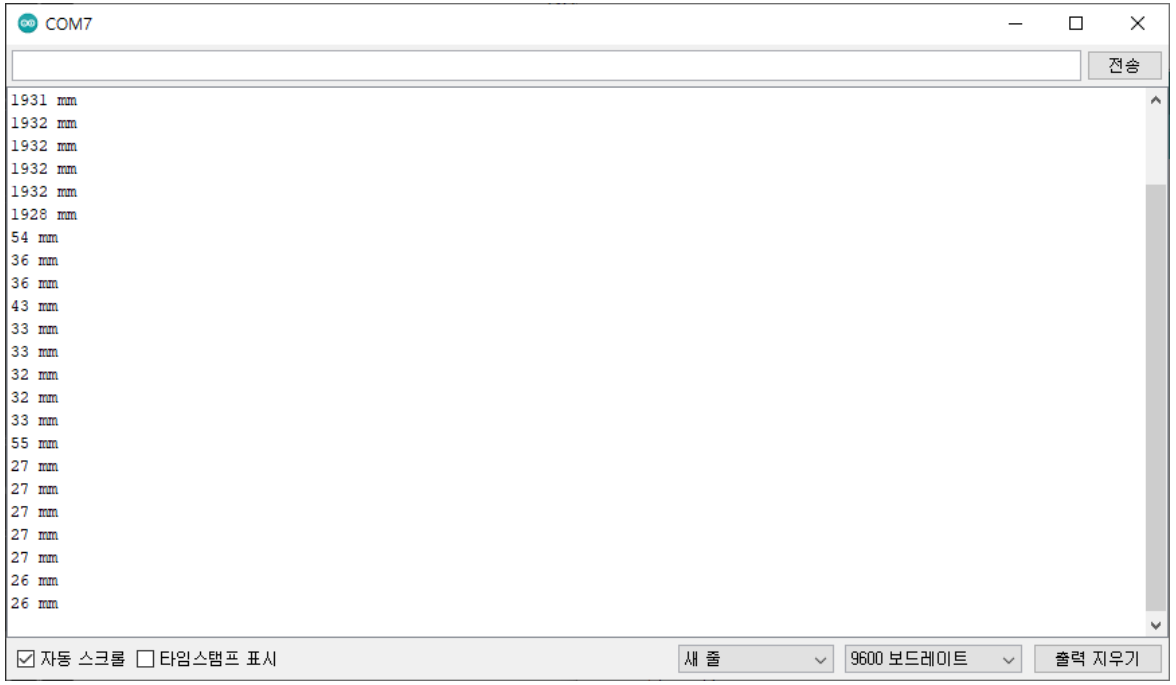
■ Measuring distances with ultrasonic sensors

→ Uses ultrasonic sensors to measure the distance to objects located in front of you

→ Output measured distance to serial communication



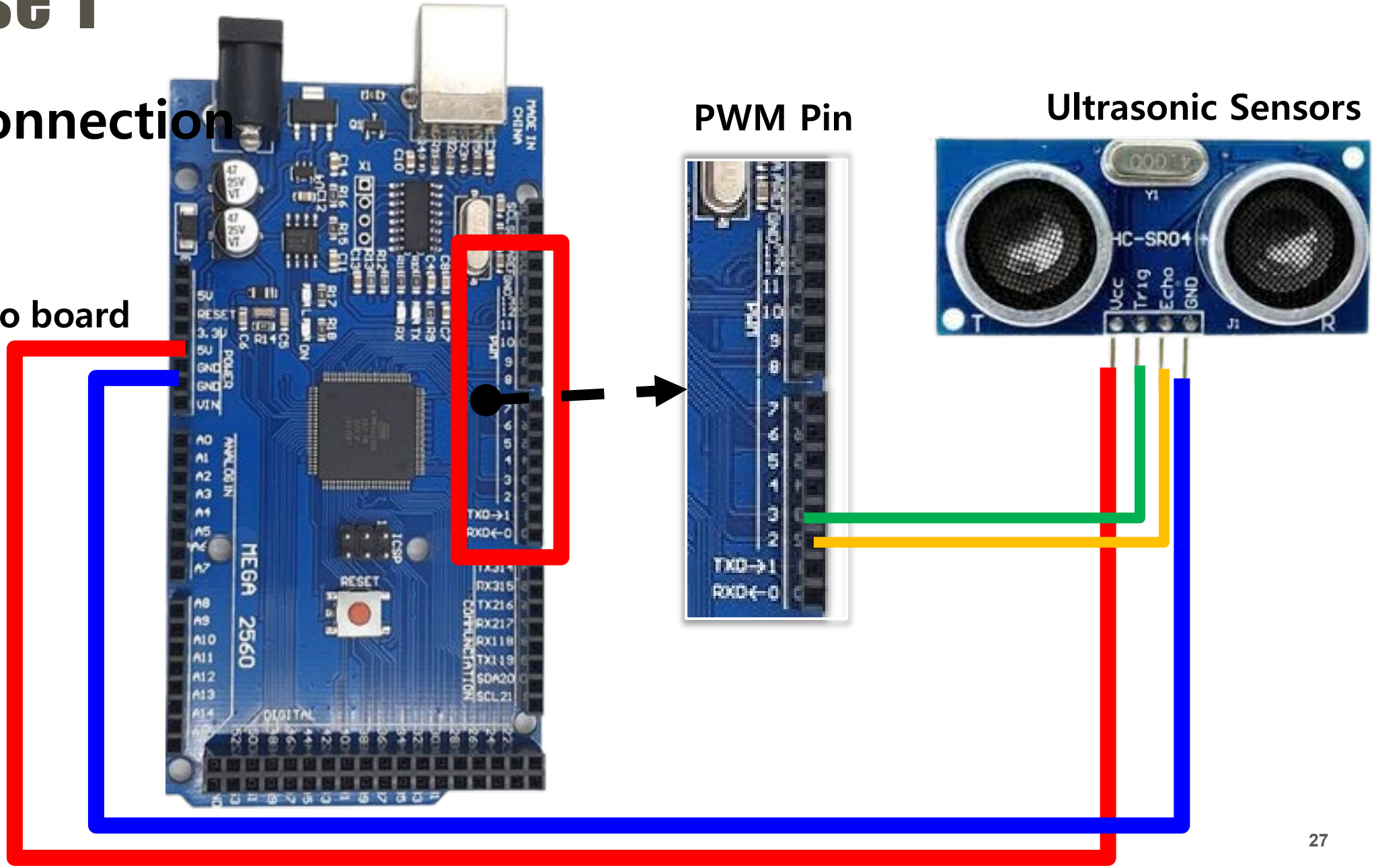
초음파 센서



결과 화면

Exercise 1

- Line connection



Exercise 1

■ Line connection (Cont'd)

→ Ultrasonic sensor VCC – connected to Arduino 5V
→ Ultrasonic sensor GND – connected to Arduino GND
→ Ultrasonic sensor Trig – Arduino PWM connected to pin 3
→ Ultrasonic sensor Echo – Arduino PWM connected to pin 2

Exercise 1

■ Declaring a Pin Number Variable

→ Trig of ultrasonic sensor, declaration of the Arduino pin number variable linked to echo

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b §

```
#include <Car_Library.h>

int trig = 3;      // trig Pin
int echo = 2;      // echo Pin

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
  pinMode(trig, OUTPUT);     // trig 핀 모드 설정
  pinMode(echo, INPUT);      // echo 핀 모드 설정
}
```

Exercise 1

■ Serial Communication and Pin Mode Settings

→ Start serial communication and set the mode of the connected pin

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b §

```
#include <Car_Library.h>

int trig = 3;      // trig Pin
int echo = 2;      // echo Pin

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
  pinMode(trig, OUTPUT);     // trig 핀 모드 설정
  pinMode(echo, INPUT);      // echo 핀 모드 설정
}
```

Exercise 1

■ Function description

- **Serial.begin(Communication speed)**

```
Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
```

→ Start serial communication at the input communication speed

- **pinMode(Pin Number, mode)**

```
pinMode(trig, OUTPUT);     // trig 핀 모드 설정
pinMode(echo, INPUT);      // echo 핀 모드 설정
```

→ Set the mode of the pin you entered.
→ Can be set to OUTPUT or INPUT
→ In that example, trig sends ultrasonic waves, so OUTPUT,
Since echo receives reflected ultrasound, it is set to INPUT

Exercise 1

■ Variable declaration and running distance measurement functions

→ Declaring a variable to store the distance value, running the ultrasonic sensor distance measurement function

```
void loop() {
  long distance;      // 거리 값 저장할 변수 선언

  distance = ultrasonic_distance(trig, echo);

  // Serial 모니터로 출력
  Serial.print(distance);
  Serial.println(" mm");

  // 1초마다 출력
  delay(1000);
}
```

Exercise 1

■ "ultrasonic_distance()" explained

→ Function to measure the time taken to receive and calculate the distance after ultrasonic transmission

①

```
float ultrasonic_distance(int trigPin, int echoPin)

distance = ((float)(340 * duration) / 1000) / 2;

return distance;
```

②

① Input: Enter the number of the connected trig pin and echo pin

② Output: Output calculated distance (mm)

Exercise 1

■ Output of measurements

→ Output measured distance values to serial monitors

```
void loop() {
  long distance;      // 거리 값 저장할 변수 선언

  distance = ultrasonic_distance(trig, echo);

  // Serial 모니터로 출력
  Serial.print(distance);
  Serial.println(" mm");

  // 1초마다 출력
  delay(1000);
}
```

Exercise 1

■Function Description

```
// Serial 모니터로 출력
Serial.print(distance);
Serial.println(" mm");
```

- **Serial.print(data)**

  → Output data via serial monitor if you want to output a → string, use double quotes ("")

  → Serial.println() is a function that wraps the line after outputting the data.

- **Delay(time)**

  → A function that stops Arduino's operation for as long as the input is received

  → units are milliseconds, ms (1/100th of a second)

```
// 1초마다 출력
delay(1000);
```

Exercise 1

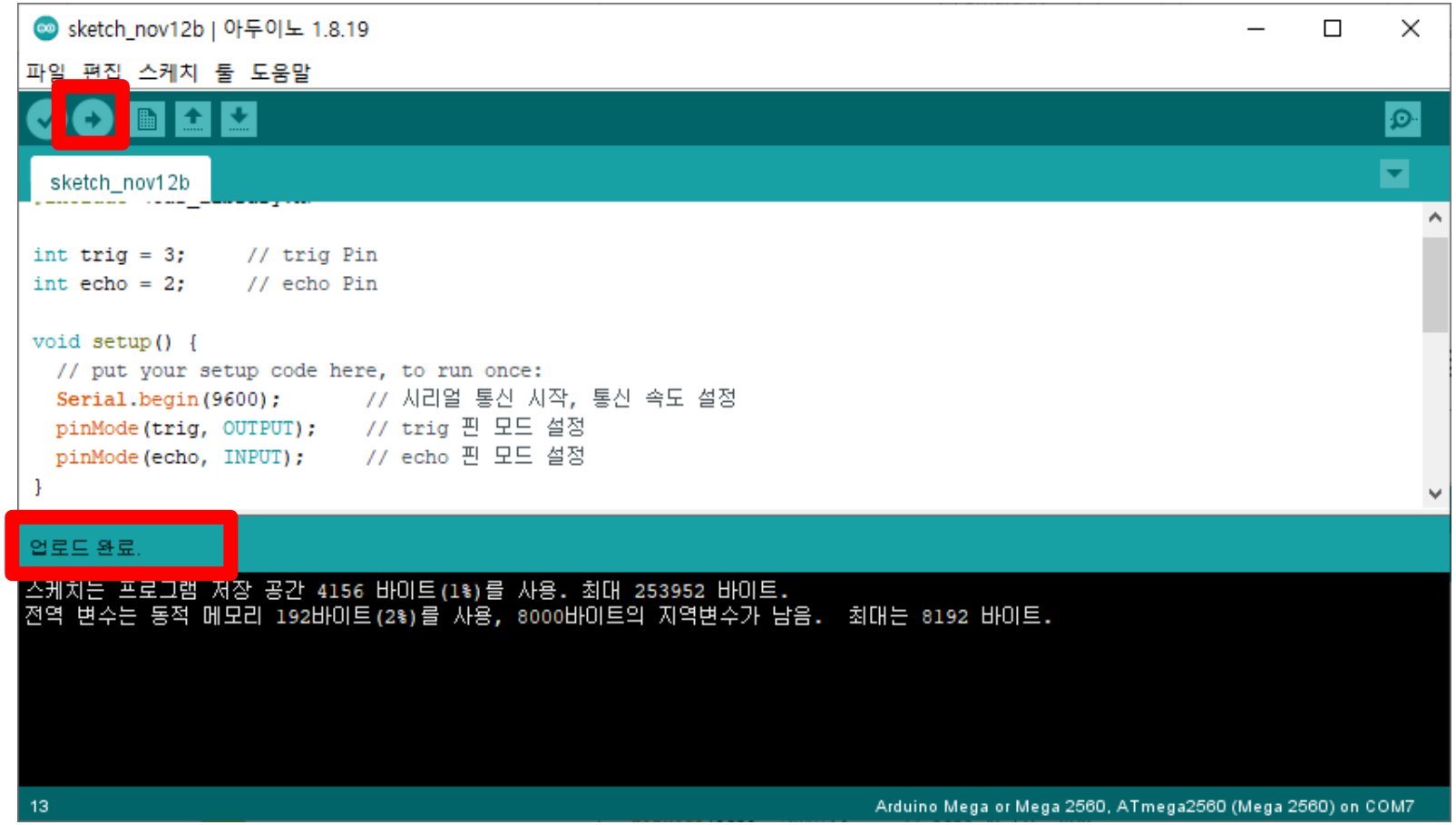
■ Compile and Upload

→ After you finish writing the source code, click the OK button to confirm the compilation and compilation completion.
→ If you get a window asking you to save, you can save or cancel it

Exercise 1

■ Compile and Upload

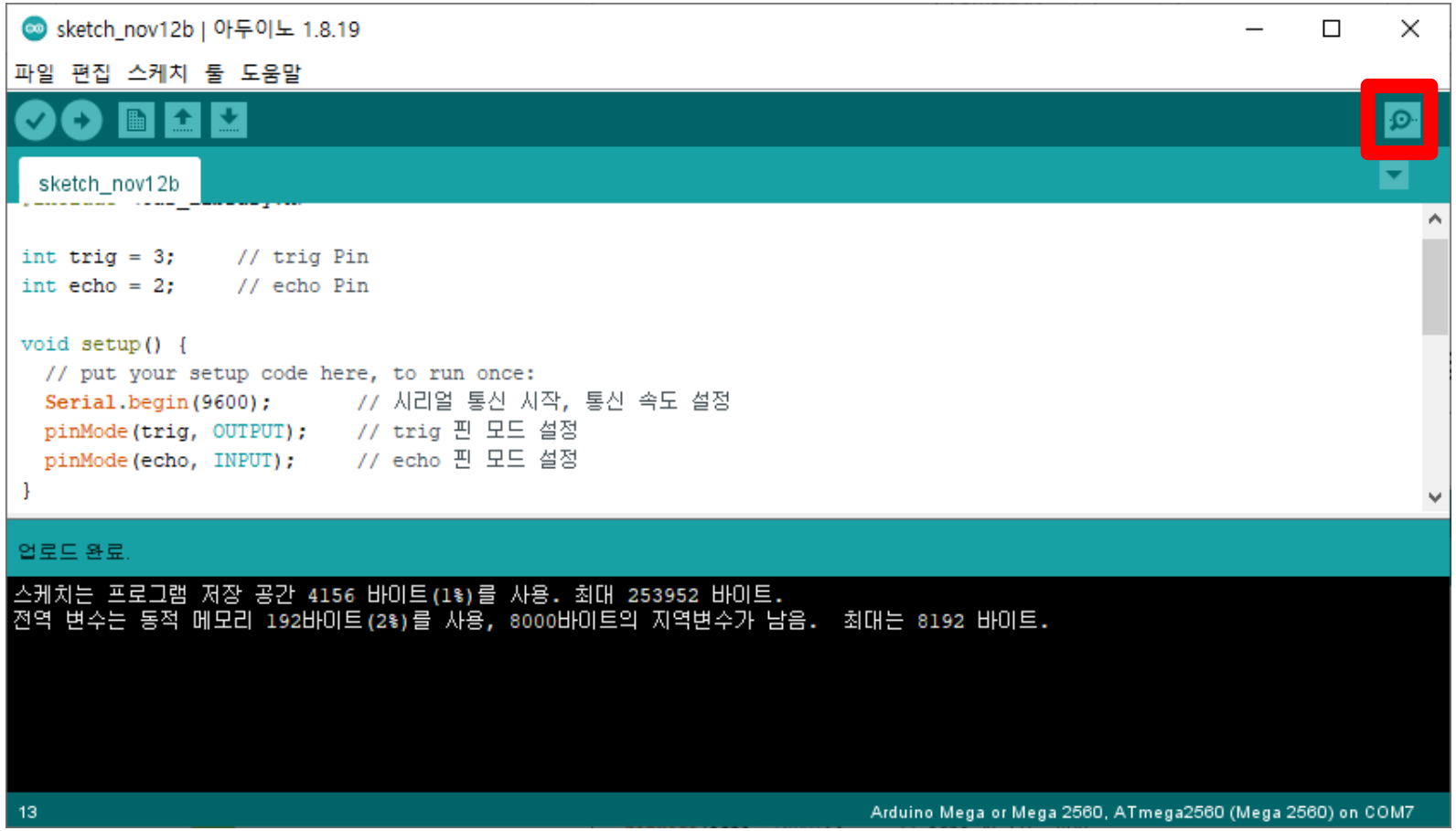
→ Click the upload button to upload to Arduino board and confirm upload completion



Exercise 1

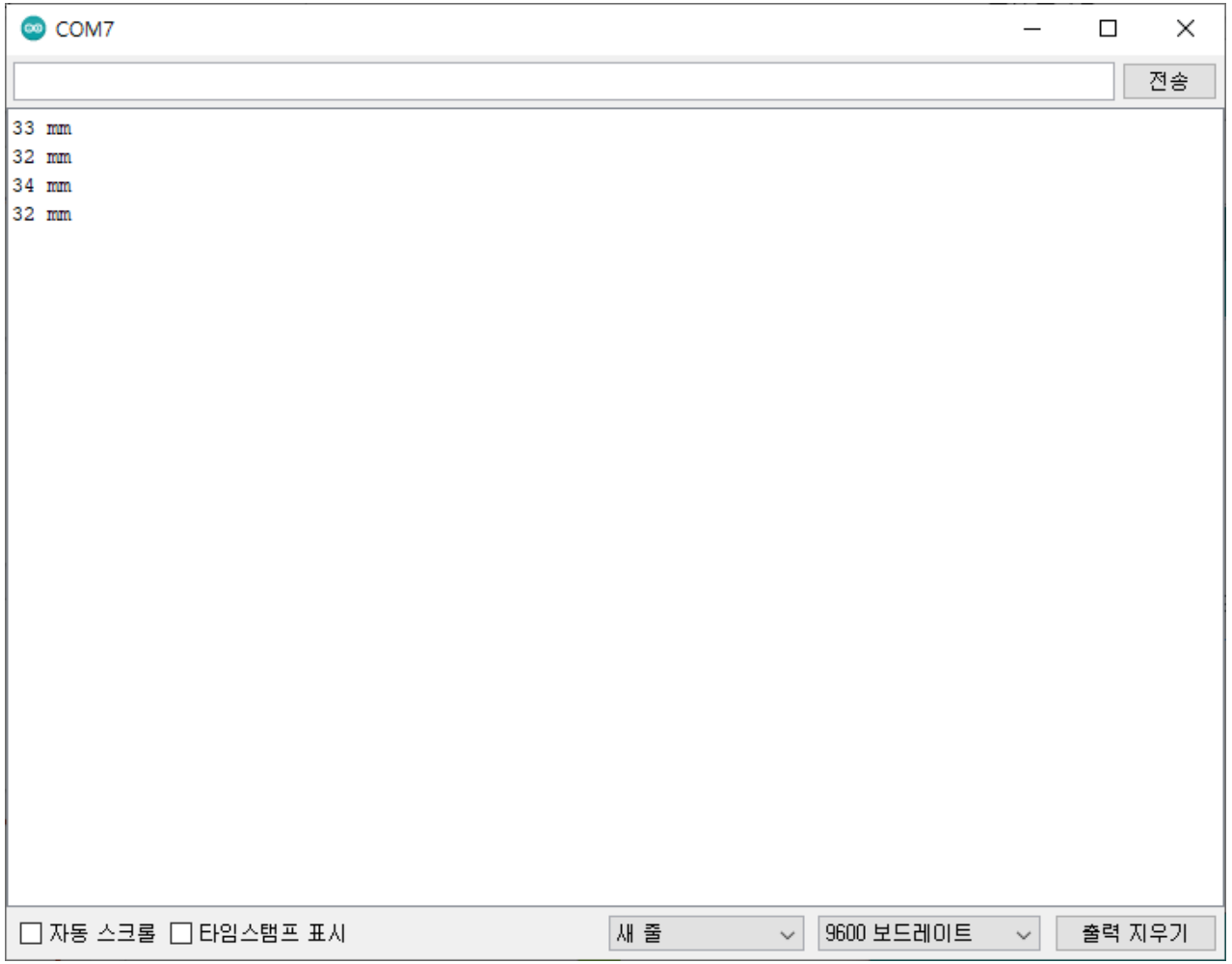
■ Check the results with the serial monitor

→ Click the Serial Monitor button or type "Ctrl + Shift + m"



Exercise 1

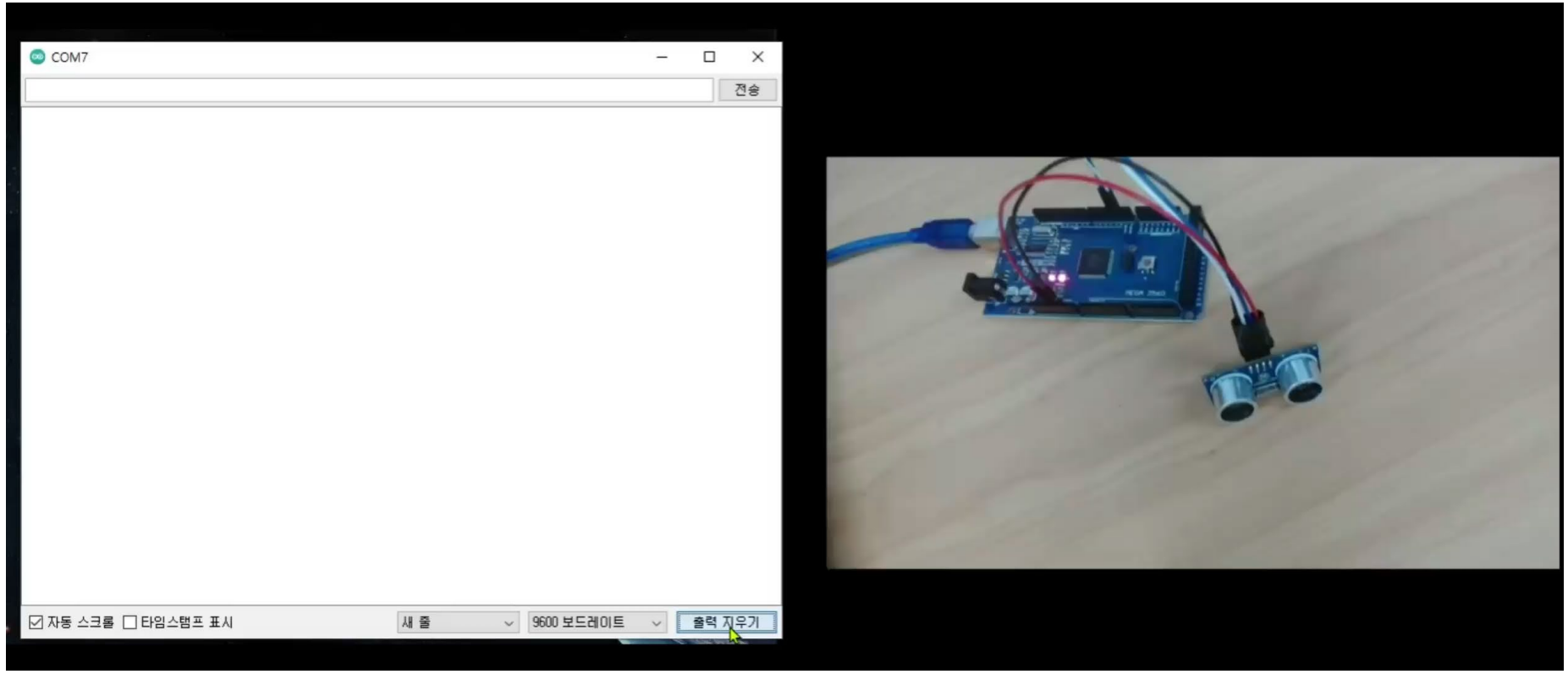
■ Serial Monitor Window



Exercise 1

■ Check the results with the serial monitor

→ Check the distance value output

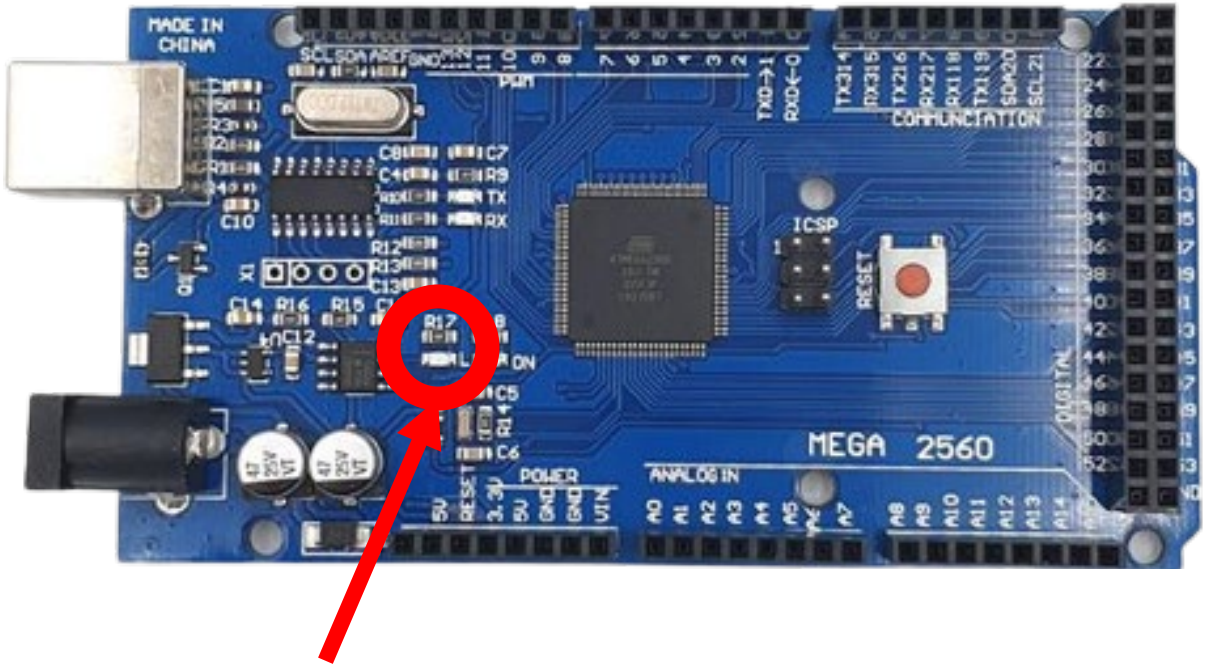


Exercise 2

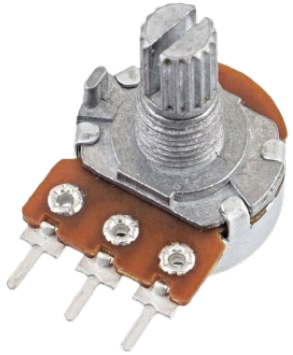
Exercise 2

■ LED brightness control using variable resistors

→ Adjustable LED brightness with analog input using variable resistor



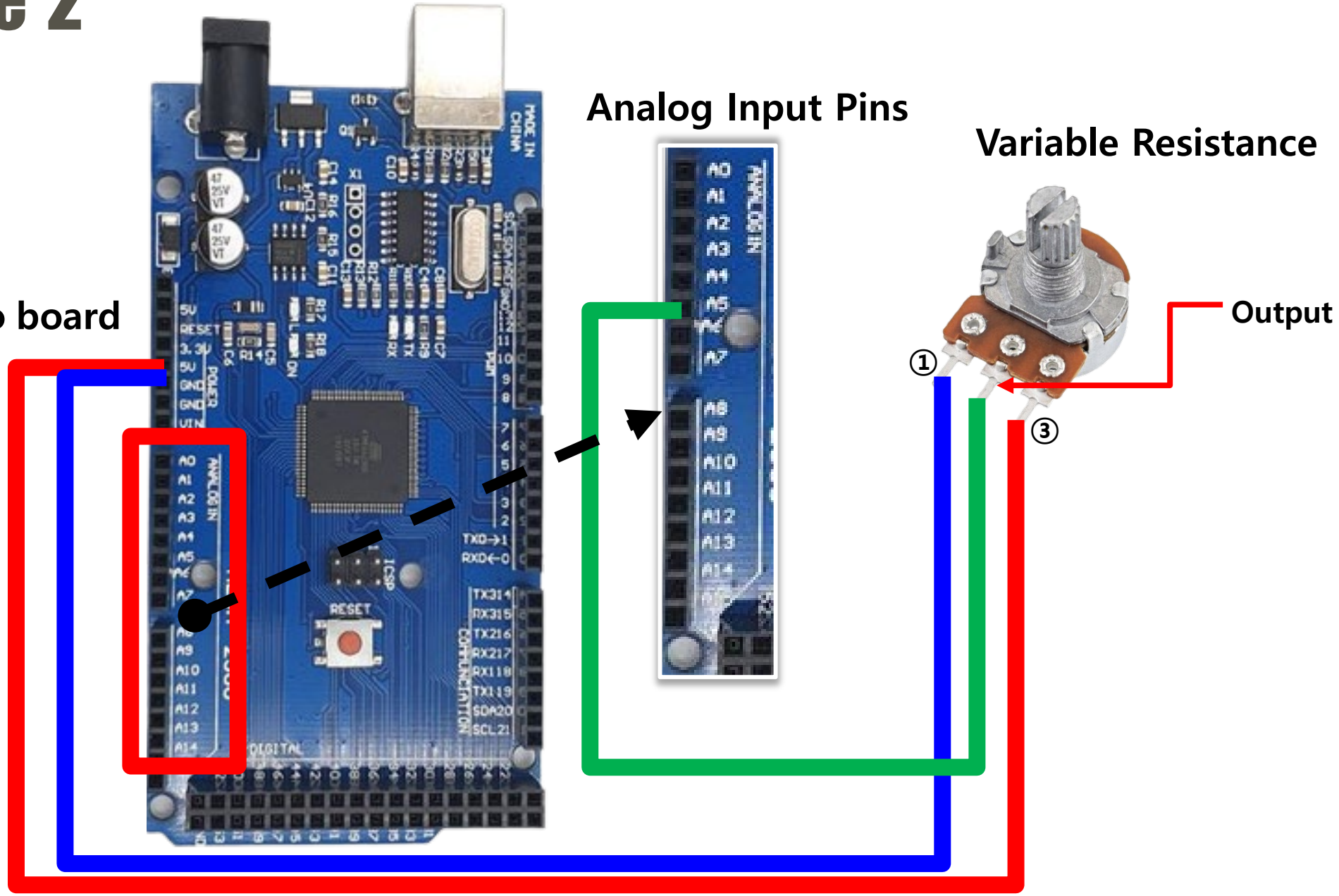
Built-in LED



Variable Resistance

Exercise 2

■ 선 연결



Exercise 2

■ Line connection(Cont'd)

→ Variable Resistor Pin 1 – Connects to Arduino GND
→ Variable Resistance Pin 3 – Connected to Arduino 5V
→ Variable Resistor Output – Connected to Arduino Analog A5 Pin

Exercise 2

■ Declaring a Pin Number Variable

→ Declaration of Arduino Pin Number Variable Linked to Variable Resistor Output Pin

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

```
#include <Car_Library.h>

int analogPin = A5;   // 가변저항 Output Pin

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
  pinMode(LED_BUILTIN, OUTPUT);     // LED 핀 모드 설정
}
```

Exercise 2

■ LED Pin Mode Settings

→ Setting the Variable Resistor Pin Mode Connected to the Arduino

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

```
#include <Car_Library.h>

int analogPin = A5;   // 가변저항 Output Pin

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
  pinMode(LED_BUILTIN, OUTPUT);     // LED 핀 모드 설정
}
```

Exercise 2

■ Running a function that reads variable declarations and resistance values

→ Declare a variable to store the resistance value, and execute a function that reads the resistance value of the variable resistor

```
void loop() {
  // put your main code here, to run repeatedly:
  int val;      // 저항값 저장할 변수 선언

  // 가변저항의 저항값을 읽어오는 함수 실행
  val = potentiometer_Read(analogPin);

  // Serial 모니터로 출력
  Serial.println(val);

  // 가변 저항 값을 LED로 보내 출력
  analogWrite(LED_BUILTIN, val);
}
```

Exercise 2

■ "potentiometer_Read()" description

→ A function that reads the resistance value of a variable resistor

```
int potentiometer_Read(int pin)
```

①

```
value = analogRead(pin) / 4;

return value;
```

②

① Input: Input the pin number associated with the output pin of the variable resistor

② Output: Output resistance value mapped to 255

Exercise 2

■ Resistive value serial output and LED output

→ Output the resistance value of the variable resistor to the serial monitor

→ Send the resistor value to the LED to adjust the brightness

```
void loop() {
  // put your main code here, to run repeatedly:
  int val;      // 저항값 저장할 변수 선언

  // 가변저항의 저항값을 읽어오는 함수 실행
  val = potentiometer_Read(analogPin);

  // Serial 모니터로 출력
  Serial.println(val);

  // 가변 저항 값을 LED로 보내 출력
  analogWrite(LED_BUILTIN, val);
}
```

Exercise 2

■ Function Description

- **analogWrite(Pin num, value)**

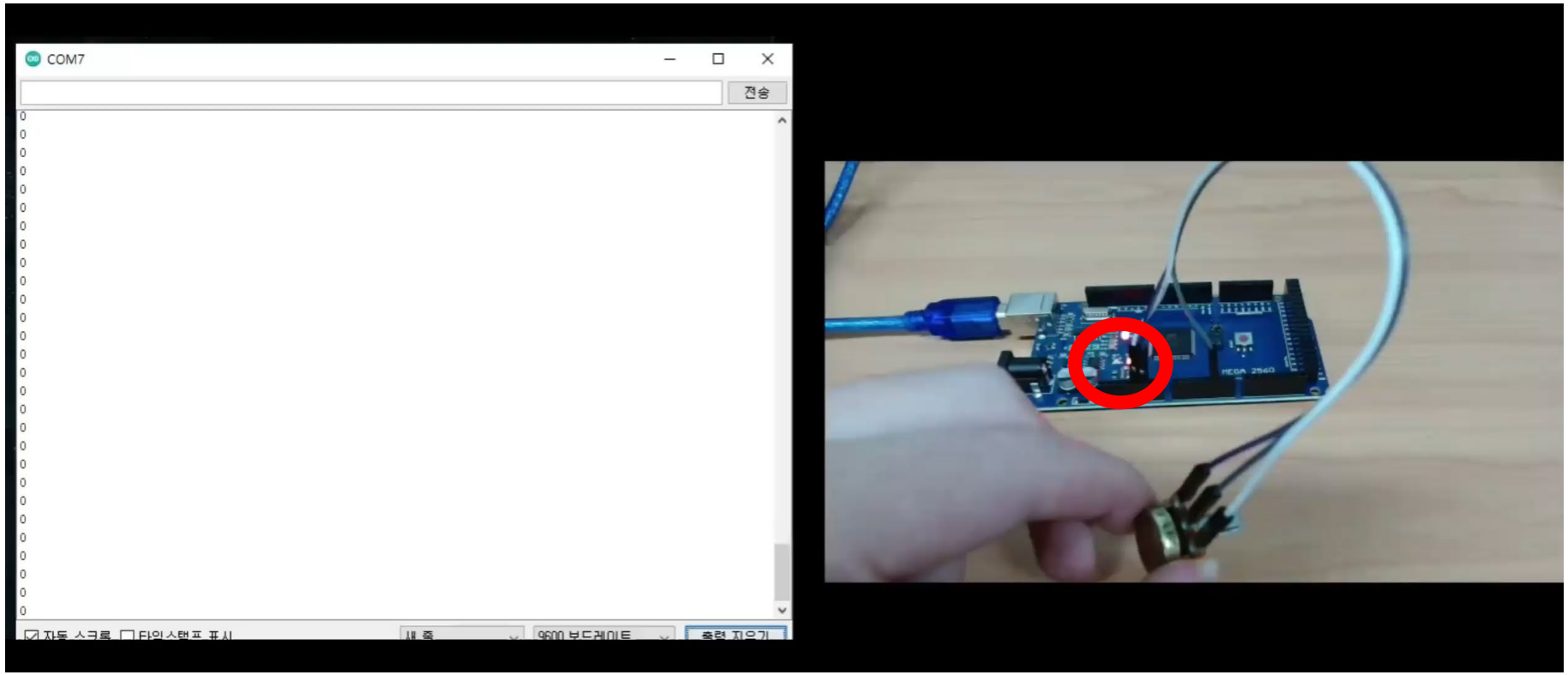
```
// 가변 저항 값을 LED로 보내 출력
analogWrite(LED_BUILTIN, val);
```

→ Set the output voltage of the pin to the input value (0~255)

Exercise 2

■ Check LED Brightness Control

→ Check LED brightness control with variable resistors



Exercise 3

Exercise 3

■ Motor Control

→ Connect the motor driver and gear motor to the Arduino to control the motor

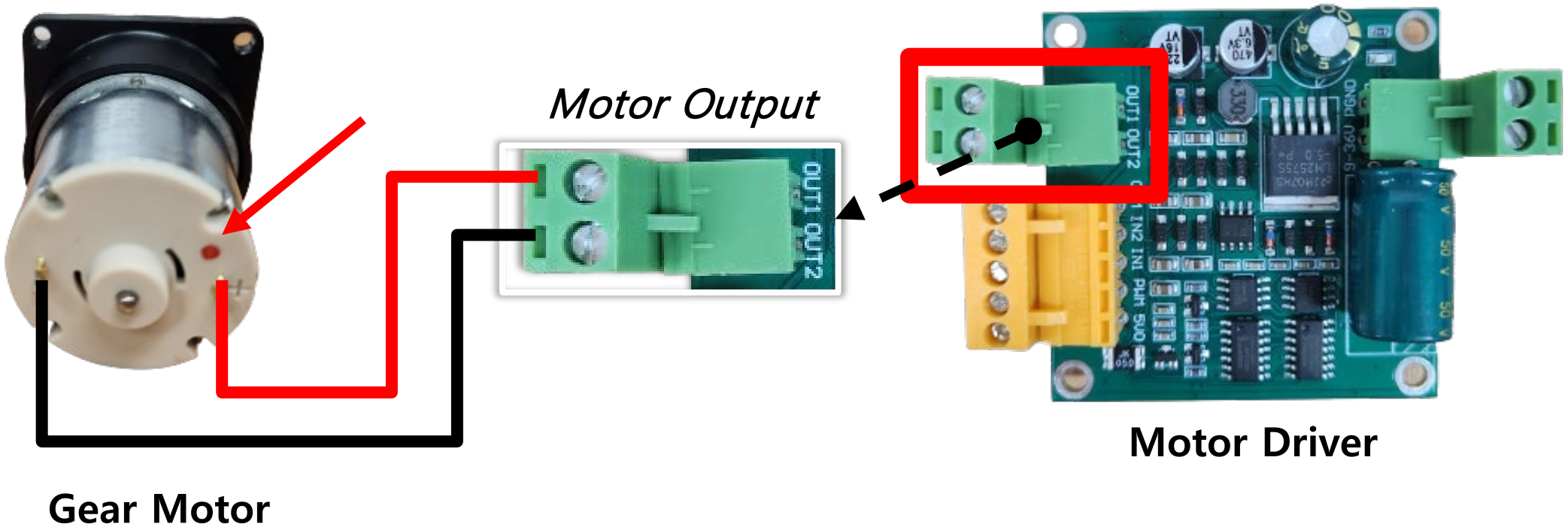
Gear Motor

Motor Driver

Exercise 3

■ Line connection– ①Motor Driver:Gear Motor

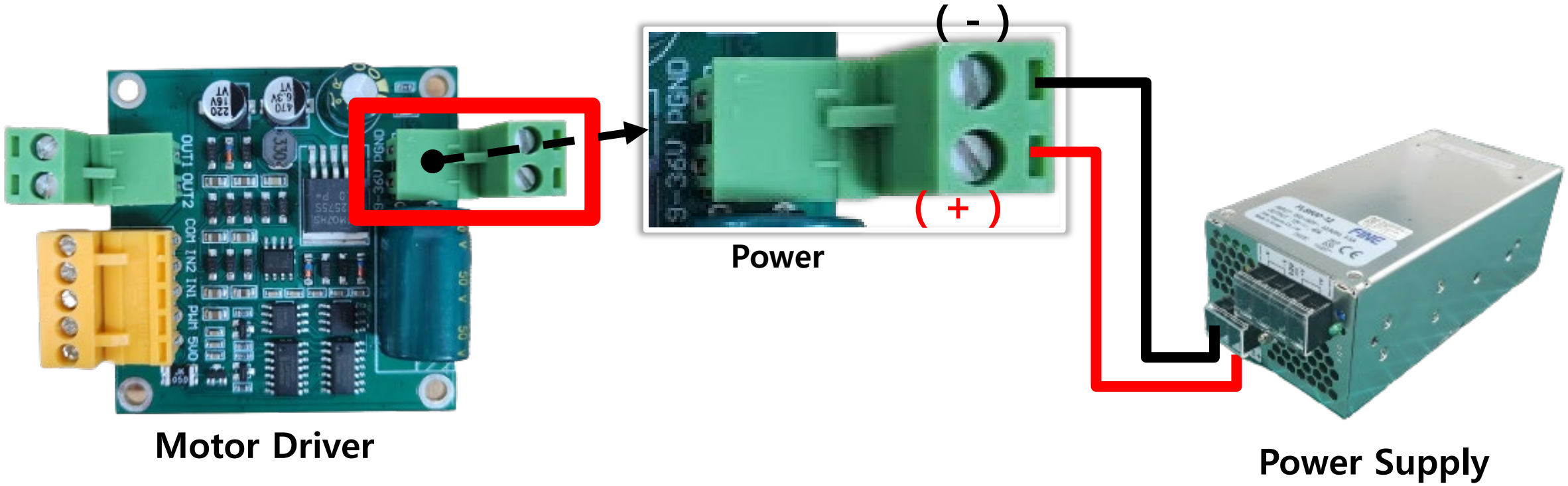
→ Gear motor red marked part – connected to motor driver OUT1
→ Gear Motor Unmarked Part – Connected to Motor Driver OUT2



Introduction

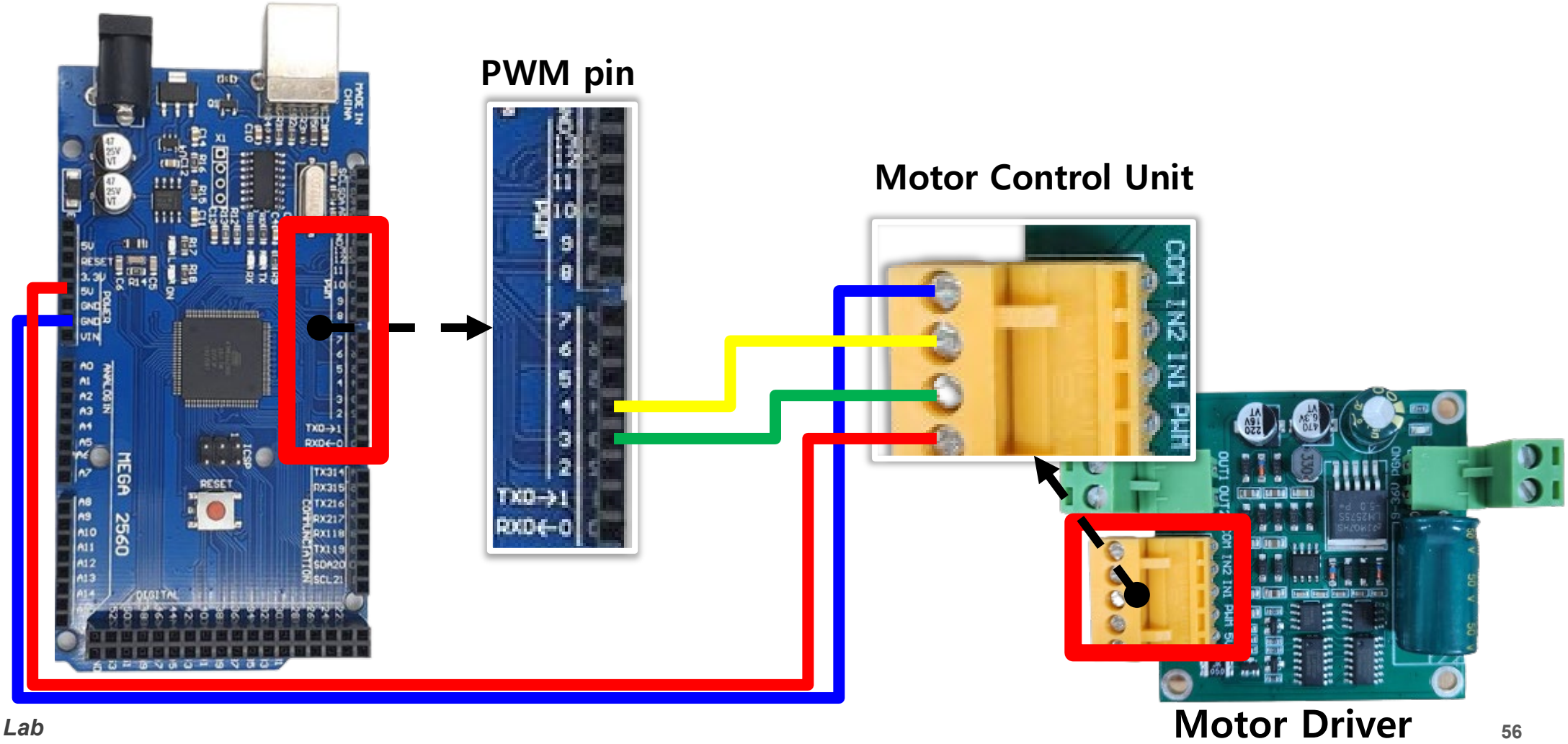
■ Line connection- ② Motor Driver:Power Supply

→ Requires 12V power because it uses a 12V motor
→ Power supply positive (+) – connected to the motor driver power portion 9-36 V portion → power supply cathode (-) – connected to the motor driver power portion PGND portion



Introduction

■ Line connection- ③ Arduino:Motor Driver



Introduction

■ Line connection- ③ 아두이노:모터 드라이버 (Cont'd)

→ COM of motor driver control unit– Arduino's GND Pin
→ PWM in the motor driver control unit– Arduino's 5V Pin
→ IN1 of motor driver control unit– Arduino's PWM Pin 3
→ IN2 of motor driver control unit– Arduino's PWM Pin 4

Exercise 3

■ Declaring Pin Variables

→ Declaration of the Arduino pin number variable associated with the driver's control pin

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

```
#include <Car_Library.h>

int motorA1 = 3;    // 모터 드라이버 IN1
int motorA2 = 4;    // 모터 드라이버 IN2


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);       // 시리얼 통신 시작, 통신 속도 설정
  pinMode(motorA1, OUTPUT);
  pinMode(motorA2, OUTPUT);
}
```

Exercise 3

■ Pin Mode Settings

→ Motor Control Pin Mode Settings

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

```
#include <Car_Library.h>

int motorA1 = 3;    // 모터 드라이버 IN1
int motorA2 = 4;    // 모터 드라이버 IN2

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);       // 시리얼 통신 시작, 통신 속도 설정
  pinMode(motorA1, OUTPUT);
  pinMode(motorA2, OUTPUT);
}
```

Exercise 3

■ loop() code

```
void loop() {
  // put your main code here, to run repeatedly:

  // Forward
  Serial.println("Motor Forward");
  motor_forward(motorA1, motorA2, 75);
  delay(3000);

  // Backward
  Serial.println("Motor Backward");
  motor_backward(motorA1, motorA2, 150);
  delay(3000);

  // Hold
  Serial.println("Motor Hold");
  motor_hold(motorA1, motorA2);
  delay(3000);
}
```

Exercise 3

■ Motor Forward Rotation

→ Run the "motor_forward()" function, enter the control pin number and motor speed

```
void loop() {
  // put your main code here, to run repeatedly:

  // Forward
  Serial.println("Motor Forward");
  motor_forward(motorA1, motorA2, 175);
  delay(3000);

  // Backward
  Serial.println("Motor Backward");
  motor_backward(motorA1, motorA2, 175);
  delay(3000);

  // Hold
  Serial.println("Motor Hold");
  motor_hold(motorA1, motorA2);
  delay(3000);
}
```

Exercise 3

■ "motor_forward()" description

→ Function to rotate the motor forward

```
void motor_forward(int IN1, int IN2, int speed)
```

①

```
{
    analogWrite(IN1, speed);
    analogWrite(IN2, LOW);
}
```

②

① Input: Motor driver IN1, input pin number associated with IN2, motor rotation speed (0~255)

② The motor rotates at the input rotational speed in the forward direction

Exercise 3

■ Motor Reverse Rotation

→ “motor_backward()” Execute the function, enter the control pin number and motor speed

```
void loop() {
  // put your main code here, to run repeatedly:

  // Forward
  Serial.println("Motor Forward");
  motor_forward(motorA1, motorA2, 175);
  delay(3000);

  // Backward
  Serial.println("Motor Backward");
  motor_backward(motorA1, motorA2, 175);
  delay(3000);

  // Hold
  Serial.println("Motor Hold");
  motor_hold(motorA1, motorA2);
  delay(3000);
}
```

Exercise 3

■ "motor_backward()" description

→ Function to rotate the motor in reverse direction

```
void motor_backward(int IN1, int IN2, int speed)
```

①

```
{
    analogWrite(IN1, LOW);
    analogWrite(IN2, speed);
}
```

②

① Input: Input the pin number and motor speed associated with the motor driver IN1 and IN2

② The motor rotates at the input rotational speed in the reverse direction

Exercise 3

■ Motor Stop

→ “motor_hold()” Run the function, enter the control pin number

```
void loop() {
  // put your main code here, to run repeatedly:

  // Forward
  Serial.println("Motor Forward");
  motor_forward(motorA1, motorA2, 175);
  delay(3000);

  // Backward
  Serial.println("Motor Backward");
  motor_backward(motorA1, motorA2, 175);
  delay(3000);

  // Hold
  Serial.println("Motor Hold");
  motor_hold(motorA1, motorA2);
  delay(3000);
}
```

Exercise 3

■ "motor_hold()" description

→ Function to stop the motor

```
void motor_hold(int IN1, int IN2)
```

①

```
{
    analogWrite(IN1, LOW);
    analogWrite(IN2, LOW);
}
```

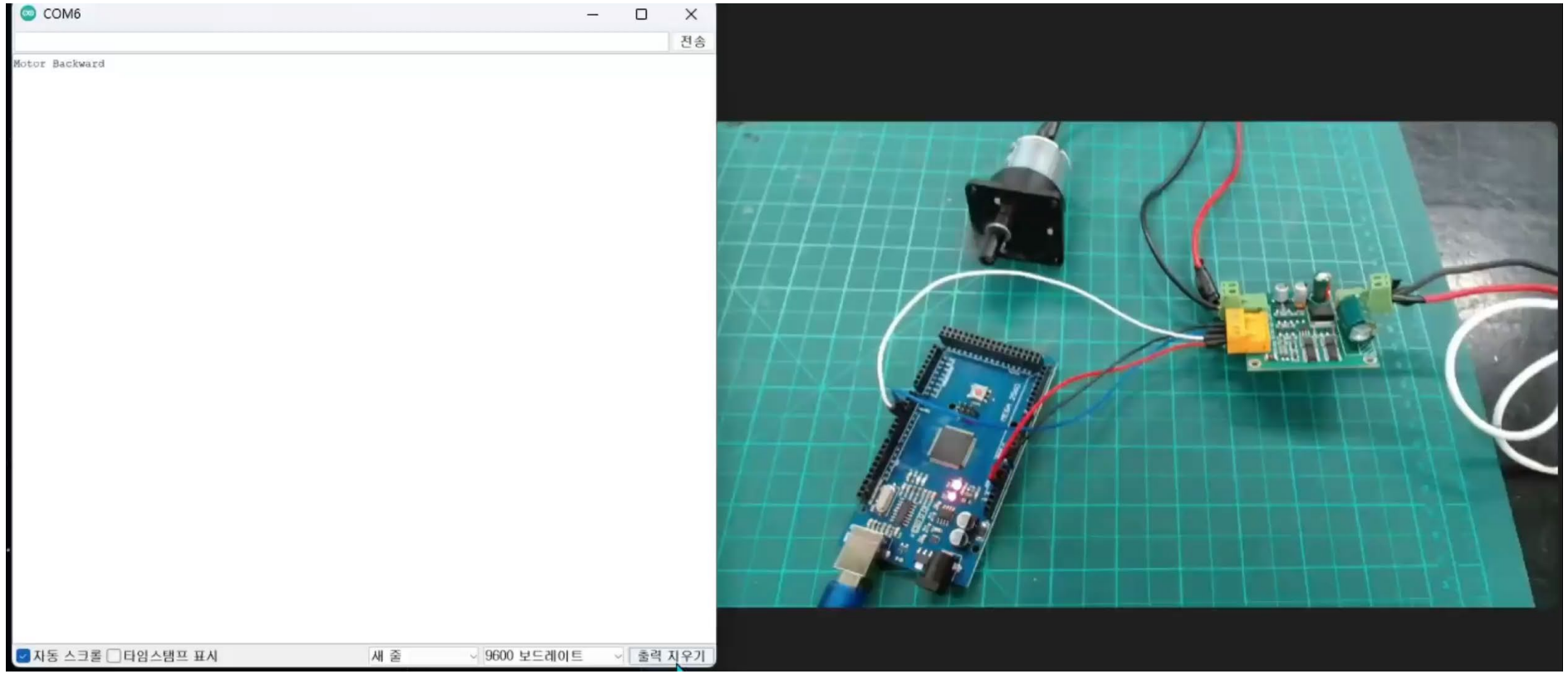
②

① Input: Enter the pin number associated with the motor driver IN1, IN2

② Motor rotation stop

Exercise 3

■ Check the results



Exercise 4

Exercise 4

■ Arduino control with a Python program using serial communication

→ Using serial communication, an integer value is sent to Arduino in a Python program to control LED brightness

```
while True › if input_value == 'q'
Run: main
C:\Users\autolab\anaconda3\envs\Test\python.exe C:\Users\autolab\PycharmProjects\pythonPro
아두이노로 전송할 저항 값(0~255): 255
아두이노로 전송할 저항 값(0~255): 100
아두이노로 전송할 저항 값(0~255): 75
아두이노로 전송할 저항 값(0~255): 10
아두이노로 전송할 저항 값(0~255): 1
아두이노로 전송할 저항 값(0~255): 255
아두이노로 전송할 저항 값(0~255): 50
아두이노로 전송할 저항 값(0~255): 255
아두이노로 전송할 저항 값(0~255): 10
아두이노로 전송할 저항 값(0~255): 0
아두이노로 전송할 저항 값(0~255):
Version Control  Run  Python Packages  TODO  Python Console  Problems  Terminal  Services
Localized PyCharm 2022.2.3 is available // Switch and restart // Do... (6 minutes ago)  9:27  CRLF  UTF-8  4 spaces  Python 3.9 (Test)
```

Exercise 4

- **Variable Declaration and Serial Communication Settings** → Declare a variable to store the value sent to the serial

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

```
#include <Car_Library.h>

int val;            // 수신된 값 저장할 변수

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
  pinMode(LED_BUILTIN, OUTPUT);      // LED 핀 모드 설정
}
```

Exercise 4

■ Variable Declaration and Serial Communication Settings

→ Start serial communication and set communication speed, set LED pin mode

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

```
#include <Car_Library.h>

int val;            // 수신된 값 저장할 변수

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);        // 시리얼 통신 시작, 통신 속도 설정
  pinMode(LED_BUILTIN, OUTPUT);      // LED 핀 모드 설정
}
```

Exercise 4

■ Receive serial communication data

→ Read serial data and convert data to integers
→ Send the transmitted value to the LED to output the LED

```
void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available()) {
    val = Serial.parseInt();

    if(val >= 0) {
      analogWrite(LED_BUILTIN, val);
    }
  }
}
```

Exercise 4

■ Function Description

```
if(Serial.available()) {
  val = Serial.parseInt();
```

- **Serial.available()**

  → Check if there is data received from the serial port

- **Serial.parseInt()**

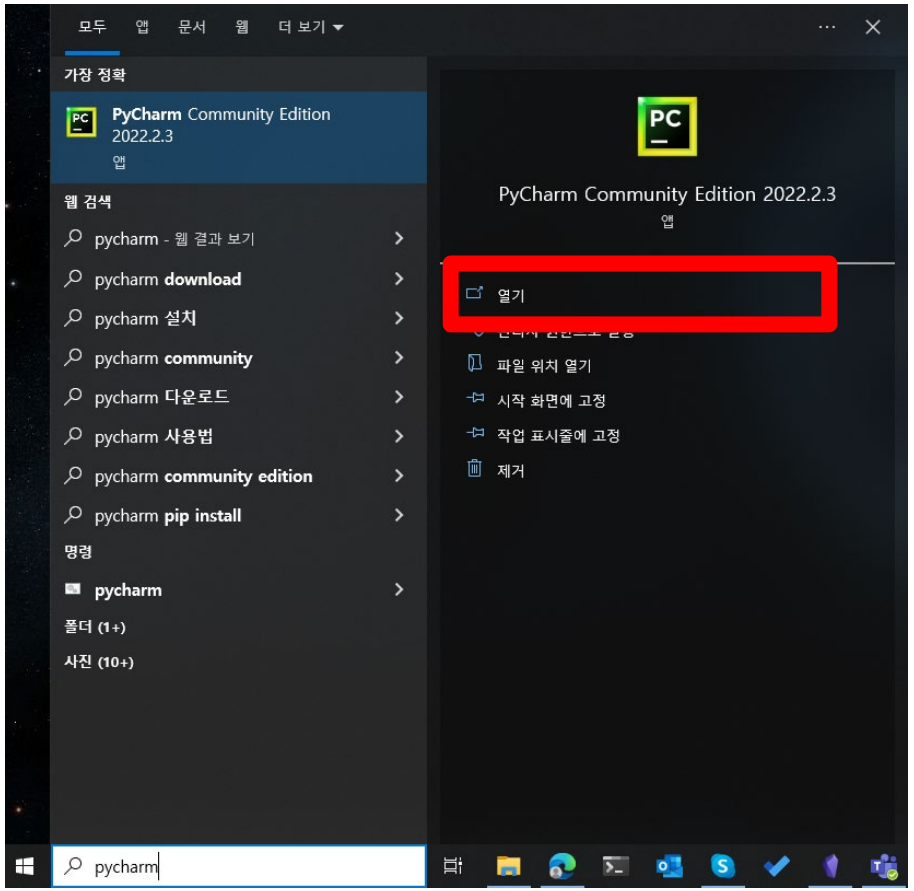
  → Converts data received from serial ports into integers

Exercise 4

■ Launch the Pycharm app

→ Type "Win+q" on your keyboard, then type "pycharm" and click "Open"

→ To install the app, refer to the "Environment Settings" course material!



Exercise 4

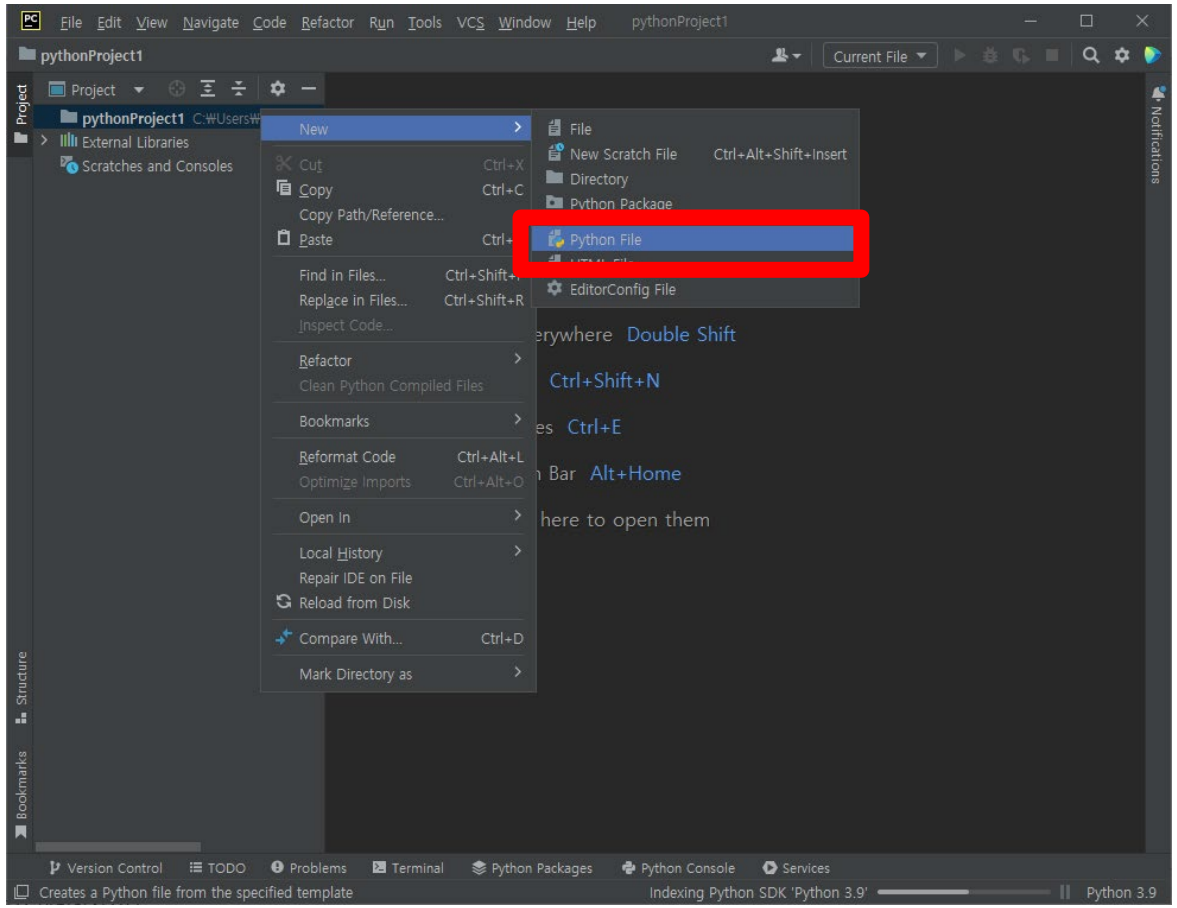
■ Create a Pycharm project and install the library

→ Requires installation of the Python "pyserial" library to proceed with that example
→ For information on how to create a project and install libraries, refer to the "Preferences" course!

Exercise 4

■ main.py File Creation

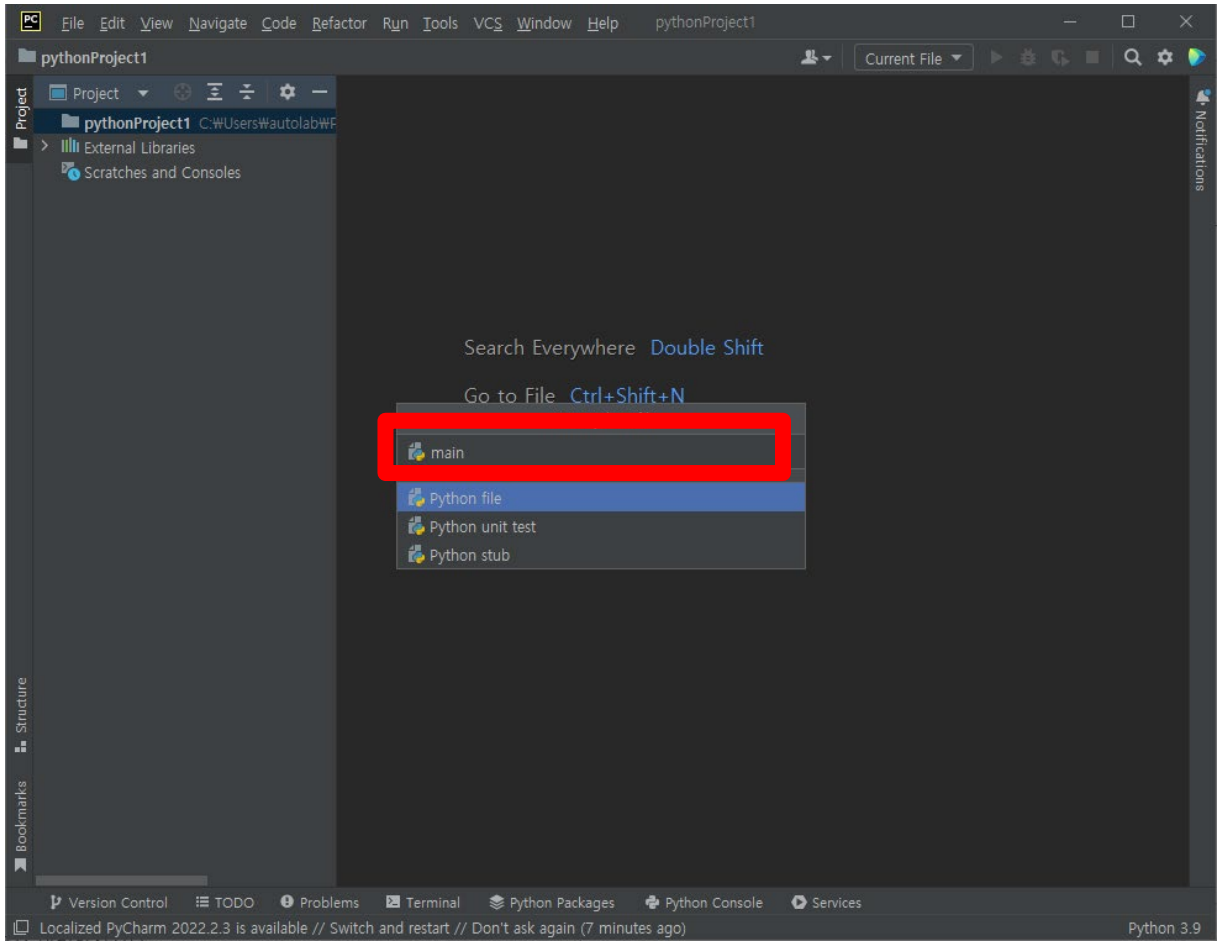
→ Create a main.py file for your project
→ Right-click on the project folder on the left -> select New -> click Python File



Exercise 4

■ main.py File Creation

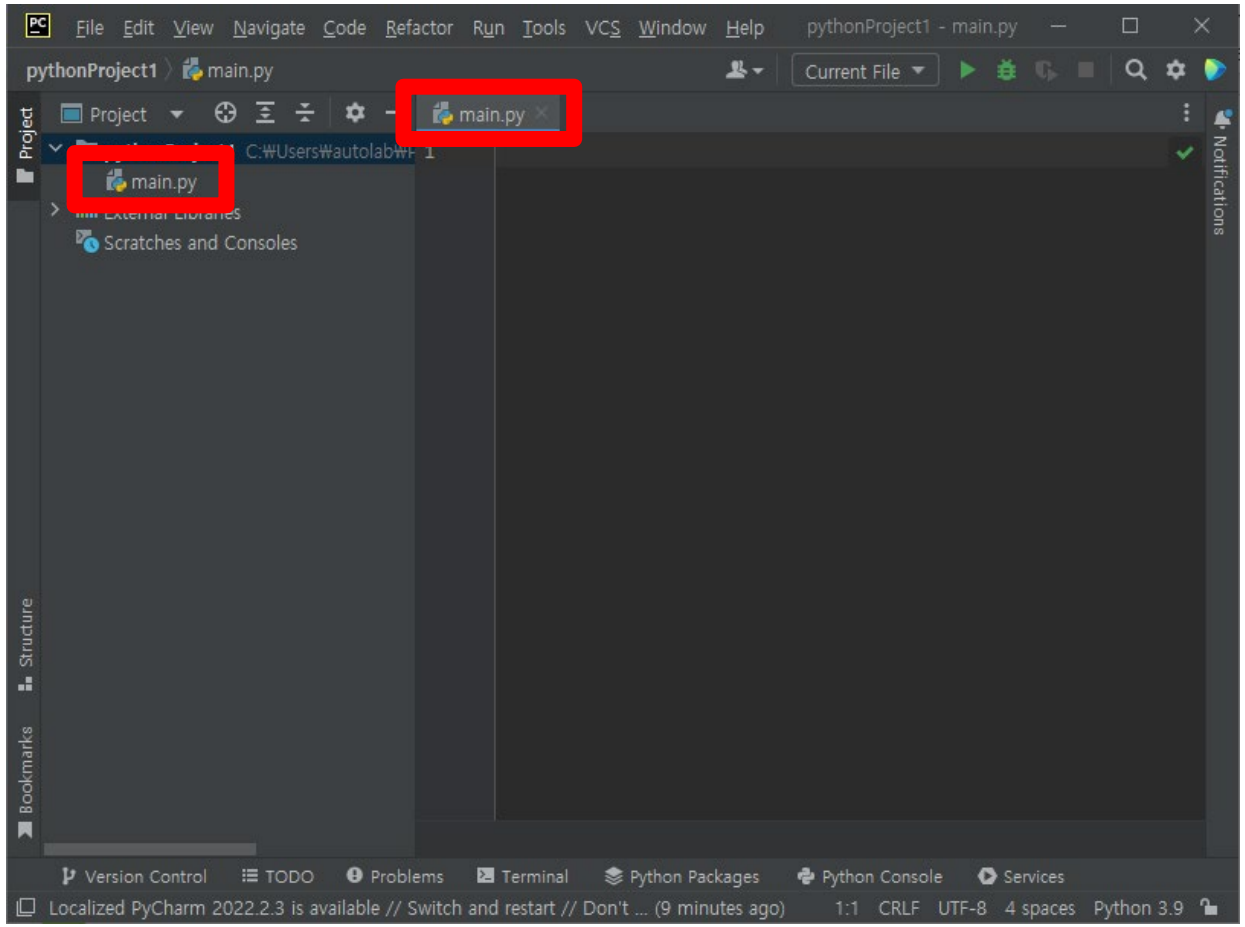
→ Type "main" on your keyboard and then press Enter



Exercise 4

■ main.py File Creation

→ Verify file creation “main.py “

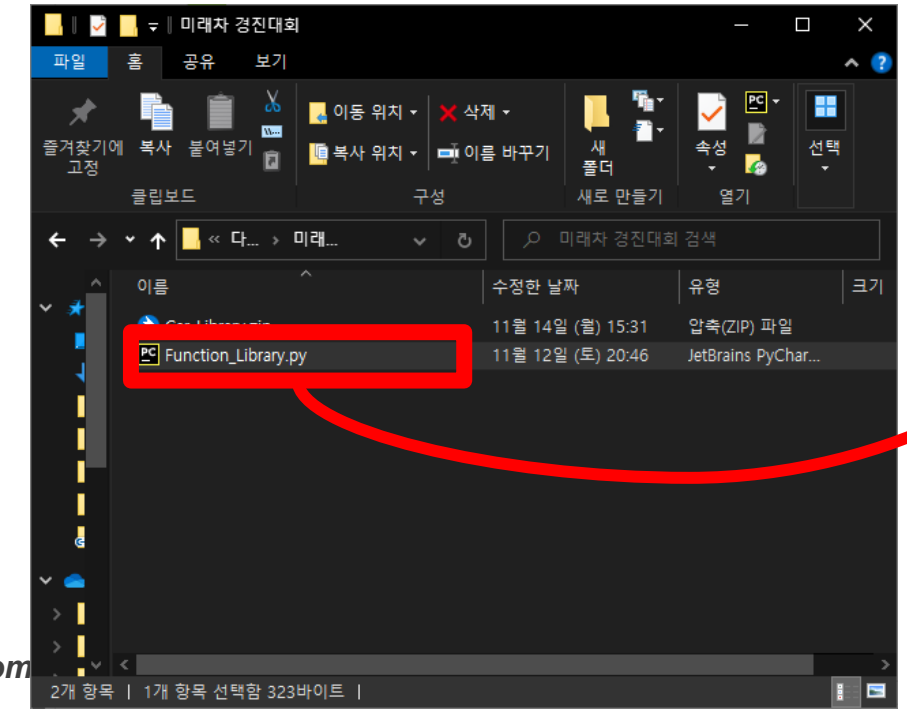


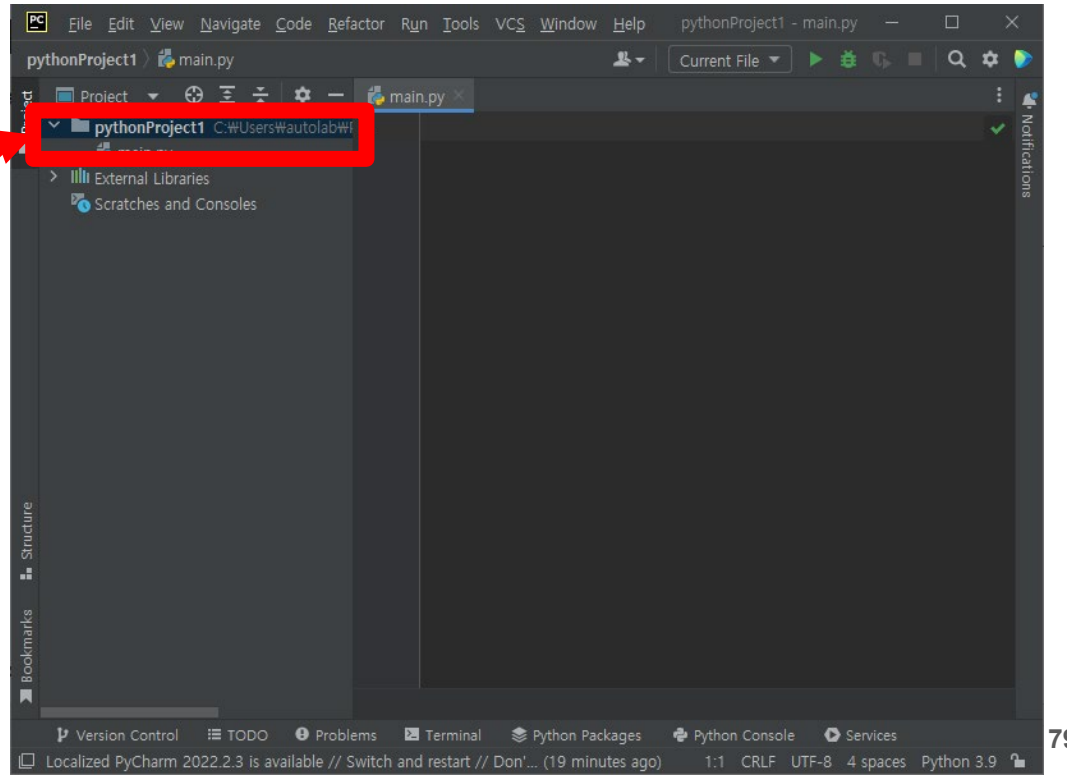
Exercise 4

■ Function_Library.py Add files

→ Add the "Function_Library.py" downloaded from GitHub to your Python Project

→ Select the file with the mouse and drag and drop it into the PythonProject

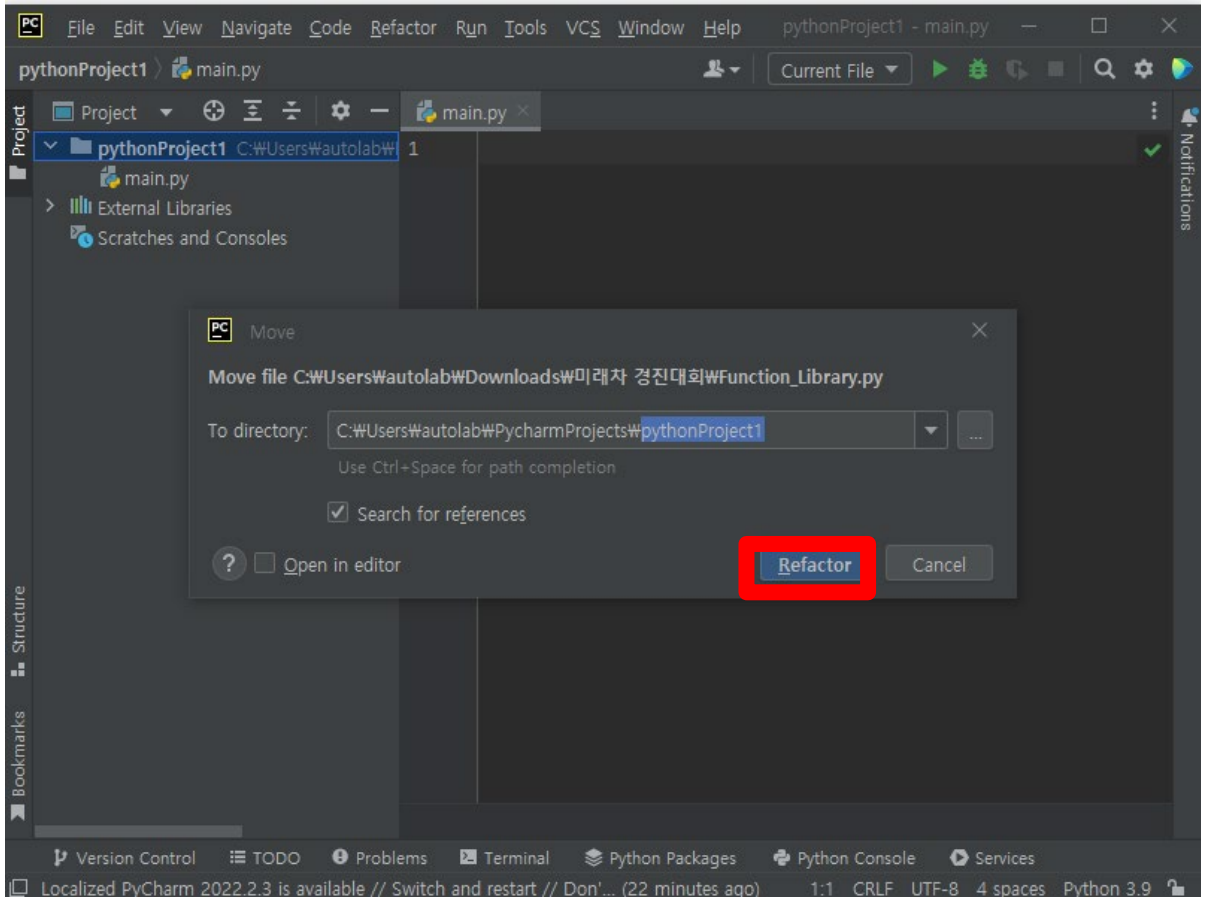




Exercise 4

■ Function_Library.py Add files

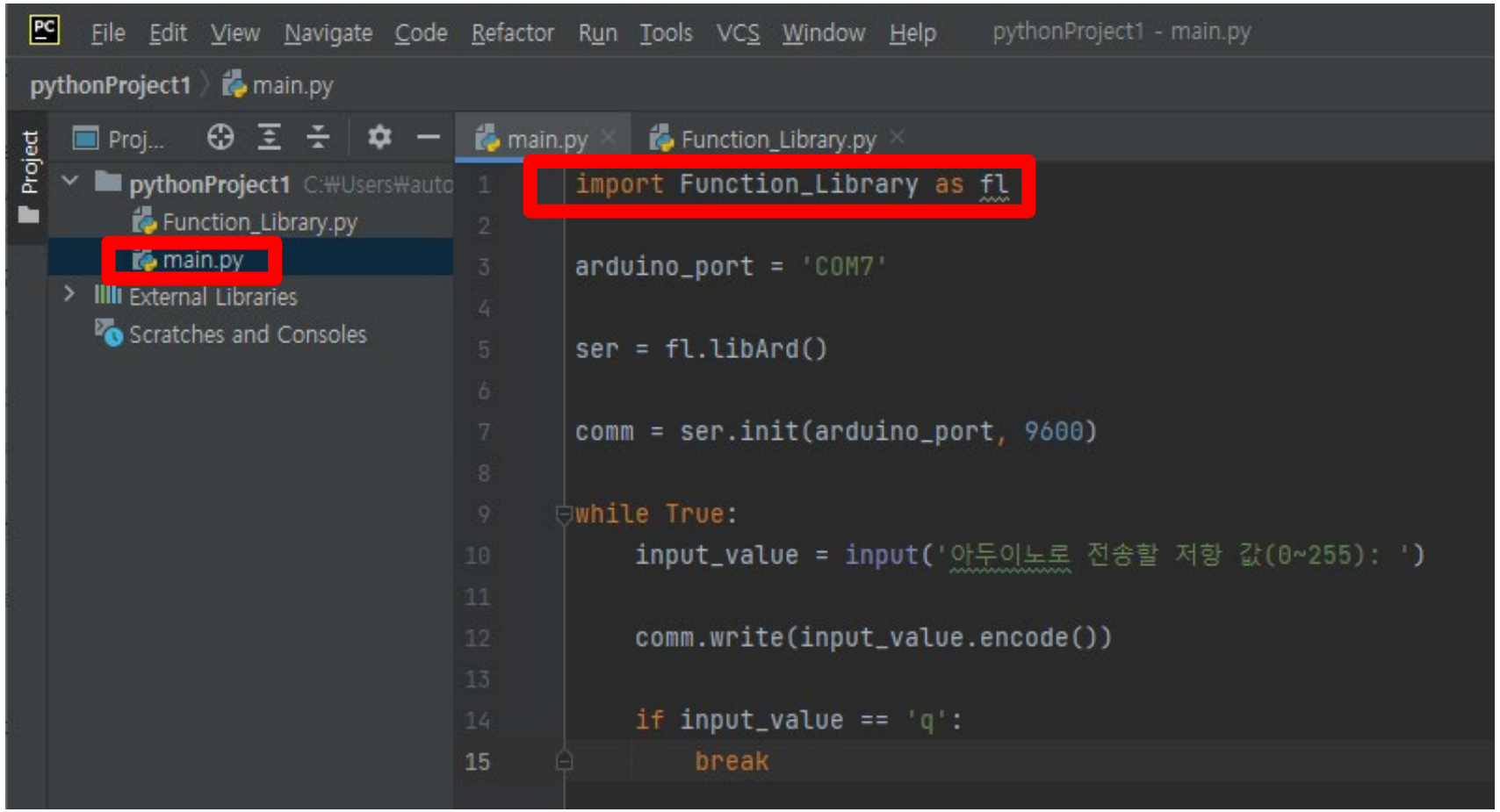
→ When the window appears, press Enter or click the Refactor button



Exercise 4

■ (Python) Library Import

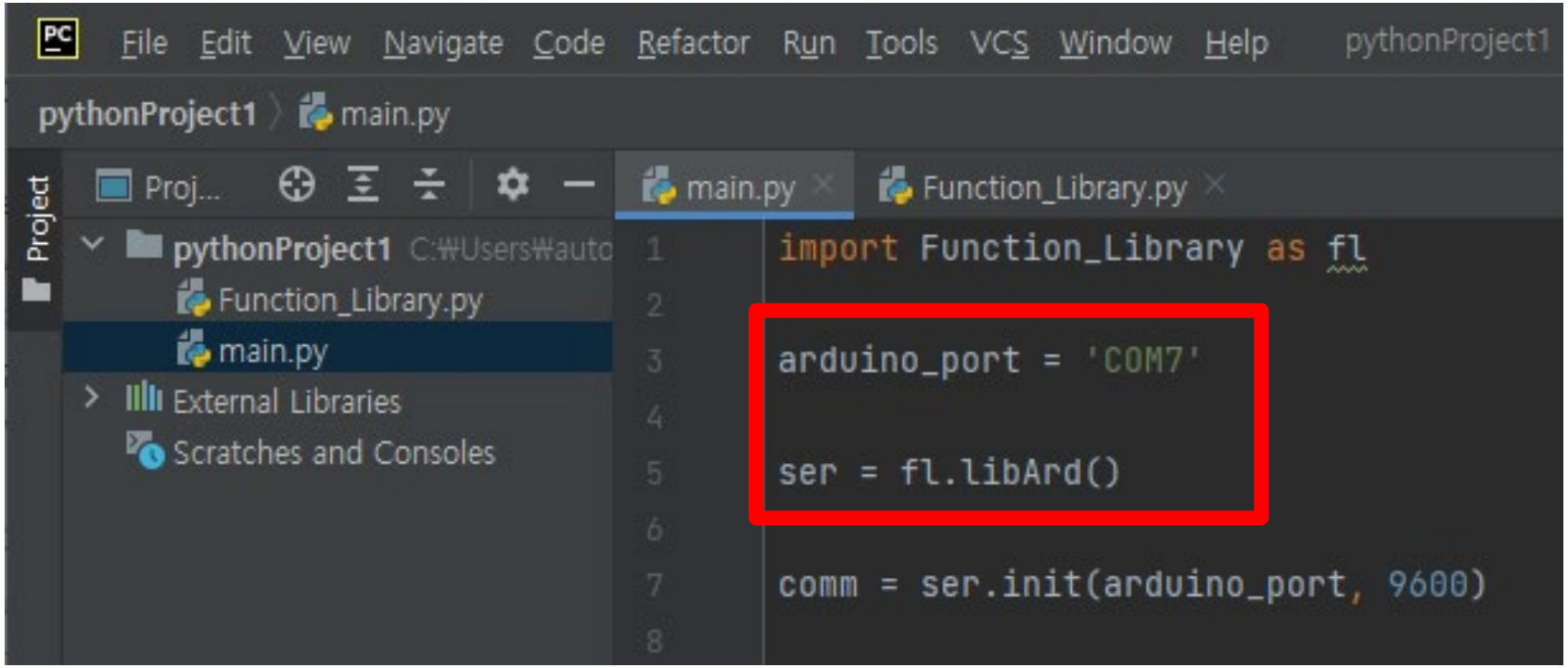
→ main.py Enter "import Function_Library as fl" in the source code



Exercise 4

■ Declaring variables and loading classes

→ Declare Serial Port Number Variables
→ Load the libArd() class of "Function_Library.py"
→ libArd() class is a collection of functions needed for practice.

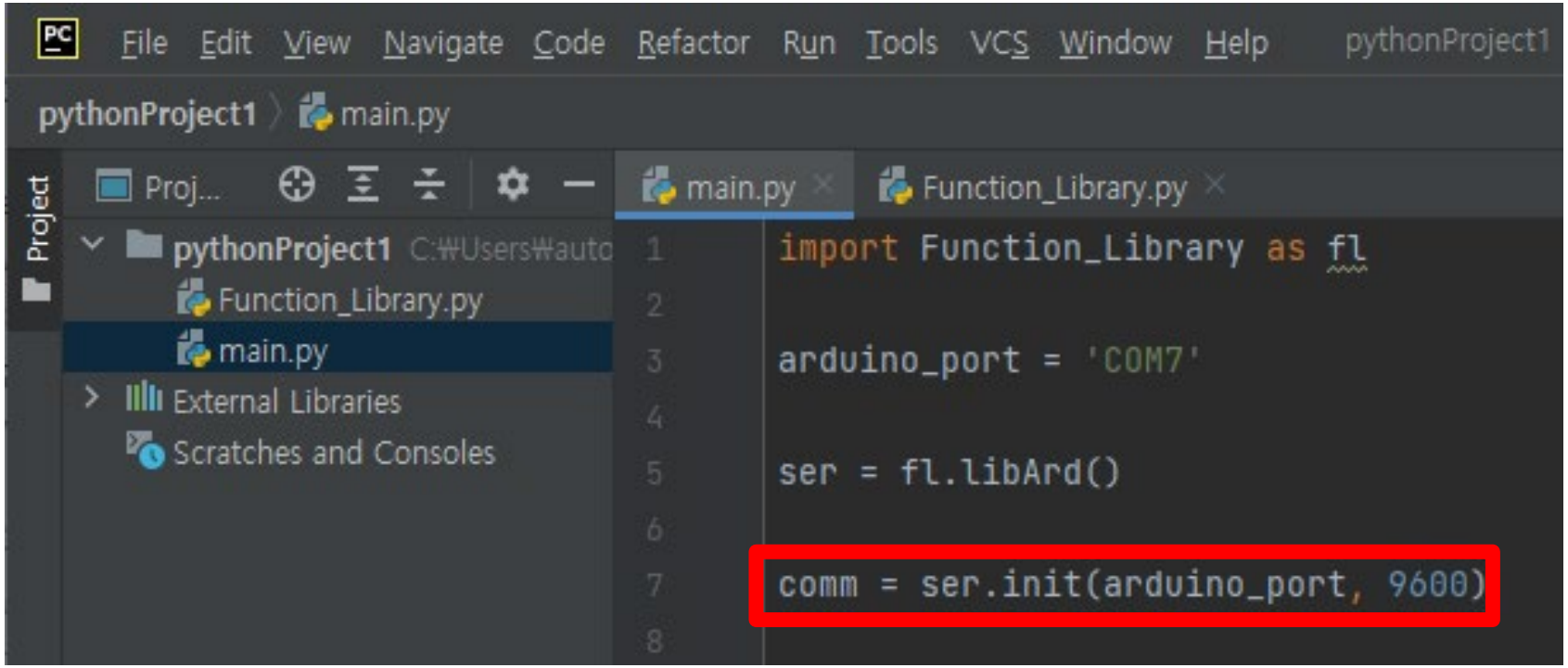


Exercise 4

■ Serial communication settings

→ ser.init() is a function that sets serial port number and sets communication speed in Python

→ Enter the port number variable and communication speed



Exercise 4

■ Serial communication transmission

→ input(): Output the specified sentence and receive the value input from the user

→ comm.write(): Transmitting input to serial communication

```
while True:
    input_value = input('아두이노로 전송할 저항 값(0~255): ')

    comm.write(input_value.encode())

    if input_value == 'q':
        break
```

Exercise 4

■ Arduino Compile and Upload

→ Compare the source code and upload it to the Arduino board

sketch_nov12b | 아두이노 1.8.19

파일 편집 스케치 툴 도움말

sketch_nov12b

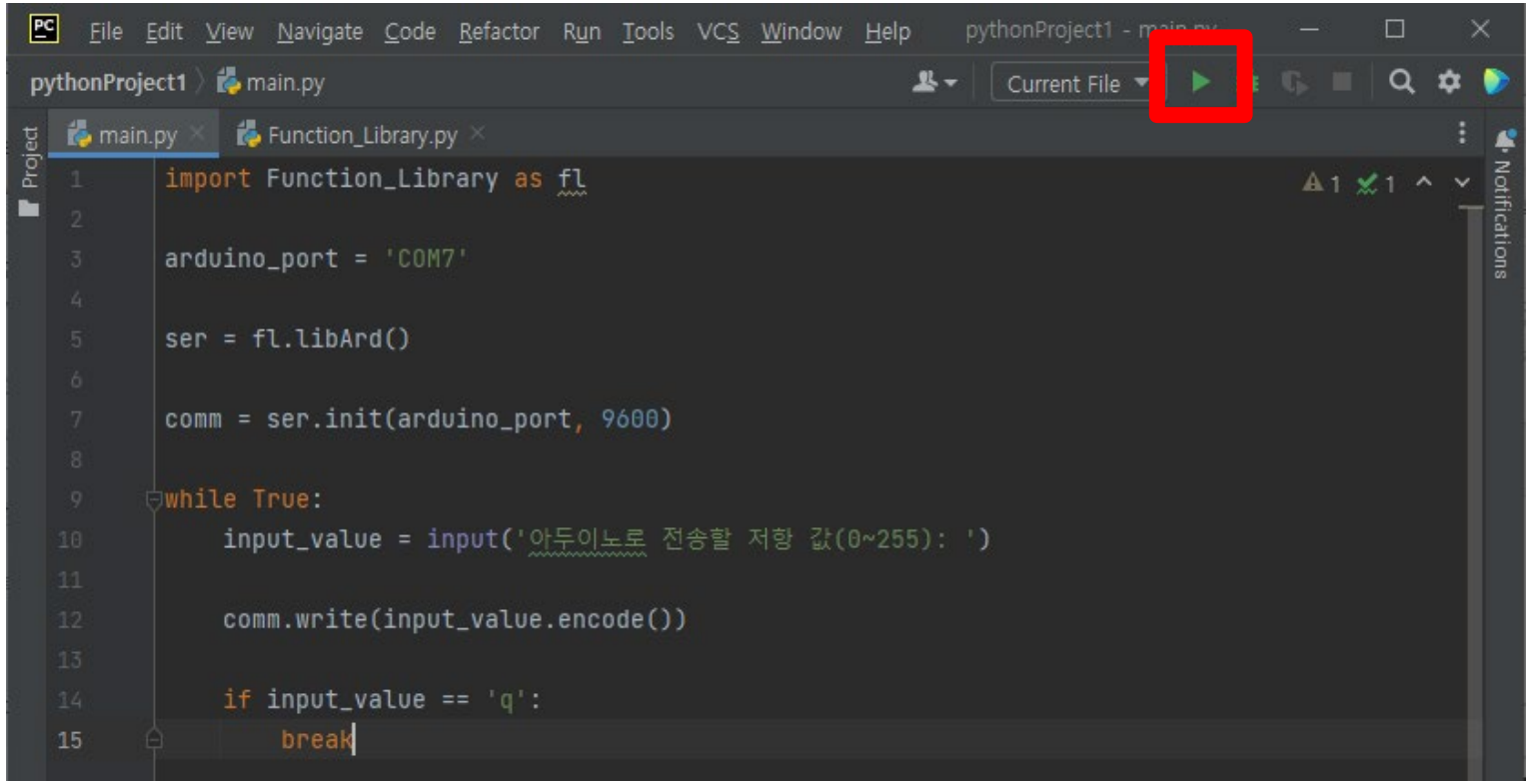
```
#include <Car_Library.h>

int val;          // 수신된 값 저장할 변수

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);       // 시리얼 통신 시작, 통신 속도 설정
  pinMode(LED_BUILTIN, OUTPUT);     // LED 핀 모드 설정
}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available()) {
    val = Serial.parseInt();

    if(val >= 0) {
      analogWrite(LED_BUILTIN, val);
    }
  }
}
```

Exercise 4

■ Running Python source code

→ Click the "Run" button or press Shift + F10 to run the source code

→ Make sure the Arduino serial monitor is closed (if it is open, an error will occur)

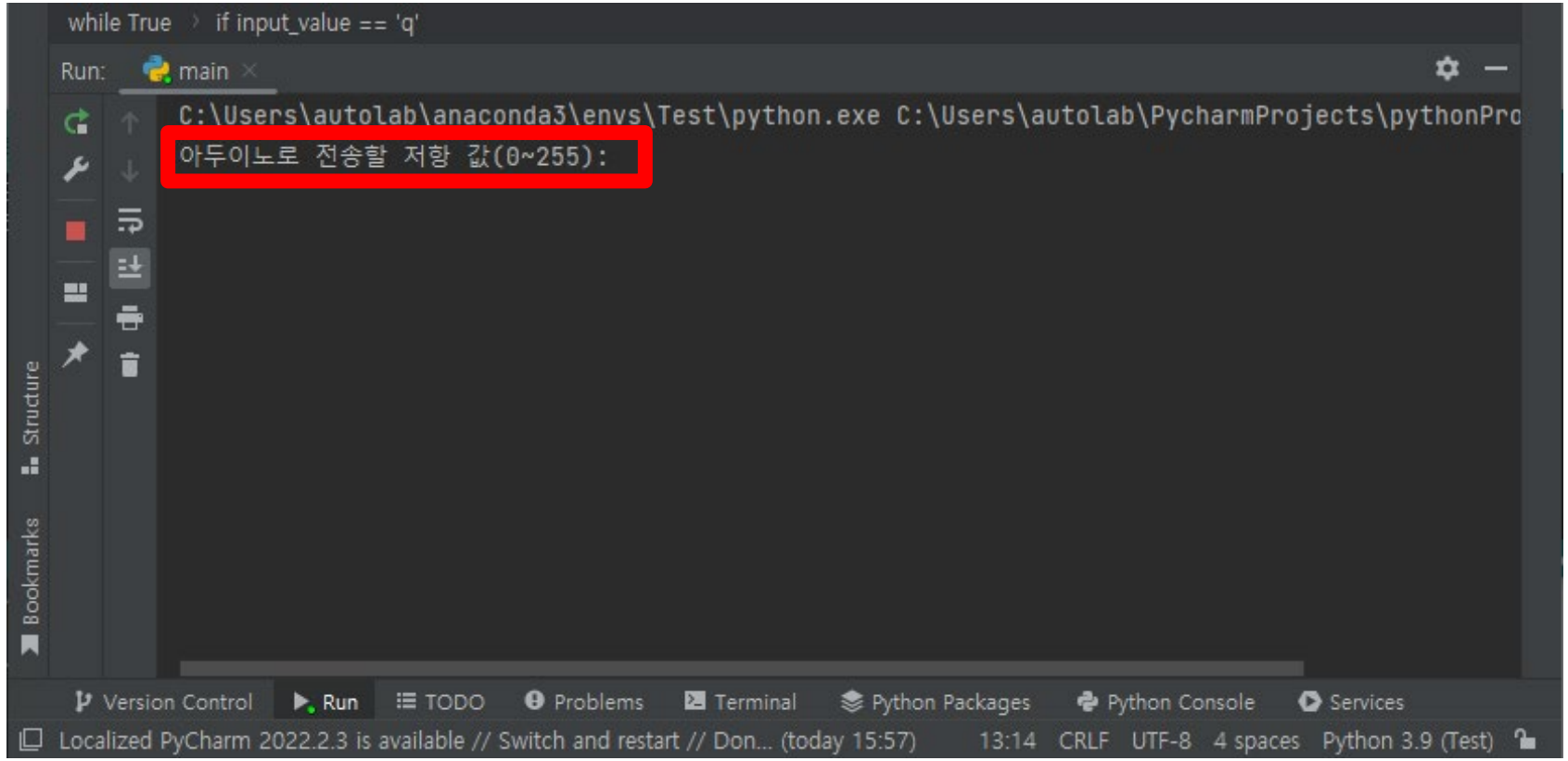


Exercise 4

■ Enter Resistance Values

When you press Run →, a window like the picture will appear at the bottom of the pycharm program.

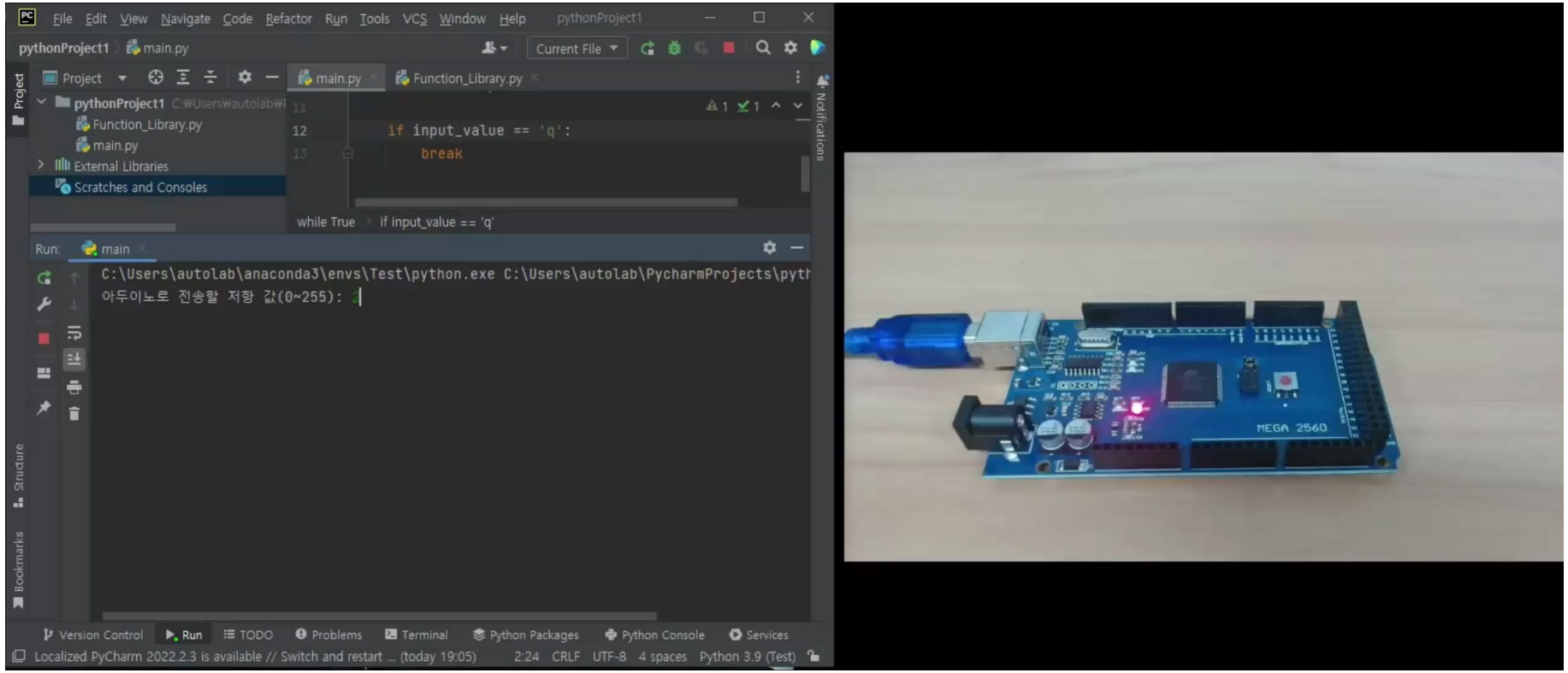
→ Enter the resistance value you want to send to Arduino and press Enter



Exercise 4

■ Check the results

→ Check LED brightness changes based on input values



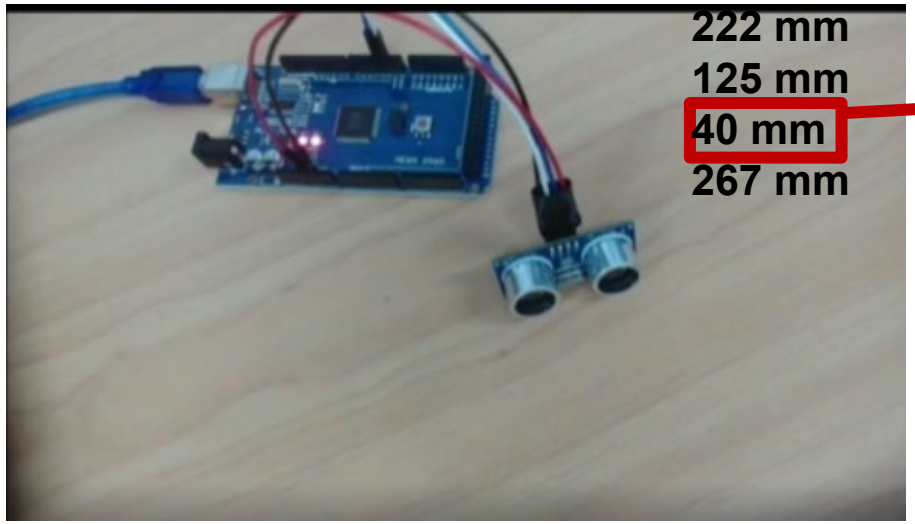
Thank You!

Automation Lab.

Temp – More Problem1

- Stopping the servo motor through an ultrasonic sensor (sudden stop)

- Equipment used: , ultrasonic sensor, servo motor



Motor speed stop when less than 50mm

Temp – More Problem2

■ Implementation of a safe start system for child

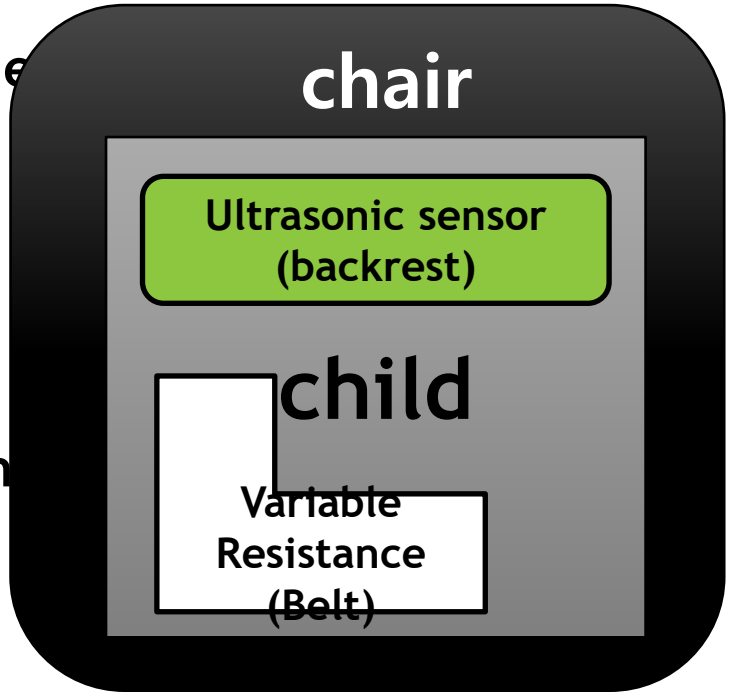
- Equipment used: Variable resistance, ultrasonic sensor
- Condition

1. It is assumed that the seat belt of a baby seat for infants is a tightening method, that is, variable resistance.
2. Detected by ultrasonic sensors to check if the child is close to the backrest.
3. Only when the seat belt is tightly fastened in the baby seat and the child is close to the backrest
   Motor controllable
4. The motor control signal uses serial communication.
5. The motor control signal is activated when "G" is sent as serial communication, and stopped when "S" is sent.

Temp – More Problem2

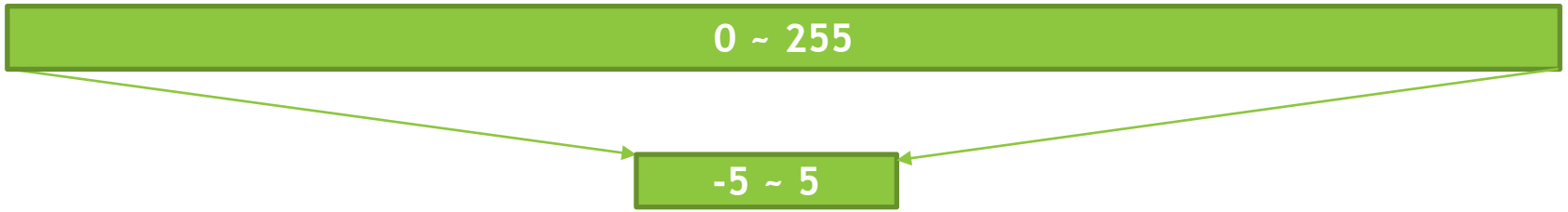
■ Case

1. If your child is safely seated in a baby seat
   -> The variable resistance value read is a very large value
   -> The ultrasonic sensor value read is large.
   - > LED lights up.
   -> Send a motor control signal to control the motor.
2. Other cases
   -> The LED lights out.
   -> Even if the motor control signal is sent, the motor can
   -> If the motor is running, stop immediately.



Temp – More Problem3

- Rotating the motor by the angle of variable resistance=Readjust the variable resistance to a value between 0~255 to a value between -5 0 5 (Negative - Reverse / Positive - Forward / 0 - Stop)



- Accelerate as the absolute value of the variable resistance increases
(0→-1 acceleration, 0→3 3x acceleration)