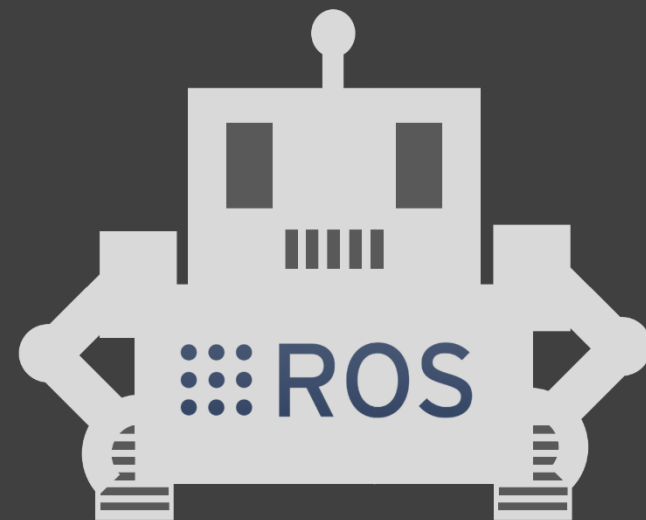


라즈베리 파이 기초

Chapter 1. LED & 스위치 제어

구선생 로보틱스



강의 자료 다운로드



라즈베리 파이 강의 노트

<https://github.com/DoveSensei/RaspberryPiNote>

라즈베리 파이란 무엇인가?

라즈베리 파이란 무엇인가?

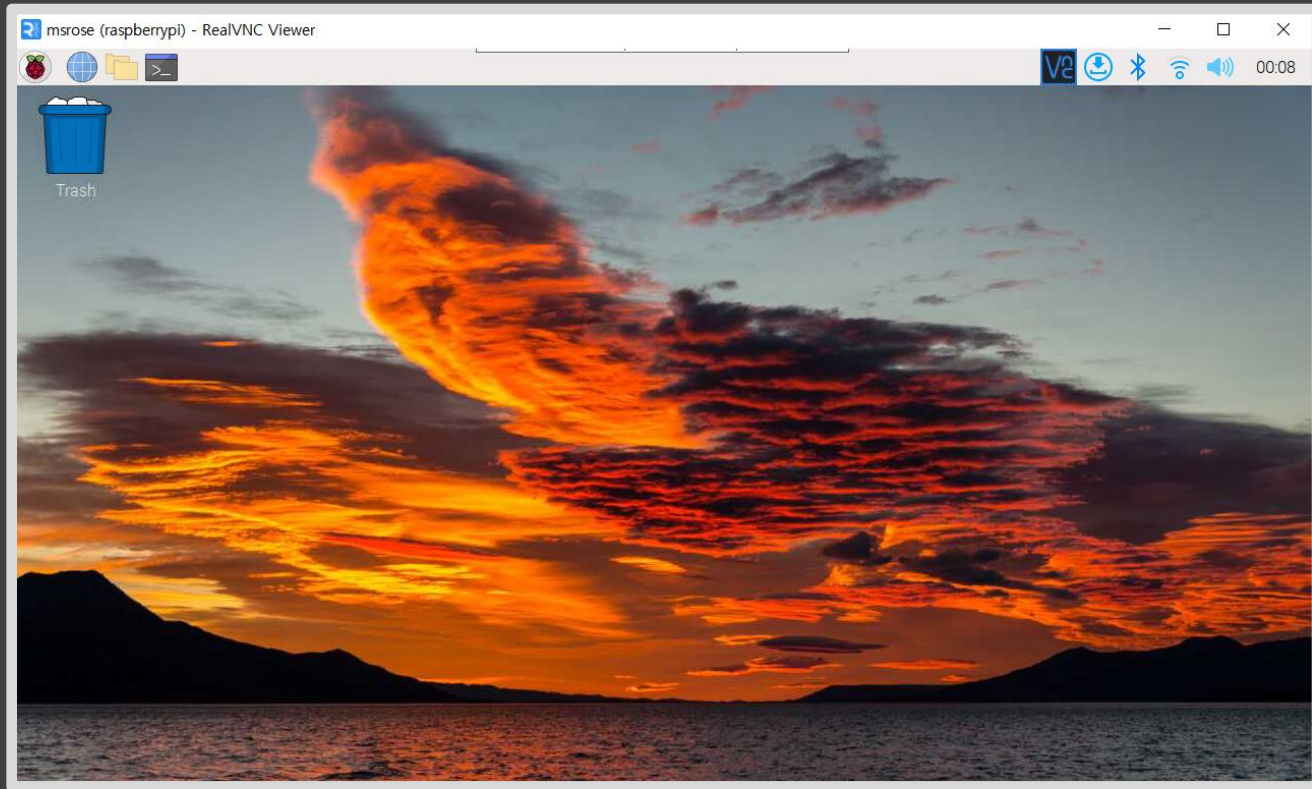
라즈베리 파이 개요



- 저렴한 가격과 작은 크기의 싱글 보드 컴퓨터
- 리눅스 기반 운영체제
- ARM CPU

라즈베리 파이란 무엇인가?

원격 접속 방법



RealVNC Viewer를 사용하여 접속



라즈베리파이-원격접속-방법.PDF

문서 참고

Linux 기초 명령어

Linux 기초 명령어

간단한 Python 소스코드 작성

1) Python 소스코드 생성

```
nano my_first_python.py
```

2) 아래 내용 작성 후 저장

```
1 print("Hello World!!")  
2
```

3) 실행

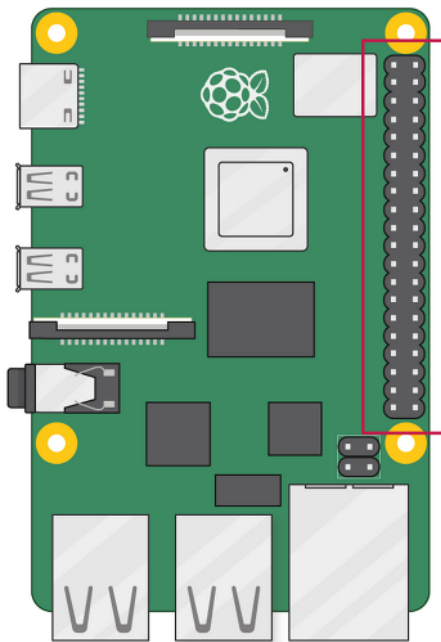
```
python3 my_first_python.py
```

```
msrose@raspberrypi:~ $ python3 my_first_python.py  
Hello World!!
```

LED & 스위치 제어

LED & 스위치 제어

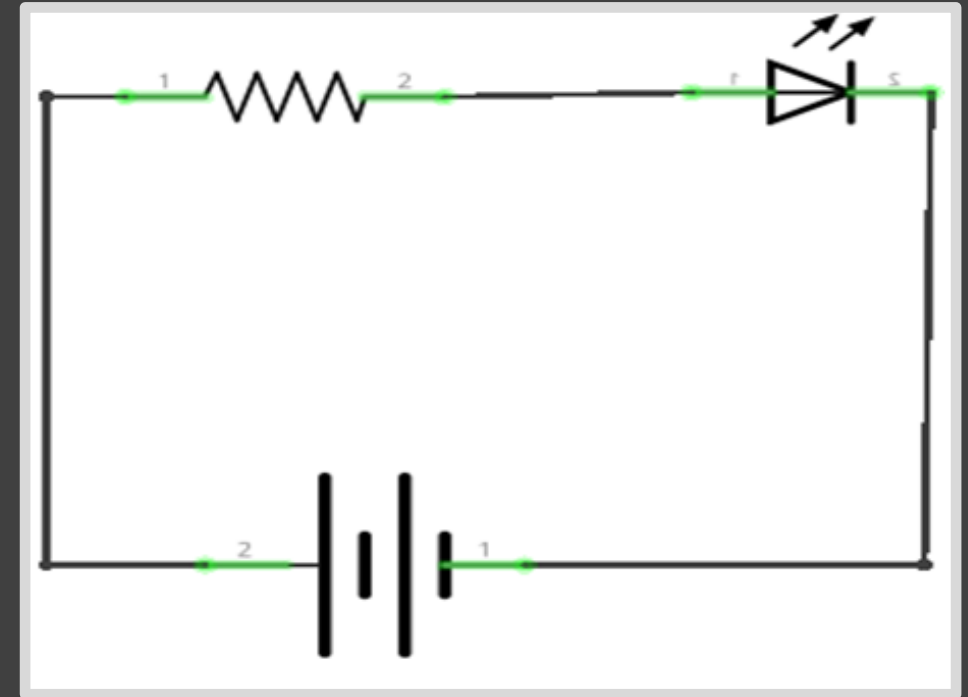
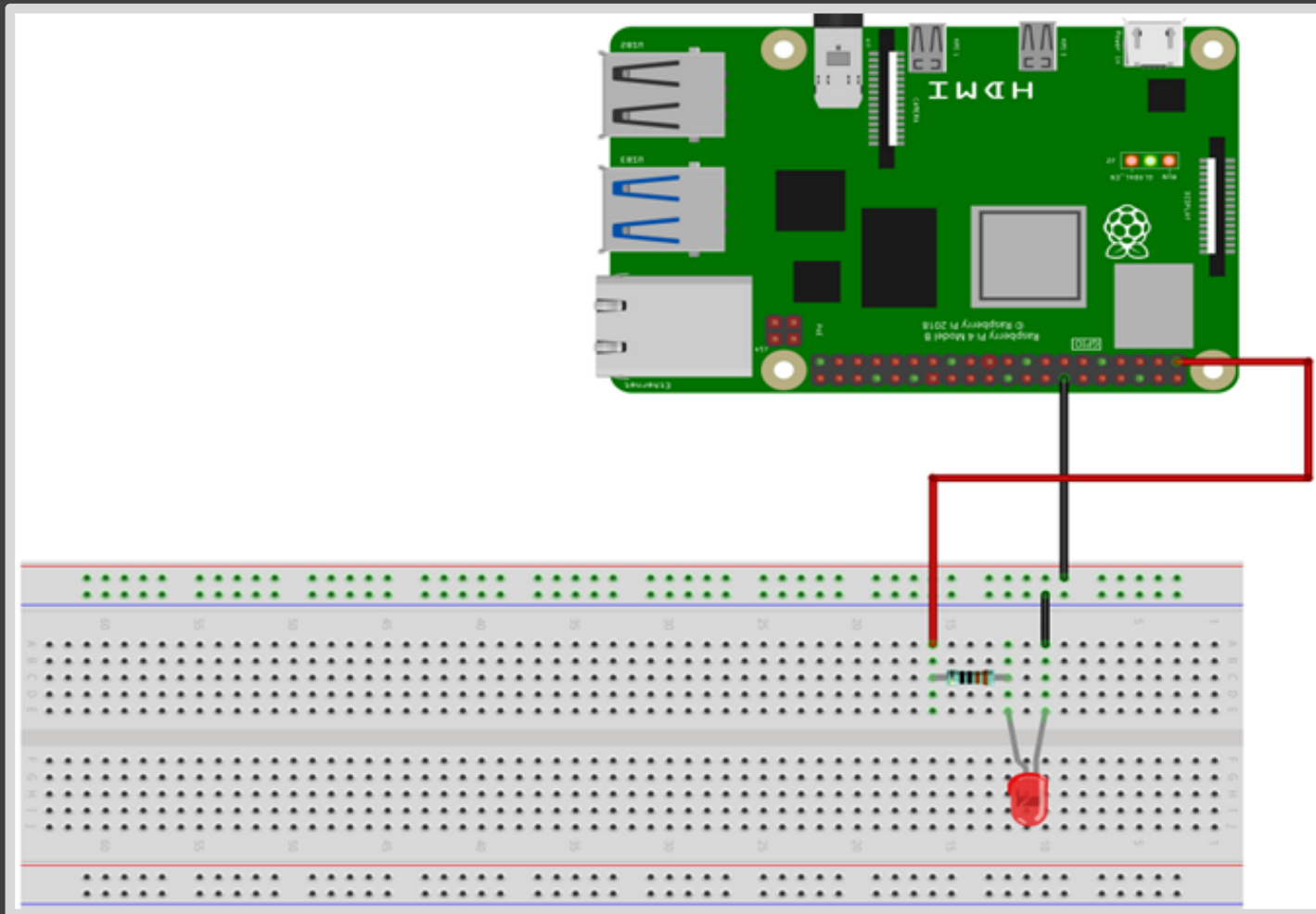
라즈베리 파이 PIN Map



3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)

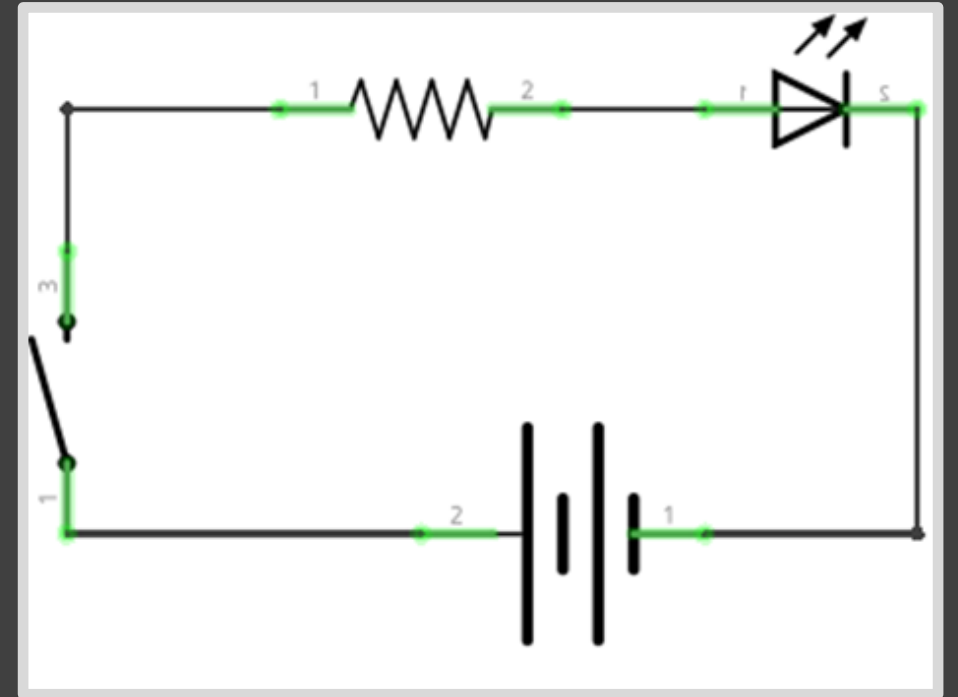
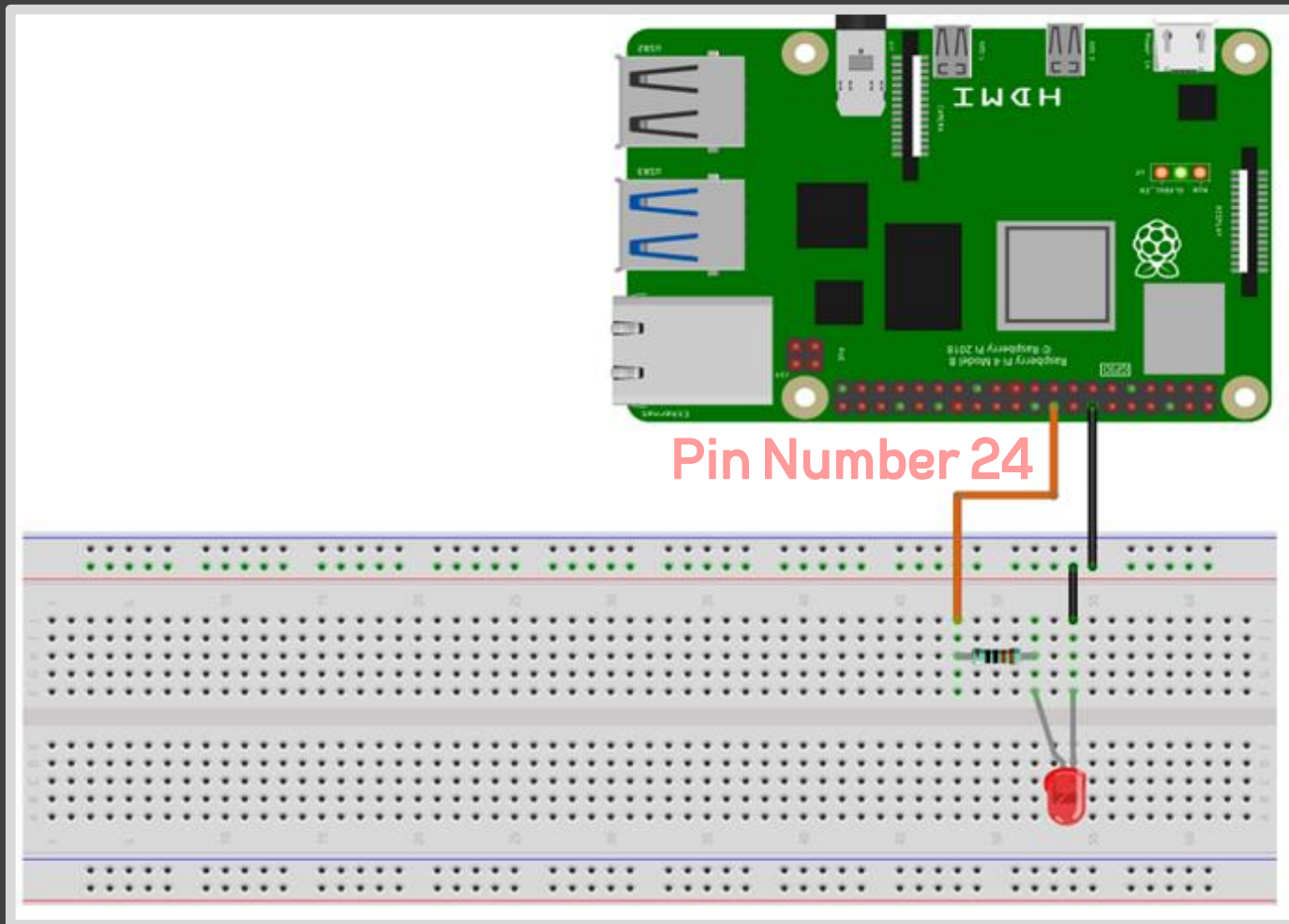
LED & 스위치 제어

LED 제어 기초 회로도



LED & 스위치 제어

LED 제어 회로도



LED & 스위치 제어

LED 제어 원리

아래 경로의 파일을 이용하여 Gpio를 제어

```
msrose@raspberrypi: /sys/class/gpio $ ls  
export  gpiochip0  gpiochip504  unexport
```

기능	명령어
LED 메모리 할당	echo 24> /sys/class/gpio/export
LED 모드 설정	echo out > /sys/class/gpio/gpio24/direction
LED 켜기	echo 1 > /sys/class/gpio/gpio24/value
LED 메모리 할당 해제	echo 24 > /sys/class/gpio/unexport

LED & 스위치 제어

LED 제어 Python 코드

```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BCM) # GPIO 모드 초기화
5  GPIO.setup(24,GPIO.OUT) # 24번 핀 OUT 모드 설정 / 메모리 할당
6
7  GPIO.output(24,GPIO.HIGH) # 24번 핀 1
8  time.sleep(1.0) # 1초 대기
9  GPIO.output(24,GPIO.LOW) # 24번 핀 0
10
11 GPIO.cleanup() # 모든 GPIO 메모리 할당 해제
```

LED & 스위치 제어

PWM 제어

5V

0V

5V

0V

5V

0V

5V

0V

5V

0V

PWM(Pulse Width Modulation) 제어는 펄스의 폭을 변화시켜 디지털 신호로 아날로그 같은 연속적인 제어를 할 수 있는 방법

0% Duty Cycle

25% Duty Cycle

50% Duty Cycle

75% Duty Cycle

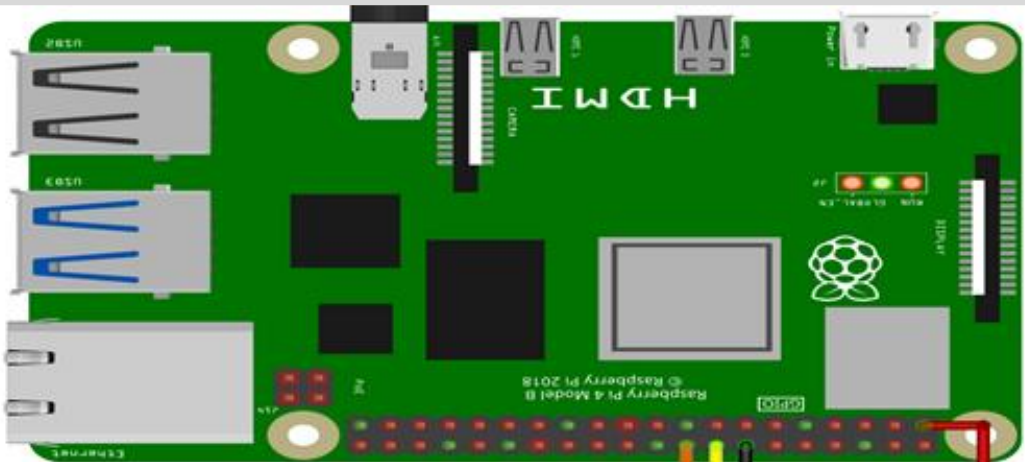
100% Duty Cycle

LED & 스위치 제어

PWM 제어 Python 코드

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM) # GPIO 모드 초기화
5 GPIO.setup(24, GPIO.OUT) # 24번 핀 OUT 모드 설정 / 메모리 할당
6
7 ledWhitePwm = GPIO.PWM(24, 500) # 24번 핀 PWM 500Hz 설정
8 ledWhitePwm.start(0) # 24번 핀 PWM LOW
9
10 ledWhitePwm.ChangeDutyCycle(0) # 24번 핀 PWM 시작 0% Duty 설정
11 time.sleep(1.0) # 1초 대기
12 ledWhitePwm.ChangeDutyCycle(30) # 24번 핀 PWM 시작 30% Duty 설정
13 time.sleep(1.0) # 1초 대기
14 ledWhitePwm.ChangeDutyCycle(60) # 24번 핀 PWM 시작 60% Duty 설정
15 time.sleep(1.0) # 1초 대기
16 ledWhitePwm.ChangeDutyCycle(100) # 24번 핀 PWM 시작 100% Duty 설정
17
18 GPIO.cleanup() # 모든 GPIO 메모리 할당 해제
```


스위치 제어 회로도



LED & 스위치 제어

스위치 제어 소스코드

```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BCM) # GPIO 모드 초기화
5  GPIO.setup(24, GPIO.OUT) # 24번 핀 OUT 모드 설정 / 메모리 할당
6  GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) # 23번 핀 IN 모드 설정 / 메모리 할당
7
8  try:
9      while True:
10         switch_state = GPIO.input(23) # 23번 스위치 값 읽기
11         if switch_state:
12             GPIO.output(24, GPIO.HIGH) # 24번 핀 1
13         else:
14             GPIO.output(24, GPIO.LOW) # 24번 핀 0
15
16     # Ctrl+C 입력시 프로그램 종료
17 except KeyboardInterrupt:
18     pass
19
20
21 GPIO.cleanup()
```

LED & 스위치 제어

작품 제작하기

작품 01, 05 제작하기

감사합니다

구선생 로보틱스

