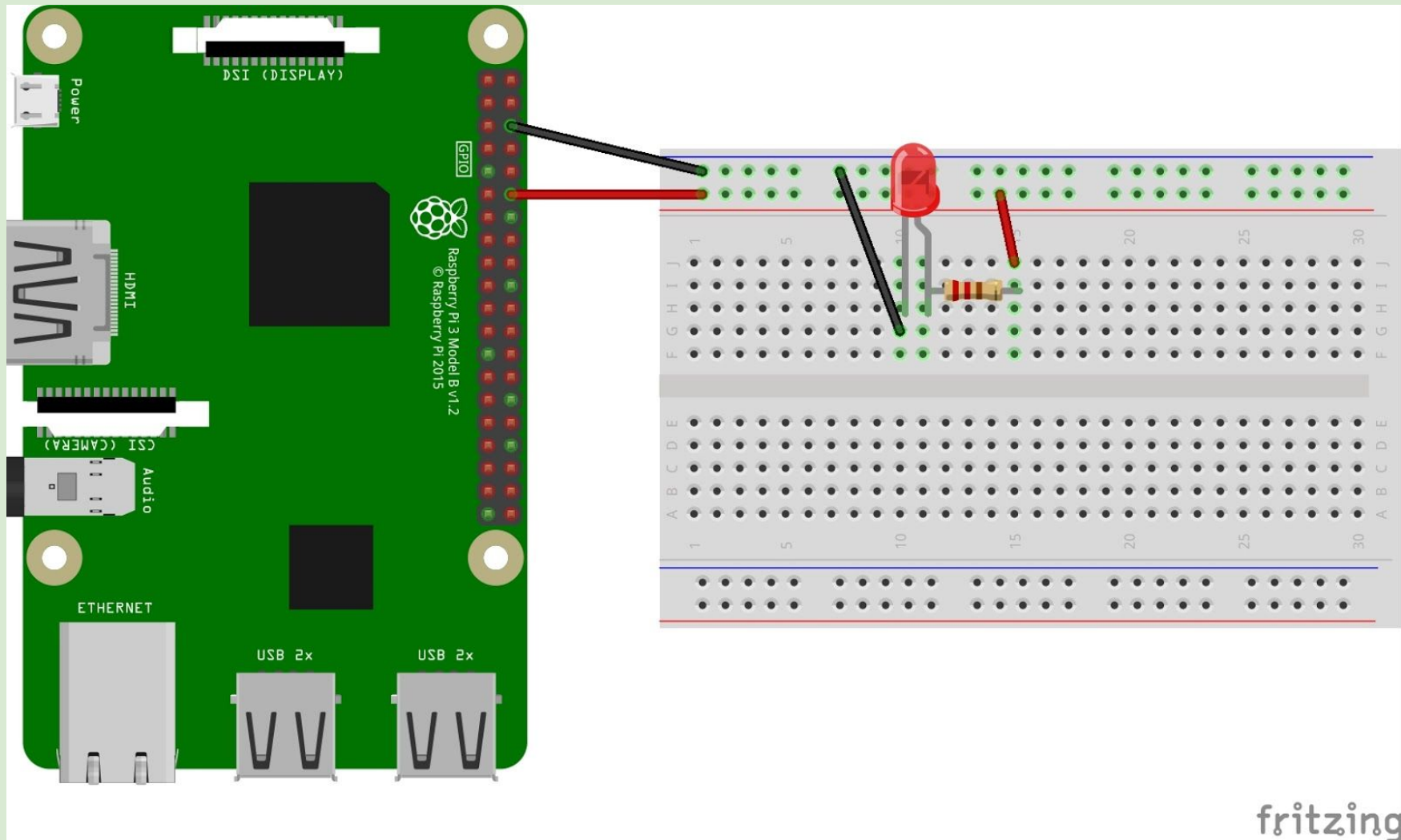


Fritzing Layout :	1
Usage :	2
Java File :	3
Result Image :	9

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

01. Fritzing Layout :



RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

02. Usage :

라즈베리파이에서 직접 Java 실행해서 LED ON/OFF 하기¹ :

0201. download and unzip Pi4J² :

권장방법1.

<http://pi4j.com/download.html> 에서 Download :: Direct Download :: LATEST SNAPSHOT BUILDS :: pi4j-1.2-SNAPSHOT.zip 파일을 다운로드 받고, 압축을 해제하고 /home/pi/pi4j-1.2 폴더를 생성합니다.

오래된방법1(권장방법1을 따라 해주세요.)

pi@raspberrypi:~ \$ wget <http://get.pi4j.com/download/pi4j-1.0.zip>
VNC를 이용하여 라즈베리파이 탐색기에서 pi4j-1.0.zip 파일을 압축해제함.
/home/pi/pi4j-1.0 폴더 생성.

이 방법으로 Java 파일을 실행하게 되면 RaspberryPi3B, RaspberryPi3B+ 모델에서 Firmware 버전과 Pi4J-1.0 버전이 문제를 일으켜 오류가 발생합니다.
RaspberryPi3B모델의 경우는 주석5의 해결책 대로 '\$ sudo rpi-update 52241088c1da59a359110d39c1875cda56496764' 를 실행하여 펌웨어 업데이트를 하면 해결이 되었지만, RaspberryPi3B+ 모델의 경우 해결이 되지 않아, 3B, 3B+ 모델 모두 잘 실행되는 권장방법1을 따라 실습해주세요.

¹ <http://bluexmas.tistory.com/454> [출처] RaspberryPi - GPIO로 LED 켜고 끄기 OS/Raspberry Pi 2015.08.13 23:55 Posted by 파란크리스마스

² <https://github.com/Pi4J/pi4j> 실습에서 사용한 1.0버전 Repository : <https://github.com/Pi4J/pi4j/tree/release/1.0>

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

0202. upload, compile and run Java File :

목차 Java File 의 Ch01012_JavaRunOnRaspberryPi3_LedOnOff.java를 생성해서
Filezilla를 이용하여
/home/pi/NetBeansProjects/Ch01012_JavaRunOnRaspberryPi3_LedOnOff 폴더에
업로드함.

↓

compile Java File :

```
pi@raspberrypi:~/NetBeansProjects/Ch01012_JavaRunOnRaspberryPi3_LedOnOff $ javac -cp3 ../../pi4j-1.2/lib/pi4j-core.jar  
Ch01012_JavaRunOnRaspberryPi3_LedOnOff.java
```

↓

run Java File :

```
pi@raspberrypi:~/NetBeansProjects/Ch01012_JavaRunOnRaspberryPi3_LedOnOff $ sudo java -cp ../../pi4j-1.2/lib/pi4j-core.jar  
Ch01012_JavaRunOnRaspberryPi3_LedOnOff
```

03. Java File :

Lec_JavaRasPi3_0140/Ch01012_JavaRunOnRaspberryPi3_LedOnOff/Ch01012_JavaRunOnRaspberryPi3_LedOnOff.java

³ <https://blog.naver.com/haskim0716n/221245472615> [출처] [JAVA] Javac 옵션 | [작성자] 02160415

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

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

/*
...NetBeans :: wiringPiSetup: Must be root (Did you forget sudo ?)4
*/

/*
...NetBeans :: WiringPi에서 "Unable to determine hardware version. I see : Hardware :
BCM2835 '오류'5 (이하 번역문임) :
증상 :
```

⁴ <https://raspberrypi.stackexchange.com/questions/54208/wiringpisetup-must-be-root-did-you-forget-sudo> wiringPiSetup: Must be root (Did you forget sudo ?)

⁵ <https://qiita.com/jollyjoester/items/ba59e5d43e28b701f120> WiringPiで「Unable to determine hardware version. I see: Hardware : BCM2835」エラー

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

```
$ sudo python3 led.py
Unable to determine hardware version. I see : Hardware : BCM2835
```

```
,
- expecting BCM2708 or BCM2709.
If this is a genuine Raspberry Pi then please report this
to projects@drogon.net. If this is not a Raspberry Pi then you
are on your own as wiringPi is designed to support the
Raspberry Pi ONLY.
```

해결 방법 :

제목의 오류를 그 그는 후에 "Kernel를 upgrade하기 전에 움직이고 있었어!"라고 기입이 어딘가에 있었으므로 4.9 계의 이전 4.4 계까지 다운 그레이드 보니 움직였다.

당초 Kernel 버전은 4.9.24

```
$ uname6 -a
Linux rasp-jolly 4.9.24-v7 + # 993 SMP Wed Apr 26 18:01:23 BST 2017 armv7l GNU / Linux
4.4 시스템의 최신 같아 녀석 의 해시를 사용하여 다운 그레이드
```

```
$ sudo rpi-update 52241088c1da59a359110d39c1875cda56496764
```

⁶ <https://terms.naver.com/entry.nhn?docId=4125871&cid=59321&categoryId=59321> 유닉스 리눅스 명령어 사전 uname

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

다시 시작

```
$ sudo reboot
```

재부팅 후 Kernel 버전

```
$ uname -a
```

```
Linux rasp-jolly 4.4.50-v7+ # 970 SMP Mon Feb 20 19:18:29 GMT 2017 armv7l GNU / Linux
```

움직였다! (L 치카에 1 시간 이상 걸렸다 orz)

*/

```
public class Ch01012_JavaRunOnRaspberryPi3_LedOnOff {  
  
    public static void main(String[] args) throws InterruptedException {  
  
        System.out.println("<--Pi4J--> GPIO Control Example ... started.");  
  
        // create gpio controller  
        final GpioController gpio = GpioFactory.getInstance();  
  
        // provision gpio pin #01 as an output pin and turn on  
        final GpioPinDigitalOutput pin = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01,  
"MyLED", PinState.HIGH);
```

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

```
// set shutdown state for this pin7 :  
//  configure the pin shutdown behavior; these settings will be  
//  automatically applied to the pin when the application is terminated.  
pin.setShutdownOptions(true, PinState.LOW);  
  
System.out.println("--> GPIO state should be: ON");  
  
Thread.sleep(5000);  
  
// turn off gpio pin #01  
pin.low();  
System.out.println("--> GPIO state should be: OFF");  
  
Thread.sleep(5000);  
  
// toggle the current state of gpio pin #01 (should turn on)  
pin.toggle();  
System.out.println("--> GPIO state should be: ON");
```

⁷ <https://www.programcreek.com/java-api-examples/?class=com.pi4j.io.gpio.GpioPinDigitalOutput&method=setShutdownOptions> Java Code Examples for
`com.pi4j.io.gpio.GpioPinDigitalOutput.setShutdownOptions()`

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 :: Ch01012_JavaRunOnRaspberryPi3_LedOnOff

```
Thread.sleep(5000);

// toggle the current state of gpio pin #01 (should turn off)
pin.toggle();
System.out.println("--> GPIO state should be: OFF");

Thread.sleep(5000);

// turn on gpio pin #01 for 1 second and then off
System.out.println("--> GPIO state should be: ON for only 1 second");
pin.pulse(1000, true); // set second argument to 'true' use a blocking call

// stop all GPIO activity/threads by shutting down the GPIO controller
// (this method will forcefully shutdown all GPIO monitoring threads and scheduled
tasks)
gpio.shutdown();

System.out.println("Exiting ControlGpioExample");
}
}
```

RaspberryPi3 :: Java :: Lec_JavaRasPi3_0140 ::
Ch01012_JavaRunOnRaspberryPi3_LedOnOff

04. Result Image :

