

교육 소개 및 신청은 아래 URL을 참조해주세요 :

<https://blog.naver.com/roboholic84/221375246133> 또는

https://www.devicelab.kr/pro/edu_l.asp?idx=376&pg=1&bgbn=R

올 겨울 자바로 크리스마스 트리 만들고 싶은 자바 웹개발자에게
라즈베리파이가 준비한 사물인터넷 종합선물세트



라즈베리파이로 실습하는 **Pi4J** 사물인터넷

10월 22/23/24/25/26일
18시~21시
5일 총 15시간

신청대상 학생, 직장인, 예비/기창업자 관심있는사람 모두	신청기간 10월 10일 ~ 10월 19일 (선정자 10월 20일 문자발송)	신청방법 www.devicelab.kr (프로그램신청 - 교육세미나)	신청문의 메카솔루션 교육 담당자 : 070-7705-5557
--	--	---	--

주최 |  과학기술정보통신부

주관 |  서울대학교
모바일융합센터

후원 |  MechaSolution
collaborative innovation

Raspberry Pi3B :: 수업 둘째날 미리 보기!

Share_Lec_JavaRasPi3_0140_Ch01010_HiRaspberry :	2
Fritzing 소개 :	2
Share_Lec_JavaRasPi3_0140_Ch01011_ManualLedOnOffOnRaspberryPi3 :	3
Share_Lec_JavaRasPi3_0140_Ch01012_JavaRunOnRaspberryPi3_LedOnOff :	3
Share_Lec_JavaRasPi3_0140_Ch01013_JavaRunOnRaspberryPi3_OneRelay :	13
Share_Lec_JavaRasPi3_0140_Ch02010_LedOnOff :	14

Raspberry Pi3B :: 수업 둘째날 미리 보기!

1. Share_Lec_JavaRasPi3_0140_Ch01010_HiRaspberry :

1.1. NetBeans IDE를 이용하여 첫번째 Java¹ 프로젝트를 생성하고 실행해 봅니다.

2. Fritzing² 소개 :

2.1. 실습 자료에 있는 Fritzing Layout의 이해를 위해 Fritzing을 간략하게 봅니다.

2.2. 라즈베리파이에서 설치해서 사용하는 방법이 있다고 하지만 오류가 발생한다고 하며, 집에서 VNC, Putty, NetBeans를 원격으로 라즈베리파이에 연결하고 문서 작성시에 Fritzing을 함께 이용하는 것을 전제로 간단히 소개합니다 .

¹ <https://books.trinket.io/thinkjava/> Think Java by Allen Downey and Chris Mayfield

² <http://fritzing.org/home/>

<http://fritzing.org/learning/get-started>

<https://youtu.be/9E779EfPLI4> Learning Arduino Episode 001 - Electronic Basics

<https://www.youtube.com/playlist?list=PL0648F7957AB08CDB&feature=plcp> Fritzing Killer Tips

Raspberry Pi3B :: 수업 둘째날 미리 보기!

3. Share_Lec_JavaRasPi3_0140_Ch01011_ManualLedOnOffOnRaspberryPi3 :

3.1. LED, 저항, BreadBoard를 이용하여 라즈베리파이에서 직접 GPIO 실행해서 LED ON/OFF 해봅니다.

4. Share_Lec_JavaRasPi3_0140_Ch01012_JavaRunOnRaspberryPi3_LedOnOff :

4.1. 라즈베리파이에서 직접 Java 파일을 작성, 컴파일, 실행해서 LED ON/OFF 해봅니다.

4.2. Pi4j 공식 홈페이지(<http://pi4j.com/>)에서 Pi4j Installation, Download 부분을 간략하게 봅니다.

4.3. 가장 간단한 형태의 소스 구성을 먼저 보고 Pi4j를 알아 봅니다³.

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

³ [출처] <http://bluexmas.tistory.com/823> LED 켜고 끄기 참조 | [작성자] 파란크리스마스

Raspberry Pi3B :: 수업 둘째날 미리 보기!

```
public class LEDTest {  
  
    private static GpioController gpio = null;  
  
    static GpioPinDigitalOutput led_pin;  
  
    public static void main(String[] args) throws Exception {  
        gpio = GpioFactory.getInstance();  
        led_pin = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_08);  
  
        // LED 켜기  
        led_pin.high();  
        Thread.sleep(2000); // 2초 대기  
  
        // LED 끄기
```

Raspberry Pi3B :: 수업 둘째날 미리 보기!

```
    led_pin.low();  
  
    //  
    gpio.shutdown();  
}  
}
```

출처: <http://bluexmas.tistory.com/823> [파란크리스마스]

4.4. Pi4J 공식 홈페이지(<http://pi4j.com/>)에서 Usage 부분을 소개합니다.

4.4.1. Pi4J의 사용법 구성 :

4.4.1.1. Import Pi4J Packages

4.4.1.2. Create Controller Instance :

```
final GpioController gpio = GpioFactory.getInstance();
```

```
final GpioController gpio = GpioFactory.getInstance();
```

Raspberry Pi3B :: 수업 둘째날 미리 보기!

4.4.1.3. Pin Numbering :

4.4.1.3.1. 라즈베리파이 모델별로 핀 구성이 다를 수 있음을 유의합니다.

4.4.1.3.2. Pin Numbering - Raspberry Pi 3 Model B⁴의 핀 구성을 알아보고
터미널에서 gpio readall 와 pinout을 실행하여 출력한 핀의 위치를
비교해봅니다.

⁴ <http://pi4j.com/pins/model-3b-rev1.html> Pin Numbering - Raspberry Pi 3 Model B

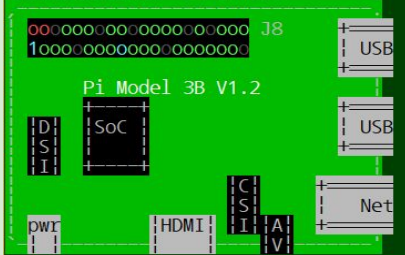
Raspberry Pi3B :: 수업 둘째날 미리 보기!

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ gpio readall
```

Pi 3											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	IN	1	3	4		5v			
3	9	SCL.1	IN	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	0	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

```
pi@raspberrypi:~$
```

```
pi@raspberrypi:~$ pinout
```



```
Revision      : a02082  
SoC           : BCM2837  
RAM           : 1024Mb  
Storage       : MicroSD  
USB ports     : 4 (excluding power)  
Ethernet ports: 1  
Wi-fi         : True  
Bluetooth     : True  
Camera ports (CSI): 1  
Display ports (DSI): 1
```

J8:

Pin	Signal	Pin	Signal
3V3	(1)	2	5V
GPIO2	(3)	4	5V
GPIO3	(5)	6	GND
GPIO4	(7)	8	GPIO14
GND	(9)	10	GPIO15
GPIO17	(11)	12	GPIO18
GPIO27	(13)	14	GND
GPIO22	(15)	16	GPIO23
3V3	(17)	18	GPIO24
GPIO10	(19)	20	GND
GPIO9	(21)	22	GPIO25
GPIO11	(23)	24	GPIO8
GND	(25)	26	GPIO7
GPIO0	(27)	28	GPIO1
GPIO5	(29)	30	GND
GPIO6	(31)	32	GPIO12
GPIO13	(33)	34	GND
GPIO19	(35)	36	GPIO16
GPIO26	(37)	38	GPIO20
GND	(39)	40	GPIO21

For further information, please refer to <https://pinout.xyz>

4.4.1.3.3. BCM 모드의 핀번호 또는 wPi 번호 중 방식을 정해서 핀구성을 합니다.

⁵ <https://projects.raspberrypi.org/en/projects/raspberry-pi-using/9>

Raspberry Pi3B :: 수업 둘째날 미리 보기!

4.4.1.4. Provision Pins :

```
// provision gpio pin #02 as an input pin with its internal pull down resistor enabled
// (configure pin edge to both rising and falling to get notified for HIGH and LOW state
// changes)
GpioPinDigitalInput myButton
    = gpio.provisionDigitalInputPin(RaspiPin.GPIO_02,    // PIN NUMBER
                                    "MyButton",          // PIN FRIENDLY NAME (optional)
                                    PinPullResistance.PULL_DOWN); // PIN RESISTANCE
(optional)

// provision gpio pins #04 as an output pin and make sure is is set to LOW at startup
GpioPinDigitalOutput myLed
    = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_04, // PIN NUMBER
                                    "My LED",          // PIN FRIENDLY NAME (optional)
                                    PinState.LOW);     // PIN STARTUP STATE (optional)
```

Raspberry Pi3B :: 수업 둘째날 미리 보기!

4.4.1.5. Control Pin State :

```
// explicitly set a state on the pin object
myLed.setState(PinState.HIGH);

// use convenience wrapper method to set state on the pin object
myLed.low();
myLed.high();

// use toggle method to apply inverse state on the pin object
myLed.toggle();

// use pulse method to set the pin to the HIGH state for
// an explicit length of time in milliseconds
myLed.pulse(1000);
```

Raspberry Pi3B :: 수업 둘째날 미리 보기!

4.4.1.6. Read Pin State :

```
// get explicit state enumeration for the GPIO pin associated with the button
PinState myButtonState = myButton.getState();

// use convenience wrapper method to interrogate the button state
boolean buttonPressed = myButton.isHigh();
```

4.4.1.7. Listen for Pin Changes :

You must create a class that implements the GpioListener interface. This interface simply contain a single method `pinStateChanged(GpioPinStateChangeEvent event)` that is used to notify the listener class when a pin change event has occurred.

```
// The following snippet demonstrates a simple listener implementation:
public static class GpioUsageExampleListener implements GpioPinListenerDigital {
```

Raspberry Pi3B :: 수업 둘째날 미리 보기!

```
@Override
public void handleGpioPinDigitalStateChangeEvent(GpioPinDigitalStateChangeEvent
event) {
    // display pin state on console
    System.out.println(" --> GPIO PIN STATE CHANGE: " + event.getPin() + " = "
        + event.getState());
}
}
```

To setup your listener, you must register the listener with a pin and instantiate your listener impl.

```
// create and register gpio pin listener
myButton.addListener(new GpioUsageExampleListener());
```

4.4.1.8. Pin Shutdown :

Raspberry Pi3B :: 수업 둘째날 미리 보기!

```
// configure the pin shutdown behavior; these settings will be
// automatically applied to the pin when the application is terminated
// ensure that the LED is turned OFF when the application is shutdown
myLed.setShutdownOptions(true, PinState.LOW, PinPullResistance.OFF);
```

4.4.1.9. Pin Triggers ::

```
// create a gpio synchronization trigger on the input pin
// when the input state changes, also set LED controlling gpio pin to same state
myButton.addTrigger(new GpioSyncStateTrigger(myLed));
```

Raspberry Pi3B :: 수업 둘째날 미리 보기!

5. Share_Lec_JavaRasPi3_0140_Ch01013_JavaRunOnRaspberryPi3_OneRelay :

- 5.1. 릴레이 부품을 확보하고 안전조치 후 문서 업데이트가 필요합니다.
- 5.2. '릴레이'에 대해서 알아보고 아두이노를 이용한 예제 자료를 봅니다⁶.
- 5.3. <https://github.com/OlivierLD/raspberry-pi4j-samples>

OlivierLD/raspberry-pi4j-samples 를 다운받아서 컴파일 및 실행해봅니다.

⁶ <https://blog.naver.com/roboholic84/220636740499> [출처] 모스펫과 릴레이의 차이점이 뭘까? (모스펫VS릴레이)(릴레이 회로/릴레이 구조/모스펫 회로/모스펫 원리)|작성자 오픈랩

<https://blog.naver.com/roboholic84/220460268471> [출처] 릴레이란? (릴레이 구조, 릴레이 원리, 릴레이 종류, 릴레이 작동원리, 릴레이 스위치, 아두이노 릴레이, 무접점 릴레이, 릴레이 접점)|작성자 오픈랩

<https://blog.naver.com/roboholic84/220598427339> [출처] 릴레이 모듈 사용하기 (Relay)(Keyes)(전자 스위치)(LED)[아두이노 강좌]|작성자 오픈랩

<https://blog.naver.com/roboholic84/220634626349> [출처] 시간 지연 함수 delay에 대해 알아보자 [릴레이 사용][빛 감지센서(조도 센서)][LED 제어][아두이노 강좌]|작성자 오픈랩

<https://blog.naver.com/roboholic84/220904712531> [출처] 아두이노 릴레이로 조명제어하기 / 1채널 릴레이 사용하기 / 아두이노강좌|작성자 오픈랩

Raspberry Pi3B :: 수업 둘째날 미리 보기!

6. Share_Lec_JavaRasPi3_0140_Ch02010_LedOnOff :

- 6.1. 넷빈IDE에서 Java 프로젝트를 생성하고 Pi4J 라이브러리를 추가해봅니다.
- 6.2. 노트북에 넷빈IDE를 설치했을 때 원격으로 라즈베리파이에 있는 Java 프로젝트를 실행하는 방법을 알아봅니다.