

Smart Software Project

Lab: Week 2
SmartCAR Platform
& Programming
Environment Setup

Prof. HyungJune Lee
hyungjune.lee@ewha.ac.kr

Today

- Lab announcement
- SmartCAR platform
- Arduino programming environment setup
- Lab Assignments #1-1, #1-2
- Course announcement



Class Schedule

| Week | Lecture Contents | Lab Contents |
|---------|---|--|
| Week 1 | Course introduction | Arduino introduction: platform & programming environment |
| Week 2 | Embedded system overview & source management in collaborative repository (using GitHub) | Lab 1: Arduino Mega 2560 board & SmartCAR platform |
| Week 3 | ATmega2560 Micro-controller (MCU): architecture & I/O ports, Analog vs. Digital, Pulse Width Modulation | Lab 2: SmartCAR LED control |
| Week 4 | Analog vs. Digital & Pulse Width Modulation | Lab 3: SmartCAR motor control (Due: HW on creating project repository using GitHub) |
| Week 5 | ATmega2560 MCU: memory, I/O ports, UART | Lab 4: SmartCAR control via Android Bluetooth |
| Week 6 | ATmega2560 UART control & Bluetooth communication between Arduino platform and Android device | Lab 5: SmartCAR control through your own customized Android app (Due: Project proposal) |
| Week 7 | Midterm exam | |
| Week 8 | ATmega2560 Timer, Interrupts & Ultrasonic sensors | Lab 6: SmartCAR ultrasonic sensing |
| Week 9 | Infrared sensors & Buzzer | Lab 7: SmartCAR infrared sensing |
| Week 10 | Acquiring location information from Android device & line tracing | Lab 8: Implementation of line tracer |
| Week 11 | Gyroscope, accelerometer, and compass sensors | Lab 9: Using gyroscope, accelerometer, and compass sensors |
| Week 12 | Project | Team meeting (for progress check) |
| Week 13 | Project | Team meeting (for progress check) |
| Week 14 | Course wrap-up & next steps | |
| Week 15 | Project presentation & demo I (Due: source code, presentation slides, & poster slide) | Project presentation & demo II |
| Week 16 | Final week (no final exam) | |



Lab Session

- Practice in-lab programming exercises based on the lecture materials
- Upload source codes for lab assignments in Ewha Cyber Campus after the lab session
 - Due: 11:59pm on the lab day
- Once you are done, you can leave the session after checking with me or TA
- Or, continue to work on programming for other homework assignments



Lab Policy

- 1) Please check out your SmartCAR (& Nexus 7 tablet) as soon as you arrive at the classroom
- 2) Please complete lab assignments
- 3) Upload required files to Ewha Cyber Campus
- 4) Check with me or TA
- 5) Please remove files that you created or downloaded in your computer after you are done
- 6) Please shut down your computer before you leave
- 7) Return the checked-out SmartCAR (& Nexus 7 tablet) to TA



Today

- Lab announcement
- **SmartCAR platform**
- Arduino programming environment setup
- Lab Assignments #1-1, #1-2
- Course announcement



SmartCAR Platform

- SmartCAR platform
 - Mainboard and sensor board installed inside the body
 - Sensors and extension connector can be accessed from outside



SmartCAR Components

- SmartCAR TOP side
 - ① Power Switch : Controlling power to ON/OFF
 - ② Power LED & Reset Switch
 - ③ Extension connector : Compass Sensor(CO, CO₂, NO, Dust, Ozone, VOC, Weather, etc.) can be installed
 - ④ Voltmeter : Returns the current battery voltage



SmartCAR Components

- SmartCAR front side

- ① Ultrasonic sensor (Transmitter)

- Transmit an ultrasonic wave to detect an object

- ② Ultrasonic sensor (Receiver)

- Receive the ultrasonic wave transmitted from TX

- ③ LED

- Use for showing software state or debugging

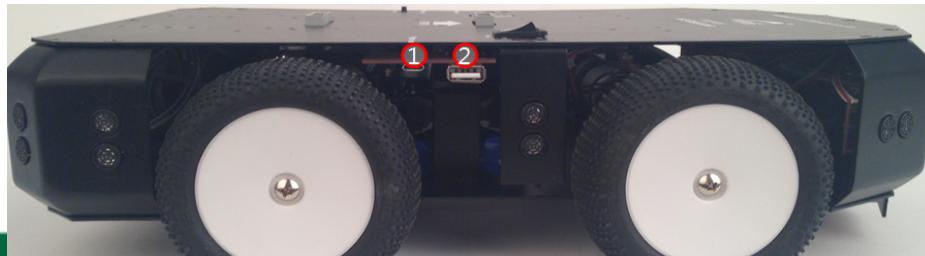


SmartCAR Components

- SmartCAR rear side
 - ① LED & Ultrasonic sensor



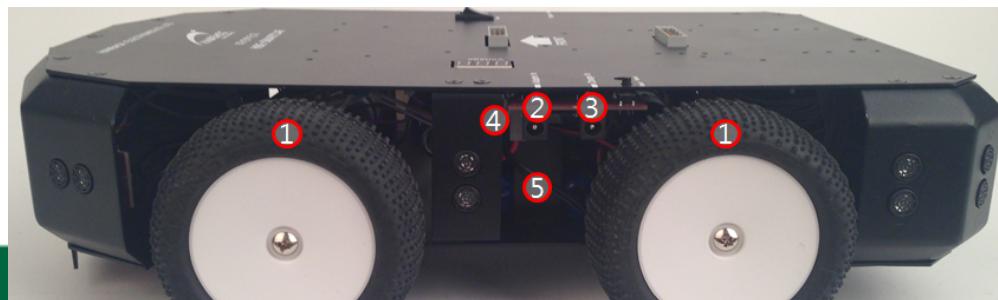
- SmartCAR right side
 - ① Program Port
For programming ATmega2560
 - ② Host USB Port
For connecting Android tablet or smartphone



SmartCAR Components

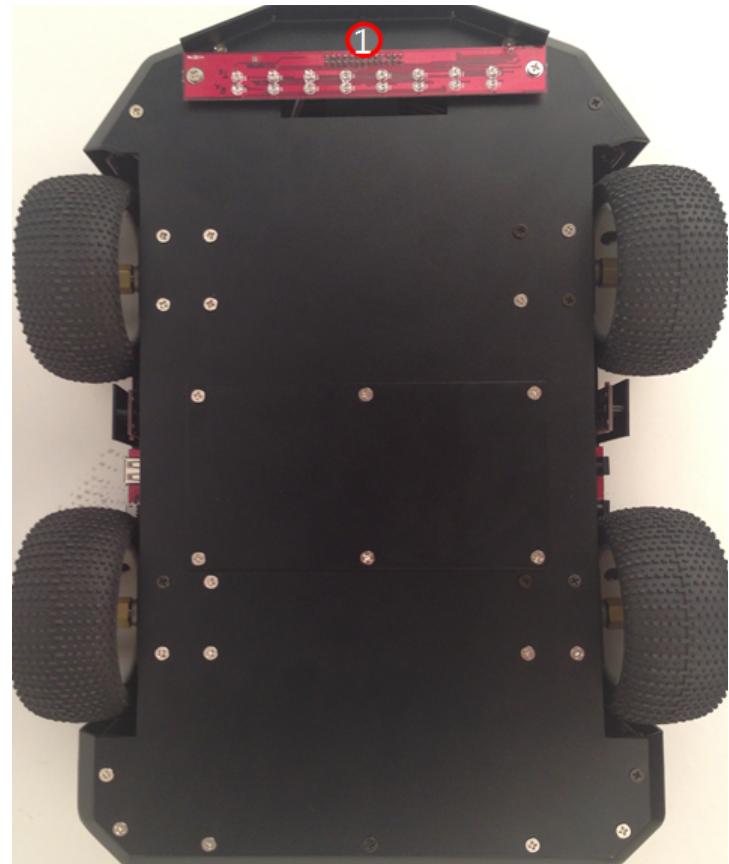
- SmartCAR Left side

- ① Motor and wheel
 - One wheel driven by one motor
 - Two wheels and motors in the same side operate together
- ② Adapter In Port: For connecting adapter
- ③ Charge In Port: For charging battery
- ④ Power In Port: For connecting to battery
- ⑤ Battery



SmartCAR Components

- SmartCAR
Bottom side
① Infrared sensor
: For detecting lines on
the floor e.g., line tracer

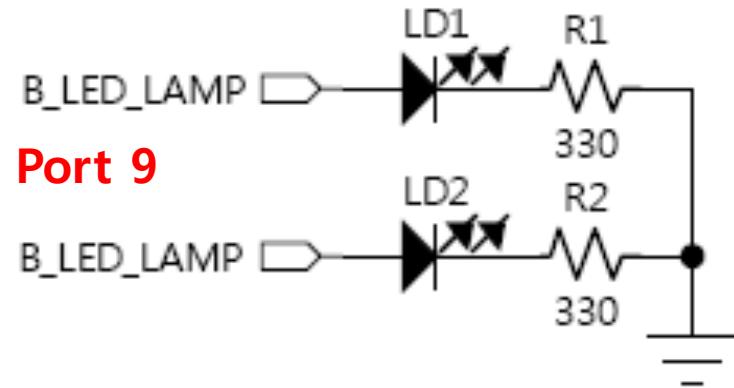
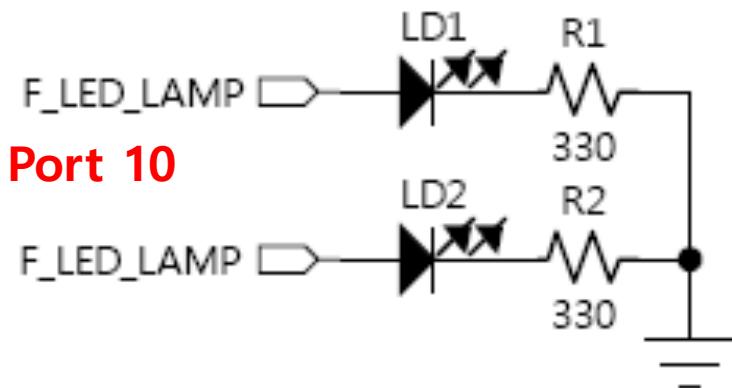


Circuit for Front LED & Rear LED

- Front LED(Left) & Rear LED (Right)



- Front LED (Left: White), Rear LED(Right : Red) LED circuit



Today

- Lab announcement
- SmartCAR platform
- **Arduino programming environment setup**
- Lab Assignments #1-1, #1-2
- Course announcement



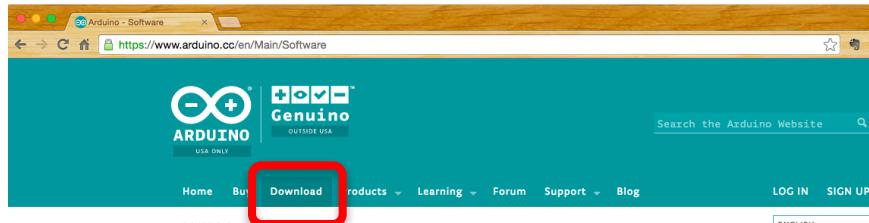
Arduino Programming Environment

- Two IDEs
 - 1. Official Arduino IDE
 - Download at <http://arduino.cc>
 - 2. Androx Studio IDE (based on Eclipse)
 - Download at
<https://www.dropbox.com/s/1jooitdit1dg8g3/Androx.zip?dl=0>



1-1. Download the Arduino IDE

- Go to <http://arduino.cc>
 - Click on Download in the above menu
 - Click on Windows Installer of Arduino 1.6.8

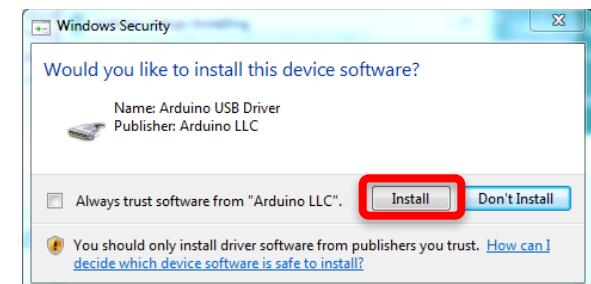
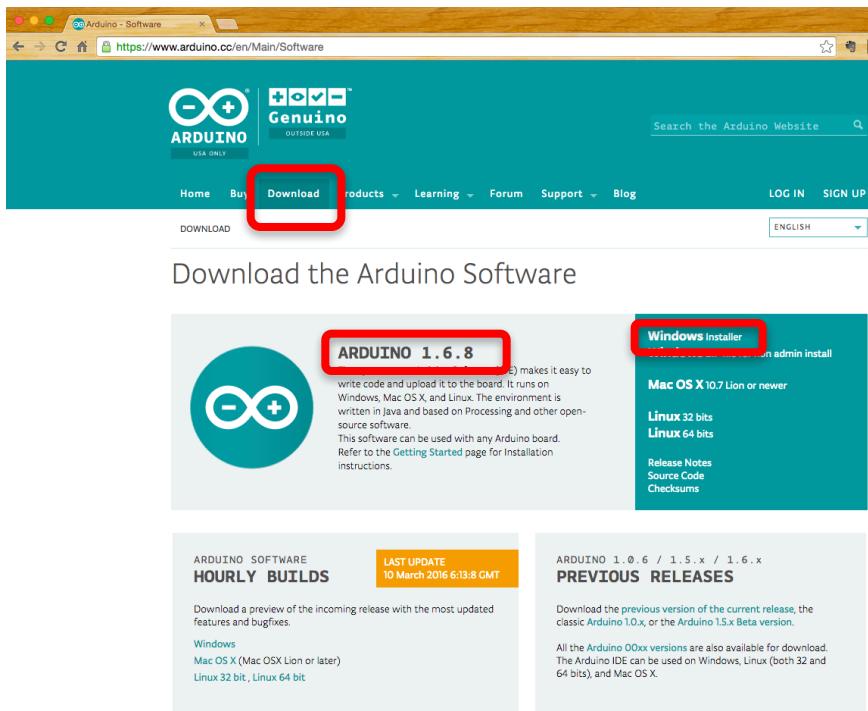


Download the Arduino Software



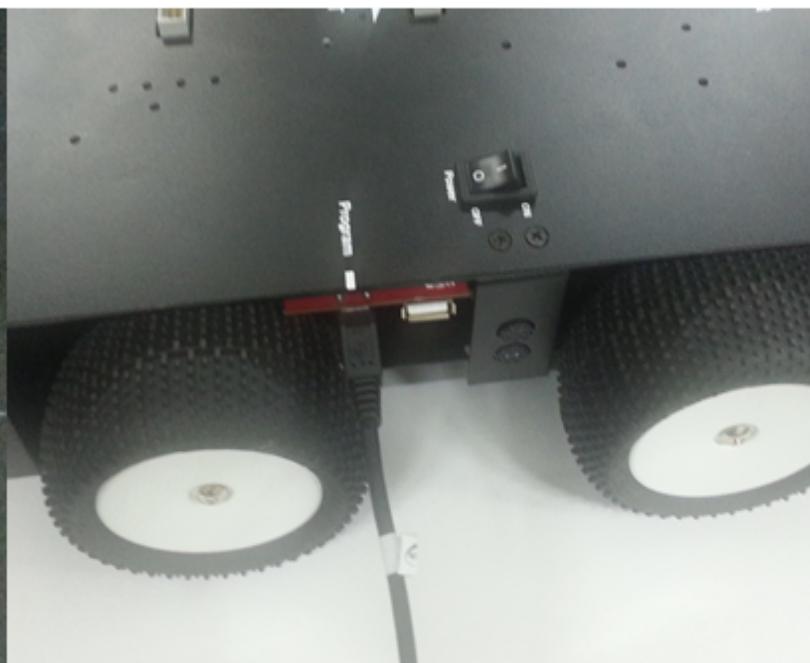
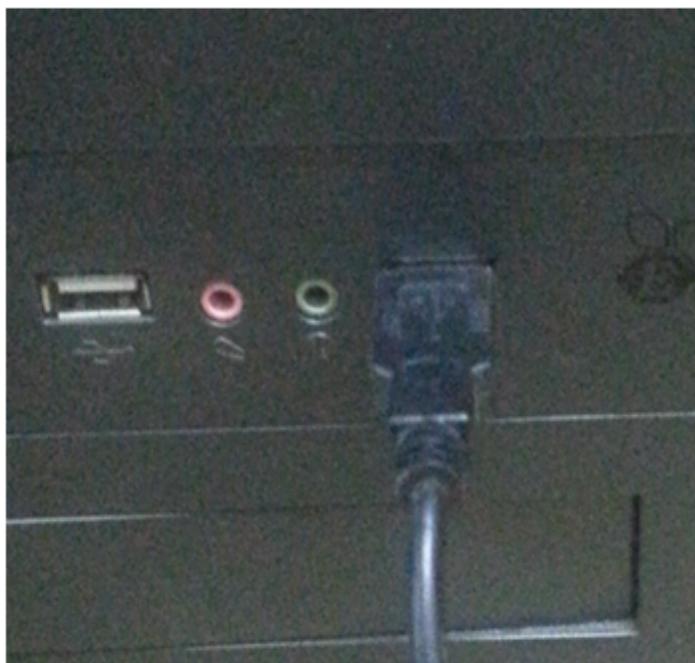
1-2. Install the Arduino IDE & Driver

- Install the Arduino IDE
- Also, install the necessary driver if asked



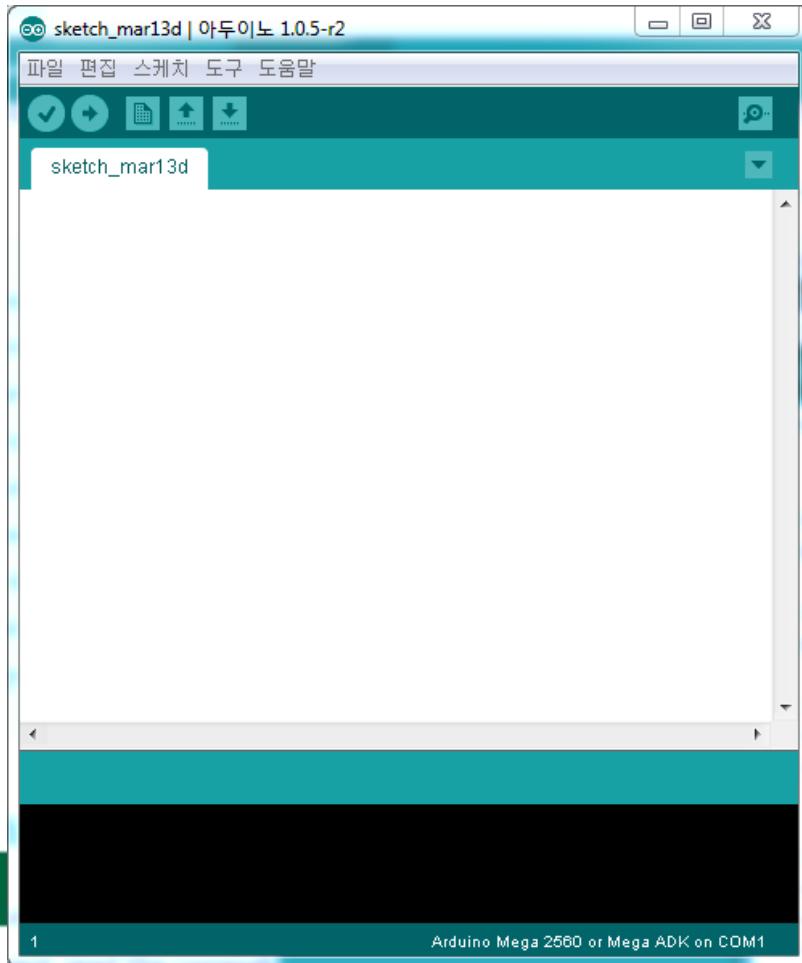
1-3. Connect your PC to SmartCAR

- Use USB cable to connect SmartCAR to your PC



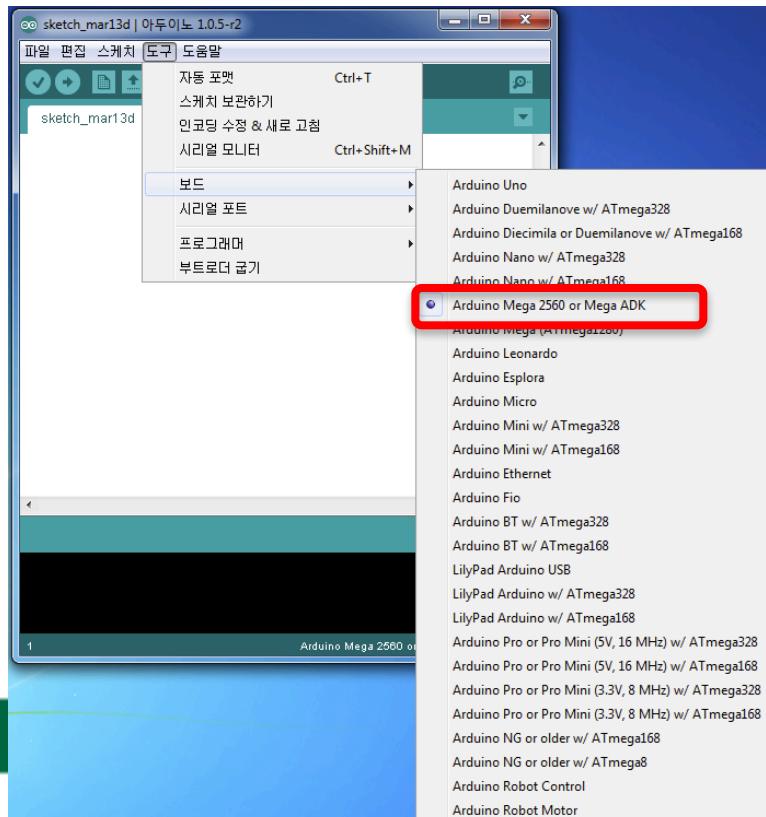
1-4. Launch the Arduino IDE

- Launch the Arduino IDE



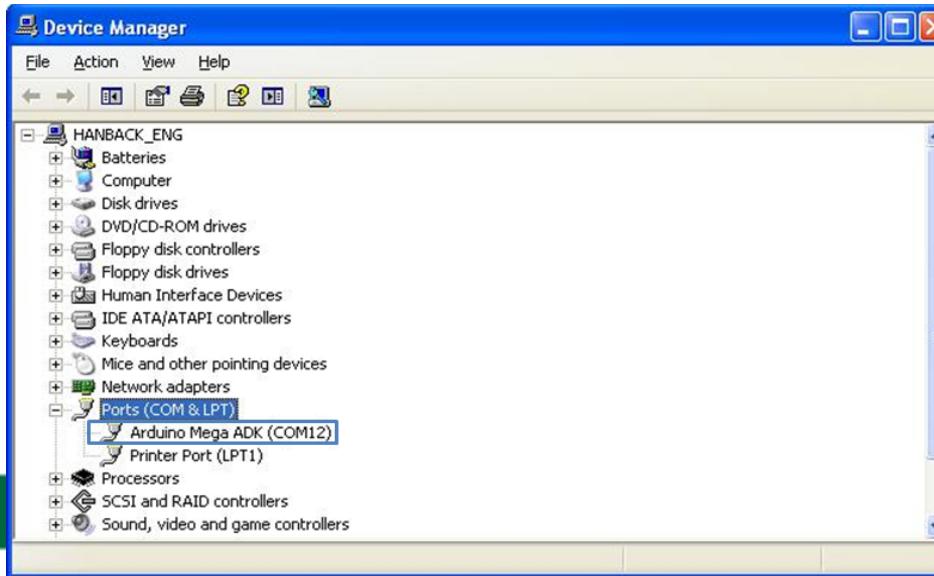
1-5. Select your board

- Go to “Tools”
- For “Board”
 - Select “Arduino Mega 2560 or Mega ADK



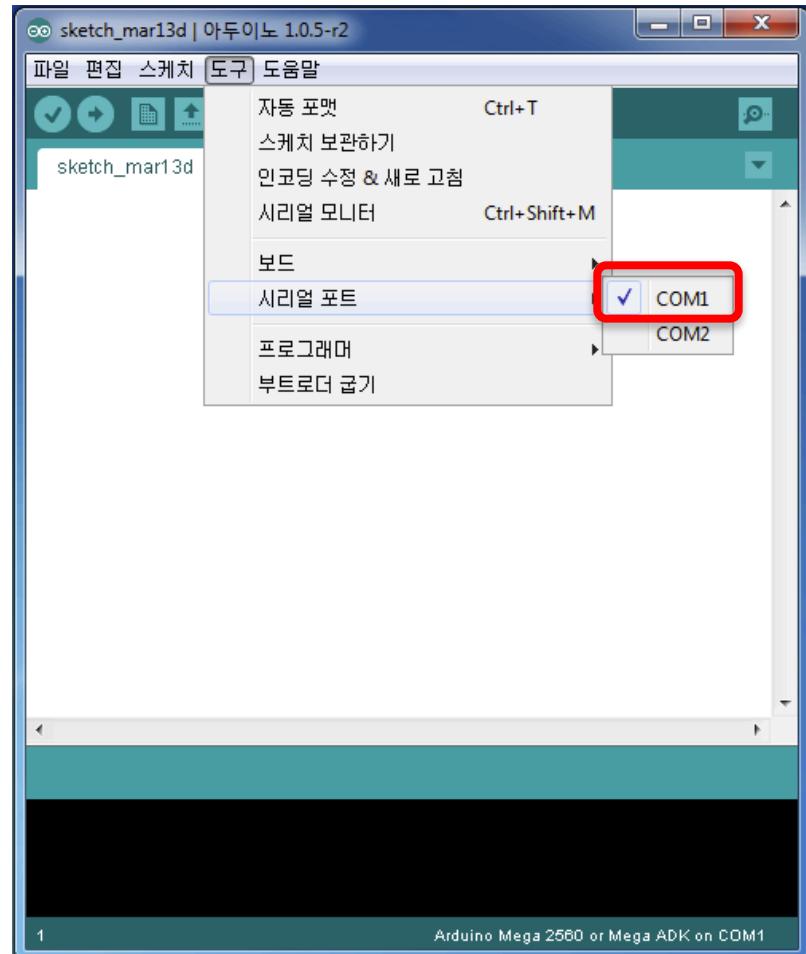
1-6. Check serial port number

- Launch Control Panel
- Click on “Hardware & Sound”
- Under Devices and Printers, click on “Device Manager”
 - Arduino Mega ADK (COMXX)



1-7. Select your serial port

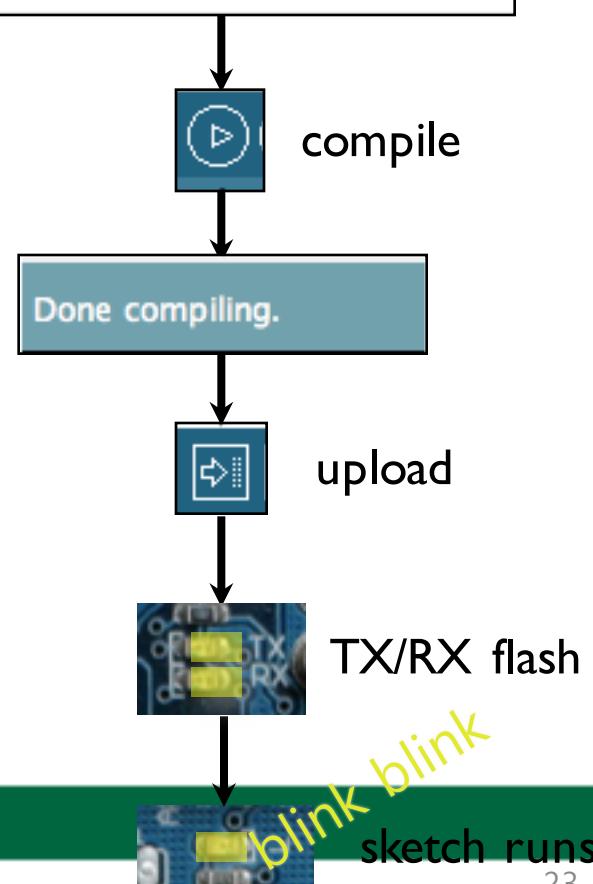
- Go to “Tools”
- For “Serial Port”
 - Select **COMxx**
- Should display the “**V**” mark



1-8. Program Using Arduino IDE

- Write your sketch
- Press Compile button
(to check for errors)
- Press Upload button to
program Arduino board
with your sketch

```
void setup() {  
    pinMode(ledPin, OUTPUT);      // sets t  
}  
void loop() {  
    digitalWrite(ledPin, HIGH);   // sets t  
    delay(1000);                // waits  
    digitalWrite(ledPin, LOW);    // sets t  
    delay(1000);                // waits  
}
```



LED Example

- Front LED should be turned ON for 1 second and OFF for 1 second, repeatedly

```
#define FRONT_LED_PIN 10
#define REAR_LED_PIN 9

void setup(){
    pinMode(FRONT_LED_PIN, OUTPUT);
}

void loop(){
    digitalWrite(FRONT_LED_PIN, HIGH);
    delay(1000);
    digitalWrite(FRONT_LED_PIN, LOW);
    delay(1000);
}
```



Arduino Programming Environment

- Two IDEs
 - 1. Official Arduino IDE
 - Download at <http://arduino.cc>
 - 2. Androx Studio IDE (based on Eclipse)
 - Download at
<https://www.dropbox.com/s/1jooitdit1dg8g3/Androx.zip?dl=0>



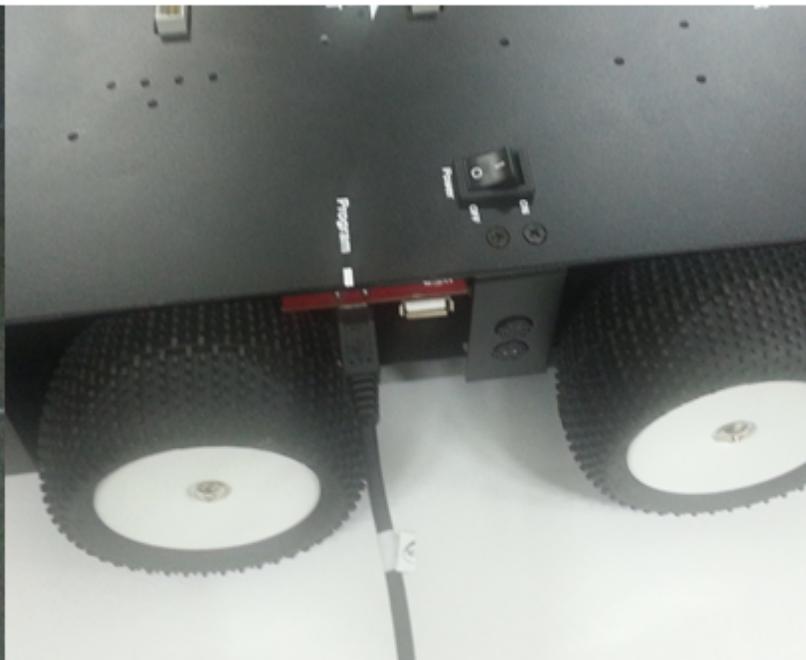
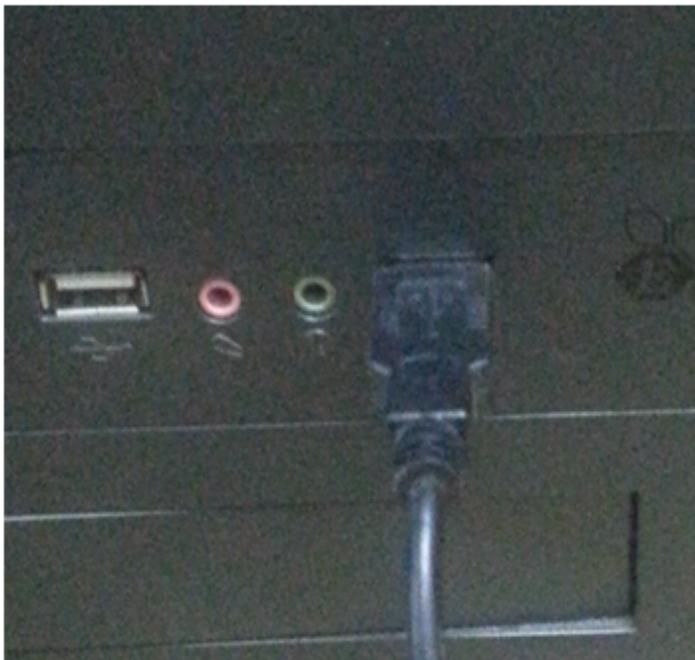
2-1. Download the Androx Studio

- Go to
<https://www.dropbox.com/s/1jooitdit1dg8g3/Androx.zip?dl=0>
- Unzip the file
- Install AndroXStudio_Setup_2.0.4.exe file
 - Will take about 10 minutes



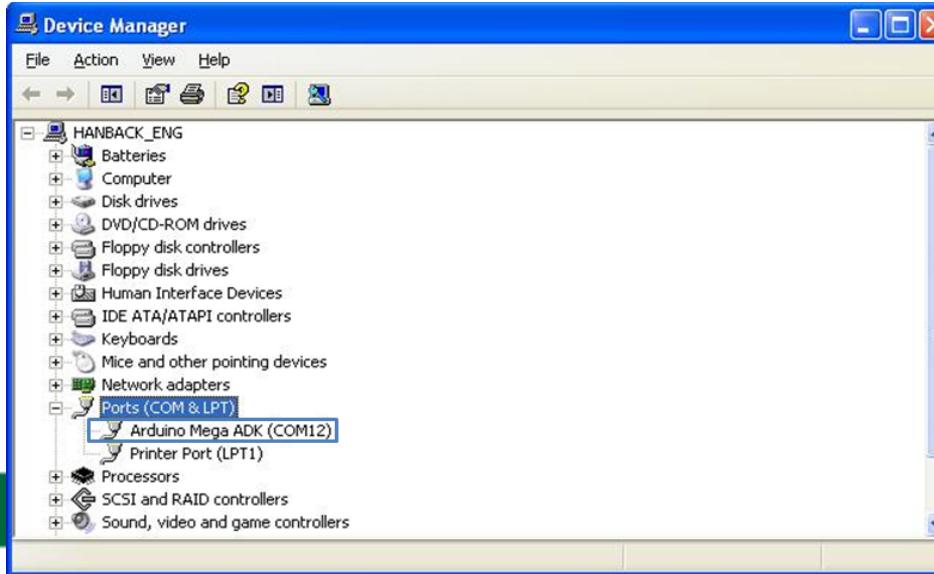
2-2. Connect your PC to SmartCAR

- Use USB cable to connect SmartCAR to your PC

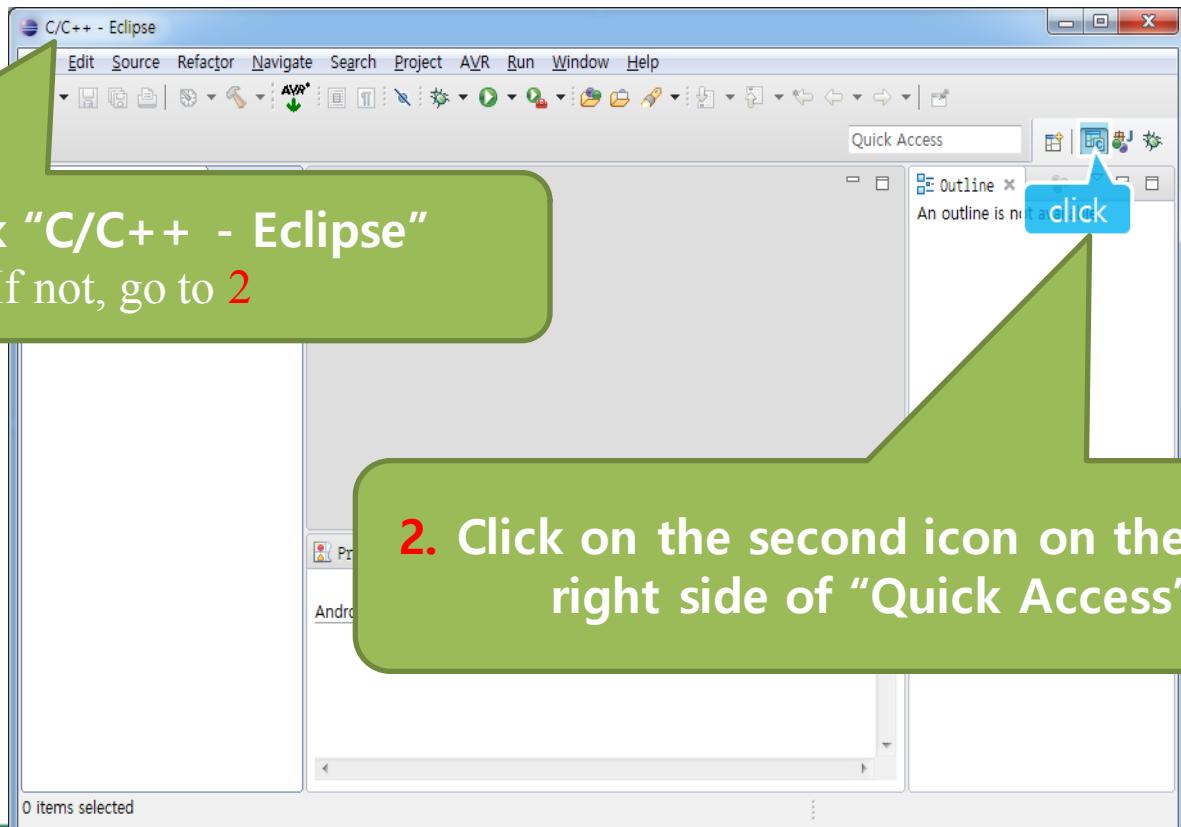


2-3. Check serial port number

- Launch Control Panel
- Click on “Hardware & Sound”
- Under Devices and Printers, click on “Device Manager”
 - Arduino Mega ADK (COMXX)

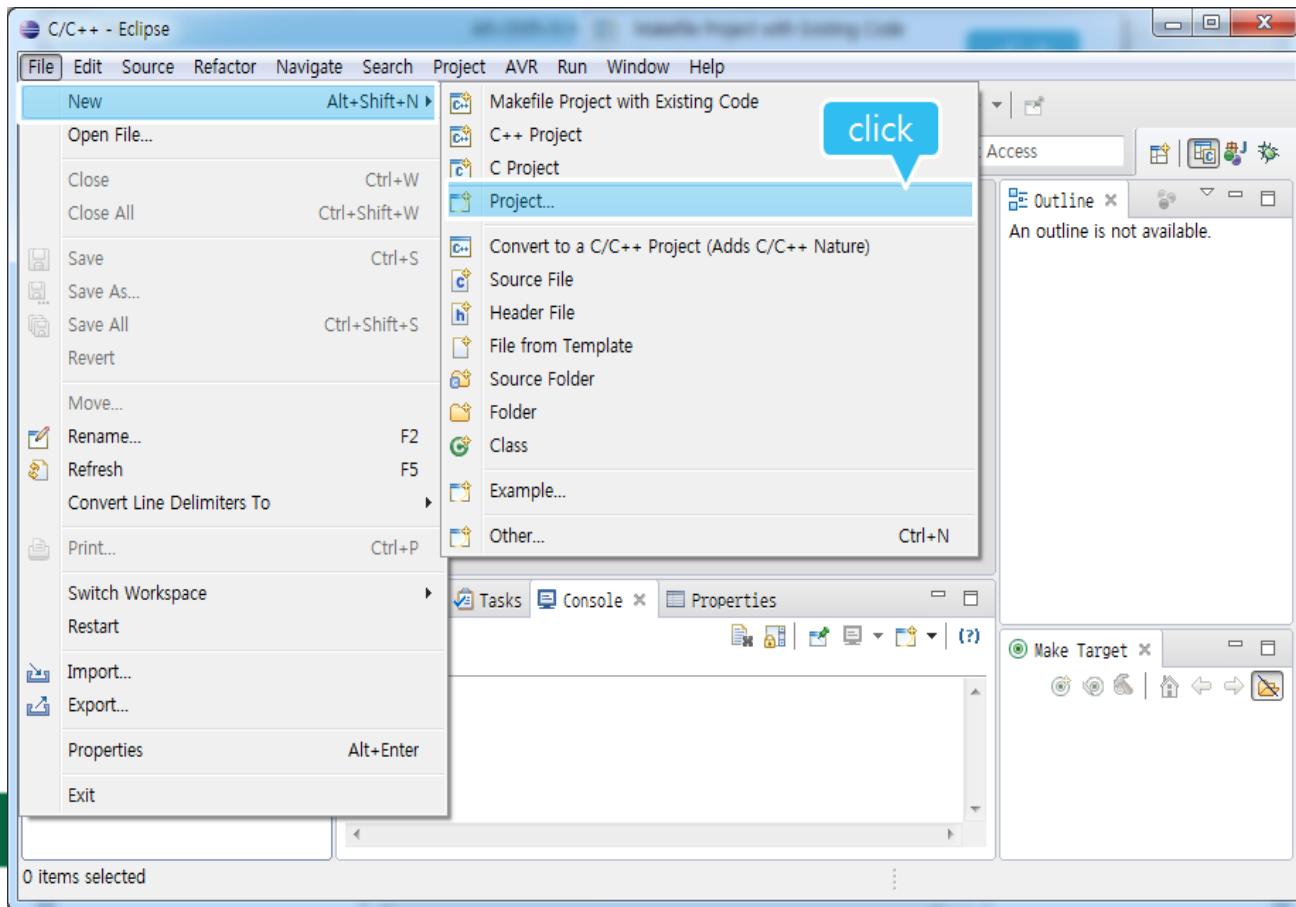


2-4. Launch the Androx Studio



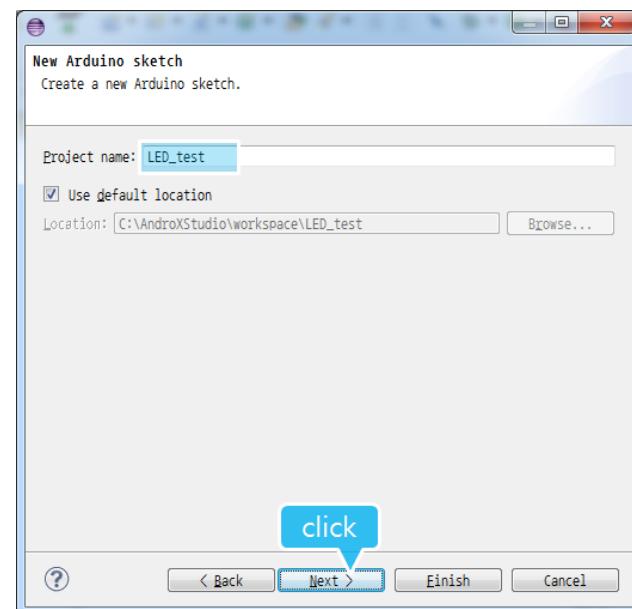
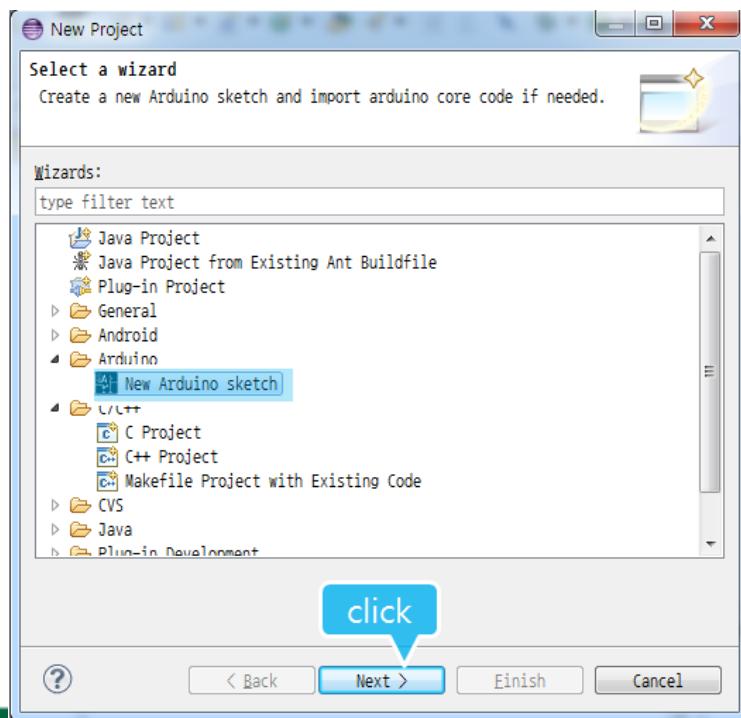
2-5. Create a new Project

- Create a new Project
 - Click on “File – New – Project”



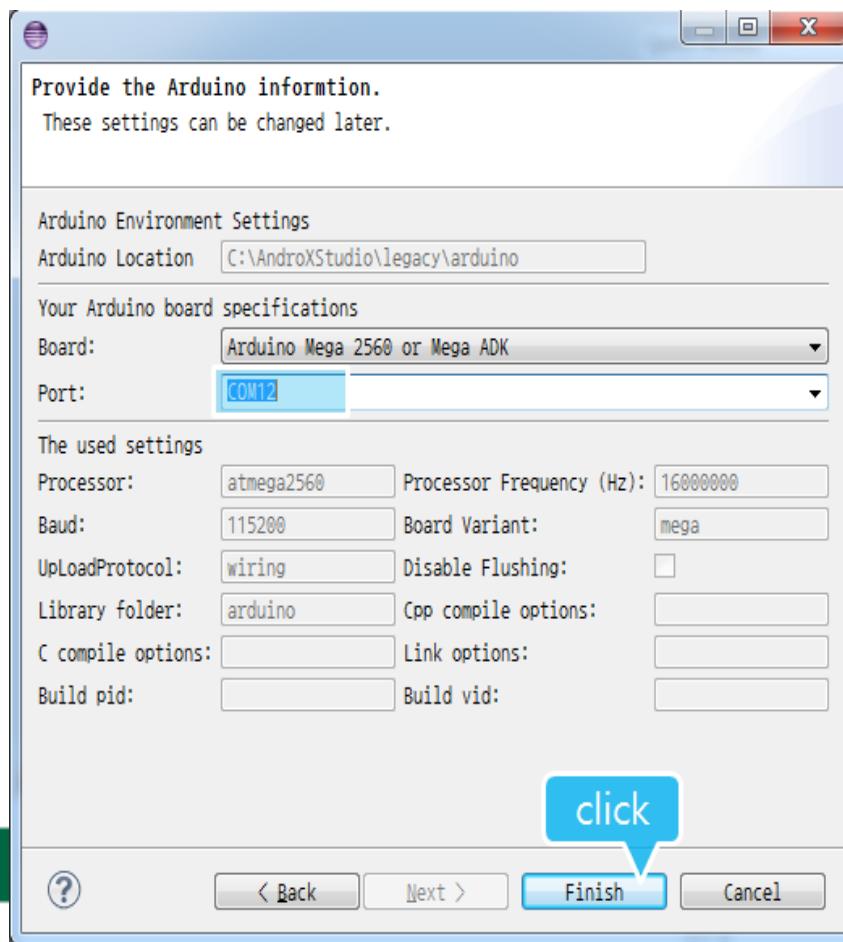
2-6. Set up the new Project

- Select “Arduino – New Arduino sketch”
- Click on “Next”
- In “Project name”, enter “LED_test”
- Click on “Next”



2-7. Select your serial port

- Select your serial port
- Click on “Finish”



2-8. Write your code

- Double click on LED_test.cpp
- Write your code

The screenshot shows the Eclipse C/C++ IDE interface. The title bar reads "C/C++ - LED_test/LED_test.cpp - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, AVR, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, and Build. The Project Explorer view on the left shows a project named "Arduino_Mega_2560_or_Mega" containing "Includes", "arduino", and "mega" folders, and a "LED_test" folder containing "Includes", "LED_test.cpp", and "LED_test.h". The central editor window displays the "LED_test.cpp" file with the following code:

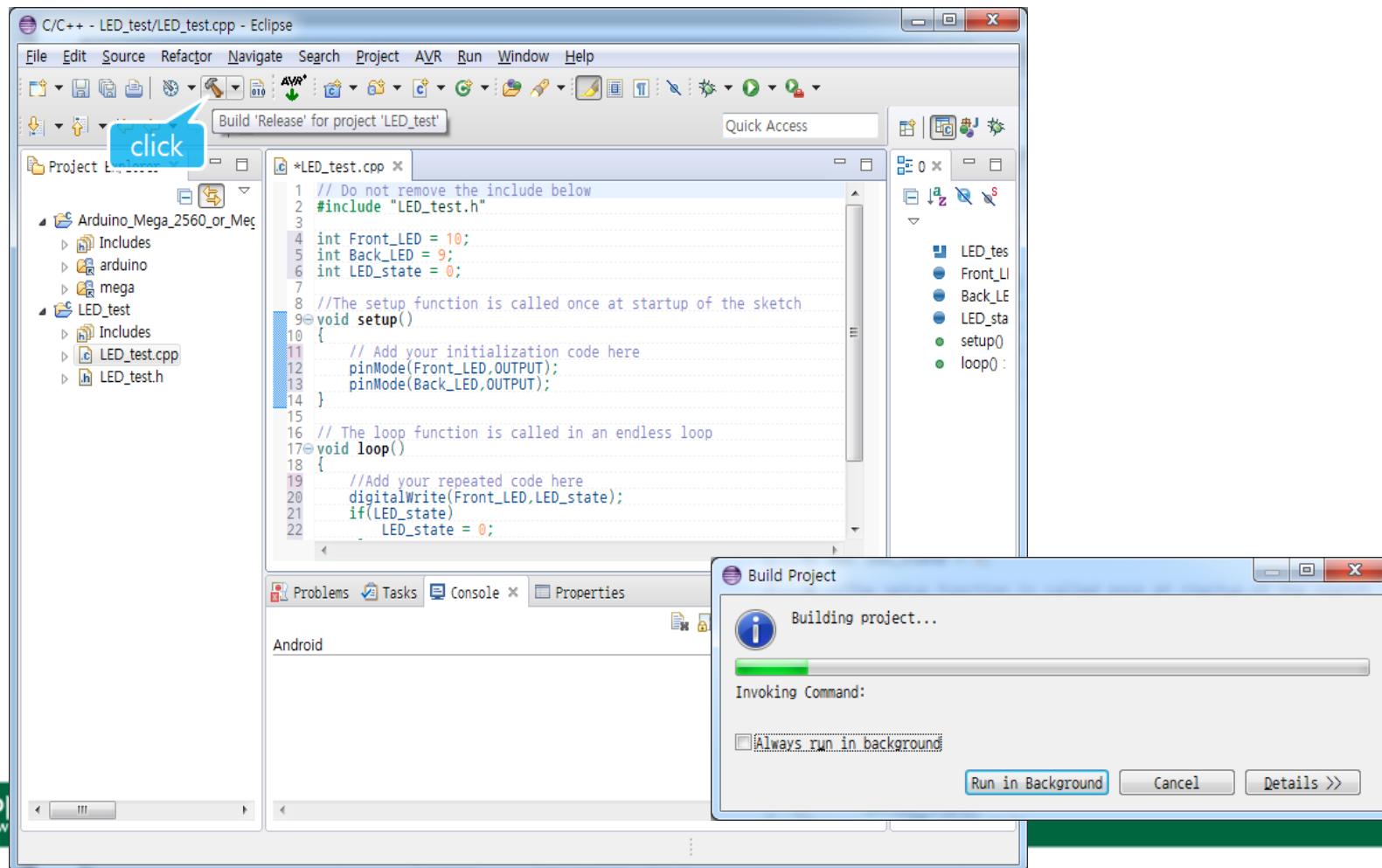
```
1 // Do not remove the include below
2 #include "LED_test.h"
3
4 int Front_LED = 10;
5 int Back_LED = 9;
6 int LED_state = 0;
7
8 //The setup function is called once at startup of the sketch
9 void setup()
10 {
11     //Add your initialization code here
12     pinMode(Front_LED,OUTPUT);
13     pinMode(Back_LED,OUTPUT);
14 }
15
16 // The loop function is called in an endless loop
17 void loop()
18 {
19     //Add your repeated code here
20     digitalWrite(Front_LED,LED_state);
21     if(LED_state)
22         LED_state = 0;
23     else
24         LED_state = 1;
25     delay(1000);
26 }
```

The right-hand margin shows code completion suggestions: LED_tes, Front_LL, Back_LE, LED_sta, setup(), and loop(). Below the editor are tabs for Problems, Tasks, Console, and Properties, with the Properties tab selected. At the bottom, there are buttons for Writable, Smart Insert, and a status bar showing "1 : 1".



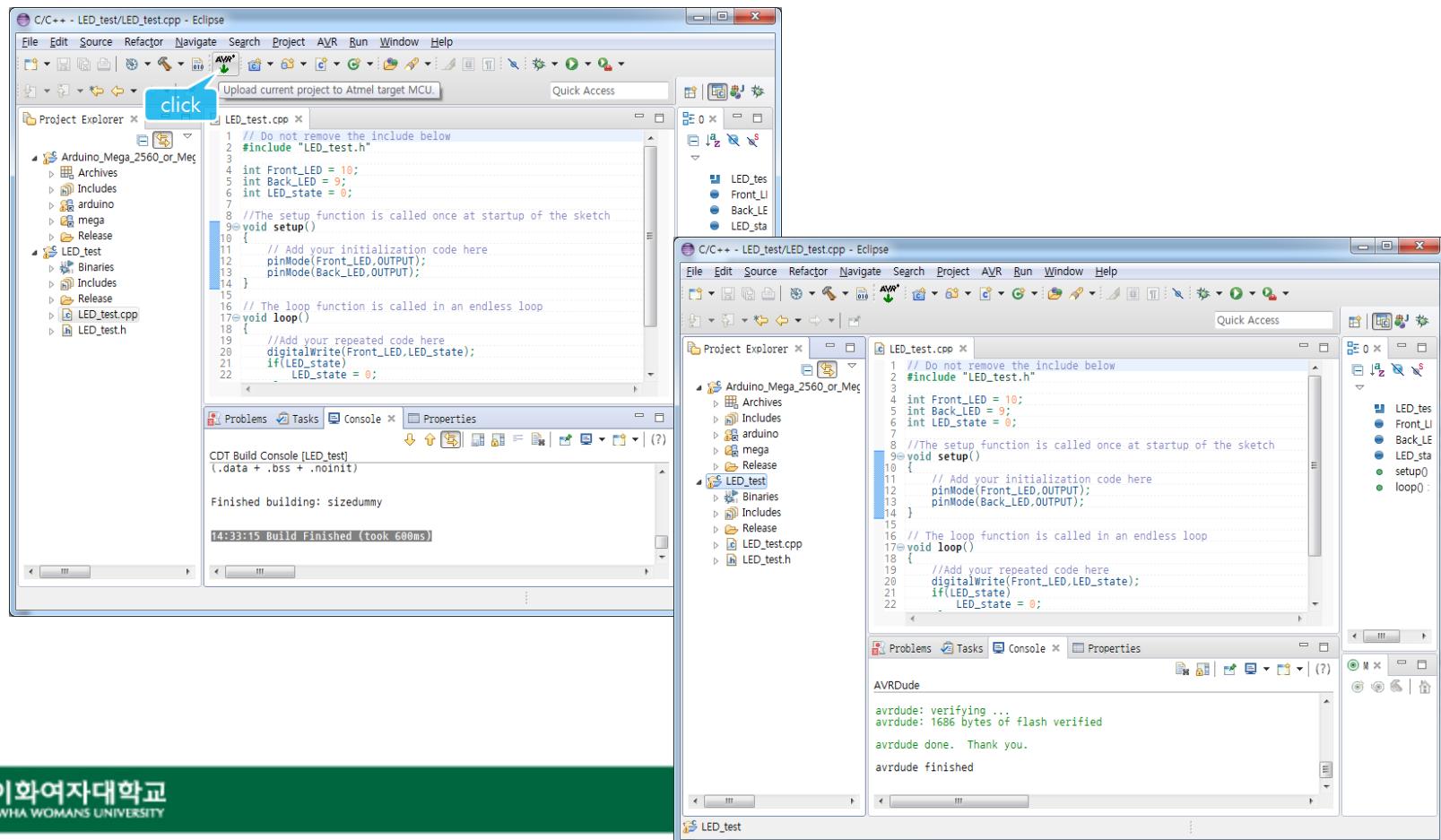
2-9. Compile your code

- Compile your code



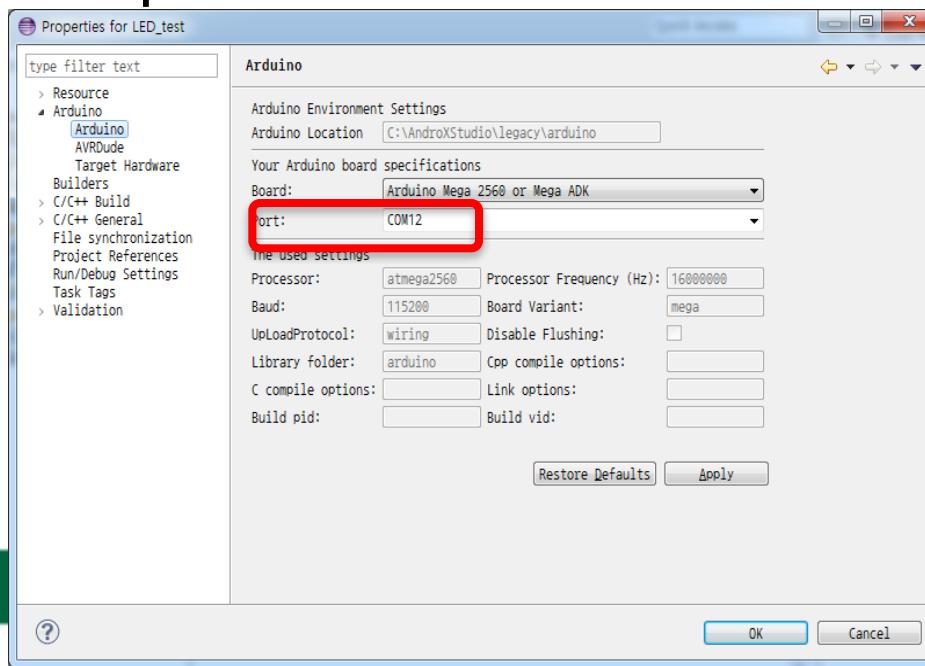
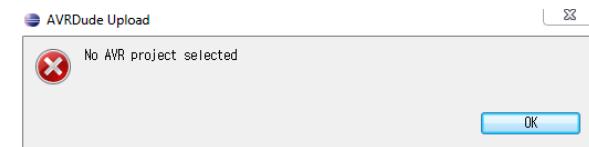
2-10. Upload and Run your program

- Upload your code to Arduino board



2-11. Configure your serial port

- If the port is not properly configured,
 - Right click on the “LED_test” project
 - Select “Properties”
- Click on “Arduino – Arduino”
- Select your port and click on “OK”



Today

- Lab announcement
- SmartCAR platform
- Arduino programming environment setup
- **Lab Assignments #1-1, #1-2**
- Course announcement



Lab Assignment #1-1

- Program using **Arduino IDE**
- Front LED should be turned ON, while rear LED should be turned OFF for 1 second
- Then, front LED should be turned OFF, while rear LED should be turned ON for 1 second
- Repeat the above operations
- Upload your sketch file to Cyber Campus
- Show your result to TA or instructor



Lab Assignment #1-2

- Program using **Androx Studio**
- Front LED is initially OFF
- Rear LED is initially ON
- Front LED should be incrementally brightening for about 3 seconds, and incrementally darkening for about 3 seconds, repeatedly
- At the same time, rear LED should be incrementally darkening for about 3 seconds, and incrementally brightening for about 3 seconds, repeatedly
- Hint) Use “**analogWrite()**”
- Upload your cpp file to Cyber Campus
- Show your result to TA or instructor



Course Announcement

- Next lecture, we will continue to study
 - ATmega2560 MCU
 - Architecture
 - I/O ports
- Next lab session, we will cover
 - Advanced LED control
 - Bluetooth communication between SmartCAR and Nexus 7 tablet

