



라즈베리 파이 GPIO

로봇SW 교육원

최상훈(shchoi82@gmail.com)

학습 목표

2

- 라즈베리파이 GPIO 제어
- wiringPi 라이브러리
- BCM2835 라이브러리

라즈베리 파이 GPIO

3

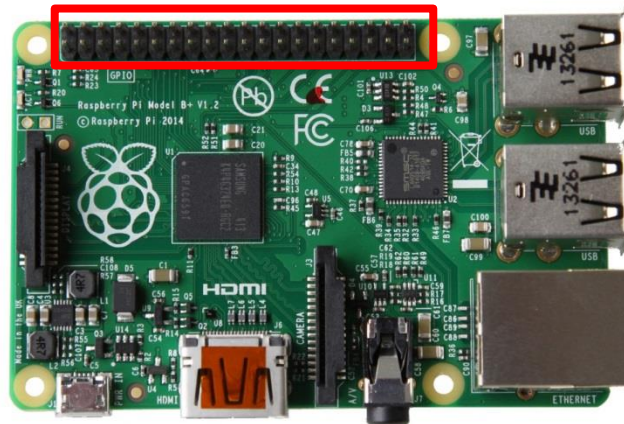
- GPIO
 - General Purpose Input / Output
 - 범용적으로 사용 가능
 - 입/출력 신호를 제어할 수 있는 전용 포트
- OS(리눅스) 기반 동작으로 인한 실시간 처리의 제한
 - 다중 프로세스 실행에 따른 프로세스 스케줄링 또는 시스템 부하에 의해 실행속도와 실행시점을 정확히 파악할 수 없음
 - 그러므로 실시간 처리를 요구하는 응용에는 부적합함

라즈베리 파이 GPIO

4

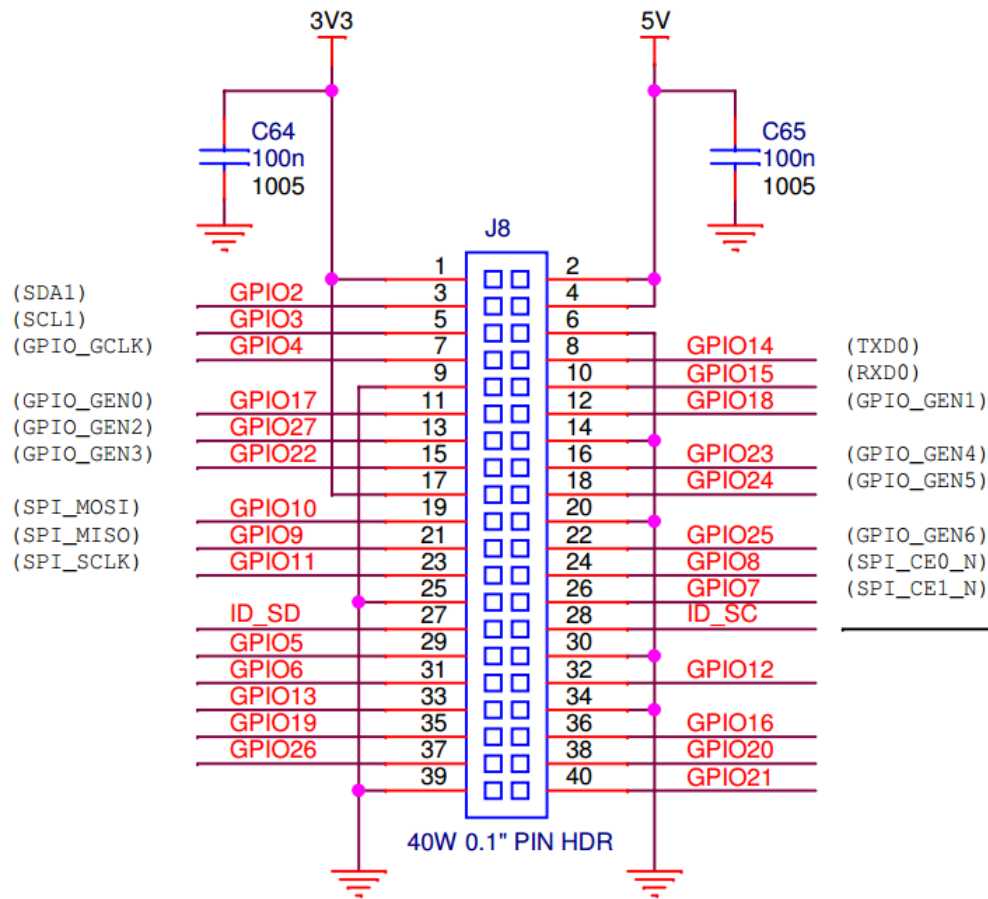
〈라즈베리 파이 B+의 GPIO〉

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



라즈베리 파이 GPIO

• Raspberry-Pi-B-Plus-V1.2-Schematics



ID_SD and ID_SC PINS:

These pins are reserved for ID EEPROM.

At boot time this I2C interface will be interrogated to look for an EEPROM that identifies the attached board and allows automatic setup of the GPIOs (and optionally, Linux drivers).

DO NOT USE these pins for anything other than attaching an I2C ID EEPROM. Leave unconnected if ID EEPROM not required.

GPIO EXPANSION

wiringPi 라이브러리

6

- 라즈베리파이 GPIO 라이브러리
- *GPIO Interface library for the Raspberry Pi*
- 주소 : <http://wiringpi.com/>



실습 1-1 : wiringPi 라이브러리 설치

7

- wiringPi 라이브러리 다운로드

```
$ sudo apt-get install git-core
```

```
$ git clone git://git.drogon.net/wiringPi
```

- wiringPi 라이브러리 빌드

```
$ cd wiringPi
```

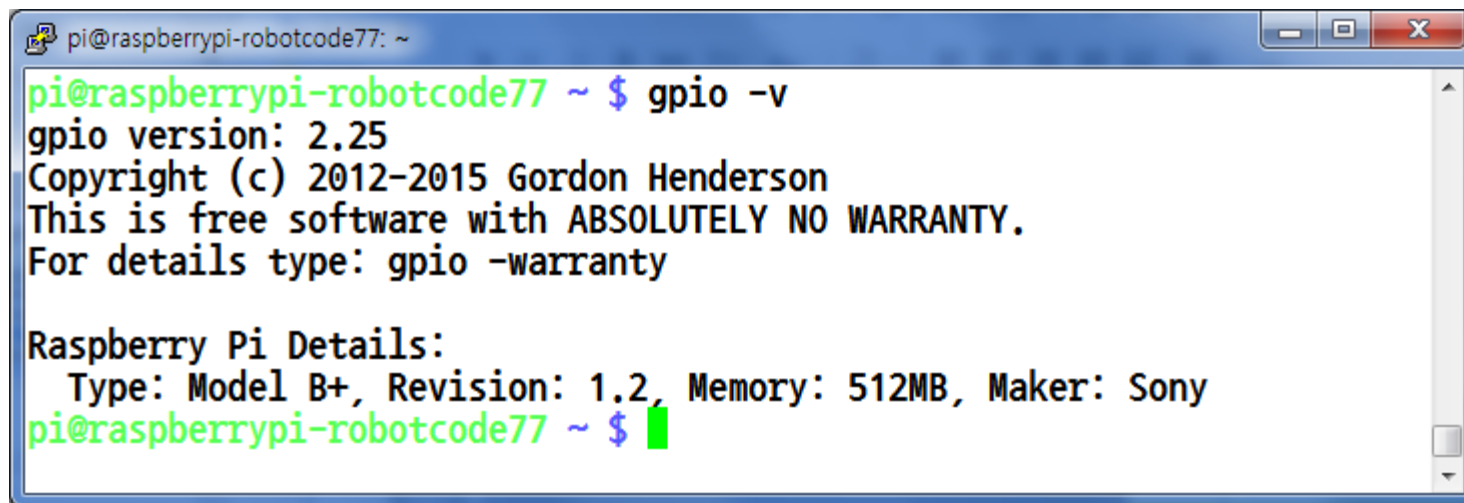
```
$ ./build
```

실습 1-2 : wiringPi 라이브러리 설치

8

- 설치 확인

```
$ gpio -v
```



```
pi@raspberrypi-robotcode77: ~  
pi@raspberrypi-robotcode77 ~ $ gpio -v  
gpio version: 2.25  
Copyright (c) 2012-2015 Gordon Henderson  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type: gpio -warranty  
  
Raspberry Pi Details:  
Type: Model B+, Revision: 1.2, Memory: 512MB, Maker: Sony  
pi@raspberrypi-robotcode77 ~ $
```


실습 1-3 : wiringPi 라이브러리 설치

9

• GPIO 핀 정보 확인

```
$ gpio readall
```

pi@raspberrypi-robotcode77: ~
pi@raspberrypi-robotcode77 ~\$ gpio readall

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	1	ALT0 TxD	15	14
		0v			9	10	1	ALT0 RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v		
22	3	GPIO. 3	IN	0	15	16	0	GPIO. 4	4	23
		3.3v			17	18	0	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v		
9	13	MISO	IN	0	21	22	0	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	CE0	10	8
		0v			25	26	1	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v		
6	22	GPIO.22	IN	1	31	32	0	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v		
19	24	GPIO.24	IN	0	35	36	0	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	GPIO.28	28	20
		0v			39	40	0	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM

pi@raspberrypi-robotcode77 ~\$



wiringPi 라이브러리

10

- wiringPi 를 이용한 라즈베리 파이 GPIO 제어
 - 제어 명령어 : gpio
 - 프로그래밍 인터페이스 제공 : 라이브러리

wiringPi : 제어 명령어

11

- GPIO 핀 제어 명령어
 - 핀의 모드 (디지털 출력) 설정
 - 핀 모드의 종류 : 디지털 입력, 디지털 출력, PWM 출력
 - 핀 모드 설정 방법
 - gpio mode 핀번호 out
 - ex) GPIO 3을 **디지털 출력**으로 설정

```
$ gpio -g mode 3 out
```

- 디지털 출력
 - 디지털 출력 값 설정 방법
 - gpio write 핀번호 0/1
 - ex) GPIO 3에 디지털 출력

```
$ gpio -g write 3 1
```

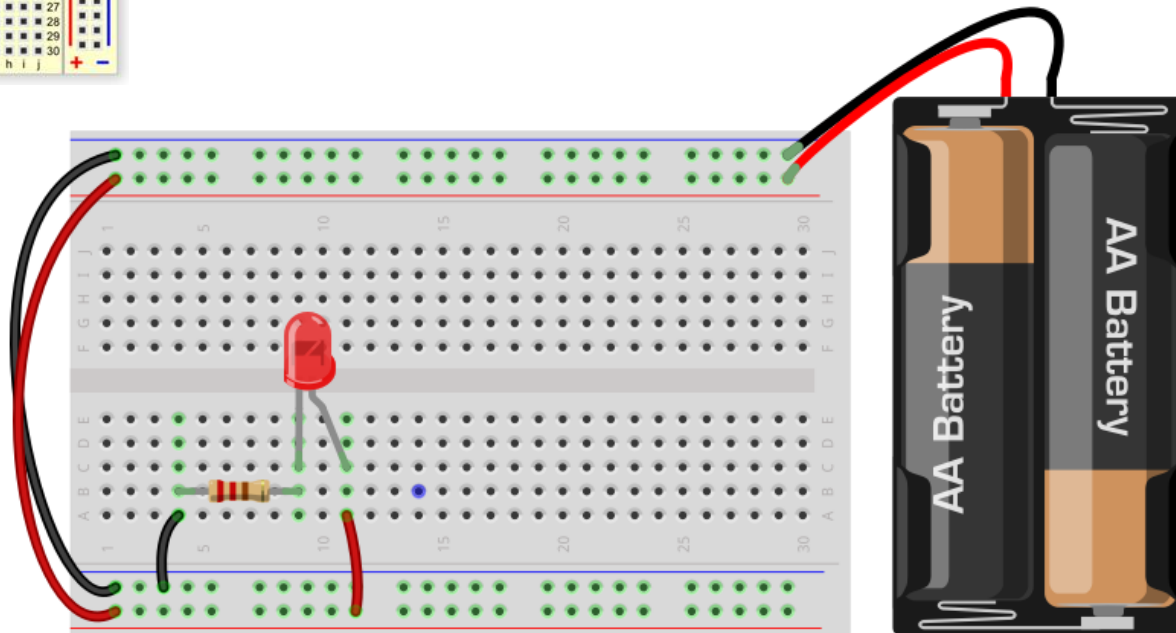
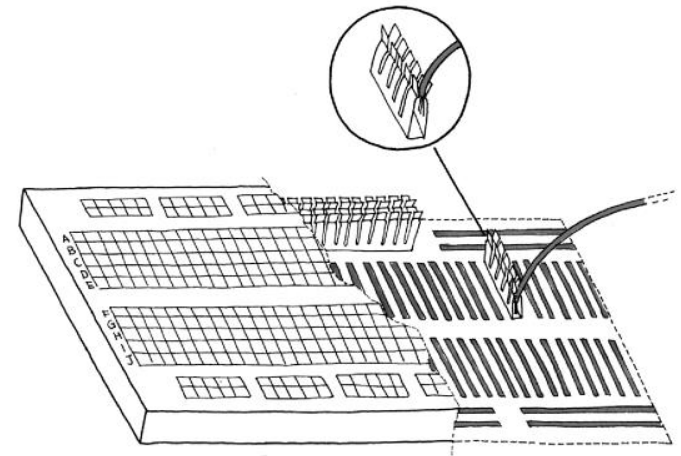
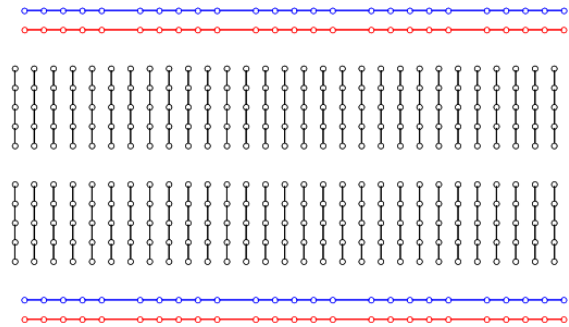
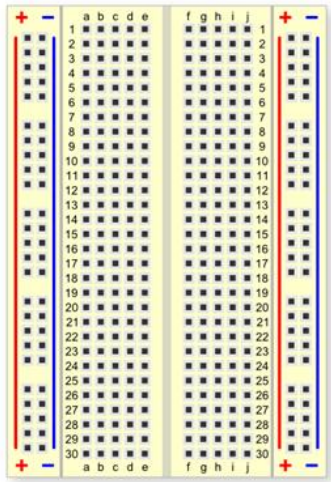
HIGH

```
$ gpio -g write 3 0
```

LOW

Bread board

12

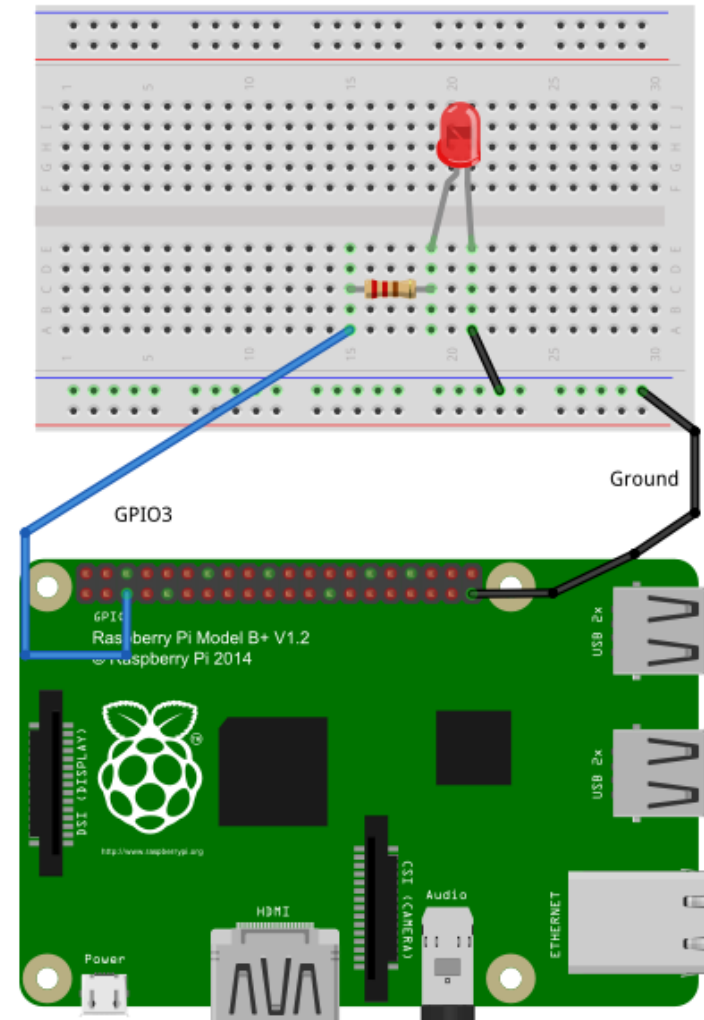


실습 2-1: 제어 명령어로 LED On/Off

13

1. GPIO 3핀에 LED 연결

- 220 Ω 저항
- LED



실습 2-2: 제어 명령어로 LED On/Off

14

2. GPIO 3 핀 모드 출력으로 설정

```
$ gpio -g mode 3 out
```

3. GPIO 3 핀 모드 확인

```
$ gpio readall
```

COM8 - PuTTY

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

BCM	wPi	Name	Mode	V	Phys
		3.3v			1
2	8	SDA	IN	1	3
3	9	SCL	OUT	0	5
4	7	GPIO. 7	IN	1	7
		0v			9
17	0	GPIO. 0	IN	0	11
27	2	GPIO. 2	IN	0	13
22	3	GPIO. 3	IN	0	15
		3.3v			17
10	12	MOSI	IN	0	19
9	13	MISO	IN	0	21
11	14	SCLK	IN	0	23
		0v			25

실습 2-3: 제어 명령어로 LED On/Off

15

4. LED ON

```
$ gpio -g write 3 1
```

5. LED OFF

```
$ gpio -g write 3 0
```

wiringPi 프로그래밍 인터페이스

16

- 소스코드의 헤더파일
 - 소스코드에서 wiringPi 라이브러리를 사용하기 위해서는 wiringPi 헤더를 추가해야 함
`#include <wiringPi.h>`
- 컴파일
 - 컴파일 할때 링커 옵션 `-lwiringPi` 를 추가해야 함
ex) `gcc -lwiringPi source.c`

wiringPi : 프로그래밍 인터페이스

17

- wiringPi 초기화 함수
 - wiringPi 라이브러리를 사용하기 전 다음 3가지 함수 중 반드시 하나가 호출되어야 함
 - wiringPiSetupPhys()
 - 핀제어시 물리번호 사용
 - wiringPiSetupGpio()
 - 핀제어시 GPIO번호를 사용함
 - wiringPiSetupSys()
 - 핀제어시 하드웨어를 직접 접근하지 않고
sys파일시스템의 /sys/class/gpio 를 사용 root 권한 필요 없음

wiringPi : 프로그래밍 인터페이스

18

- GPIO제어 관련 함수

- pinMode(int pin, int mode)

- mode의 종류 : INPUT, OUTPUT, PWM_OUTPUT

- ex) GPIO 3 핀을 출력으로 하고 싶을 경우

- pinMode(3, OUTPUT)

- ex) GPIO 4 핀을 입력으로 하고 싶을 경우

- pinMode(4, INPUT)

pin 모드 설정

- digitalWrite(int pin, int value)

- value의 종류 : 1 또는 0 (HIGH, LOW)

- ex) GPIO 3 핀을 1(HIGH)로 설정

- digitalWrite(3, 1); 또는 digitalWrite(3, HIGH);

- ex) GPIO 3 핀을 0(LOW)로 설정

- digitalWrite(3, 0); 또는 digitalWrite(3, LOW);

디지털 출력
(OUTPUT)

- int digitalRead(int pin)

- GPIO 4핀의 상태(HIGH 또는 LOW) 읽기

- ex) status = digitalRead(4)

디지털 입력
(INPUT)

wiringPi : 프로그래밍 인터페이스

19

– pullUpDnControl(int pin, int pud);

- pud **종류** : PUD_OFF, PUD_DOWN(pull to ground), PUD_UP(pull to 3.3v)

**GPIO 핀
상태 설정**

– void pwmWrite(int pin, int value);

- value **범위** : 0~1024

PWM 출력 설정

– pwmSetRange (unsigned int range) ;

PWM 범위 설정

wiringPi : 프로그래밍 인터페이스

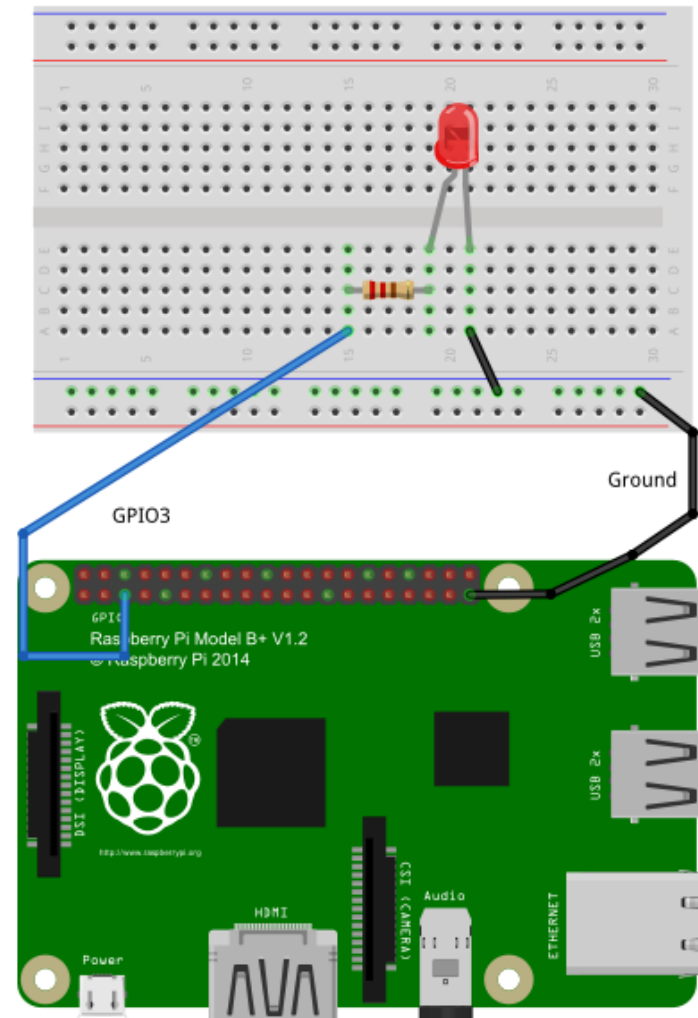
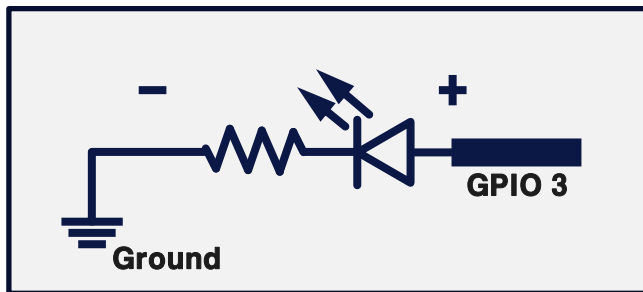
20

- 타이밍 관련 함수
 - unsigned intmills()
 - 초기화된 후 지난 밀리초를 구함
 - unsigned intmicros()
 - 초기화된 후 지난 마이크로초를 구함
 - void delay(unsigned int howLong)
 - 프로그램을 지연시킴(millisecond)
 - void delayMicroseconds(unsigned int howLong)
 - 프로그램을 지연시킴(microsecond)

실습 3-1 : LED

21

- wiringPi 라이브러리를 이용한 LED ON / OFF
- 구성
 - 220Ω 저항
 - LED
- GPIO : 3



실습 3-2 : LED

22

- 파일명 : gpioLed1.c

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

#define LED1 3          // gpio 3

#define EXIT_SUCC  0
#define EXIT_FAIL  1

int
main(void)
{
    if(wiringPiSetupGpio() == -1)
        return EXIT_FAIL;

    pinMode(LED1, OUTPUT);

    while(1) {
        digitalWrite(LED1, 1);

        delay(1000);

        digitalWrite(LED1, 0);

        delay(1000);
    }

    return EXIT_SUCC;
}
```

실습 3-3 : LED

23

- 컴파일

```
$ gcc -lwiringPi gpioLed1.c -o gpioLed1
```

- 실행

```
$ sudo ./gpioLed1
```

- 프로그램 강제 종료

```
[Ctrl + c]
```

미션!

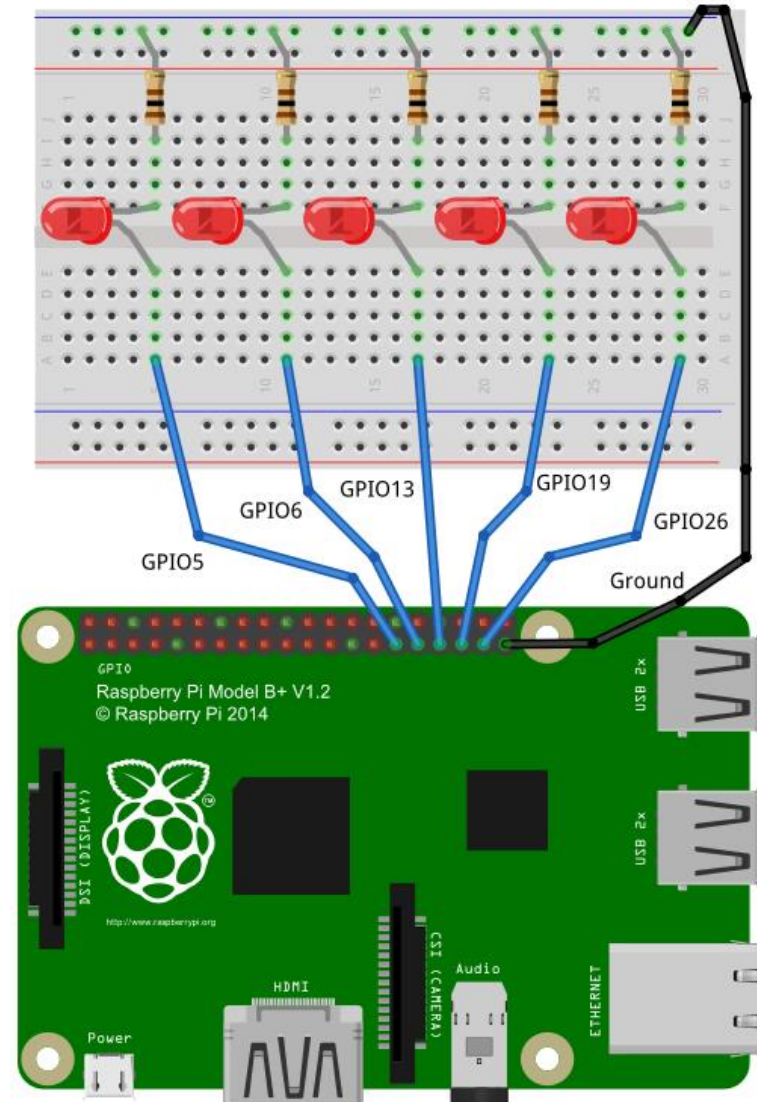
24

1. 실습 3 프로그램은 1초 단위로 On/Off 된다.
이 프로그램이 2초 단위로 On/Off 되도록 변경해라.
 - 0.5초 단위로 변경해라.
 - 0.1초 단위로 변경해라.
 - 0.05초 단위로 변경해라.
2. 실습 3 프로그램은 무한대로 반복된다. 이 프로그램 이 10번 반복 후 종료 되도록 프로그램을 수정하시오.
3. GPIO12 핀에 LED와 저항을 연결하고 wiringPi 제어명령어를 이용해 LED를 On/Off 하시오.

실습 4-1 : LED

25

- wiringPi 라이브러리를 이용한 LED ON / OFF
- 구성
 - 100Ω 저항 5개
 - LED 5개
- GPIO : 5, 6, 13, 19, 26



실습 4-2 : LED

26

- wiringPi 제어명령어를 이용해 LED를 On/Off

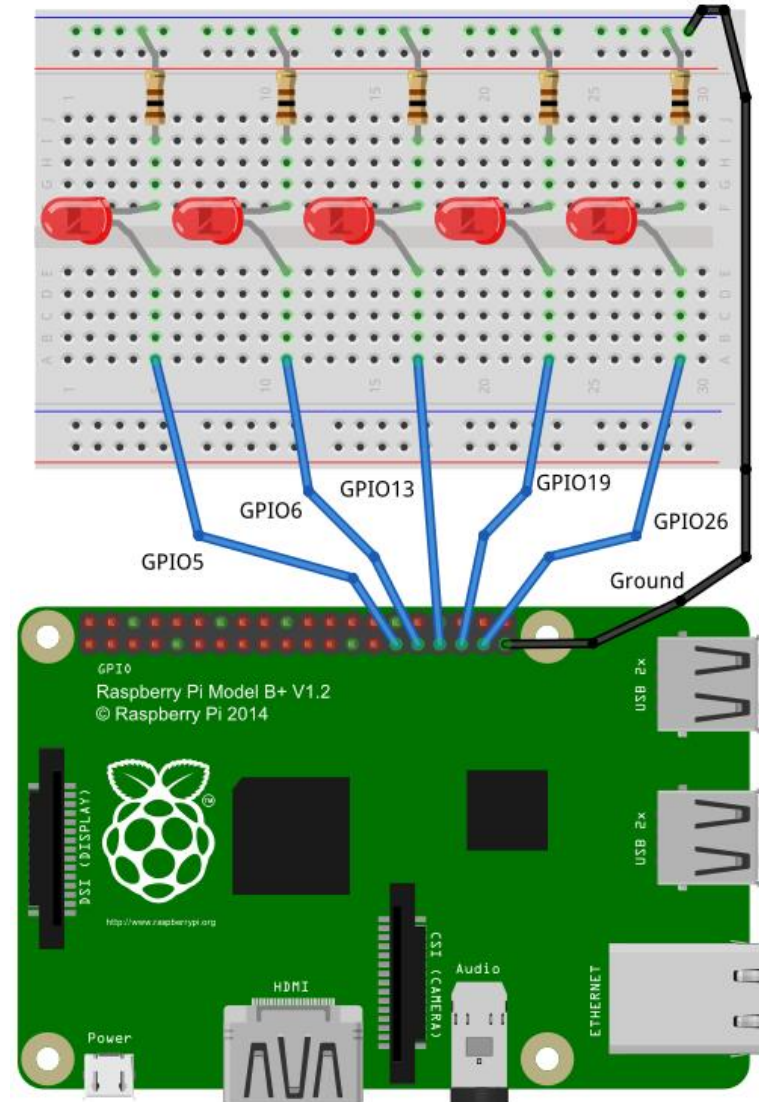
- GPIO : 5, 6, 13, 19, 26

- On

```
$ gpio -g write 5 1
$ gpio -g write 6 1
$ gpio -g write 13 1
$ gpio -g write 19 1
$ gpio -g write 26 1
```

- Off

```
$ gpio -g write 5 0
$ gpio -g write 6 0
$ gpio -g write 13 0
$ gpio -g write 19 0
$ gpio -g write 26 0
```



실습 4-3 : LED

27

• 파일명 : gpioLed2.c

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

#define LED1 5
#define LED2 6
#define LED3 13
#define LED4 19
#define LED5 26

#define EXIT_SUCC 0
#define EXIT_FAIL 1

int
main(void)
{
    int i;
    wiringPiSetupGpio();

    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    pinMode(LED5, OUTPUT);
```

```
    digitalWrite(LED1, 0);
    digitalWrite(LED2, 0);
    digitalWrite(LED3, 0);
    digitalWrite(LED4, 0);
    digitalWrite(LED5, 0);

    delay(1000);

    digitalWrite(LED1, 1);
    delay(200);
    digitalWrite(LED1, 0);

    digitalWrite(LED2, 1);
    delay(200);
    digitalWrite(LED2, 0);

    digitalWrite(LED3, 1);
    delay(200);
    digitalWrite(LED3, 0);

    digitalWrite(LED4, 1);
    delay(200);
    digitalWrite(LED4, 0);

    digitalWrite(LED5, 1);
    delay(200);
    digitalWrite(LED5, 0);
    return 0;
```

}

실습 4-4 : LED

28

- 컴파일

```
$ gcc -lwiringPi gpioLed2.c -o gpioLed2
```

- 실행

```
$ sudo ./gpioLed2
```

미션!

29

1. 실습4 프로그램은 GPIO 핀번호를 매크로로 정의하여 사용하였다.
int 배열과 반복문을 사용하여 프로그램을 간단하게 수정해보자.
2. 5개의 LED가 0.2초 간격으로 순서대로 On/Off 된다.
0.1초 간격으로 변경해보자.
3. 5개의 LED가 0.1초 간격으로 켜진후 역순으로 켜지도록 변경해보자.
-> 1번(LED) 2번 3번 4번 5번 **5번 4번 3번 2번 1번**
4. 프로그램 시작할 때 전체 LED를 0.5초 간격으로 5번 깜빡이는 코드를 추가해보자.
5. 위 프로그램은 한번 수행되고 종료된다.
이 프로그램을 무한대로 반복하도록 변경해보자.