

PWM 출력 제어

컴퓨터 네트워크 설계

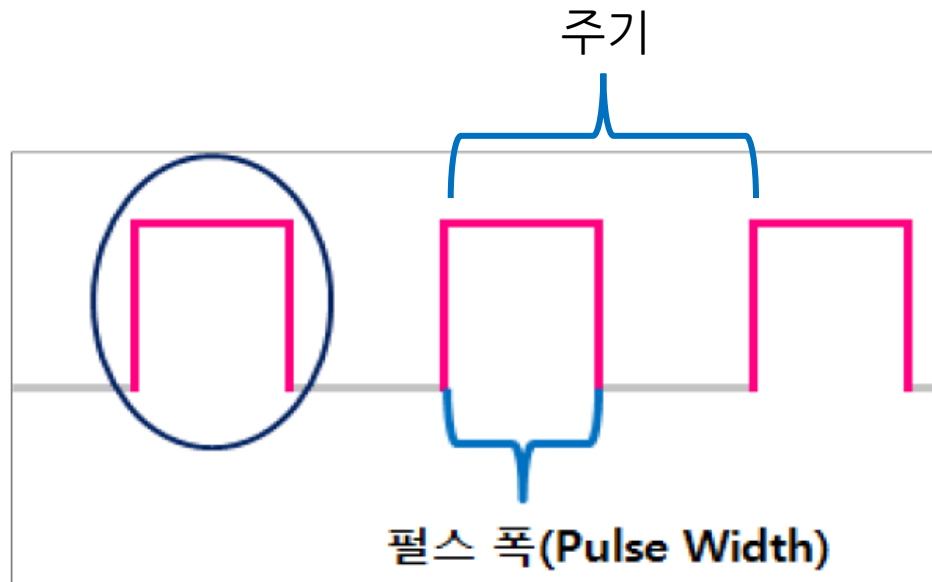
청주대학교 전자공학과
한철수

목차

- PWM 출력 제어
- 하드웨어 방식의 PWM 출력 제어 함수
- 소프트웨어 방식의 PWM 출력 제어 함수
- 제어 연습

PWM 출력 제어

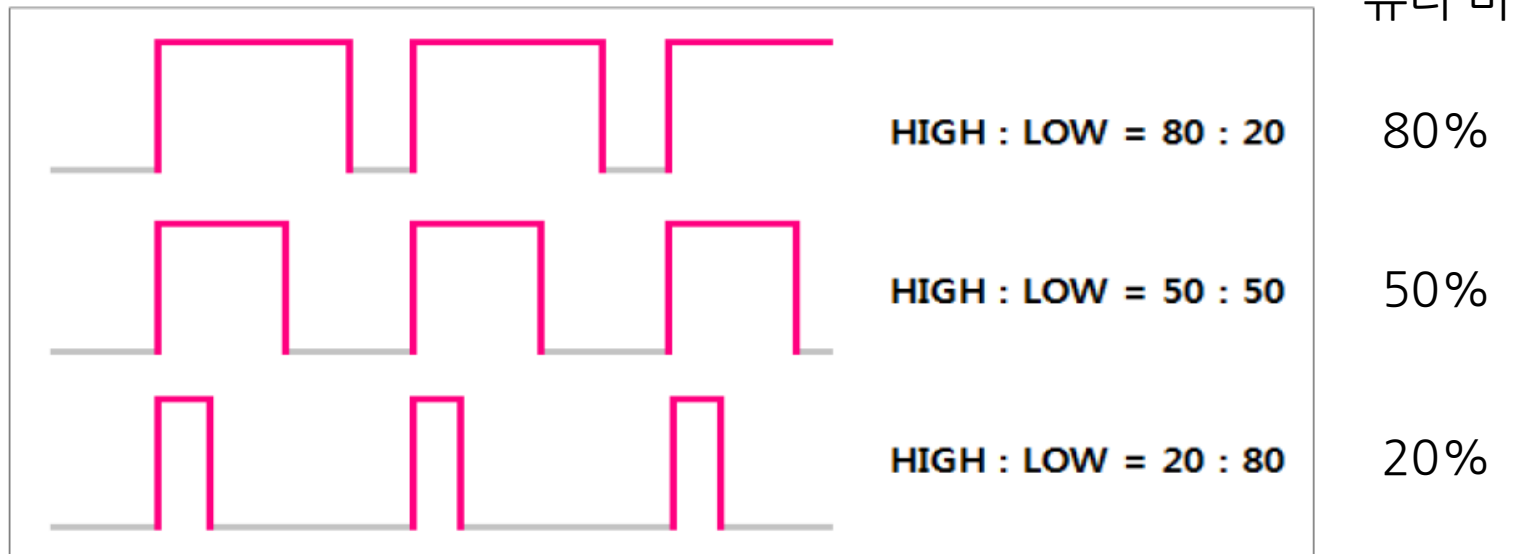
- PWM 출력 제어란 GPIO핀을 통해 출력되는 주기적인 펄스 신호의 펄스 폭을 조절하는 것을 말함.
 - PWM(Pulse Width Modulation)은 펄스 폭 변조를 뜻함.
- PWM 신호



PWM 신호의 듀티 비

- 듀티 비(Duty Ratio)

$$\text{듀티 비} = \frac{\text{펄스 폭}}{\text{주기}}$$



- PWM 출력 제어는 **조명의 밝기 제어, 모터의 속도 제어** 등에 널리 이용됨.

라즈베리 파이의 PWM 핀

- 라즈베리 파이에 2개의 PWM 채널이 있음.
 - PWM0
 - PWM1

PWM1
PWM1

GPIO#	NAME		NAME	GPIO#
	3.3 VDC Power	1	5.0 VDC Power	2
8	GPIO 8 SDA1 (I2C)	3	5.0 VDC Power	4
9	GPIO 9 SCL1 (I2C)	5	Ground	6
7	GPIO 7 GPCLK0	7	GPIO 15 TxD (UART)	15
	Ground	9	GPIO 16 RxD (UART)	16
0	GPIO 0	11	GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13	Ground	14
3	GPIO 3	15	GPIO 4	4
	3.3 VDC Power	17	GPIO 5	5
12	GPIO 12 MOSI (SPI)	19	Ground	20
13	GPIO 13 MISO (SPI)	21	GPIO 6	6
14	GPIO 14 SCLK (SPI)	23	GPIO 10 CE0 (SPI)	10
	Ground	25	GPIO 11 CE1 (SPI)	11
	SDA0 (I2C ID EEPROM)	27	SCL0 (I2C ID EEPROM)	
21	GPIO 21 GPCLK1	29	Ground	30
22	GPIO 22 GPCLK2	31	GPIO 26 PWM0	26
23	GPIO 23 PWM1	33	Ground	34
24	GPIO 24 PCM_FS/PWM1	35	GPIO 27	27
25	GPIO 25	37	GPIO 28 PCM_DIN	28
	Ground	39	GPIO 29 PCM_DOUT	29

PWM0

PWM0

PWM 핀의 wPi 핀 번호

- 1번, 26번 핀이 PWM0을, 23번, 24번 핀이 PWM1을 위한 핀임.

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

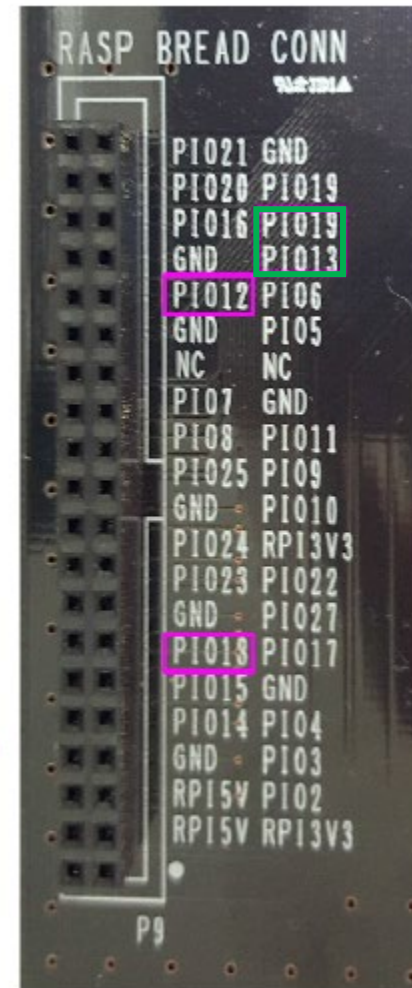
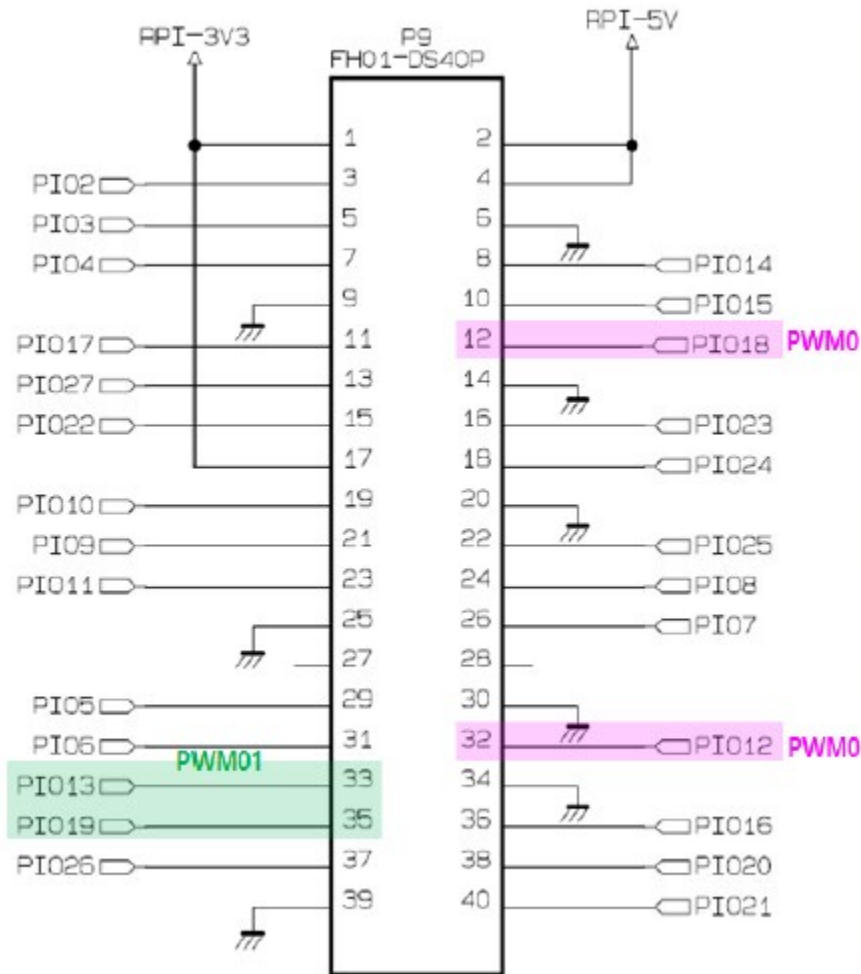
Pi 3											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	ALT0	1	3	4		5V			
3	9	SCL.1	ALT0	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	IN	TxD	15	14
		0v			9	10	1	IN	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	1	13	14		0v			
22	3	GPIO. 3	IN	1	15	16	1	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19	20		0v			
9	13	MISO	ALT0	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23	24	1	OUT	CE0	10	8
		0v			25	26	1	OUT	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	1	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	

PWM0

PWM0

PWM1
PWM1

키트의 PWM 핀



하드웨어 방식의 PWM 출력 제어 함수

- 내장된 전용 하드웨어에서 PWM 신호를 만들어 PWM 출력 전용 핀을 통해 출력시킴.
- `void pwmWrite(int pin, int value);`
 - pin에 `value/1024`의 듀티 비를 갖는 PWM 파형을 출력시킴.
 - 매개 변수
 - Pin에는 1만 넣을 것. 23, 24, 26번 핀은 현재 지원하지 않음.
 - Value에는 0~1024의 값을 넣음.
- 주의 사항
 - `sudo ./test` // 관리자 권한으로 실행

함수 사용 예

- 사용 예 1
 - `pwmWrite(1, 0);`
 - 1번 핀에서 듀티 비 0%의 PWM 신호를 출력시킴. ($0/1024=0$)
- 사용 예 2
 - `pwmWrite(1, 512);`
 - 1번 핀에서 듀티 비 50%의 PWM 신호를 출력시킴. ($512/1024=0.5$)
- 사용 예 3
 - `pwmWrite(1, 1024);`
 - 1번 핀에서 듀티 비 100%의 PWM 신호를 출력시킴. ($1024/1024=1$)

예제 1

```
#include <wiringPi.h>


int main()
{
    const int pwm_pin=1;

    wiringPiSetup();
    pinMode(pwm_pin, PWM_OUTPUT);

    while(1){
        for(int i=0; i<1024; ++i){
            pwmWrite(pwm_pin, i);
            delay(1);
        }
        for(int i=1024; i>0; --i){
            pwmWrite(pwm_pin, i);
            delay(1);
        }
    }

    return 0;
}
```

주의!



퀴즈

- PWM 출력 제어를 통해 아래 순서와 같이 LED의 밝기를 제어하라.
 - ① 듀티 비 0%의 PWM 신호를 1초간 출력함.
 - ② 듀티 비 50%의 PWM 신호를 1초간 출력함.
 - ③ 듀티 비 100%의 PWM 신호를 1초간 출력함.
 - ④ 이후에는 ①~③의 패턴을 반복함.

소프트웨어 방식의 PWM 출력 제어 함수

- PWM 출력 전용 핀이 아닌 일반 GPIO 핀에서 PWM 신호를 소프트웨어 방식으로 만들고 출력시킴.
- `int softPwmCreate(int pin, int initialValue, int pwmRange);`
 - pin을 소프트웨어 PWM 핀으로 초기화함.
 - initialValue는 초기 펄스 폭이고, pwmRange는 주기임.
- `int softPwmWrite(int pin, int value);`
 - `softPwmCreate()` 함수로 초기화 시킨 pin에 `value/pwmRange`의 듀티 비를 갖는 PWM 신호를 출력시킴.
 - value에는 0~pwmRange의 값을 넣음.
- 주의 사항
 - `#include <softPwm.h>` // 코드에 추가

함수 사용 예

- 사용 예 1

- `softPwmCreate(25, 50, 100);`
 - 25번 핀을 소프트웨어 PWM 핀으로 설정함.
 - 초기 펄스 폭을 50으로, 주기를 100으로 설정함. 따라서 듀티 비 50%의 PWM 신호가 즉시 출력됨. ($50/100=0.5$)
- `softPwmWrite(25, 25);`
 - 25번 핀에서 출력되는 PWM 신호의 듀티 비를 25%로 변경함.

- 사용 예 2

- `softPwmCreate(29, 0, 100);`
- `softPwmWrite(29, 50);`
 - 29번 핀에서 출력하는 PWM 신호의 듀티 비를 50%로 변경함.

- 사용 예 3

- `softPwmCreate(28, 0, 1024);`
- `softPwmWrite(28, 512);`
 - 28번 핀에서 출력하는 PWM 신호의 듀티 비를 50%로 변경함.

예제 2

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

int main()
{
    const int soft_pwm_pin=25;

    wiringPiSetup();
    softPwmCreate(soft_pwm_pin, 0, 100);

    while(1){
        for(int i=0; i<100; ++i){
            softPwmWrite(soft_pwm_pin, i);
            delay(10);
        }
        for(int i=100; i>0; --i){
            softPwmWrite(soft_pwm_pin, i);
            delay(10);
        }
    }

    return 0;
}
```

질문

Q&A