

---

# 임베디드 시스템

LED 및 스위치 제어

---

Inhyeok Kang

M23522@hallym.ac.kr

연구실: 공학관 1321호

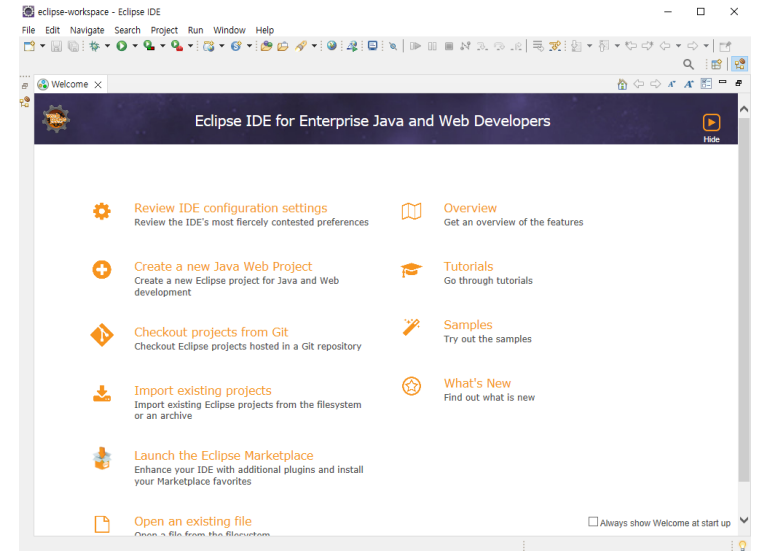
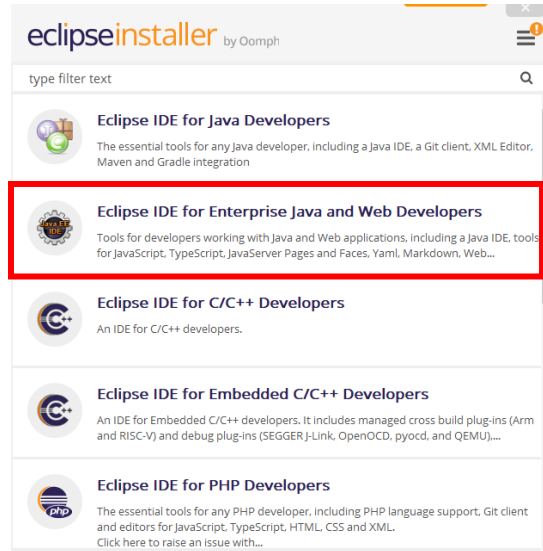
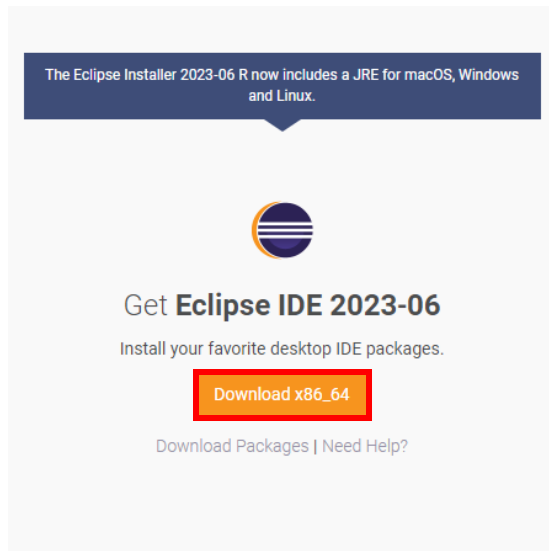
---

# Contents

1. Eclipse 설치
2. JDK 설치
3. Raspberry Pi GPIO Pinout
4. WiringPi
5. LED 및 스위치 연결
6. Eclipse 세팅
7. LED 및 스위치 제어

# Eclipse 설치

- 홈페이지(<https://www.eclipse.org/downloads/>) 접속 후 다운로드 및 설치
  - 2번째 위치한 [Eclipse IDE for Enterprise Java and Web Developers] 선택



---

# JDK 설치

- 1. 스마트리드에서 [pi4j.zip], [JDK 1.8.zip] 파일 다운로드
  - 임의의 폴더에 압축 해제(해당 폴더 경로 기억할 것)

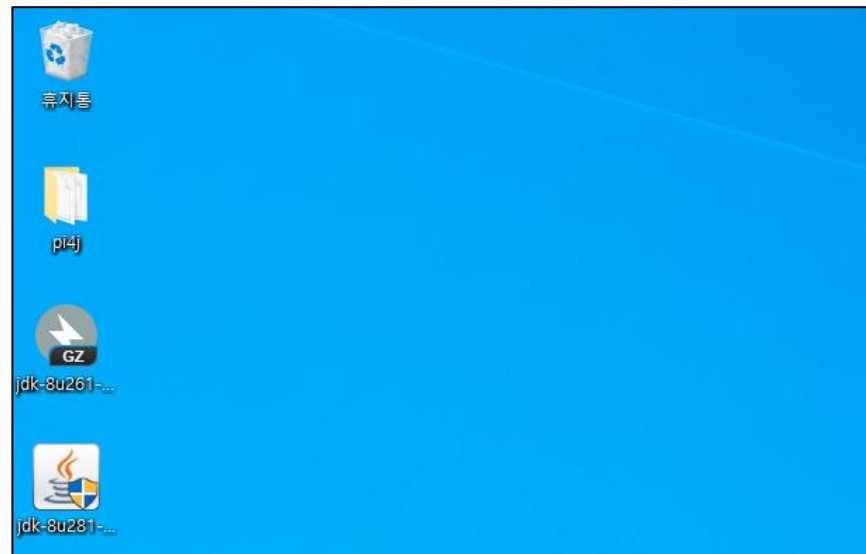
3주차 [9월11일 - 9월17일]



pi4j.zip



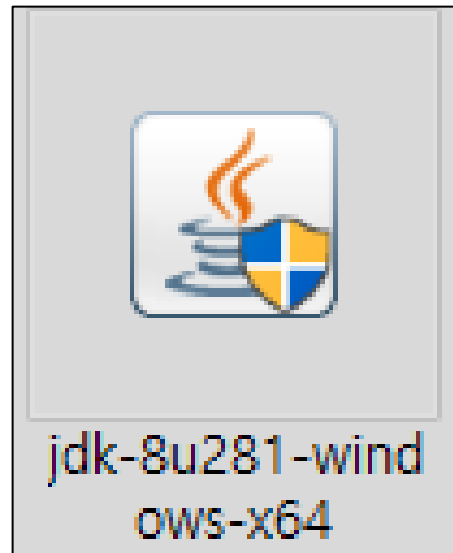
JDK 1.8.zip



---

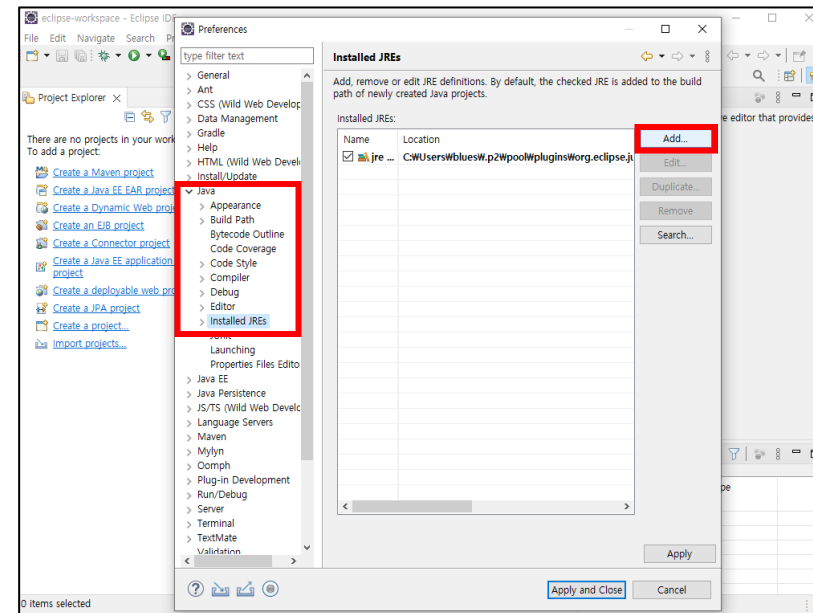
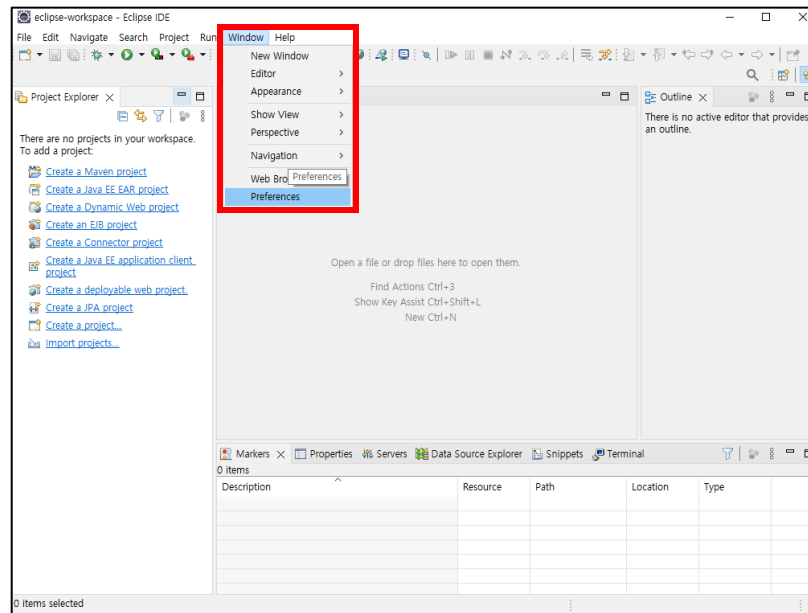
# JDK 설치

- 2. [jdk-8u281-windows-x64.exe] 실행
  - JDK 1.8 설치 완료



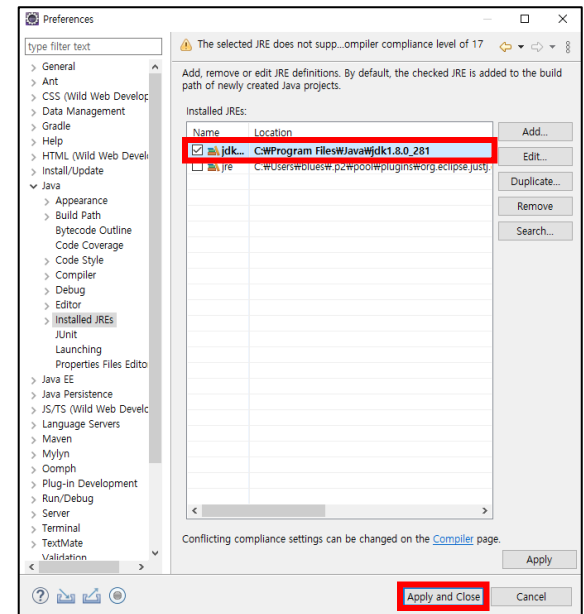
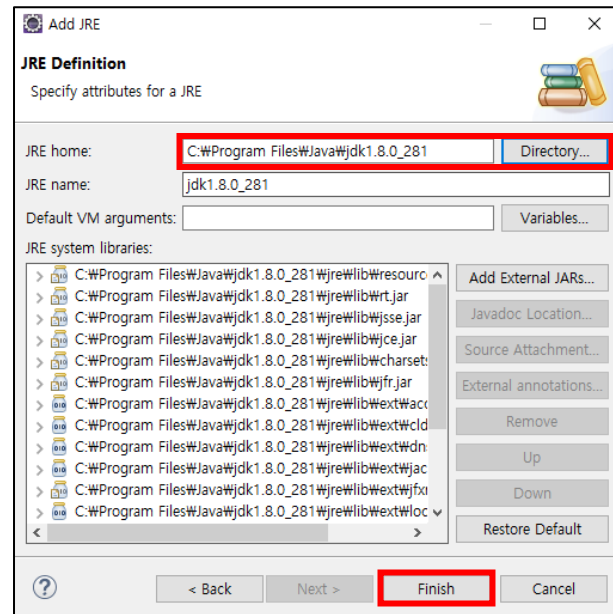
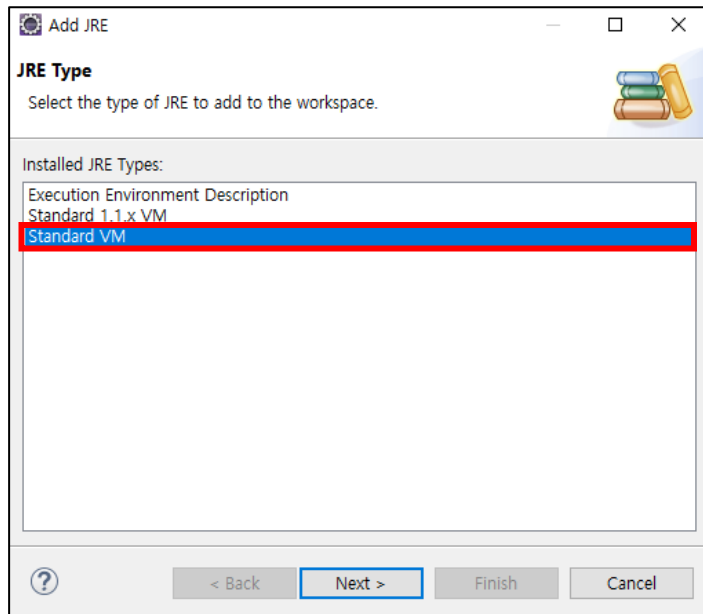
# JDK 설치

- 3. Eclipse에 JDK 1.8 추가
  - Eclipse 실행
  - **[Window]** 클릭 → **[Preferences]** 클릭
  - **[Java]** 클릭 → **[Installed JREs]** 클릭 → **[Add]** 클릭



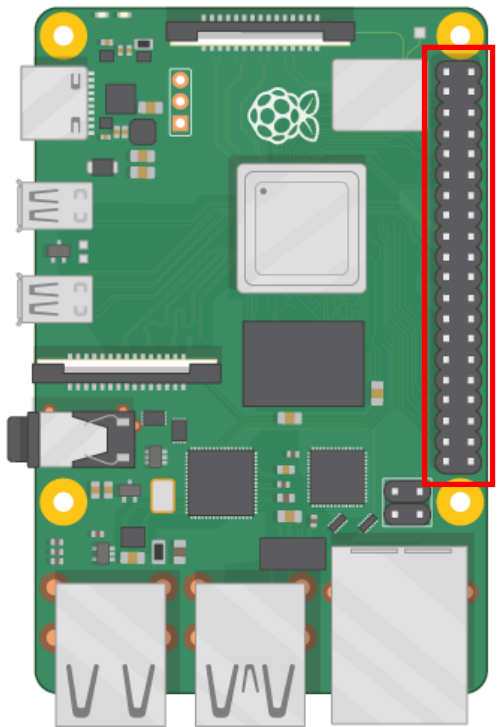
# JDK 설치

- 3. Eclipse에 JDK 1.8 추가
  - [Standard VM] 선택
  - [Directory] 선택 → [C:\Program Files\Java\jdk1.8.0\_281] 추가 → [Finish] 클릭
  - [jdk1.8.0\_281] 선택 → [Apply and Close] 클릭



# Raspberry Pi GPIO Pinout

- Raspberry Pi 3B+ & 4B GPIO Pinout
  - 두 모델 모두 동일



GPIO Pins

Raspberry Pi 3 Model B (J8 Header)					
GPIO#	NAME		NAME	GPIO#	
	3.3 VDC Power	1		2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5		6	Ground
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART)
	Ground	9		10	GPIO 16 RxD (UART)
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0
2	GPIO 2	13		14	Ground
3	GPIO 3	15		16	GPIO 4
	3.3 VDC Power	17		18	GPIO 5
12	GPIO 12 MOSI (SPI)	19		20	Ground
13	GPIO 13 MISO (SPI)	21		22	GPIO 6
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI)
	Ground	25		26	GPIO 11 CE1 (SPI)
30	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM)
21	GPIO 21 GPCLK1	29		30	Ground
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0
23	GPIO 23 PWM1	33		34	Ground
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27
25	GPIO 25	37		38	GPIO 28 PCM_DIN
	Ground	39		40	GPIO 29 PCM_DOUT

**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Raspberry Pi 4 Model B (J8 Header)					
GPIO#	NAME		NAME	GPIO#	
	3.3 VDC Power	1		2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5		6	Ground
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART)
	Ground	9		10	GPIO 16 RxD (UART)
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0
2	GPIO 2	13		14	Ground
3	GPIO 3	15		16	GPIO 4
	3.3 VDC Power	17		18	GPIO 5
12	GPIO 12 MOSI (SPI)	19		20	Ground
13	GPIO 13 MISO (SPI)	21		22	GPIO 6
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI)
	Ground	25		26	GPIO 11 CE1 (SPI)
30	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM)
21	GPIO 21 GPCLK1	29		30	Ground
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0
23	GPIO 23 PWM1	33		34	Ground
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27
25	GPIO 25	37		38	GPIO 28 PCM_DIN
	Ground	39		40	GPIO 29 PCM_DOUT





















**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>



# Raspberry Pi GPIO Pinout

- Raspberry Pi 3B+ & 4B GPIO Pinout
  - 라즈베리 파이 3B+ & 4B GPIO 핀: 40개
  - 각 핀마다 고유의 기능이 정해져 있음
    - ✓ 기본적으로 모든 핀은 디지털 신호를 입력/출력
  - 물리적인 핀 번호와 GPIO 핀 번호가 서로 다름
  - GPIO 핀 구성
    - ✓ 5V output: 2개
    - ✓ 3.3V output: 2개
    - ✓ Ground (GND): 8개
    - ✓ 나머지 핀: 입출력을 위한 핀 (각기 다른 기능 보유)
      - 향후 실습에서 각기 다른 용도의 핀을 다룰 예정

Raspberry Pi 4 Model B (J8 Header)									
GPIO#	NAME			NAME	GPIO#				
	3.3 VDC Power	1		2	5.0 VDC Power				
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power				
9	GPIO 9 SCL1 (I2C)	5		6	Ground				
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART)				
	Ground	9		10	GPIO 16 RxD (UART)				
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0				
2	GPIO 2	13		14	Ground				
3	GPIO 3	15		16	GPIO 4				
	3.3 VDC Power	17		18	GPIO 5				
12	GPIO 12 MOSI (SPI)	19		20	Ground				
13	GPIO 13 MISO (SPI)	21		22	GPIO 6				
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI)				
	Ground	25		26	GPIO 11 CE1 (SPI)				
30	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM)				
21	GPIO 21 GPCLK1	29		30	Ground				
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0				
23	GPIO 23 PWM1	33		34	Ground				
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27				
25	GPIO 25	37		38	GPIO 28 PCM_DIN				
	Ground	39		40	GPIO 29 PCM_DOUT				

**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>






# WiringPi

- WiringPi
  - 모든 Raspberry Pi 에서 사용되는 BCM2835, BCM2836 및 BCM2837 SoC 장치를 위해 C언어로 작성된 PIN 기반 GPIO 액세스 라이브러리
  - WiringPi 라이브러리를 사용해 라즈베리 파이의 GPIO 핀을 제어함

```
root@raspberrypi:/home/pi/ES_proj# gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2   | 8   | 3.3v     |      |   | 1   | 2   |      | 5v       |     |     |
| 3   | 9   | SDA.1    | OUT  | 1 | 3   | 4   |      | 5v       |     |     |
| 4   | 7   | SCL.1    | OUT  | 0 | 5   | 6   |      | 0v       |     |     |
|     |     | GPIO. 7  |      |   | 7   | 8   | 1 IN | TxD      | 15  | 14  |
|     |     | 0v       |      |   | 9   | 10  | 1 IN | RxD      | 16  | 15  |
| 17  | 0   | GPIO. 0  | IN   | 0 | 11  | 12  | 0 IN | GPIO. 1  | 1   | 18  |
| 27  | 2   | GPIO. 2  | IN   | 0 | 13  | 14  |      | 0v       |     |     |
| 22  | 3   | GPIO. 3  | IN   | 0 | 15  | 16  | 0 IN | GPIO. 4  | 4   | 23  |
|     |     | 3.3v     |      |   | 17  | 18  | 0 IN | GPIO. 5  | 5   | 24  |
| 10  | 12  | MOSI     | IN   | 0 | 19  | 20  |      | 0v       |     |     |
| 9   | 13  | MISO     | IN   | 0 | 21  | 22  | 0 IN | GPIO. 6  | 6   | 25  |
| 11  | 14  | SCLK     | IN   | 0 | 23  | 24  | 1 IN | CE0      | 10  | 8   |
|     |     | 0v       |      |   | 25  | 26  | 1 IN | CE1      | 11  | 7   |
| 0   | 30  | SDA.0    | IN   | 1 | 27  | 28  | 1 IN | SCL.0    | 31  | 1   |
| 5   | 21  | GPIO.21  | IN   | 1 | 29  | 30  |      | 0v       |     |     |
| 6   | 22  | GPIO.22  | IN   | 1 | 31  | 32  | 0 IN | GPIO.26  | 26  | 12  |
| 13  | 23  | GPIO.23  | IN   | 0 | 33  | 34  |      | 0v       |     |     |
| 19  | 24  | GPIO.24  | IN   | 0 | 35  | 36  | 0 IN | GPIO.27  | 27  | 16  |
| 26  | 25  | GPIO.25  | IN   | 0 | 37  | 38  | 0 IN | GPIO.28  | 28  | 20  |
|     |     | 0v       |      |   | 39  | 40  | 0 IN | GPIO.29  | 29  | 21  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

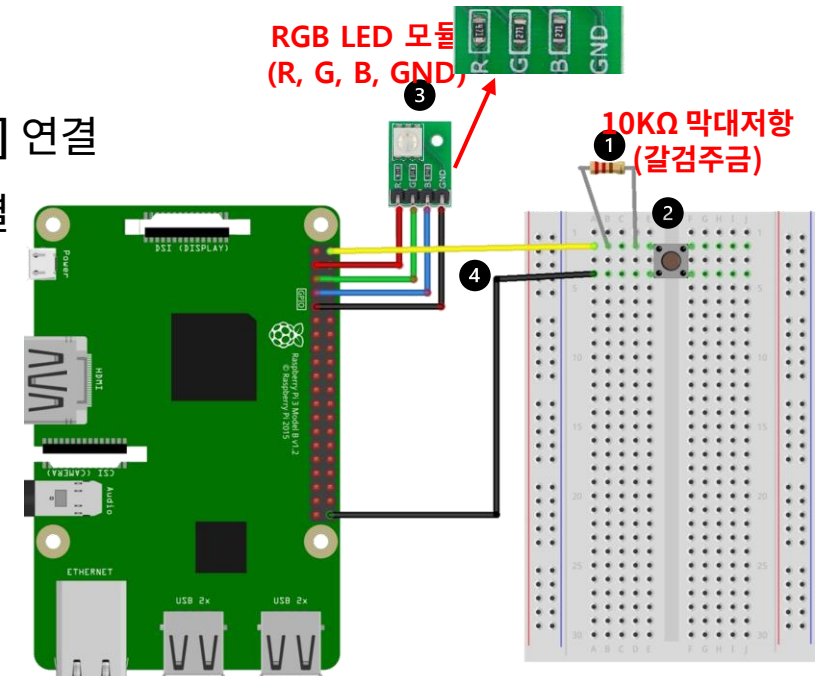
# LED 및 스위치 연결

- 1. 구성품 준비

번호	구성요소	사진
1	Raspberry Pi 본체	 <Raspberry Pi 3 Model B+> <Raspberry Pi 4 Model B>
2	점프 와이어(M/F 2개, F/F 4개)	
3	LED 센서 모듈	
4	스위치	
5	막대저항(10K $\Omega$ ) 10K $\Omega$ : 갈색-검은색-주황색-금색	

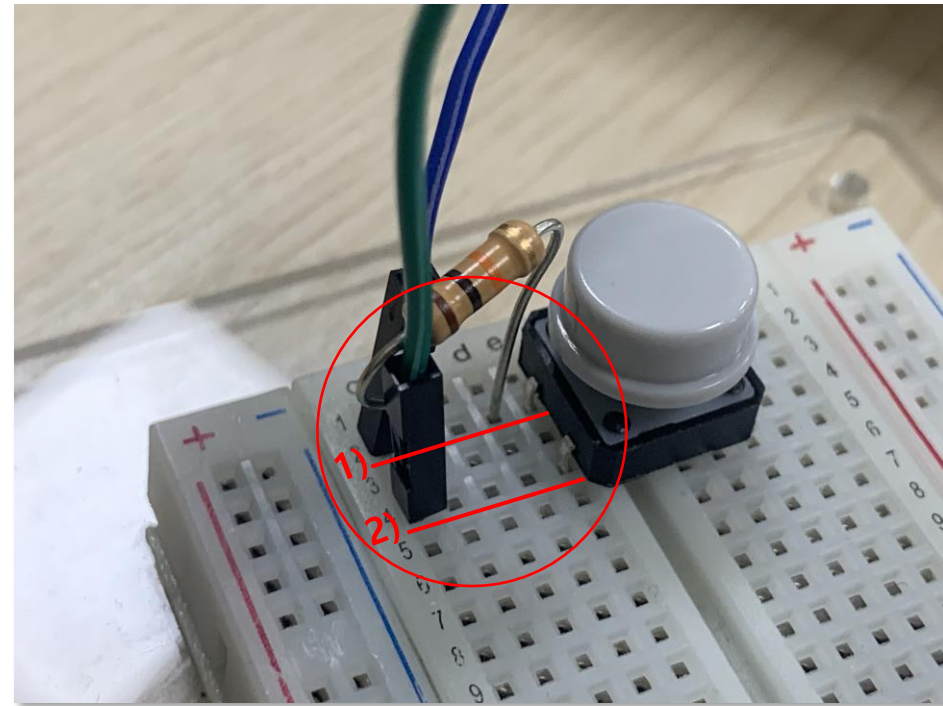
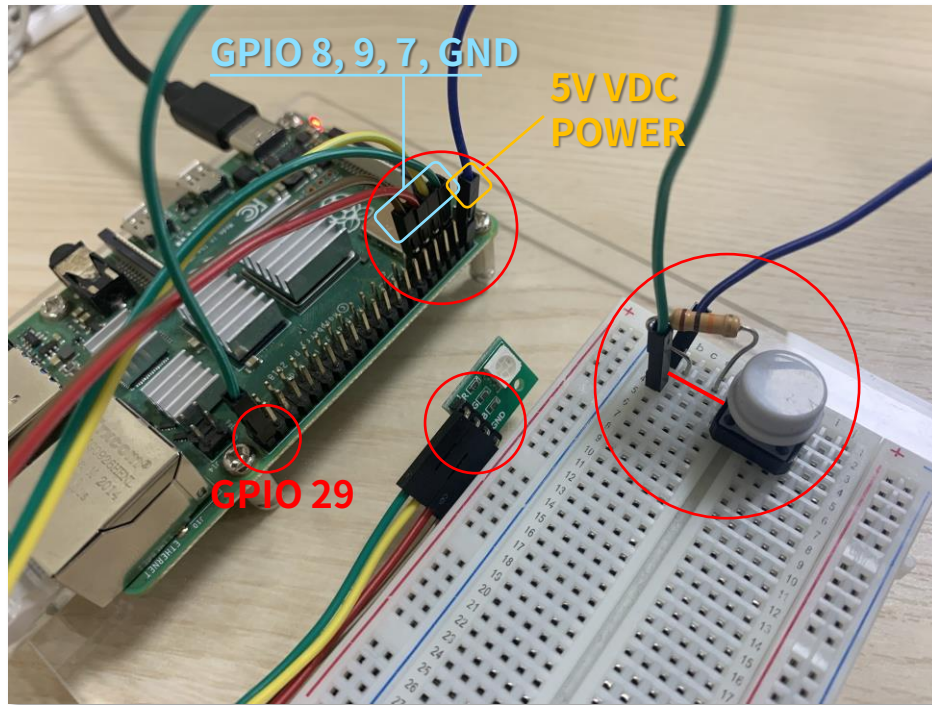
# LED 및 스위치 연결

- 2. 구성품 연결
  - [막대저항(10K $\Omega$ )]을 브레드보드에 결합
  - 각 핀마다 고유의 기능이 정해져 있음
    - ✓ 5V 입력을 통해 버튼 연결 시, 10K $\Omega$  저항 사용
    - ✓ 과도한 전압 공급으로 인한 라즈베리 파이나 센서 모듈의 고장을 막기 위함
  - [스위치]를 브레드보드에 결합
  - [점프 와이어(F/F)]로 [GPIO Pins(8, 9, 7, GND)]와 [LED 센서 모듈] 연결
  - [점프 와이어(M/F)]로 [GPIO Pins(5.0 VDC, 29)]와 브레드보드 연결



# LED 및 스위치 연결

- 2. 구성품 연결

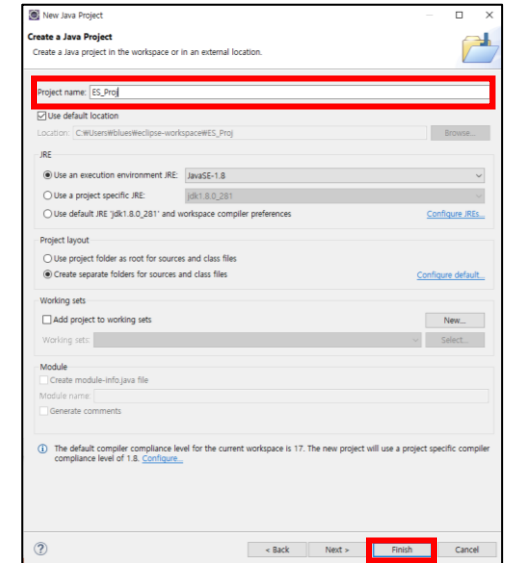
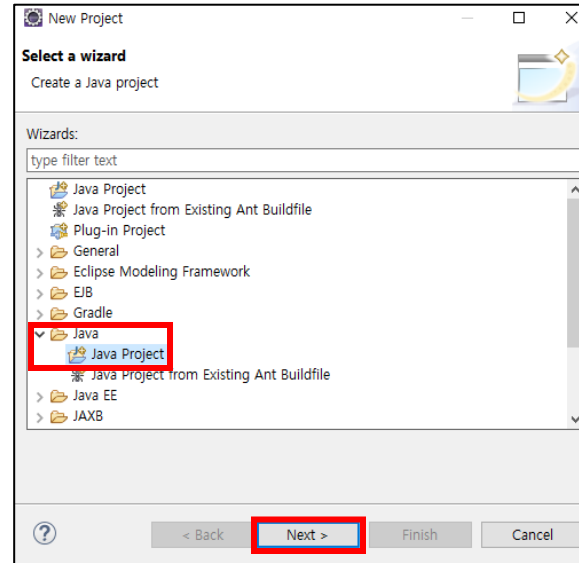
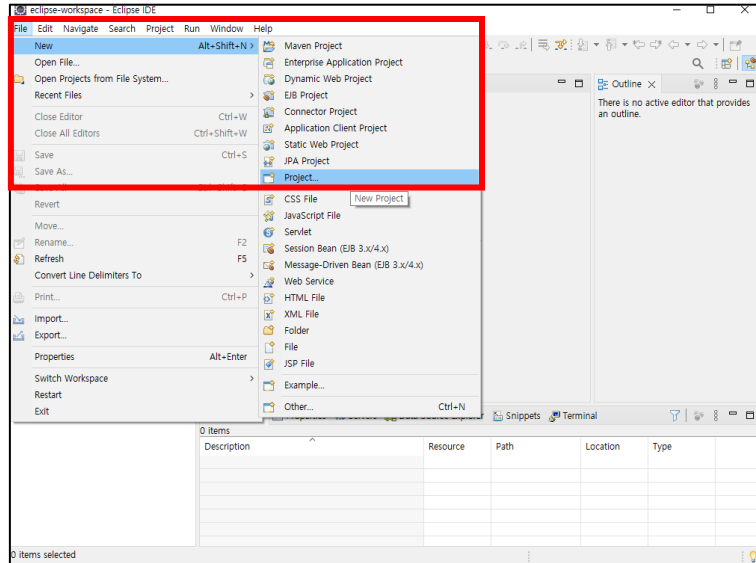


1) 5V VDC 핀 연결선, 10K $\Omega$  막대저항, 버튼 한 쪽 핀이 모두 같은 라인에 연결되어야 함

2) GPIO 핀 29번 연결선과 버튼 한 쪽 핀이 같은 라인에 연결되어야 함

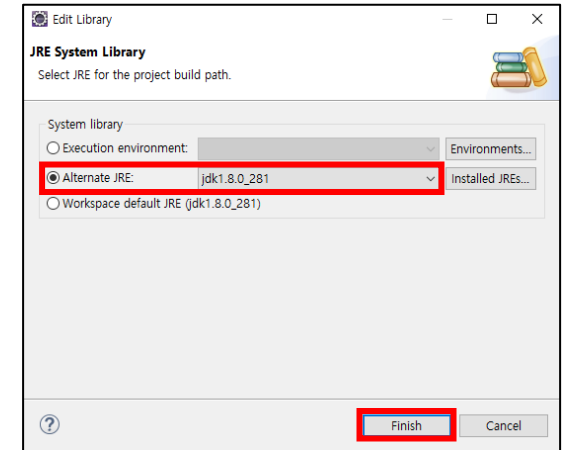
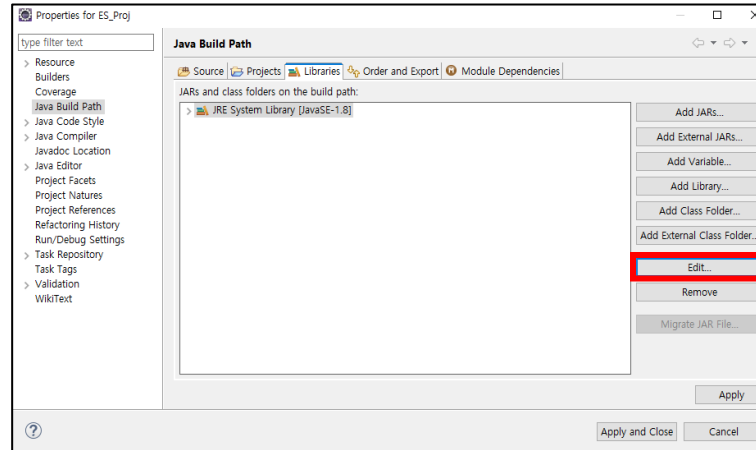
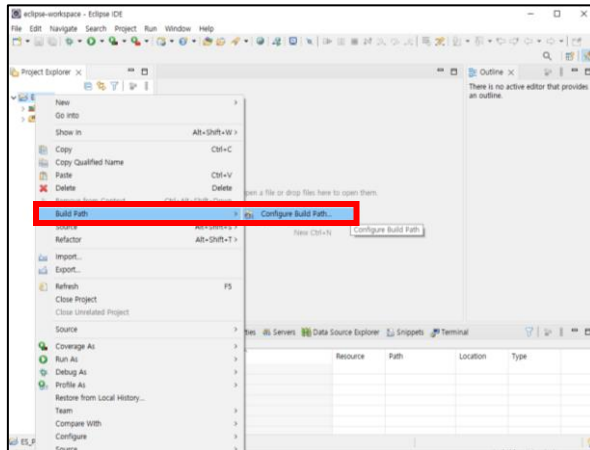
# Eclipse 세팅

- 1. Eclipse 프로젝트 생성
  - [File] → [Project]
  - [Java] → [Java Project]
  - Project Name: 'ES\_Proj' → [Finish]



# Eclipse 세팅

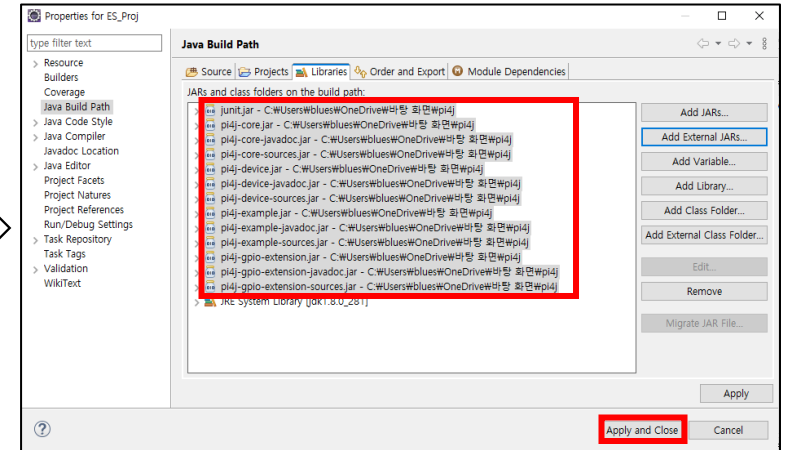
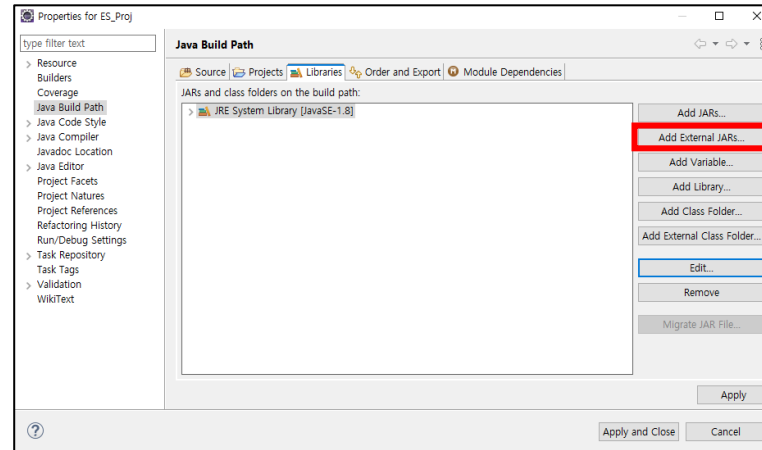
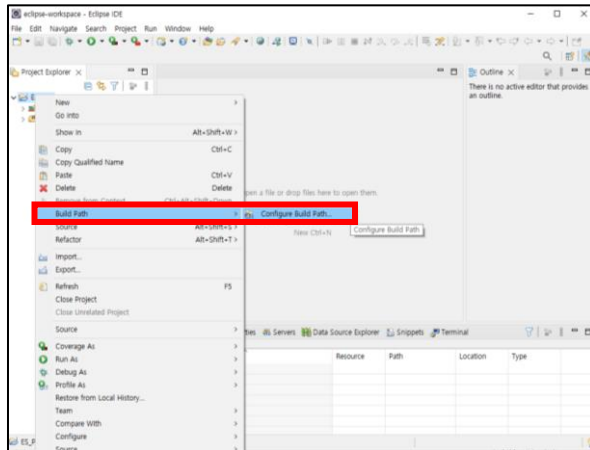
- 2. Eclipse 프로젝트 JDK 설정
  - [프로젝트 우클릭] → [Build Path] → [Configure Build Path...]
  - [Java Build Path] → [Libraries] → [Edit]
  - [Alternate JRE] → [jdk.1.8.0\_281] → [Finish]





# Eclipse 세팅

- 3. Eclipse 프로젝트 Pi4J 라이브러리 추가
  - [프로젝트 우클릭] → [Build Path] → [Configure Build Path...]
  - [Java Build Path] → [Libraries] → [Add External JARs...]
  - 바탕화면에 압축 해제한 Pi4J 폴더 선택 → 모든 .jar 파일 선택 → [Apply and Close]

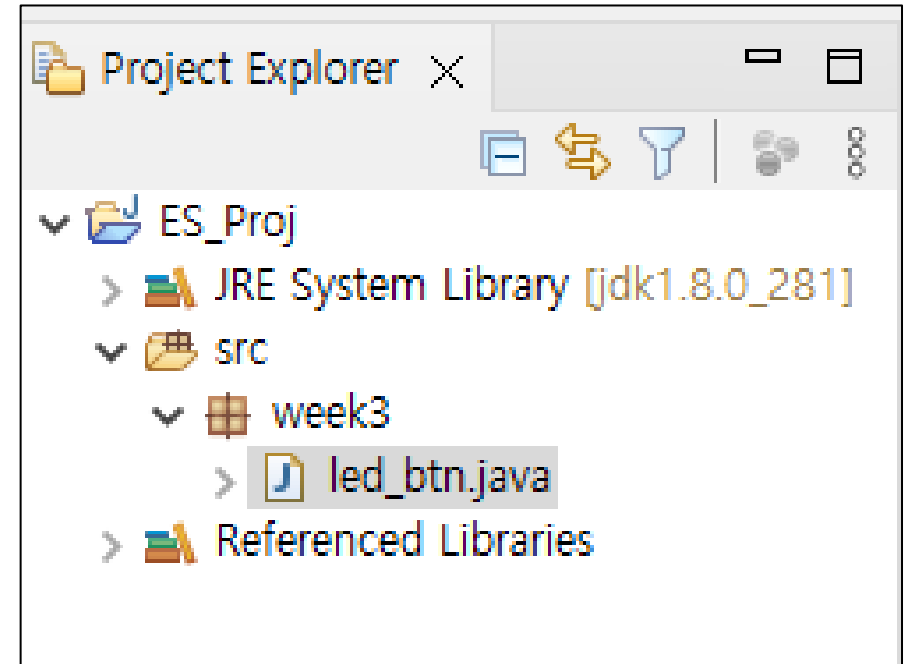




---

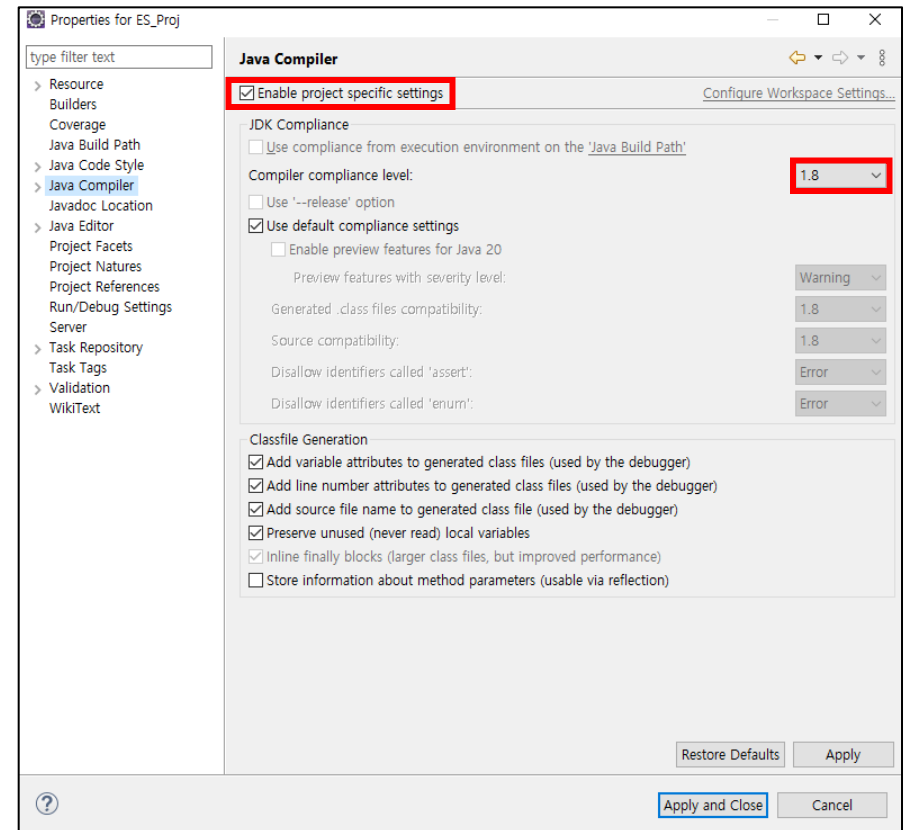
# Eclipse 세팅

- 3. Eclipse 프로젝트 Package/Class 생성
  - [프로젝트 우클릭] → [New] → [Package] → Package Name: '**week3**'
  - [프로젝트 우클릭] → [New] → [Class] → Class Name: '**led\_btn**'



# Eclipse 세팅

- 4. Eclipse 프로젝트 Java Compiler 설정(JNI 오류 발생 문제 해결)
  - [프로젝트 우클릭] → [Properties] → [Java Compiler]
  - [Enable project specific settings] ✓
  - [Compiler compliance level] → [1.8] 선택



# LED 및 스위치 제어

- 1. led\_btn.java 소스 코드
  - 버튼의 state는 4가지가 존재함
    - state가 0일 때 모든 led가 꺼짐(low)
    - state가 1일 때 빨간색 led만 켜짐(high)
    - state가 2일 때 초록색 led만 켜짐
    - state가 3일 때 파란색 led만 켜짐
    - state가 4일 때 모든 led (흰색)가 켜짐

```
package week3;

import com.pi4j.io.gpio.GpioController;
import com.pi4j.io.gpio.GpioFactory;
import com.pi4j.io.gpio.GpioPinDigitalInput;
import com.pi4j.io.gpio.GpioPinDigitalOutput;
import com.pi4j.io.gpio.PinState;
import com.pi4j.io.gpio.RaspiPin;

public class led_btn {
    public static void main(String[] args) {
        GpioController gpio = GpioFactory.getInstance(); // GPIO pin control 객체 선언
        GpioPinDigitalOutput r_led = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_08, PinState.LOW);
        GpioPinDigitalOutput g_led = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_09, PinState.LOW);
        GpioPinDigitalOutput b_led = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_07, PinState.LOW);
        GpioPinDigitalInput btn = gpio.provisionDigitalInputPin(RaspiPin.GPIO_29);

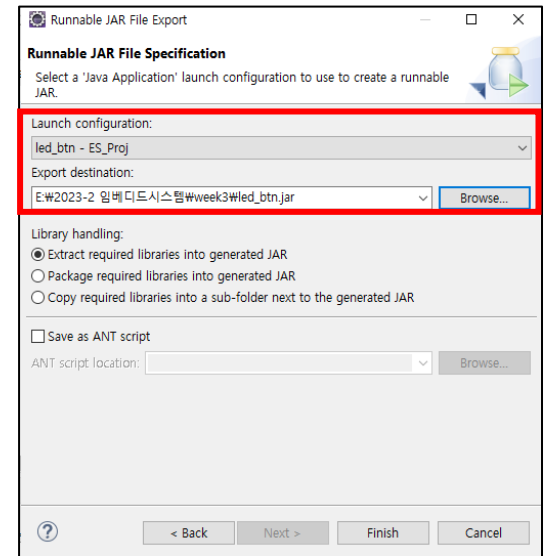
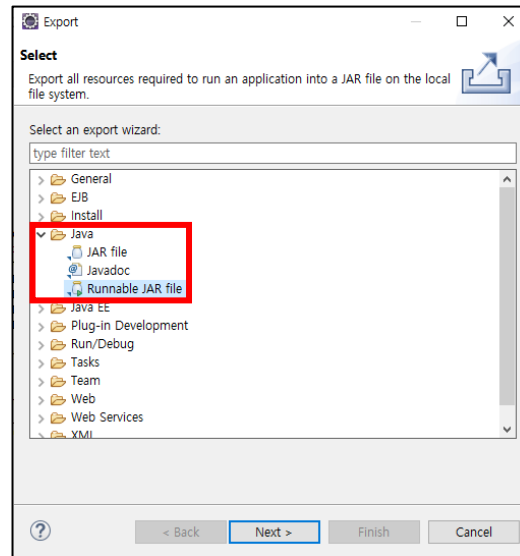
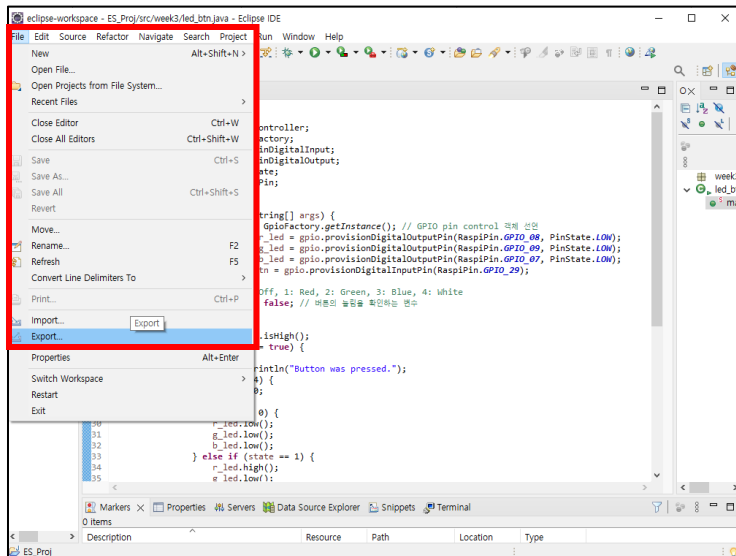
        int state = 0; // 0: Off, 1: Red, 2: Green, 3: Blue, 4: White
        boolean btn_pressed = false; // 버튼의 눌림을 확인하는 변수
    }
}
```

이어서 작성

```
while (true) {
    btn_pressed = btn.isHigh();
    if (btn_pressed == true) {
        state += 1;
        System.out.println("Button was pressed.");
        if (state > 4) {
            state = 0;
        }
    }
    try {
        Thread.sleep(300);
    } catch (Exception e) {
    }
}
```

# LED 및 스위치 제어

- 2. Java Runnable JAR 파일 생성
  - JAR 파일 생성 전 반드시 led\_btn.java를 Run한 상태여야 함 - 중요 ★
  - [File] → [Export] → [Java] → [Runnable JAR file]
  - Launch configuration: 'led\_btn - ES\_Proj'
  - Export destination: 'C:\Users\(\사용자명)\바탕 화면'\led\_btn.jar



- 3. Raspberry Pi로 JAR 및 JDK 파일 전송

- The screenshot shows a terminal window titled 'raspberrypi.local - Xshell 7 (Free for Home/School)'. The terminal displays the following text:

```

Connecting to fe80::1e98:db80:2cdc:1893:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+'.

Linux raspberrypi 6.1.21-v7+ #1642 SMP Mon Apr 3 17:20:52 BST 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 3 09:37:04 2023
pi@raspberrypi:~$
Socket error Events: 32 Error: 10053.
Connection closing...Socket close.

Connection closed by foreign host.

Disconnected from remote host(raspberrypi.local) at 22:08:05.

Type 'help' to learn how to use Xshell prompt.
(C:\>ls

Host 'raspberrypi.local' resolved to fe80::1e98:db80:2cdc:1893.
Connecting to fe80::1e98:db80:2cdc:1893:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+'.

Linux raspberrypi 6.1.21-v7+ #1642 SMP Mon Apr 3 17:20:52 BST 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
19:22:12 2023 from fe80::4441:31a5:4cbb:11e4eth0
pi@raspberrypi:~$
  
```

On the left side of the terminal window, there is a sidebar with the following text:

```

모든 세션
raspberrypi.local
이름      raspberrypi.local
호스트    raspberrypi.local
포트      22
프로토콜  SSH
사용자 이름 pi
암호      
```

At the bottom of the terminal window, there is a status bar showing the following information:

```

ssh/pi@raspberrypi.local:22  SSH2  xterm  1711x39  1.3920  1 세션  9-4  GSP  NUN
  
```

**새 세션 등록 정보** ?

**일반** 옵션

**이름(N):** raspberrypi.local

**호스트(H):** raspberrypi.local

**프로토콜(P):** SFTP **설정(S)...**

**포트 번호(P):** 22

**프록시 서버(S):** <없음> **찾아보기(W)...**

**설명(D):**

**로그인**

☐ 익명 로그인(A)

☒ SSH 키 에이전트(Xagent) 사용(G)

**방법(M):**

☒ Password **설정(T)...**

☐ Public Key

☐ Keyboard Interactive

☐ GSSAPI

☐ PKCS11

☐ CAPI **위로(U)**

**사용자 이름(U):**

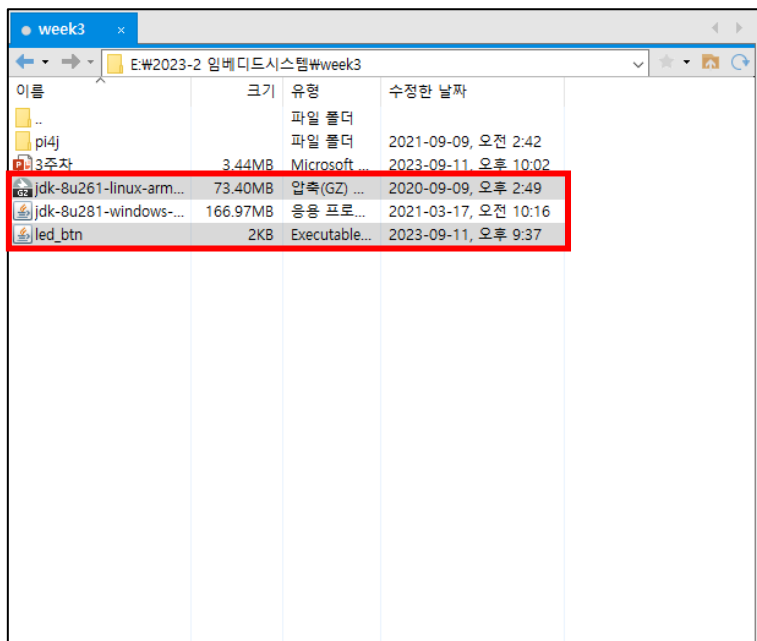
**암호(P):** **아래로(D)**

**연결** **확인** **취소**

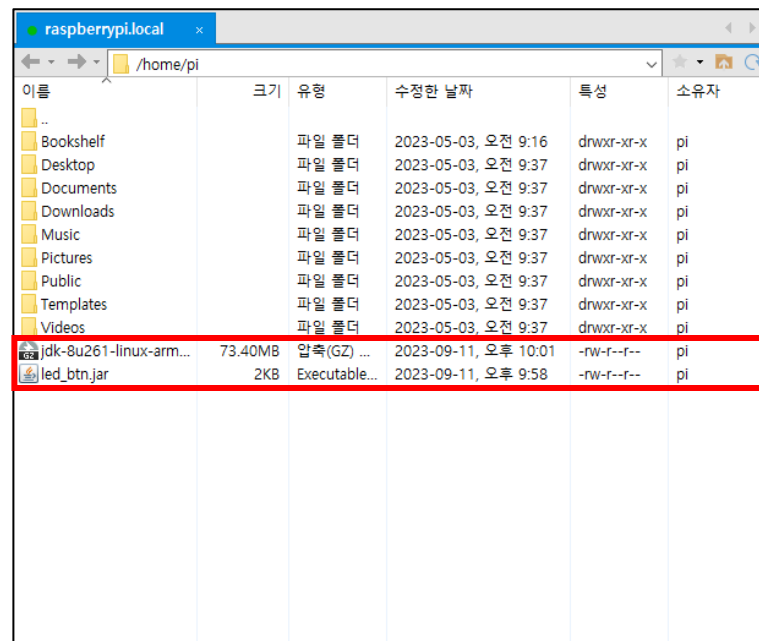
이름	크기	유형	수정한 날짜	특성	소유자
.					
Bookshelf		파일 폴더	2023-05-03, 오전 9:16	drwxr-xr-x	pi
Desktop		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Documents		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Downloads		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Music		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Pictures		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Public		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Templates		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Videos		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi

# LED 및 스위치 제어

- 3. Raspberry Pi로 JAR 및 JDK 파일 전송
  - 바탕화면에 저장된 [led\_btn.jar] 파일 더블클릭
  - 바탕화면에 저장된 [jdk-8u281-linux-arm32-vfp-hflt.tar.gz] 파일 더블 클릭
  - Raspberry Pi(오른쪽 창)로 전송되었는지 확인



이름	크기	유형	수정된 날짜
..		파일 폴더	
pi4j		파일 폴더	2021-09-09, 오전 2:42
3주차	3.44MB	Microsoft ...	2023-09-11, 오후 10:02
jdk-8u261-linux-arm...	73.40MB	압축(GZ) ...	2020-09-09, 오후 2:49
jdk-8u281-windows-...	166.97MB	응용 프로...	2021-03-17, 오전 10:16
led_btn	2KB	Executable...	2023-09-11, 오후 9:37



이름	크기	유형	수정된 날짜	특성	소유자
..					
Bookshelf		파일 폴더	2023-05-03, 오전 9:16	drwxr-xr-x	pi
Desktop		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Documents		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Downloads		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Music		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Pictures		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Public		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Templates		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
Videos		파일 폴더	2023-05-03, 오전 9:37	drwxr-xr-x	pi
jdk-8u261-linux-arm...	73.40MB	압축(GZ) ...	2023-09-11, 오후 10:01	-rw-r--r--	pi
led_btn.jar	2KB	Executable...	2023-09-11, 오후 9:58	-rw-r--r--	pi

---

# LED 및 스위치 제어

- 4. Raspberry Pi에 JDK 설치
  - XSHELL 실행 → Raspberry Pi 원격 접속
  - 아래 명령어를 순차적으로 입력하여 JDK 설치
    - ✓ `sudo tar zxvf jdk-8u261-linux-arm32-vfp-hflt.tar.gz -C /opt`
    - ✓ `sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_261/bin/javac 1`
    - ✓ `sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_261/bin/java 1`
    - ✓ `sudo update-alternatives --config java`
    - ✓ `java -version`
  - **[java version “1.8.0\_261”]**으로 출력되면 설치 성공

```
pi@raspberrypi:~ $ java -version
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b13)
Java HotSpot(TM) Client VM (build 25.261-b13, mixed mode)
```

---

---

# LED 및 스위치 제어

- 6. Raspberry Pi에서 WiringPi 업데이트
  - XSHELL 실행 → Raspberry Pi 원격 접속
  - 아래 명령어를 입력
    - ✓ `wget https://project-downloads.drogon.net/wiringpi-latest.deb` (인터넷 연결 안되어 있을 경우 에러 발생)
    - ✓ `sudo dpkg -i wiringpi-latest.deb`
    - ✓ `gpio -v` (GPIO 버전 확인)
    - ✓ `gpio readall` (GPIO 핀 맵 확인)

```
root@raspberrypi:/home/pi/ES_proj# gpio -v
gpio version: 2.52
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

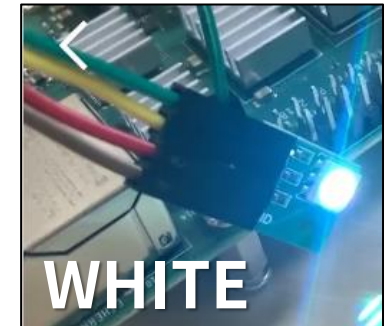
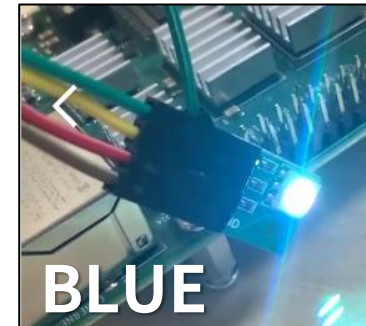
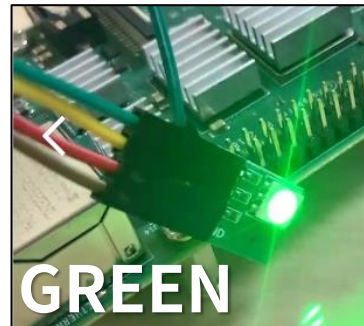
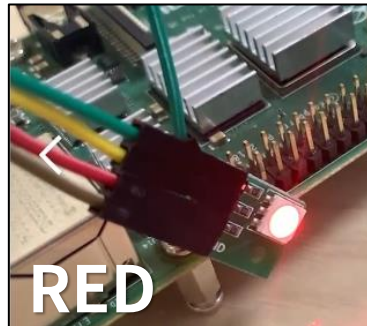
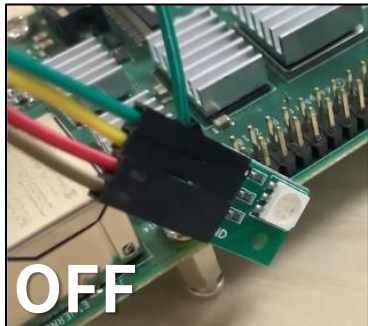
Raspberry Pi Details:
Type: Pi 4B, Revision: 02, Memory: 2048MB, Maker: Sony
* Device tree is enabled.
*--> Raspberry Pi 4 Model B Rev 1.2
* This Raspberry Pi supports user-level GPIO access.
```



# LED 및 스위치 제어

- 7. Raspberry Pi에서 JAR 파일 실행
  - XSHELL 실행 → Raspberry Pi 원격 접속
  - 아래 명령어를 입력
    - ✓ **sudo java -jar led\_btn.jar**
  - 버튼을 누를 때마다 “Button was pressed.”가 출력되면서 LED의 색이 변경됨
  - OFF → Red → Green → Blue → White

```
pi@raspberrypi:~ $ sudo java -jar led_btn.jar
Button was pressed.
Button was pressed.
Button was pressed.
Button was pressed.
```





감사합니다

Thank You

