UNIT

12

라즈베리파이 PWM

로봇SW 교육원

최상훈(shchoi82@gmail.com)

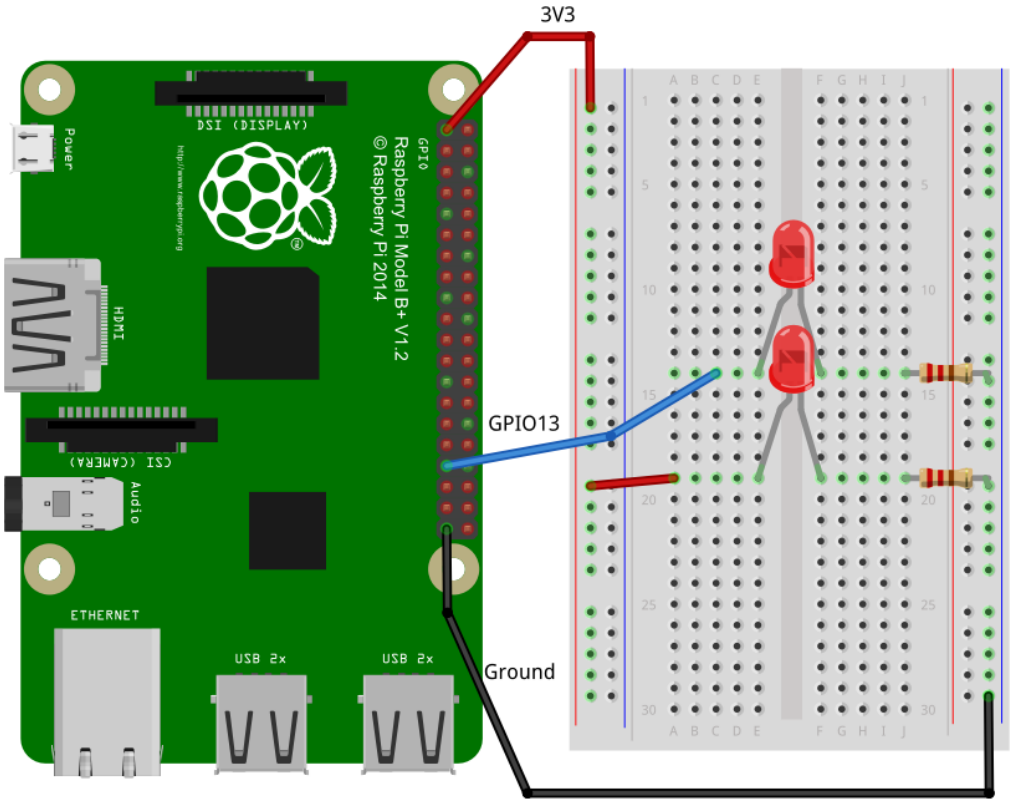
학습 목표

- PWM**제어**

실습1 : PWM

- **구성**
  - 220Ω **저항** x 2
  - LED x 2

- GPIO : 13



실습1 : PWM

- **파일명** : pwm1.c

```
#include <stdio.h>
#include <wiringPi.h>

#define LED1 13


int main(void)
{
    wiringPiSetupGpio();

    pinMode(LED1, OUTPUT);

    while(1)
    {
        digitalWrite(LED1, HIGH);
        delay(1000);
        digitalWrite(LED1, LOW);
        delay(1000);
    }

    return 0;
}
```

실습1 : PWM

- 컴파일

```
$ gcc -lwiringPi pwm1.c -o pwm1
```

- 실행

```
$ sudo ./pwm1
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

실습2 : PWM

- **파일명** : pwm2.c

```
#include <stdio.h>
#include <wiringPi.h>

#define LED1 13


int main(void)
{
    wiringPiSetupGpio();

    pinMode(LED1, OUTPUT);

    while(1)
    {
        digitalWrite(LED1, HIGH);
        delay(100);
        digitalWrite(LED1, LOW);
        delay(100);
    }

    return 0;
}
```

실습2 : PWM

- 컴파일

```
$ gcc -lwiringPi pwm2.c -o pwm2
```

- 실행

```
$ sudo ./pwm2
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

실습3 : PWM

- **파일명** : pwm3.c

```
#include <stdio.h>
#include <wiringPi.h>

#define LED1 13



int main(void)
{
    wiringPiSetupGpio();

    pinMode(LED1, OUTPUT);

    while(1)
    {
        digitalWrite(LED1, HIGH);
        delay(10);
        digitalWrite(LED1, LOW);
        delay(10);
    }

    return 0;
}
```

실습3 : PWM

- 컴파일

```
$ gcc -lwiringPi pwm3.c -o pwm3
```

- 실행

```
$ sudo ./pwm3
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

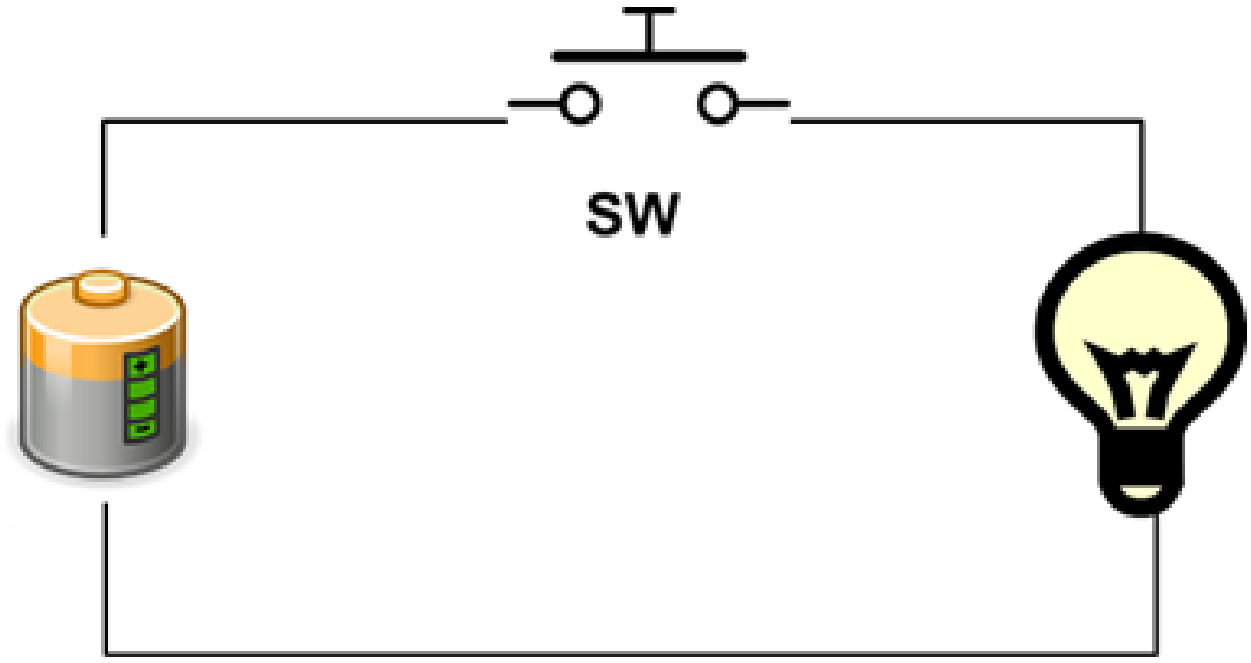
미션1

- **두 개의** LED **밝기를 비교하시오**

- **지금은** LED**가** On **상태로** 10ms, Off **상태로** 10ms **반복 되고 있다.**
  아래와 같이 delay**시간으로 변경하면서** LED**를 관찰해보시오.**
  - On **상태** 10ms, Off **상태** 10ms
  - On **상태** 11ms, Off **상태** 9ms
  - On **상태** 12ms, Off **상태** 8ms
  - On **상태** 13ms, Off **상태** 7ms
  - On **상태** 14ms, Off **상태** 6ms
  - On **상태** 15ms, Off **상태** 5ms
  - On **상태** 16ms, Off **상태** 4ms
  - On **상태** 17ms, Off **상태** 3ms
  - On **상태** 18ms, Off **상태** 2ms
  - On **상태** 19ms, Off **상태** 1ms
  - On **상태** 20ms, Off **상태** 0ms

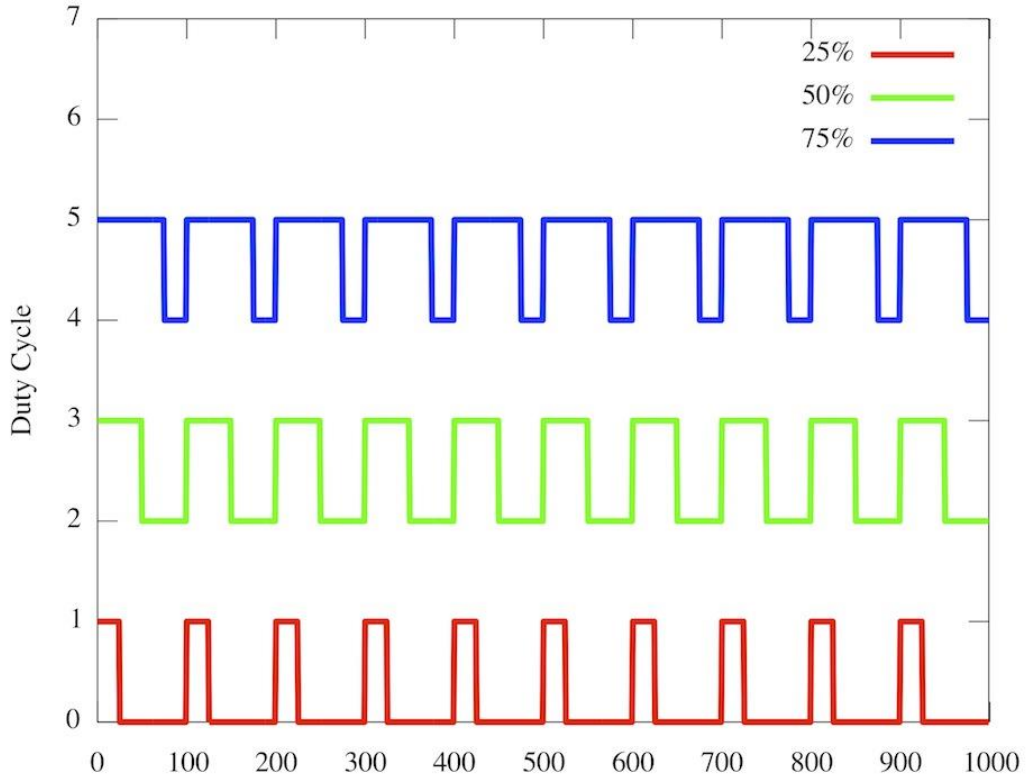
```
digitalWrite(LED1, HIGH);
delay(?);
digitalWrite(LED1, LOW);
delay(?);
```

PWM



PWM

- PWM : Pulse Width Modulator
  - **아날로그 신호를 디지털화**
  - **펄스의 폭**(width)**을 변경**(modulation) **하며 전체 전압을 제어**
  - **일반적으로 사용하는 변조 주파수** 1kHz ~ 200kHz



라즈베리 파이 PWM

BROADCOM. BCM2835 ARM Peripherals

6.2 Alternative Function Assignments

Every GPIO pin can carry an alternate function. Up to 6 alternate function are available but not every pin has that many alternate functions. The table below gives a quick over view.

	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
GPIO0	High	SDA0	SA5	<reserved>			
GPIO1	High	SCL0	SA4	<reserved>			
GPIO2	High	SDA1	SA3	<reserved>			
GPIO3	High	SCL1	SA2	<reserved>			
GPIO4	High	GPCLK0	SA1	<reserved>			ARM_TDI
GPIO5	High	GPCLK1	SA0	<reserved>			ARM_TDO
GPIO6	High	GPCLK2	SOE_N / SE	<reserved>			ARM_RTCK
GPIO7	High	SPI0_CE1_N	SWE_N / SRW_N	<reserved>			
GPIO8	High	SPI0_CE0_N	SD0	<reserved>			
GPIO9	Low	SPI0_MISO	SD1	<reserved>			
GPIO10	Low	SPI0_MOSI	SD2	<reserved>			
GPIO11	Low	[illegible]	SD3	<reserved>			
GPIO12	Low	PWM0	SD4	<reserved>			ARM_TMS
GPIO13	Low	PWM1	SD5	<reserved>			ARM_TCK
GPIO14	Low	TXD0	SD6	<reserved>			TXD1
GPIO15	Low	RXD0	SD7	<reserved>			RXD1
GPIO16	Low	<reserved>	SD8	<reserved>	CTS0	SPI1_CE2_N	CTS1
GPIO17	Low	<reserved>	SD9	<reserved>	RTS0	SPI1_CE1_N	RTS1
GPIO18	Low	PCM_CLK	SD10	<reserved>	BSCSL SDA / MOSI	SPI1_CE0_N	PWM0
GPIO19	Low	PCM_FS	SD11	<reserved>	BSCSL SCL / SCLK	SPI1_MISO	PWM1
GPIO20	Low	PCM_DIN	SD12	<reserved>	BSCSL / MISO	SPI1_MOSI	[illegible]

라즈베리 파이 PWM

GPIO20	Low	PCM_DIN	SD12	<reserved>	BSCSL / MISO	SPI1_MOSI	GPCLK0
GPIO21	Low	PCM_DOUT	SD13	<reserved>	BSCSL / CE_N	SPI1_SCLK	GPCLK1
GPIO22	Low	<reserved>	SD14	<reserved>	SD1_CLK	ARM_TRST	
GPIO23	Low	<reserved>	SD15	<reserved>	SD1_CMD	ARM_RTCK	
GPIO24	Low	<reserved>	SD16	<reserved>	SD1_DAT0	ARM_TDO	
GPIO25	Low	<reserved>	SD17	<reserved>	SD1_DAT1	ARM_TCK	
GPIO26	Low	<reserved>	<reserved>	<reserved>	SD1_DAT2	ARM_TDI	
GPIO27	Low	<reserved>	<reserved>	<reserved>	SD1_DAT3	ARM_TMS	
GPIO28	-	SDA0	SA5	PCM_CLK	<reserved>		
GPIO29	-	SCL0	SA4	PCM_FS	<reserved>		
GPIO30	Low	<reserved>	SA3	PCM_DIN	CTS0		CTS1
GPIO31	Low	<reserved>	SA2	PCM_DOUT	RTS0		RTS1
GPIO32	Low	GPCLK0	SA1	<reserved>	TXD0		TXD1
GPIO33	Low	<reserved>	SA0	<reserved>	RXD0		RXD1
GPIO34	High	GPCLK0	SOE_N / SE	<reserved>	<reserved>		
GPIO35	High	SPI0_CE1_N	SWE_N / SRW_N		<reserved>		
GPIO36	High	SPI0_CE0_N	SD0	TXD0	<reserved>		
GPIO37	Low	SPI0_MISO	SD1	RXD0	<reserved>		
GPIO38	Low	SPI0_MOSI	SD2	RTS0	<reserved>		
GPIO39	Low	SPI0_SCLK	SD3	CTS0	<reserved>		
GPIO40	Low	PWM0	SD4		<reserved>	SPI2_MISO	TXD1
	Pull	**ALT0**	**ALT1**	**ALT2**	**ALT3**	**ALT4**	**ALT5**

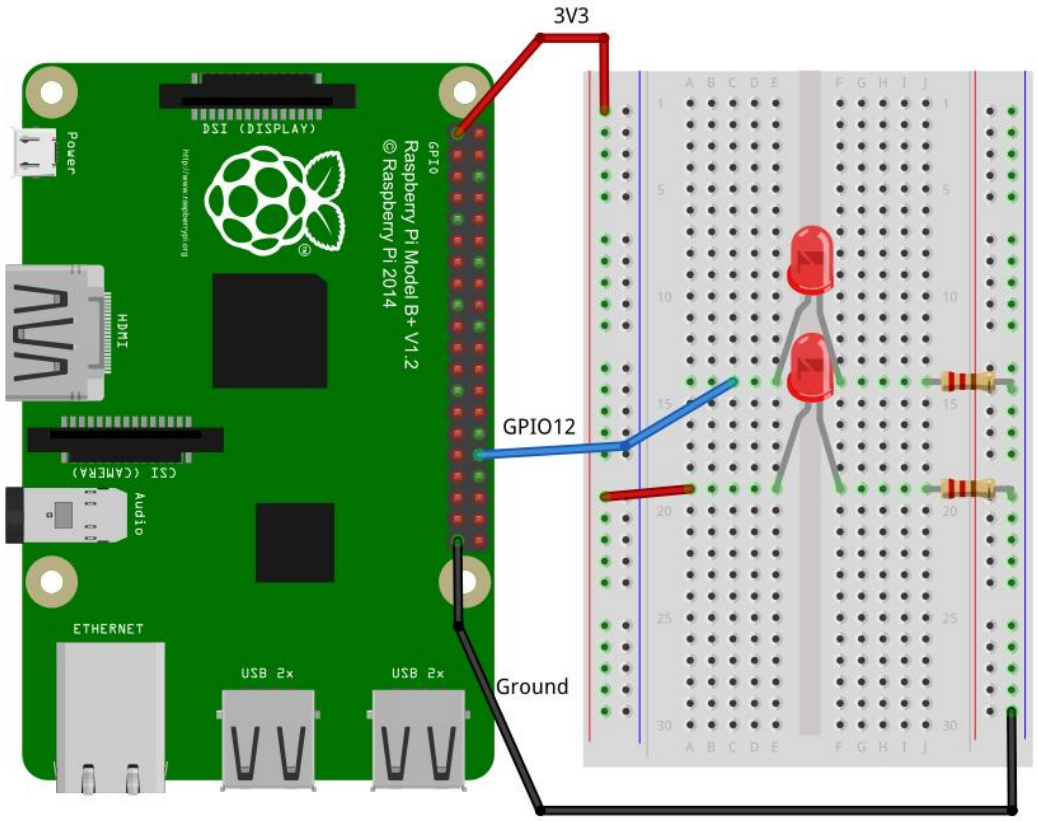
라즈베리 파이 PWM

- **라즈베리 파이**(B+) PWM
  - PWM **기능이 가능한** GPIO**핀 번호**
    - PWM 0 : GPIO 12, 18
    - PWM 1 : GPIO 13, 19

실습4 : PWM

- **구성**
  - 220Ω **저항** x 2
  - LED x 2

- GPIO : 12



실습4 : 제어 명령어를 이용한 PWM 제어

- GPIO 12번 핀을 PWM 모드로 변경

```
$ gpio -g mode 12 pwm
$ gpio readall
```

```
pi@raspberrypi:~/src$ gpio readall
 +-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
 | 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 | ALT0 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | 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 |  OUT | 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 | ALT0 | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |  OUT | 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 |
 +-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
```

실습4 : 제어 명령어를 이용한 PWM 제어

- Pulse Width **제어**
  - gpio -g pwm [**핀**#] [pulse width]
  - 100% **일 때** : 1023
  - **값의 범위** 0~1023(**기본값**)

- **다음 명령을 실행하며** LED **밝기를 확인해 보자**.

```
$ gpio -g pwm 12 0
$ gpio -g pwm 12 10
$ gpio -g pwm 12 20
$ gpio -g pwm 12 30
$ gpio -g pwm 12 100
$ gpio -g pwm 12 200
$ gpio -g pwm 12 300
$ gpio -g pwm 12 400
$ gpio -g pwm 12 500
$ gpio -g pwm 12 600
$ gpio -g pwm 12 1023
```

실습4 : 제어 명령어를 이용한 PWM 제어

- 값의 범위 변경

```
$ gpio -g pwmr 10
$
$ gpio -g pwm 12 0
$ gpio -g pwm 12 1
$ gpio -g pwm 12 2
$ gpio -g pwm 12 3
$ gpio -g pwm 12 4
$ gpio -g pwm 12 5
$ gpio -g pwm 12 6
$ gpio -g pwm 12 7
$ gpio -g pwm 12 8
$ gpio -g pwm 12 9
$ gpio -g pwm 12 10
```

실습5 : PWM

- **파일명** : pwm5.c

```
#include <stdio.h>
#include <wiringPi.h>
#define LED 12

int main(void)
{
    wiringPiSetupGpio();
    pinMode(LED, PWM_OUTPUT);

    // 범위 0 ~ 1024
    pwmWrite(LED, 0);
    delay(1000);
    pwmWrite(LED, 400);
    delay(1000);
    pwmWrite(LED, 700);
    delay(1000);
    pwmWrite(LED, 900);
    delay(1000);
    pwmWrite(LED, 1024);
    delay(1000);

    return 0;
}
```

실습5 : PWM

- 컴파일

```
$ gcc -lwiringPi pwm5.c -o pwm5
```

- 실행

```
$ sudo ./pwm5
```

실습6 : PWM

- **파일명** : pwm7.c

```
#include <stdio.h>
#include <wiringPi.h>

#define LED 12

int main(void)
{

    int i, j;
    wiringPiSetupGpio();
    pinMode(LED, PWM_OUTPUT);

    for(i = 0 ; i <= 1024 ; i++)
    {
        pwmWrite(LED, i);
        delayMicroseconds(1000);
    }

    return 0;
}
```

실습6 : PWM

- 컴파일

```
$ gcc -lwiringPi pwm7.c -o pwm7
```

- 실행

```
$ sudo ./pwm7
```

실습7 : PWM

- **파일명** : pwm8.c

```
#include <stdio.h>
#include <wiringPi.h>

#define LED 12

int main(void)
{

    int i, j;
    wiringPiSetupGpio();
    pinMode(LED, PWM_OUTPUT);

    for(j = 0 ; j < 5 ; j++)
    {
        for(i = 0 ; i <= 1024 ; i++)
        {
            pwmWrite(LED, i);
            delayMicroseconds(500);
        }
    }

    return 0;
}
```

실습7 : PWM

- 컴파일

```
$ gcc -lwiringPi pwm8.c -o pwm8
```

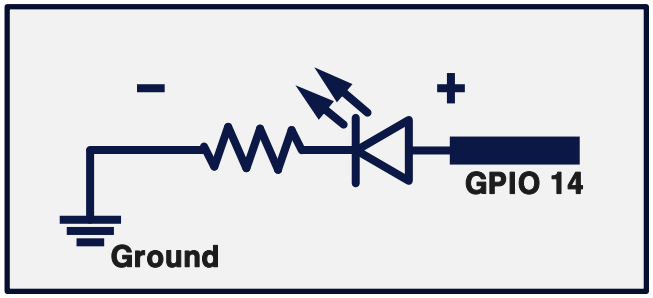
- 실행

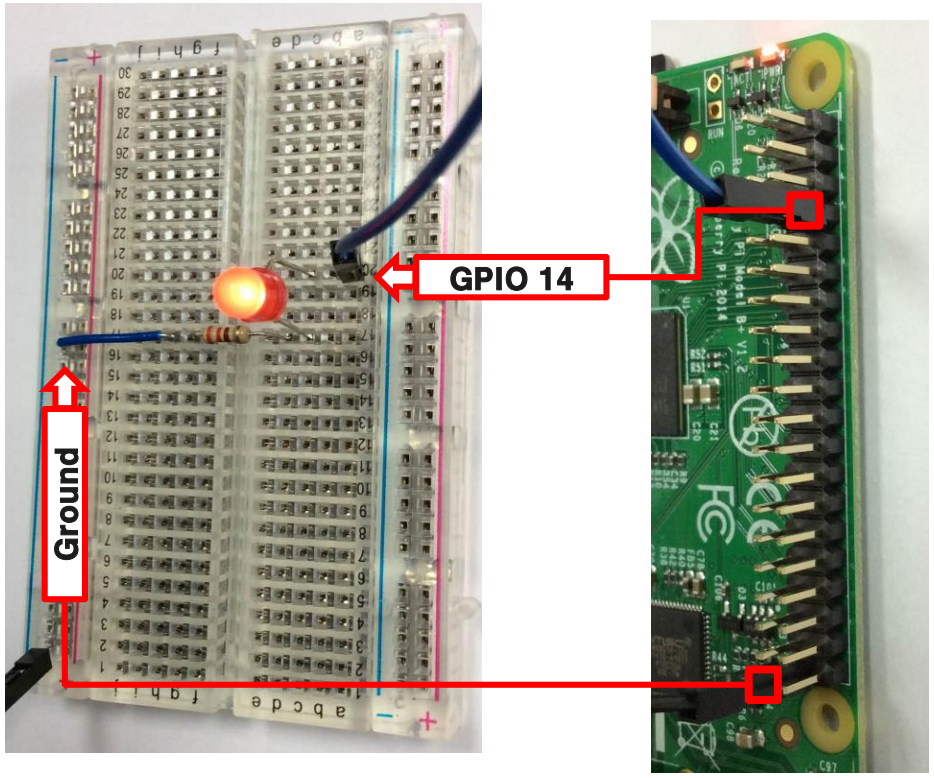
```
$ sudo ./pwm8
```

실습8-1 : Soft PWM(1/3)

- wiringPi **라이브러리는** S/W**방식으로** PWM **기능을 제공**
- **구성**
  - 220Ω **저항**
  - LED

- GPIO : 14





실습8-2 : Soft PWM(2/3)

- GPIO 를 이용한 PWM 제어
- **파일명** : gpioPwm.c

```
#include <stdio.h>
#include <wiringPi.h>
#include <softPwm.h>

#define PWM 14          // gpio 14

#define EXIT_SUCC  0
#define EXIT_FAIL  1

int
main(void)
{
    int x = 0;
    int orient = 0;

    if(wiringPiSetupGpio() == -1)
        return EXIT_FAIL;

    softPwmCreate (PWM, 0, 100) ;

    while(1) {
        if(orient == 0){
            x += 2;
            if(x == 100) orient = 1;
        } else {
            x -= 2;
            if(x == 0) orient = 0;
        }

        softPwmWrite (PWM, x);

        delay(10);
    }

    return EXIT_SUCC;
}
```

실습8-3 : Soft PWM(3/3)

- 컴파일

```
$ gcc -Wall -W -lwiringPi -lpthread gpioPwm.c -o gpioPwm
```

- 실행

```
$ sudo ./gpioPwm
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

미션2

1. LED가 점점 밝아지고 다시 점점 어두워지도록 프로그램을 작성하시오.

2. 스위치2개와 LED를 각각 라즈베리파이 GPIO에 연결하고 스위치를 이용해 LED의 밝기를 조절하는 프로그램을 작성하시오.
   - 스위치 1을 누르면 LED밝기가 밝아진다.
   - 스위치 2을 누르면 LED밝기가 어두워진다.