



라즈베리파이 스위치 제어

로봇SW 교육원

최상훈(shchoi82@gmail.com)

학습 목표

2

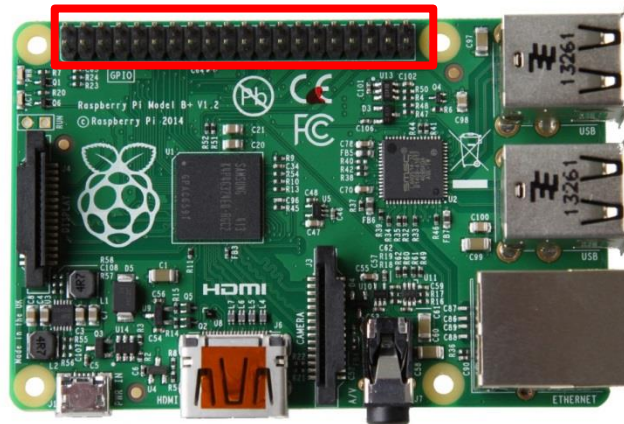
- 스위치 제어

라즈베리 파이 GPIO

3

〈라즈베리 파이 B+의 GPIO〉

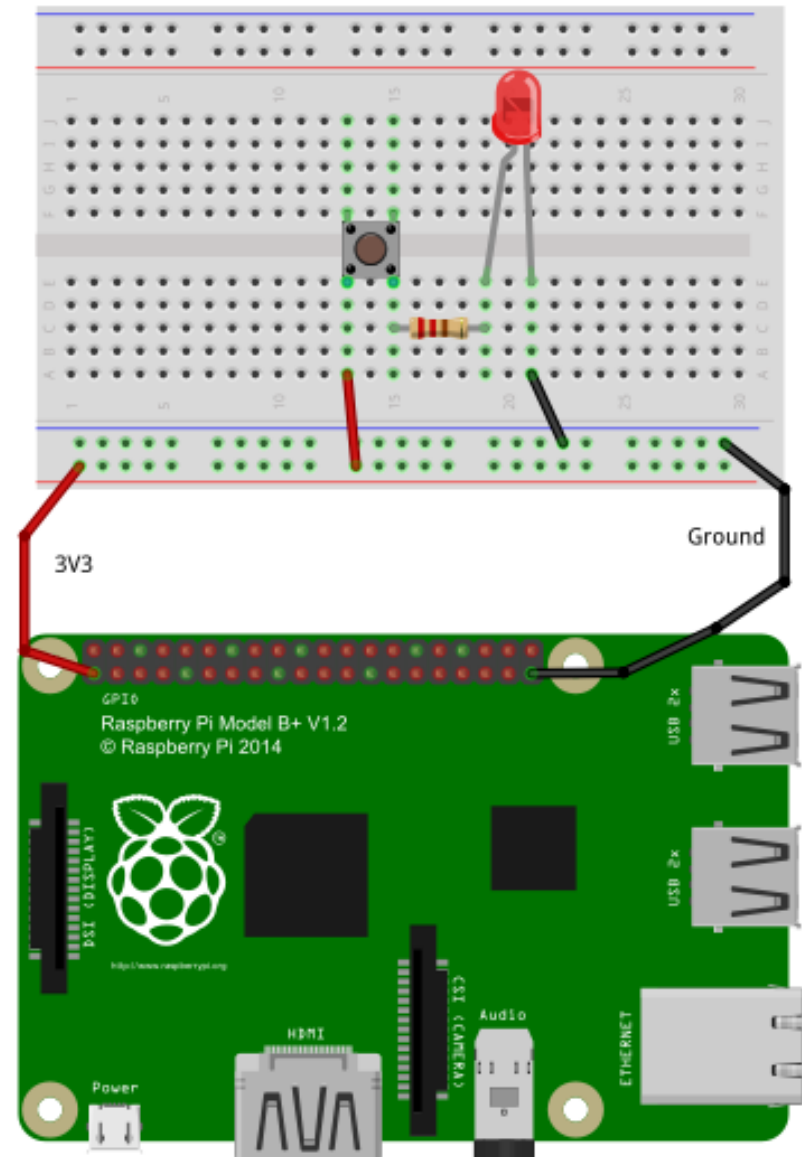
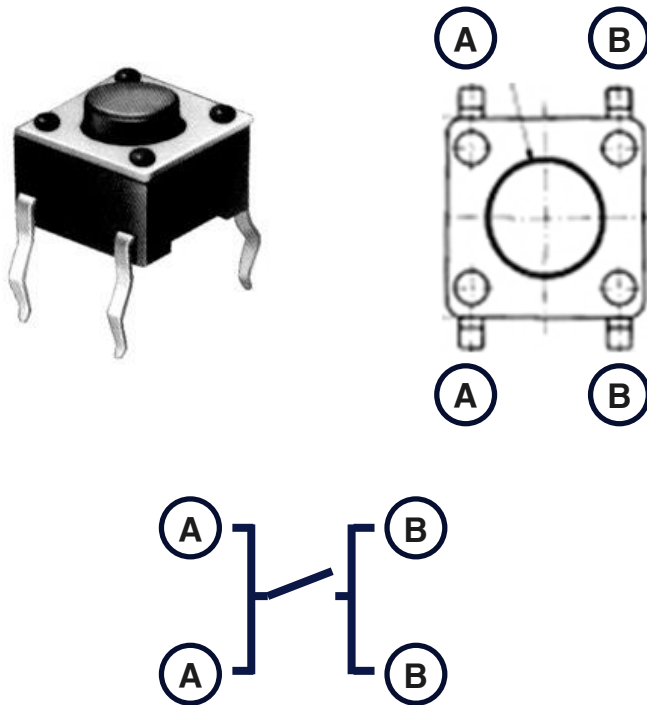
Pi Model B/B+		Pi Model B+	
3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID SD I2C ID EEPROM	27	28	ID SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21



실습1 : Switch

4

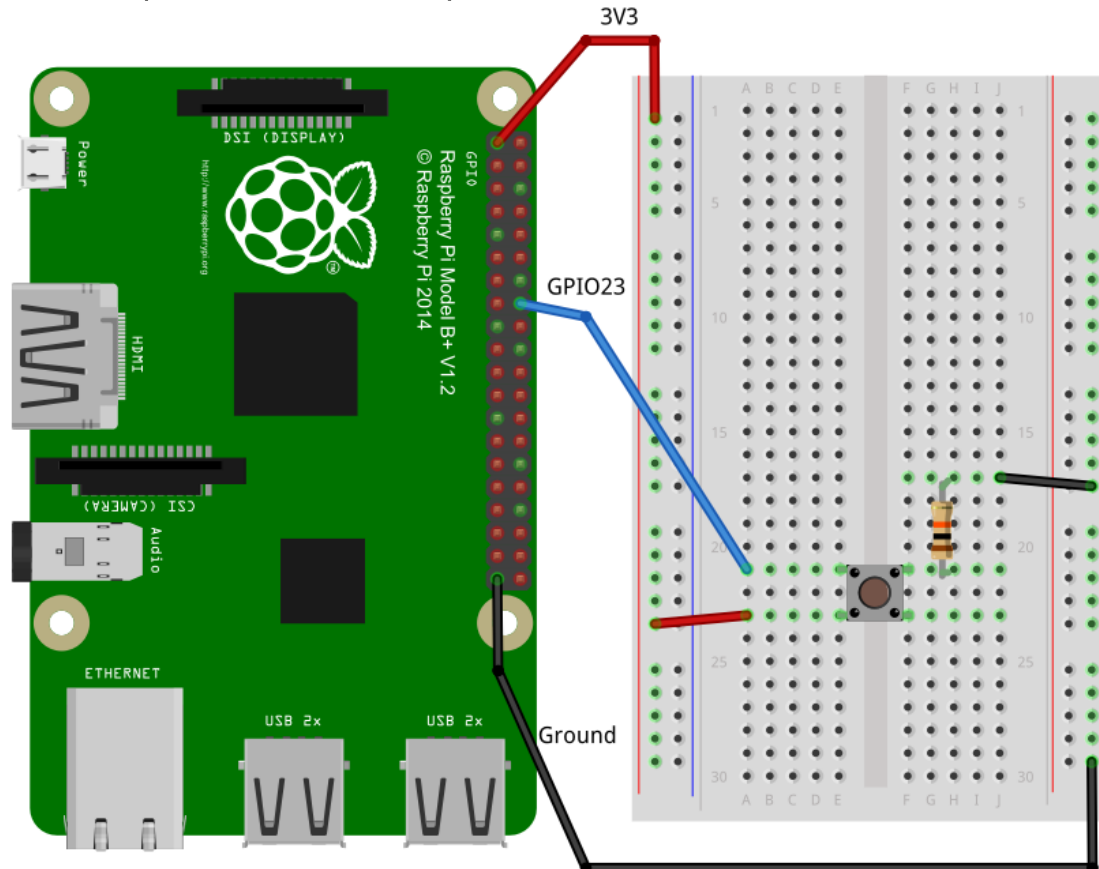
- 스위치로 LED ON/OFF
- 구성
 - 220Ω 저항
 - LED
 - 스위치



실습2-1 : 디지털 입력

5

- 스위치를 이용한 디지털 입력
- 구성
 - 스위치(GPIO 23)
 - 10k Ω 저항 (풀 다운 저항)



실습2-2

6

- 파일명 : gpioSwitchEx1.c

```
#include <stdio.h>
#include <wiringPi.h>    // wiringPi 라이브러리 헤더
#define SW    23         // gpio 23
#define EXIT_SUCC  0
#define EXIT_FAIL  1

int
main(void)
{
    int dr;
    if(wiringPiSetupGpio() == -1)    // wiringPi 초기화
        return EXIT_FAIL;

    setbuf(stdout, NULL);
    pinMode(SW, INPUT);
    pullUpDnControl(SW, PUD_OFF);    // 내부 pull down/up 저항 OFF

    while(1) {
        delay(10);
        if((dr = digitalRead(SW)) == 1){
            printf("%d:--__--\r", dr);
        } else {
            printf("%d:-----\r", dr);
        }
    }
    return EXIT_SUCC;
}
```

실습2-3


7

- 컴파일

```
$ gcc -lwiringPi gpioSwitchEx1.c -o gpioSwitchEx1
```

- 실행

```
$ sudo ./gpioSwitchEx1
```



released

```
pi@robotcode ~ $ sudo ./gpioSwitchEx1  
0:-----
```

- 프로그램 강제 종료

```
[Ctrl + c]
```



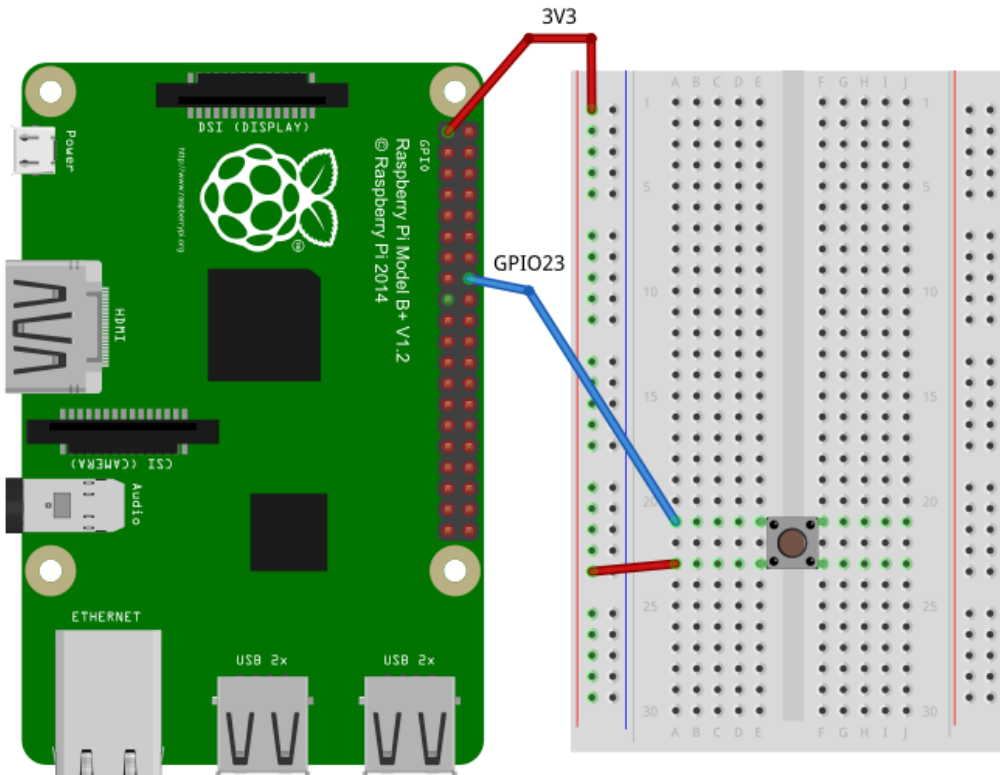
pushed

```
pi@robotcode ~ $ sudo ./gpioSwitchEx1  
1:--____--
```

실습3 : floating 상태

8

- 풀 다운 저항 제거



- released : floating 상태

released

```
pi@robotcode ~ $ sudo ./gpioSwitchEx1
0:----- or 1:--_--
```

??

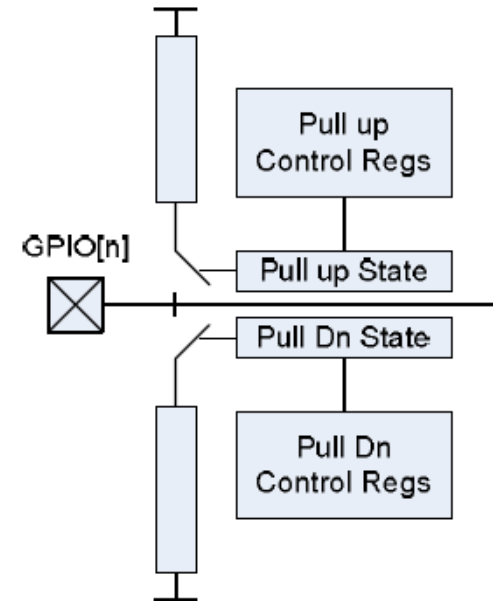
pushed

```
pi@robotcode ~ $ sudo ./gpioSwitchEx1
1:-----
```


내부 풀 업/다운 저항

- GPIO핀 내부 풀 업/다운 저항
 - 기본값 : [BCM2835-ARM datasheet](#) 89, 102page
 - Pull : High or Low
- GPIO핀 내부 풀 업/다운 저항 설정
 - Pull up Control 레지스터
 - Pull down Control 레지스터
 - BCM2835-ARM datasheet 89, 100page
- wiringPi 라이브러리 함수

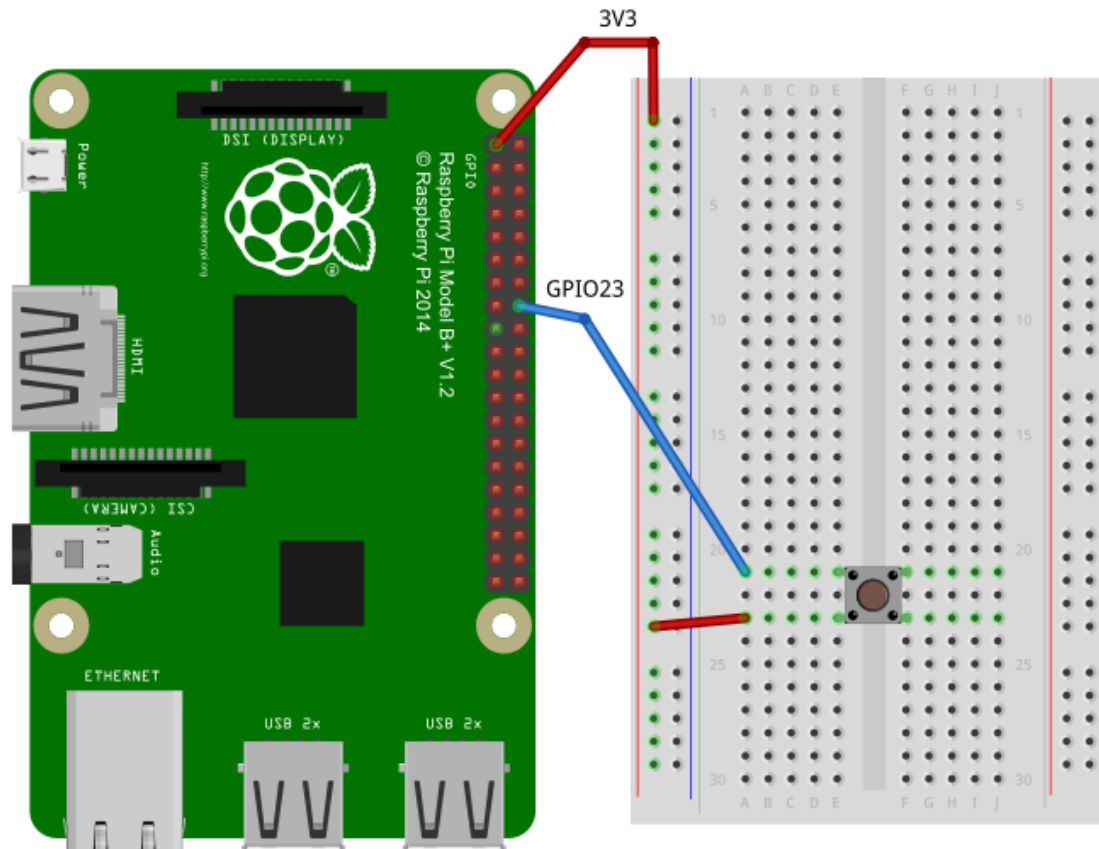

```
void pullUpDnControl(int pin, int pud)
    pin : GPIO PIN #
    pud : PUD_OFF
          PUD_DOWN(pull to ground)
          PUD_UP(pull to 3.3v)
```
- 내부 풀 업/다운 저항 값 : 50kΩ



실습4-1 : 내부 풀 다운 저항

10

- 내부 풀 다운(pull down, pull to ground) 저항 사용
- 구성
 - 스위치(GPIO 23)



실습4-2

11

• 파일명 : gpioSwitchEx2.c

```
#include <stdio.h>
#include <wiringPi.h>    // wiringPi 라이브러리 헤더
#define SW    23        // gpio 23
#define EXIT_SUCC    0
#define EXIT_FAIL    1

int
main(void)
{
    int dr;
    if(wiringPiSetupGpio() == -1)    // wiringPi 초기화
        return EXIT_FAIL;

    setbuf(stdout, NULL);
    pinMode(SW, INPUT);
    pullUpDnControl(SW, PUD_DOWN);    // 내부 pull down 저항 ON

    while(1) {
        delay(10);
        if((dr = digitalRead(SW)) == 1){
            printf("%d:--__--\r", dr);
        } else {
            printf("%d:-----\r", dr);
        }
    }

    return EXIT_SUCC;
}
```

실습4-3

12

- 컴파일

```
$ gcc -lwiringPi gpioSwitchEx2.c -o gpioSwitchEx2
```

- 실행

```
$ sudo ./gpioSwitchEx2
```



released

```
pi@robotcode ~ $ sudo ./gpioSwitchEx2  
0:-----
```

- 프로그램 강제 종료

```
[Ctrl + c]
```



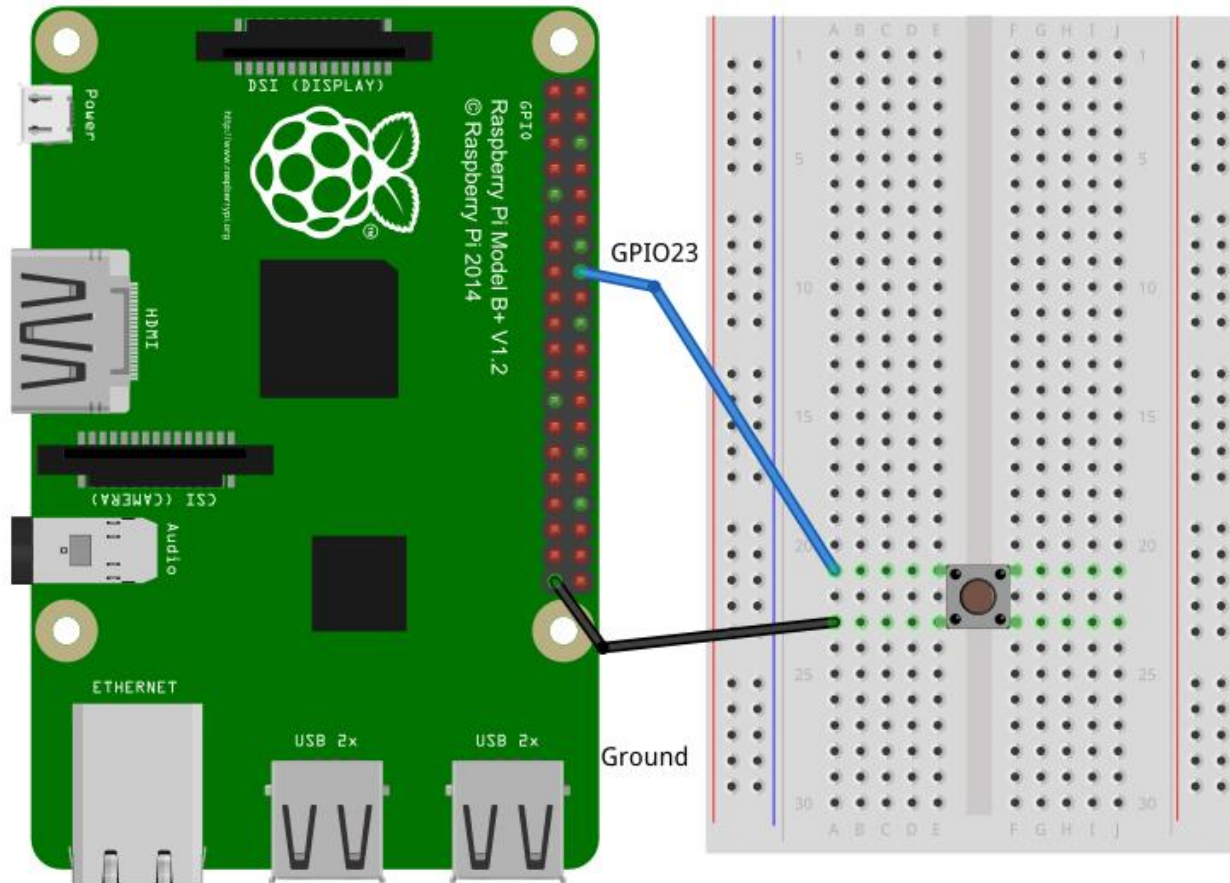
pushed

```
pi@robotcode ~ $ sudo ./gpioSwitchEx2  
1:-- ____--
```

실습5-1 : 내부 풀 업 저항

13

- 내부 풀 업(pull up) 저항 사용
- 구성
 - 스위치(GPIO 23)



실습5-2

14

• 파일명 : gpioSwitchEx3.c

```
#include <stdio.h>
#include <wiringPi.h>    // wiringPi 라이브러리 헤더
#define SW    23    // gpio 23
#define EXIT_SUCC    0
#define EXIT_FAIL    1

int
main(void)
{
    int dr;
    if(wiringPiSetupGpio() == -1)    // wiringPi 초기화
        return EXIT_FAIL;

    setbuf(stdout, NULL);
    pinMode(SW, INPUT);
    pullUpDnControl(SW, PUD_UP);    // 내부 pull UP 저항 ON

    while(1) {
        delay(10);
        if((dr = digitalRead(SW)) == 1){
            printf("%d:-----\r", dr);
        } else {
            printf("%d:--____--\r", dr);
        }
    }

    return EXIT_SUCC;
}
```

실습5-3

15

- 컴파일

```
$ gcc -lwiringPi gpioSwitchEx3.c -o gpioSwitchEx3
```

- 실행

```
$ sudo ./gpioSwitchEx3
```

released

```
pi@robotcode ~ $ sudo ./gpioSwitchEx3  
1:-----
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

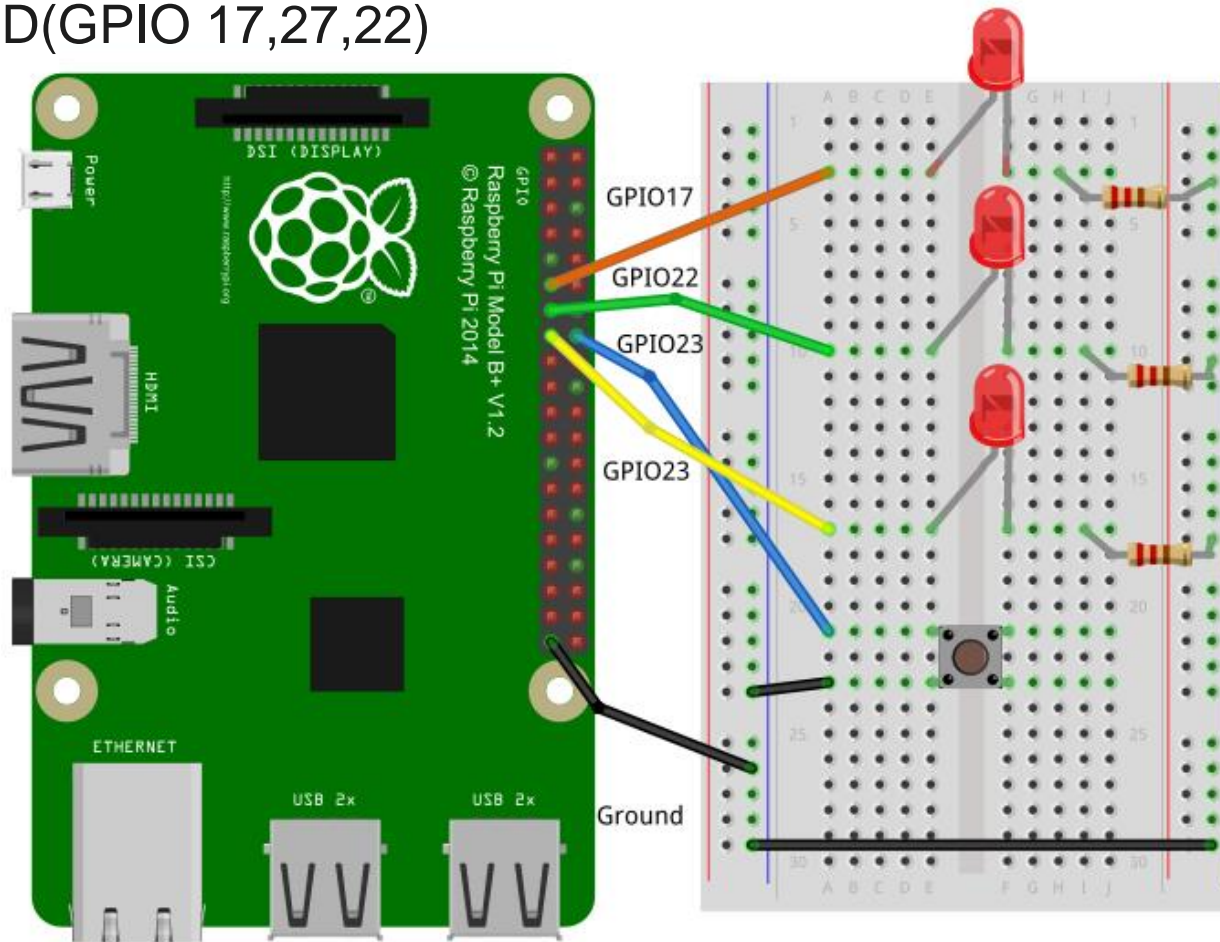
pushed

```
pi@robotcode ~ $ sudo ./gpioSwitchEx3  
0:-- ____--
```

실습6-1 : 스위치로 프로세스 실행

16

- 스위치로 프로세스 실행
- 구성
 - 스위치(GPIO 23)
 - LED(GPIO 17,27,22)



실습6-2

17

- 파일명 : gpioLed.c

```
#include <stdio.h>
#include <wiringPi.h>    // wiringPi 라이브러리 헤더
#define LED1 17
#define LED2 27
#define LED3 22
#define EXIT_SUCC  0
#define EXIT_FAIL  1
int
main(void)
{
    int i;
    if(wiringPiSetupGpio() == -1)    // wiringPi 초기화
        return EXIT_FAIL;
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    for(i = 0 ; i < 10 ; i++){
        printf("LED ON\n");
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, HIGH);
        digitalWrite(LED3, HIGH);    delay(50);
        printf("LED OFF\n");
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, LOW);    delay(50);
    }
    return EXIT_SUCC;
}
```

실습6-3

18

• 파일명 : gpioSwitchEx4.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>    // wiringPi 라이브러리 헤더
#define SW    23    // gpio 23
#define EXIT_SUCC  0
#define EXIT_FAIL  1

int
main(void)
{
    int dr;
    if(wiringPiSetupGpio() == -1)    // wiringPi 초기화
        return EXIT_FAIL;
    setbuf(stdout, NULL);
    pinMode(SW, INPUT);
    pullUpDnControl(SW, PUD_UP);

    while(1) {
        delay(10);
        if((dr = digitalRead(SW)) == 1){
            printf("%d:-----\r", dr);
        } else {
            printf("%d:--__--\n", dr);
            printf("create a new process\n");
            system("./gpioLed");
        }
    }

    return EXIT_SUCC;
}
```

실습6-4

19

- 컴파일

```
$ gcc -lwiringPi gpioLed.c -o gpioLed
```

```
$ gcc -lwiringPi gpioSwitchEx4.c -o gpioSwitchEx4
```

- 실행

```
$ sudo ./gpioSwitchEx4
```

released

```
pi@robotcode ~ $ sudo ./gpioSwitchEx4
1:-----
```

pushed

- 프로그램 강제 종료

```
[Ctrl + c]
```

```
pi@robotcode ~ $ sudo ./gpioSwitchEx4
0:--__--
create a new process
LED ON
LED OFF
...
```

미션1

20

- 실습 6-4 프로그램은 무한 반복 실행된다. 스위치를 하나 더 연결하여 스위치를 눌렀을 때 프로세스를 종료하도록 프로그램을 수정하시오.

미션2

21

- 2개의 LED를 GPIO 5,6에 각각 연결하시오.
- 스위치 버튼을 눌렀을 때 GPIO 5, 6에 연결된 LED가 모두 켜지도록 프로그램을 작성하시오.
- 두 개의 LED가 서로 번갈아 켜지도록 프로그램을 수정하시오.
 - 눌렀을 때 GPIO 5 의 LED만 ON
 - 뺐을 때 GPIO 6 의 LED만 ON

미션3

22

- 2개의 LED를 GPIO 5,6에 각각 연결하시오.
- 2개의 스위치를 연결하시오.
- 스위치 1을 눌렀을 때 GPIO 5 에 연결된 LED ON
- 스위치 2을 눌렀을 때 GPIO 6 에 연결된 LED ON
- 스위치 1,2 모두 눌렀을 때 프로그램 종료

미션4

23

- 실습 준비 : 스위치 2개
- 스위치 1
 - 스위치 1 버튼을 눌렀을 때 현재 작업 디렉토리(CWD)에 있는 모든 파일의 이름을 화면에 출력하고 filelist.out 파일에 저장하시오.
- 스위치 2
 - 스위치 2 버튼을 눌렀을 때 filelist.out 파일을 삭제하시오.