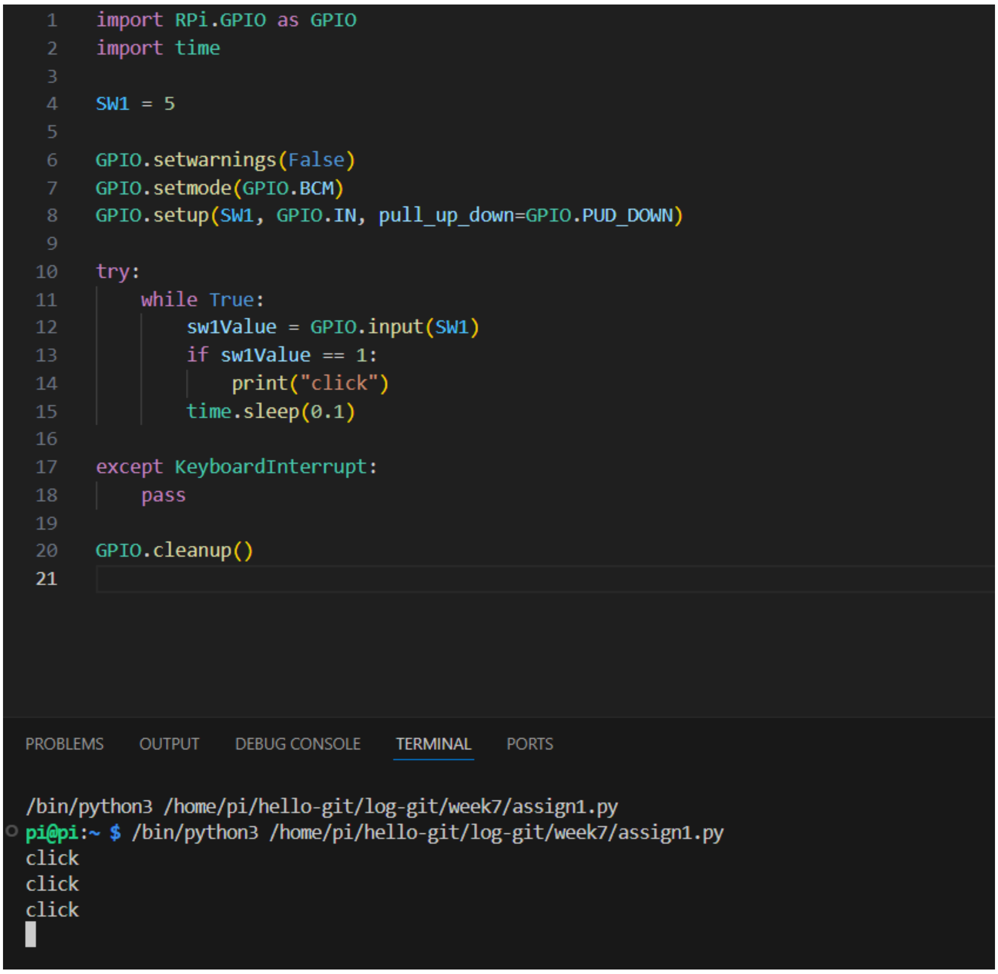
임베디드 응용 및 실습 7주차 과제

2022180034 김한별

1. 버튼 입력 받기 구현

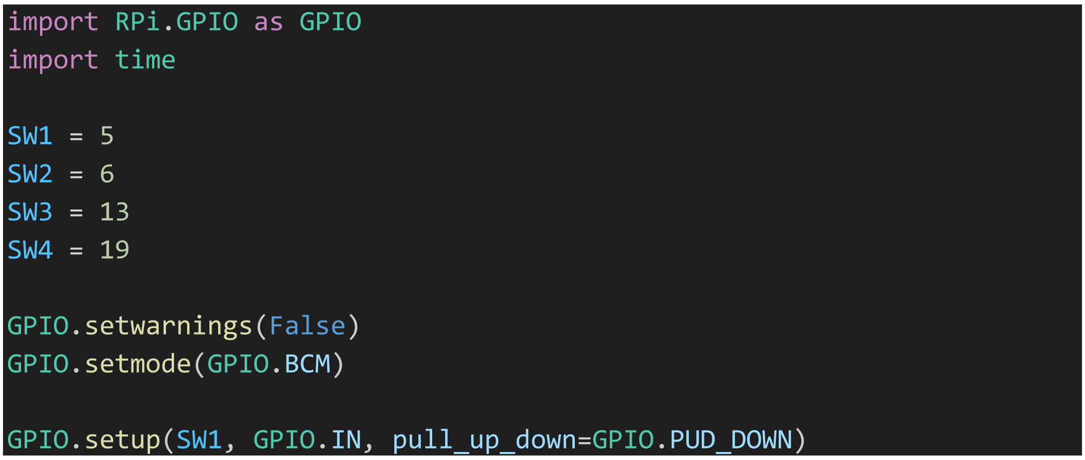
   1) 스위치 눌렸을 때만 화면에 "click"이 표기되도록 변경



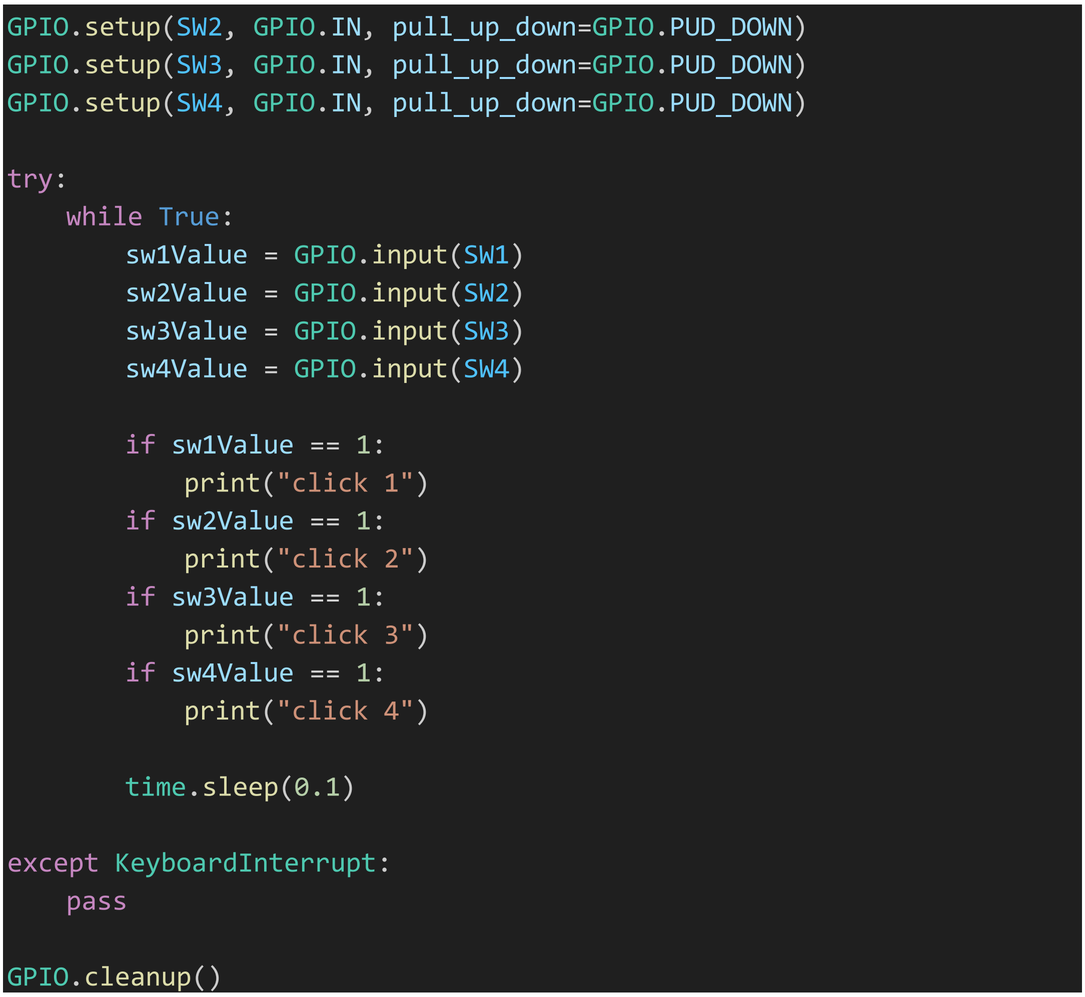
```python
import RPi.GPIO as GPIO
import time

SW1 = 5

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

try:
    while True:
        sw1Value = GPIO.input(SW1)
        if sw1Value == 1:
            print("click")
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()

```

```
/bin/python3 /home/pi/hello-git/log-git/week7/assign1.py
pi@pi:~ $ /bin/python3 /home/pi/hello-git/log-git/week7/assign1.py
click
click
click
```

   2) 몇번 스위치 눌렸는지 확인 가능하도록 "click X" 화면 출력

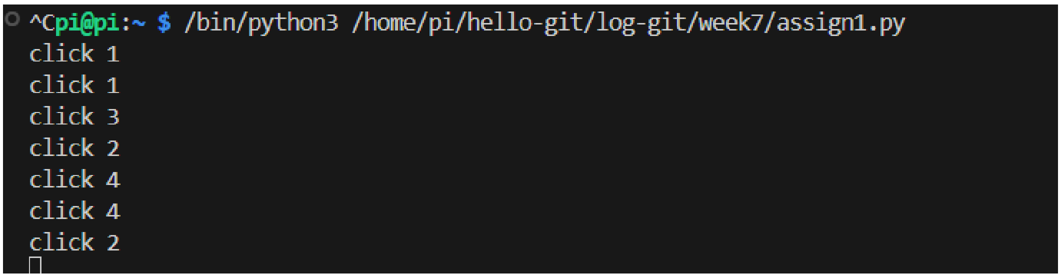


```python
import RPi.GPIO as GPIO
import time

SW1 = 5
SW2 = 6
SW3 = 13
SW4 = 19

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```



```
GPIO.setup(SW2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

try:
    while True:
        sw1Value = GPIO.input(SW1)
        sw2Value = GPIO.input(SW2)
        sw3Value = GPIO.input(SW3)
        sw4Value = GPIO.input(SW4)

        if sw1Value == 1:
            print("click 1")
        if sw2Value == 1:
            print("click 2")
        if sw3Value == 1:
            print("click 3")
        if sw4Value == 1:
            print("click 4")

        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

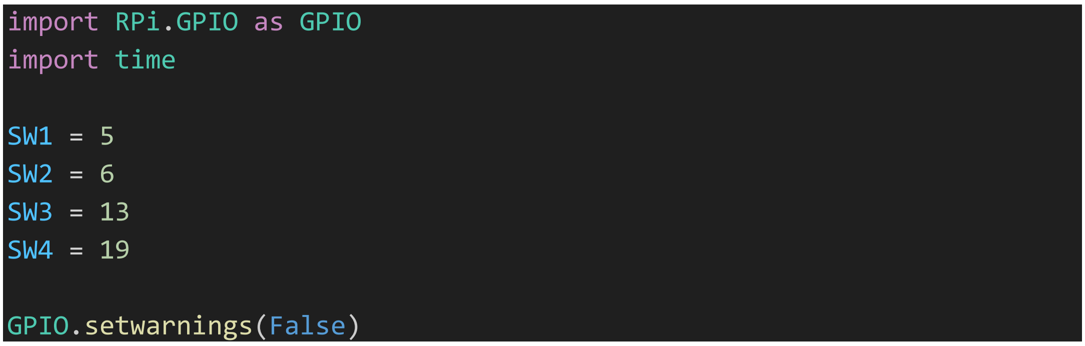
- 결과



```
^Cpi@pi:~ $ /bin/python3 /home/pi/hello-git/log-git/week7/assign1.py
click 1
click 1
click 3
click 2
click 4
click 4
click 2
```

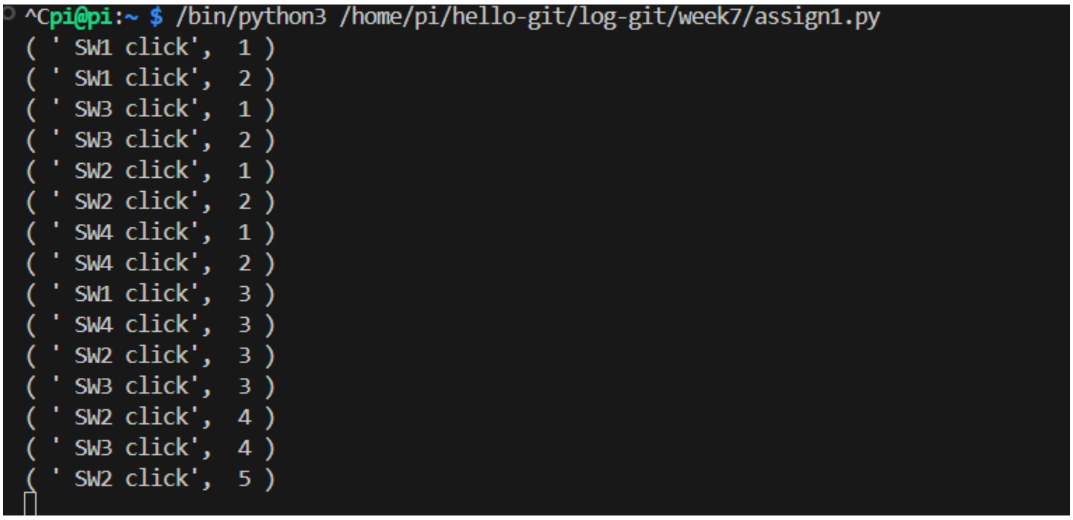
3) 0 -> 1 인 경우만 동작 : 1 일 때만 click 을 프린트할 수 있도록 이미 구현

4) 4 개의 스위치 입력 받기, 리스트 활용해 GPIO 전/후 값 저장



```
import RPi.GPIO as GPIO
import time

SW1 = 5
SW2 = 6
SW3 = 13
SW4 = 19

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

switch_names = ['SW1', 'SW2', 'SW3', 'SW4']
switch_pins = [SW1, SW2, SW3, SW4]

prev_values = [0, 0, 0, 0]
click_counts = [0, 0, 0, 0]

try:
    while True:
        for i in range(4):
            current_value = GPIO.input(switch_pins[i])
            if prev_values[i] == 0 and current_value == 1:
                click_counts[i] += 1
                print("(", "'", switch_names[i], "click', ",
click_counts[i], ")")
            prev_values[i] = current_value

        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

- 결과



```
^Cpi@pi:~ $ /bin/python3 /home/pi/hello-git/log-git/week7/assign1.py
 ( ' SW1 click',  1 )
 ( ' SW1 click',  2 )
 ( ' SW3 click',  1 )
 ( ' SW3 click',  2 )
 ( ' SW2 click',  1 )
 ( ' SW2 click',  2 )
 ( ' SW4 click',  1 )
 ( ' SW4 click',  2 )
 ( ' SW1 click',  3 )
 ( ' SW4 click',  3 )
 ( ' SW2 click',  3 )
 ( ' SW3 click',  3 )
 ( ' SW2 click',  4 )
 ( ' SW3 click',  4 )
 ( ' SW2 click',  5 )
```

2. 부저 음계 출력 구현 -> 결과 동영상 참고

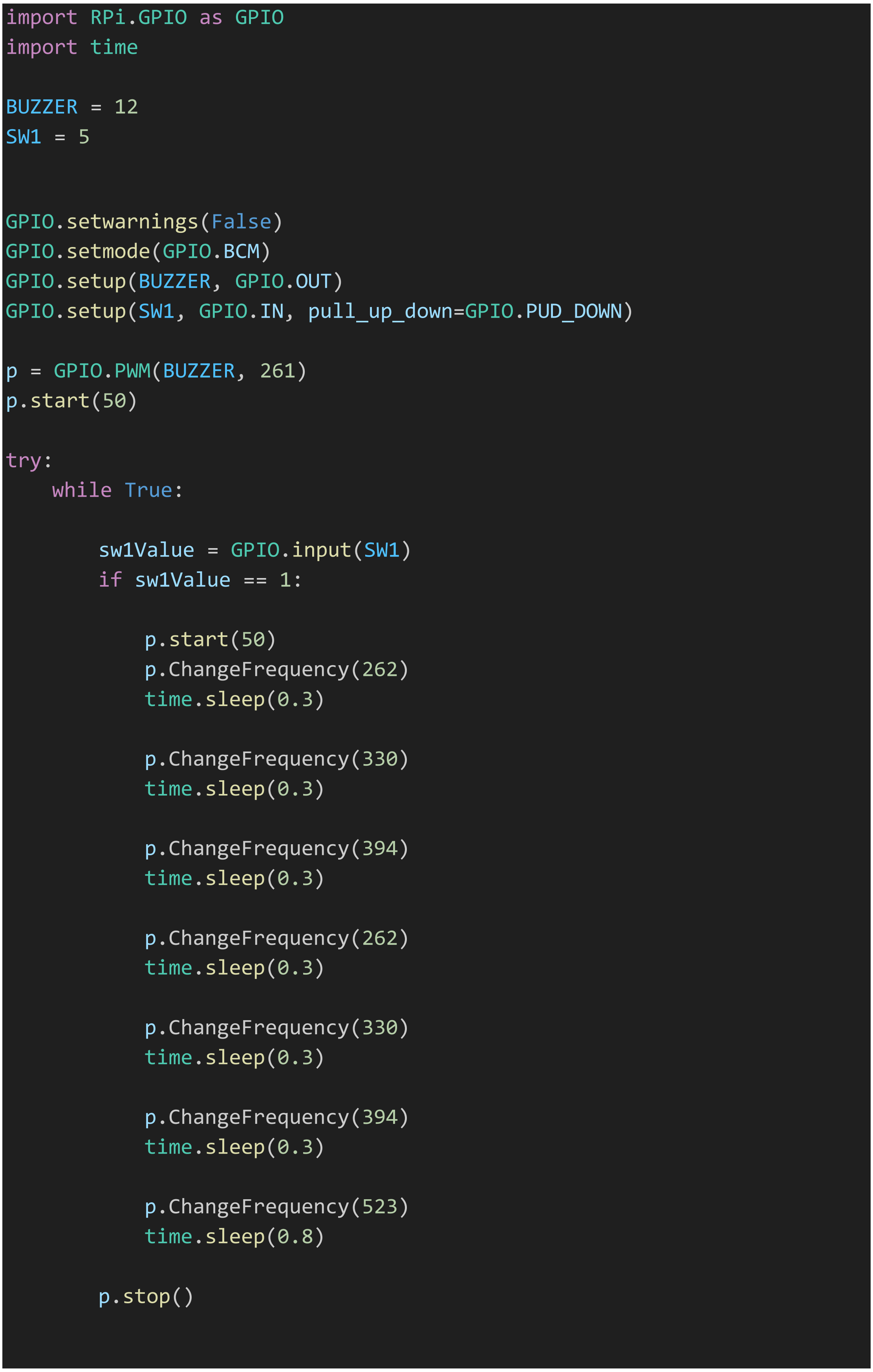
(1) “도레미파솔라시도” 음계 출력

```
import RPi.GPIO as GPIO
import time

BUZZER = 12

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)

p = GPIO.PWM(BUZZER, 261)
p.start(50)

try:
    while True:
        p.start(50)
        p.ChangeFrequency(262)
        time.sleep(1.0)
        p.ChangeFrequency(292)
        time.sleep(1.0)
        p.ChangeFrequency(330)
        time.sleep(1.0)
        p.ChangeFrequency(349)
        time.sleep(1.0)
        p.ChangeFrequency(394)
        time.sleep(1.0)
        p.ChangeFrequency(440)
        time.sleep(1.0)
        p.ChangeFrequency(494)
        time.sleep(1.0)
        p.ChangeFrequency(523)
        time.sleep(1.0)

        p.stop()
        time.sleep(1.0)

except KeyboardInterrupt:
    pass

p.stop()
GPIO.cleanup()
```

(2) 나만의 경적 소리 구현

```
import RPi.GPIO as GPIO
import time

BUZZER = 12

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)

p = GPIO.PWM(BUZZER, 261)
p.start(50)

try:
    while True:
        p.start(50)
        p.ChangeFrequency(262)
        time.sleep(0.3)

        p.ChangeFrequency(330)
        time.sleep(0.3)

        p.ChangeFrequency(394)
        time.sleep(0.3)

        p.ChangeFrequency(262)
        time.sleep(0.3)

        p.ChangeFrequency(330)
        time.sleep(0.3)

        p.ChangeFrequency(394)
        time.sleep(0.3)

        p.ChangeFrequency(523)
        time.sleep(0.8)

        p.stop()
        time.sleep(1.0)


except KeyboardInterrupt:
    pass

p.stop()
GPIO.cleanup()
```

(3) 스위치를 한번 누르면 경적 소리가 나도록 구현



```
import RPi.GPIO as GPIO
import time

BUZZER = 12
SW1 = 5


GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)
GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

p = GPIO.PWM(BUZZER, 261)
p.start(50)

try:
    while True:

        sw1Value = GPIO.input(SW1)
        if sw1Value == 1:

            p.start(50)
            p.ChangeFrequency(262)
            time.sleep(0.3)

            p.ChangeFrequency(330)
            time.sleep(0.3)

            p.ChangeFrequency(394)
            time.sleep(0.3)

            p.ChangeFrequency(262)
            time.sleep(0.3)

            p.ChangeFrequency(330)
            time.sleep(0.3)

            p.ChangeFrequency(394)
            time.sleep(0.3)

            p.ChangeFrequency(523)
            time.sleep(0.8)

        p.stop()
```

```
except KeyboardInterrupt:
    pass

p.stop()
GPIO.cleanup()
```

(4) 스위치 4 개를 사용해 나만의 음악을 연주

```
import RPi.GPIO as GPIO
import time

BUZZER = 12
SW1 = 5
SW2 = 6
SW3 = 13
SW4 = 19

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)
GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

p = GPIO.PWM(BUZZER, 261)

try:
    while True:
        sw1Value = GPIO.input(SW1)
        sw2Value = GPIO.input(SW2)
        sw3Value = GPIO.input(SW3)
        sw4Value = GPIO.input(SW4)

        if sw1Value == 1:
            p.start(50)
            p.ChangeFrequency(292)
            time.sleep(0.3)

        elif sw2Value == 1:
            p.start(50)
            p.ChangeFrequency(330)
            time.sleep(0.3)

        elif sw3Value == 1:
            p.start(50)
            p.ChangeFrequency(394)
            time.sleep(0.3)
        elif sw4Value == 1:
            p.start(50)
            p.ChangeFrequency(440)
            time.sleep(0.3)

        else:
            p.stop()

except KeyboardInterrupt:
    pass

p.stop()
GPIO.cleanup()
```

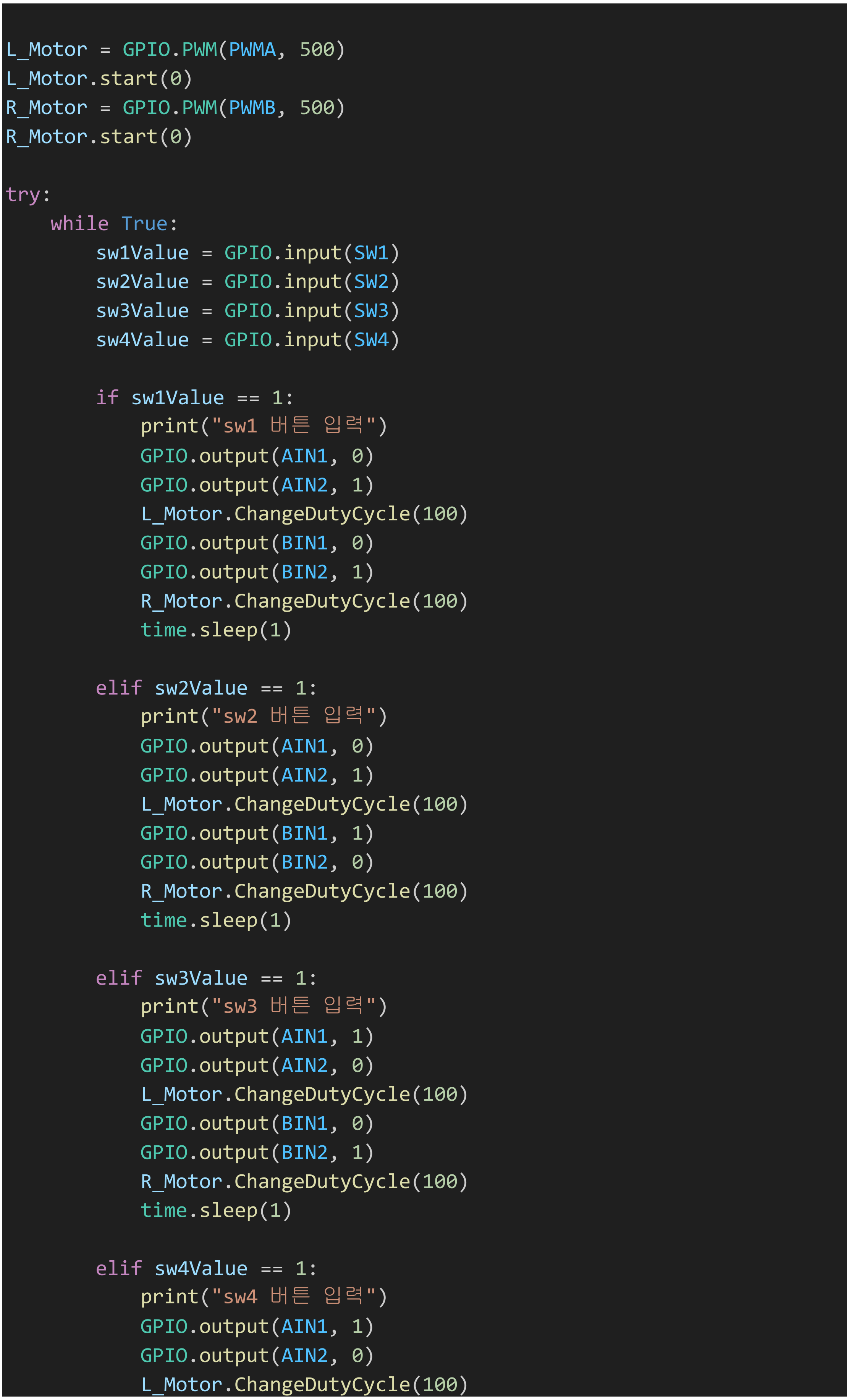
3. 자동차 움직이기 구현 -> 결과 동영상 참고

   (1) 오른쪽 모터부분의 코드를 추가해 정방향으로 50%로 동작 -> 정지 -> 동작 ...

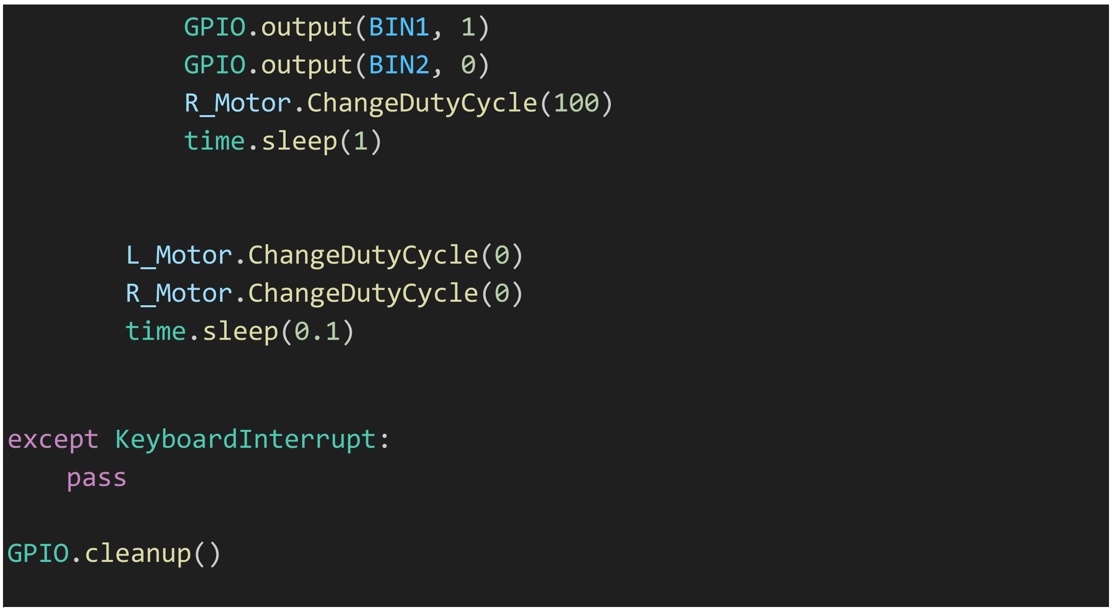
```
import RPi.GPIO as GPIO
import time

PWMA = 18
PWMB = 23
AIN1 = 22
AIN2 = 27
BIN1 = 25
BIN2 = 24

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(PWMA, GPIO.OUT)
GPIO.setup(PWMB, GPIO.OUT)
GPIO.setup(AIN1, GPIO.OUT)
GPIO.setup(AIN2, GPIO.OUT)
GPIO.setup(BIN1, GPIO.OUT)
GPIO.setup(BIN2, GPIO.OUT)

L_Motor = GPIO.PWM(PWMA, 500)
L_Motor.start(0)
R_Motor = GPIO.PWM(PWMB, 500)
R_Motor.start(0)

try:
    while True:

        GPIO.output(AIN1, 0)
        GPIO.output(AIN2, 1)
        L_Motor.ChangeDutyCycle(100)\
        GPIO.output(BIN1, 0)
        GPIO.output(BIN2, 1)
        R_Motor.ChangeDutyCycle(50)
        time.sleep(1.0)

        GPIO.output(AIN1, 0)
        GPIO.output(AIN2, 1)
        GPIO.output(BIN1, 0)
        GPIO.output(BIN2, 1)
        L_Motor.ChangeDutyCycle(0)
        R_Motor.ChangeDutyCycle(0)

        time.sleep(1.0)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

(2) 스위치를 입력 받아 자동차 조종하기 (SW1 앞, SW2 오른쪽, SW3 왼쪽, SW4 뒤)

```
import RPi.GPIO as GPIO
import time

PWMA = 18
PWMB = 23
AIN1 = 22
AIN2 = 27
BIN1 = 25
BIN2 = 24
SW1 = 5
SW2 = 6
SW3 = 13
SW4 = 19

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(PWMA, GPIO.OUT)
GPIO.setup(PWMB, GPIO.OUT)
GPIO.setup(AIN1, GPIO.OUT)
GPIO.setup(AIN2, GPIO.OUT)
GPIO.setup(BIN1, GPIO.OUT)
GPIO.setup(BIN2, GPIO.OUT)
GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SW4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

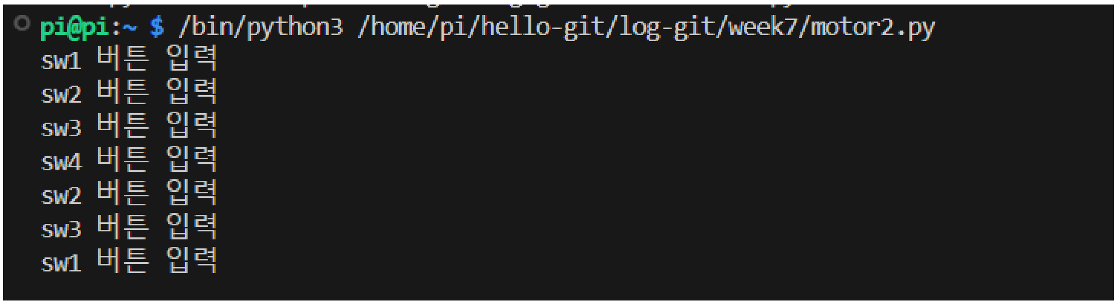


```python
L_Motor = GPIO.PWM(PWMA, 500)
L_Motor.start(0)
R_Motor = GPIO.PWM(PWMB, 500)
R_Motor.start(0)

try:
    while True:
        sw1Value = GPIO.input(SW1)
        sw2Value = GPIO.input(SW2)
        sw3Value = GPIO.input(SW3)
        sw4Value = GPIO.input(SW4)

        if sw1Value == 1:
            print("sw1 버튼 입력")
            GPIO.output(AIN1, 0)
            GPIO.output(AIN2, 1)
            L_Motor.ChangeDutyCycle(100)
            GPIO.output(BIN1, 0)
            GPIO.output(BIN2, 1)
            R_Motor.ChangeDutyCycle(100)
            time.sleep(1)

        elif sw2Value == 1:
            print("sw2 버튼 입력")
            GPIO.output(AIN1, 0)
            GPIO.output(AIN2, 1)
            L_Motor.ChangeDutyCycle(100)
            GPIO.output(BIN1, 1)
            GPIO.output(BIN2, 0)
            R_Motor.ChangeDutyCycle(100)
            time.sleep(1)

        elif sw3Value == 1:
            print("sw3 버튼 입력")
            GPIO.output(AIN1, 1)
            GPIO.output(AIN2, 0)
            L_Motor.ChangeDutyCycle(100)
            GPIO.output(BIN1, 0)
            GPIO.output(BIN2, 1)
            R_Motor.ChangeDutyCycle(100)
            time.sleep(1)

        elif sw4Value == 1:
            print("sw4 버튼 입력")
            GPIO.output(AIN1, 1)
            GPIO.output(AIN2, 0)
            L_Motor.ChangeDutyCycle(100)
```



```
            GPIO.output(BIN1, 1)
            GPIO.output(BIN2, 0)
            R_Motor.ChangeDutyCycle(100)
            time.sleep(1)


        L_Motor.ChangeDutyCycle(0)
        R_Motor.ChangeDutyCycle(0)
        time.sleep(0.1)


except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

- 프린트문 결과



```
○ pi@pi:~ $ /bin/python3 /home/pi/hello-git/log-git/week7/motor2.py
  sw1 버튼 입력
  sw2 버튼 입력
  sw3 버튼 입력
  sw4 버튼 입력
  sw2 버튼 입력
  sw3 버튼 입력
  sw1 버튼 입력
```