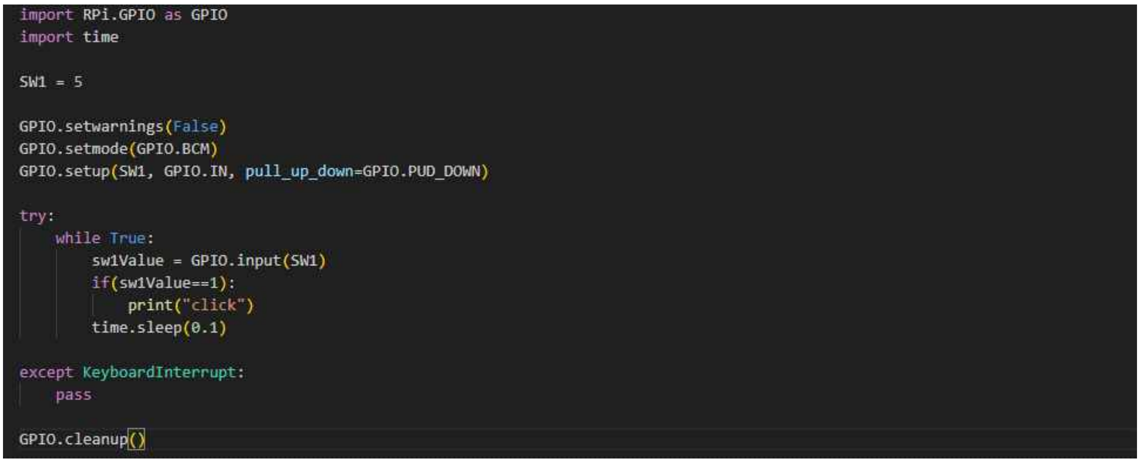
임베디드응용및실습

-7주차 과제

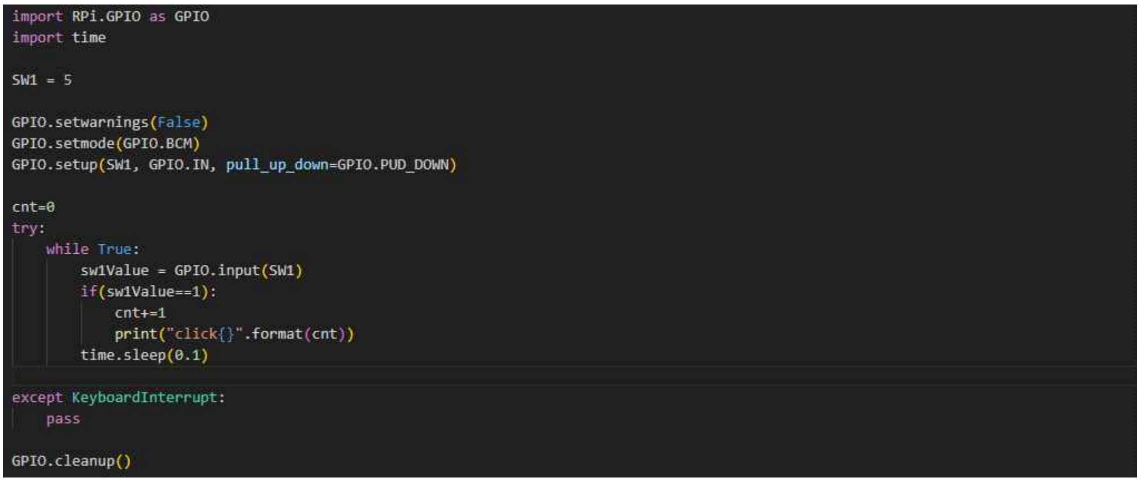
학과	정보통신공학과
학번	2020161010
이름	김기훈

1) 코드를 추가하여 스위치를 눌렀을 때만 화면에 “click”이 표기되도록 변경



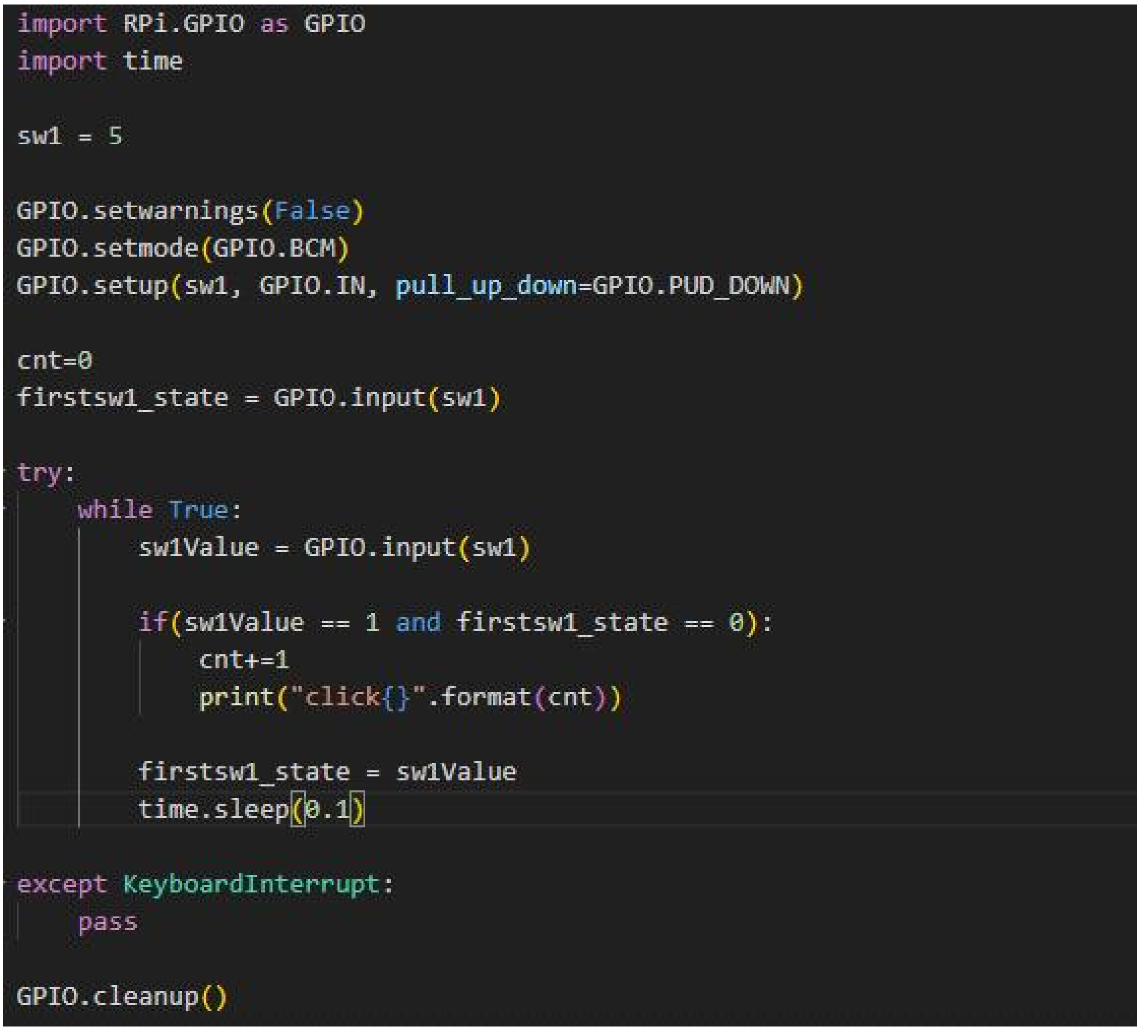
```
import RPi.GPIO as GPIO
import time

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()
```

2) 몇번 스위치가 눌렸는지 확인이 가능하도록 “click x” 등으로 화면 출력



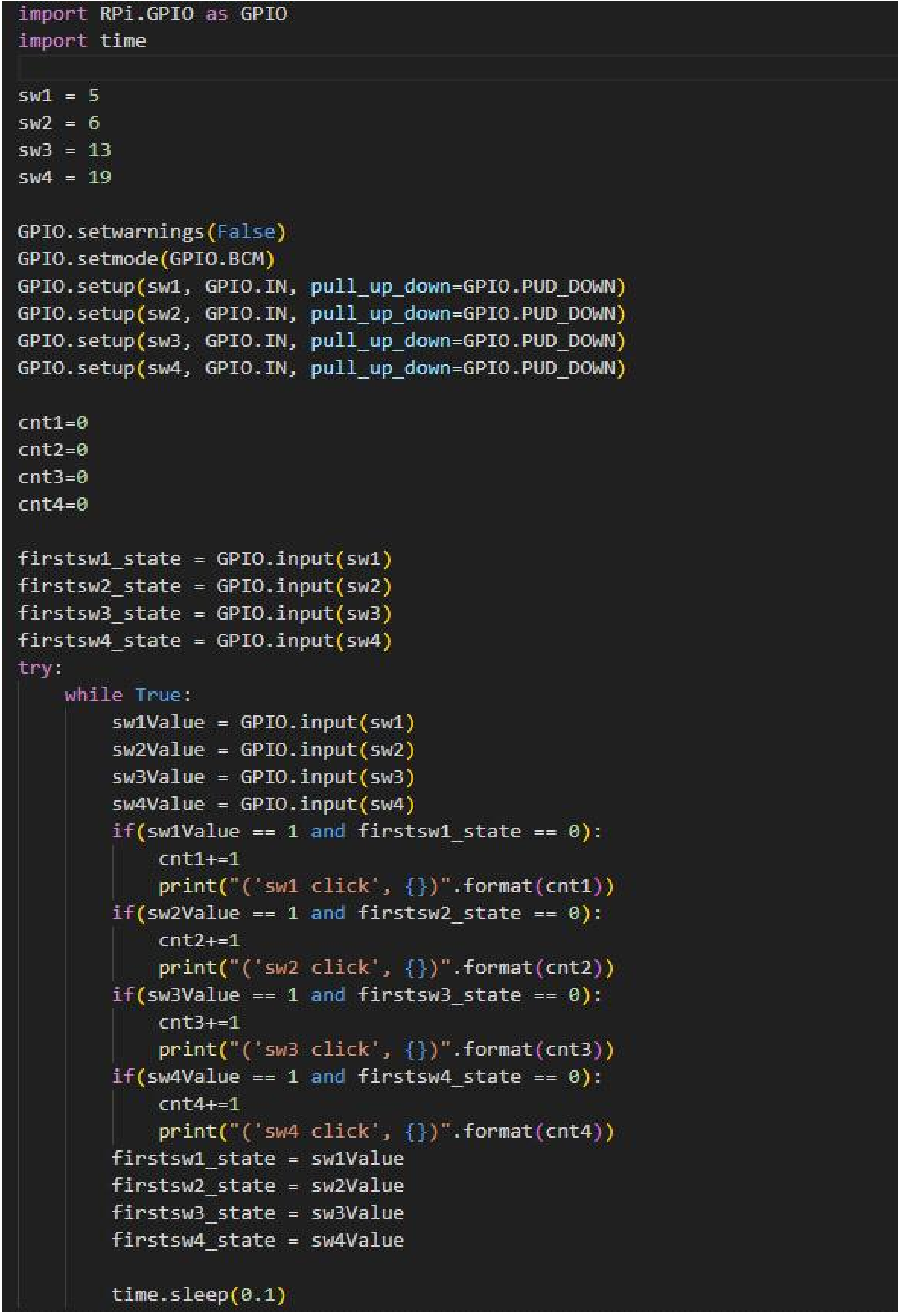
```
import RPi.GPIO as GPIO
import time

SW1 = 5

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

cnt=0
try:
    while True:
        sw1Value = GPIO.input(SW1)
        if(sw1Value==1):
            cnt+=1
            print("click{}".format(cnt))
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

3) 스위치를 눌렀을 때 0->1, 눌렀다 떼었을 때 1->0으로 값이 변경되므로 0->1인 경우만 동작되도록 변경

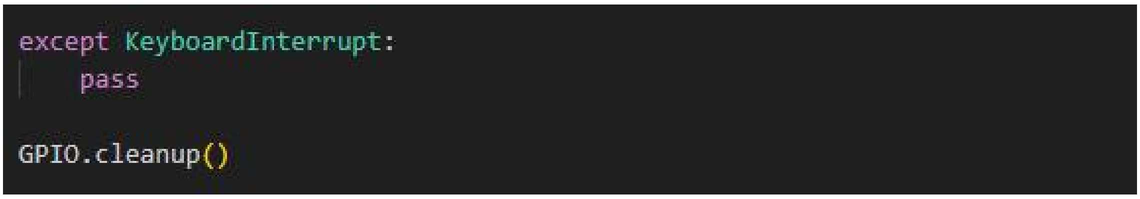


```
import RPi.GPIO as GPIO
import time

sw1 = 5

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(sw1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

cnt=0
firstsw1_state = GPIO.input(sw1)

try:
    while True:
        sw1Value = GPIO.input(sw1)

        if(sw1Value == 1 and firstsw1_state == 0):
            cnt+=1
            print("click{}".format(cnt))

        firstsw1_state = sw1Value
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

4) 4개의 스위치 입력을 받도록 해보자. 화면에 아래와 같이 출력되도록 한다. 단, 리스 트를 최대한 활용하여 GPIO 전/후 값을 저장한다.

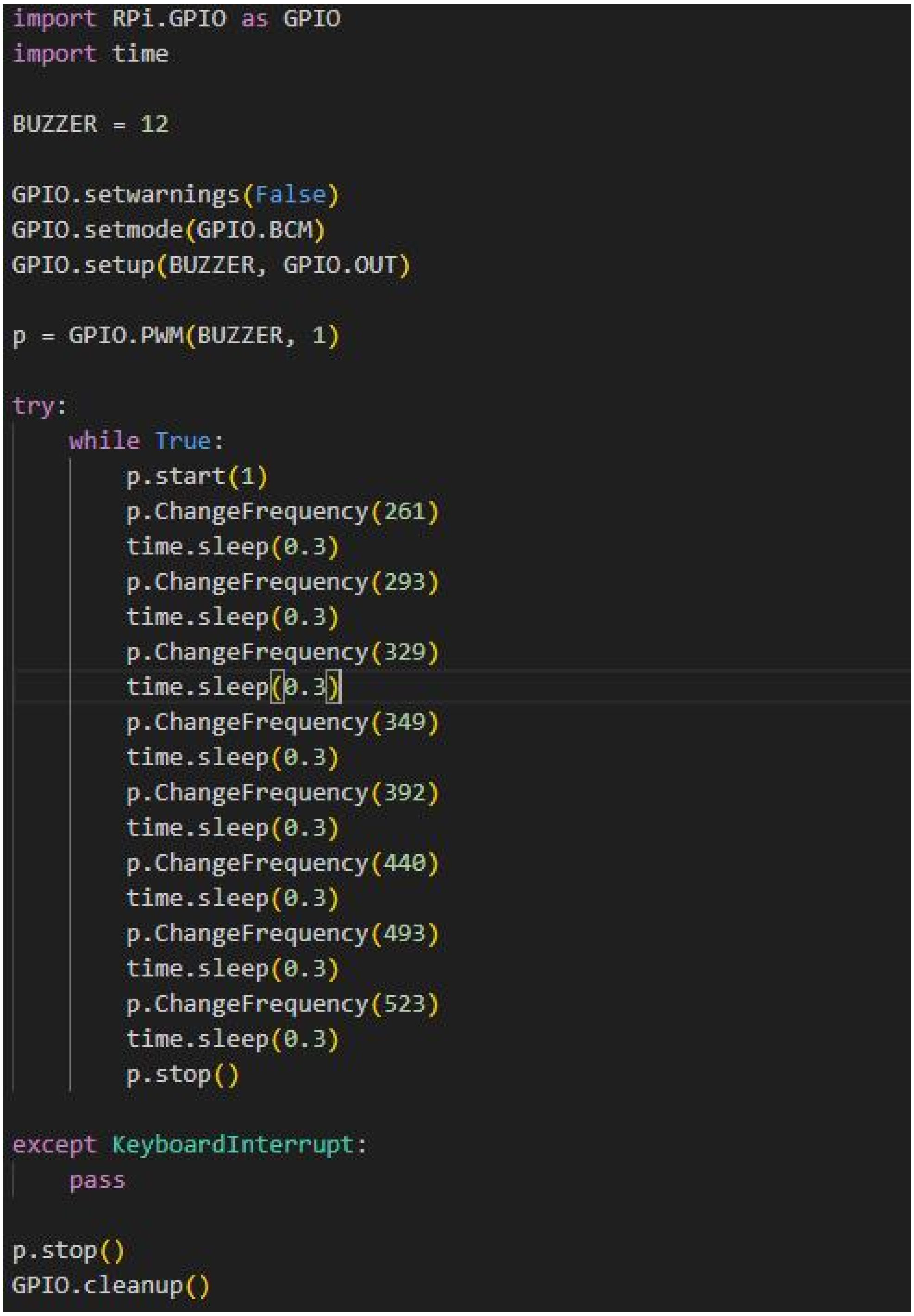


```python
import RPi.GPIO as GPIO
import time

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)

cnt1=0
cnt2=0
cnt3=0
cnt4=0

firstsw1_state = GPIO.input(sw1)
firstsw2_state = GPIO.input(sw2)
firstsw3_state = GPIO.input(sw3)
firstsw4_state = GPIO.input(sw4)
try:
    while True:
        sw1Value = GPIO.input(sw1)
        sw2Value = GPIO.input(sw2)
        sw3Value = GPIO.input(sw3)
        sw4Value = GPIO.input(sw4)
        if(sw1Value == 1 and firstsw1_state == 0):
            cnt1+=1
            print("('sw1 click', {})".format(cnt1))
        if(sw2Value == 1 and firstsw2_state == 0):
            cnt2+=1
            print("('sw2 click', {})".format(cnt2))
        if(sw3Value == 1 and firstsw3_state == 0):
            cnt3+=1
            print("('sw3 click', {})".format(cnt3))
        if(sw4Value == 1 and firstsw4_state == 0):
            cnt4+=1
            print("('sw4 click', {})".format(cnt4))
        firstsw1_state = sw1Value
        firstsw2_state = sw2Value
        firstsw3_state = sw3Value
        firstsw4_state = sw4Value

        time.sleep(0.1)
```



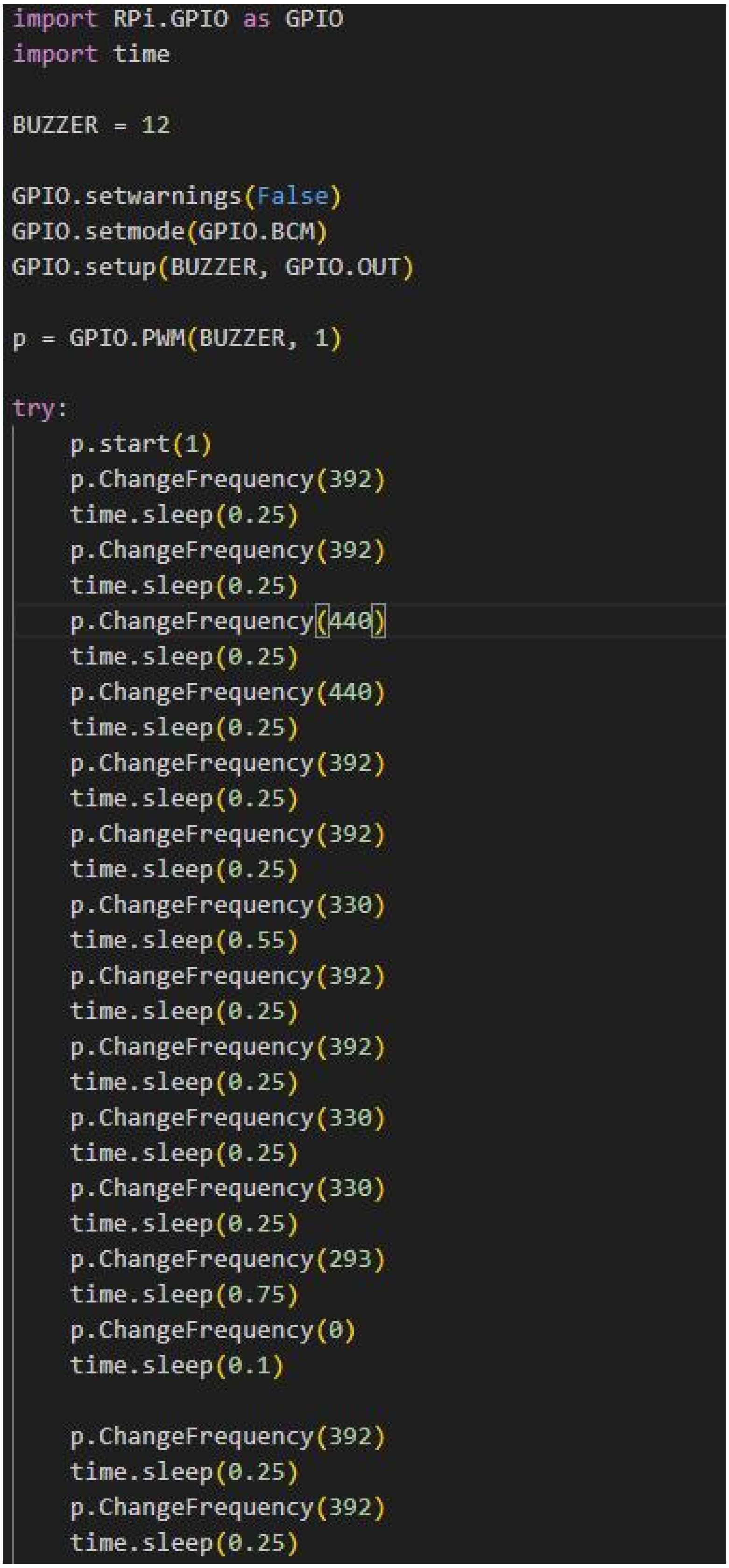
```python
except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

1) “도레미파솔라시도” 음계를 출력

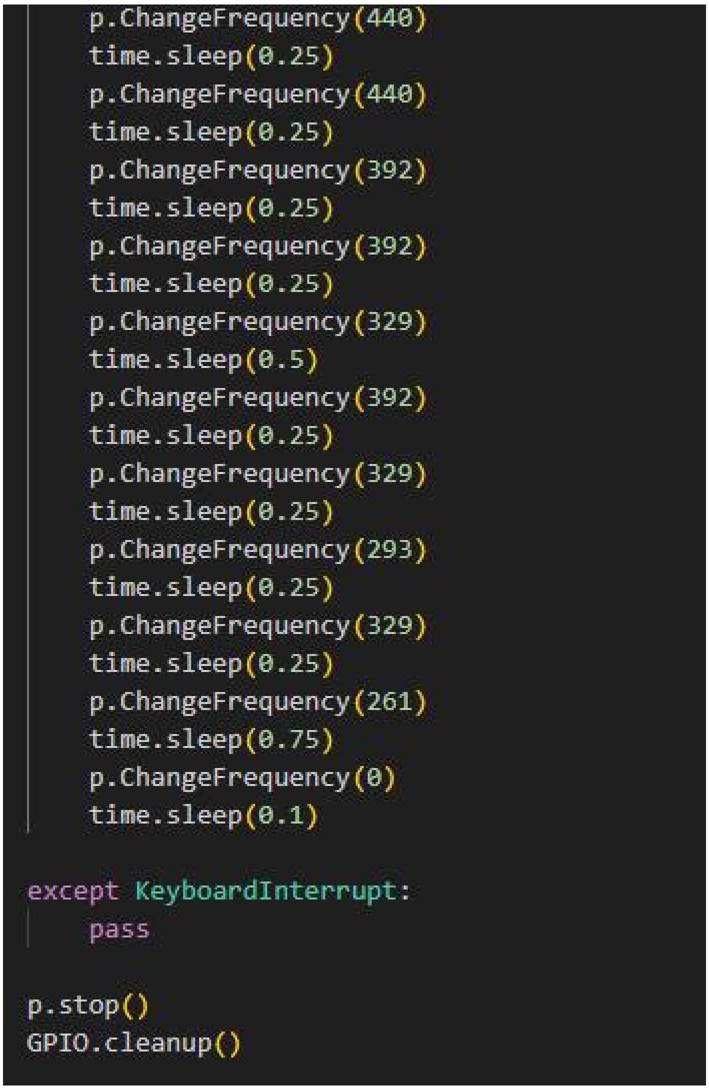


```python
import RPi.GPIO as GPIO
import time

BUZZER = 12

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

p = GPIO.PWM(BUZZER, 1)

try:
    while True:
        p.start(1)
        p.ChangeFrequency(261)
        time.sleep(0.3)
        p.ChangeFrequency(293)
        time.sleep(0.3)
        p.ChangeFrequency(329)
        time.sleep(0.3)
        p.ChangeFrequency(349)
        time.sleep(0.3)
        p.ChangeFrequency(392)
        time.sleep(0.3)
        p.ChangeFrequency(440)
        time.sleep(0.3)
        p.ChangeFrequency(493)
        time.sleep(0.3)
        p.ChangeFrequency(523)
        time.sleep(0.3)
        p.stop()

except KeyboardInterrupt:
    pass

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

2) 나만의 경적 소리 구현

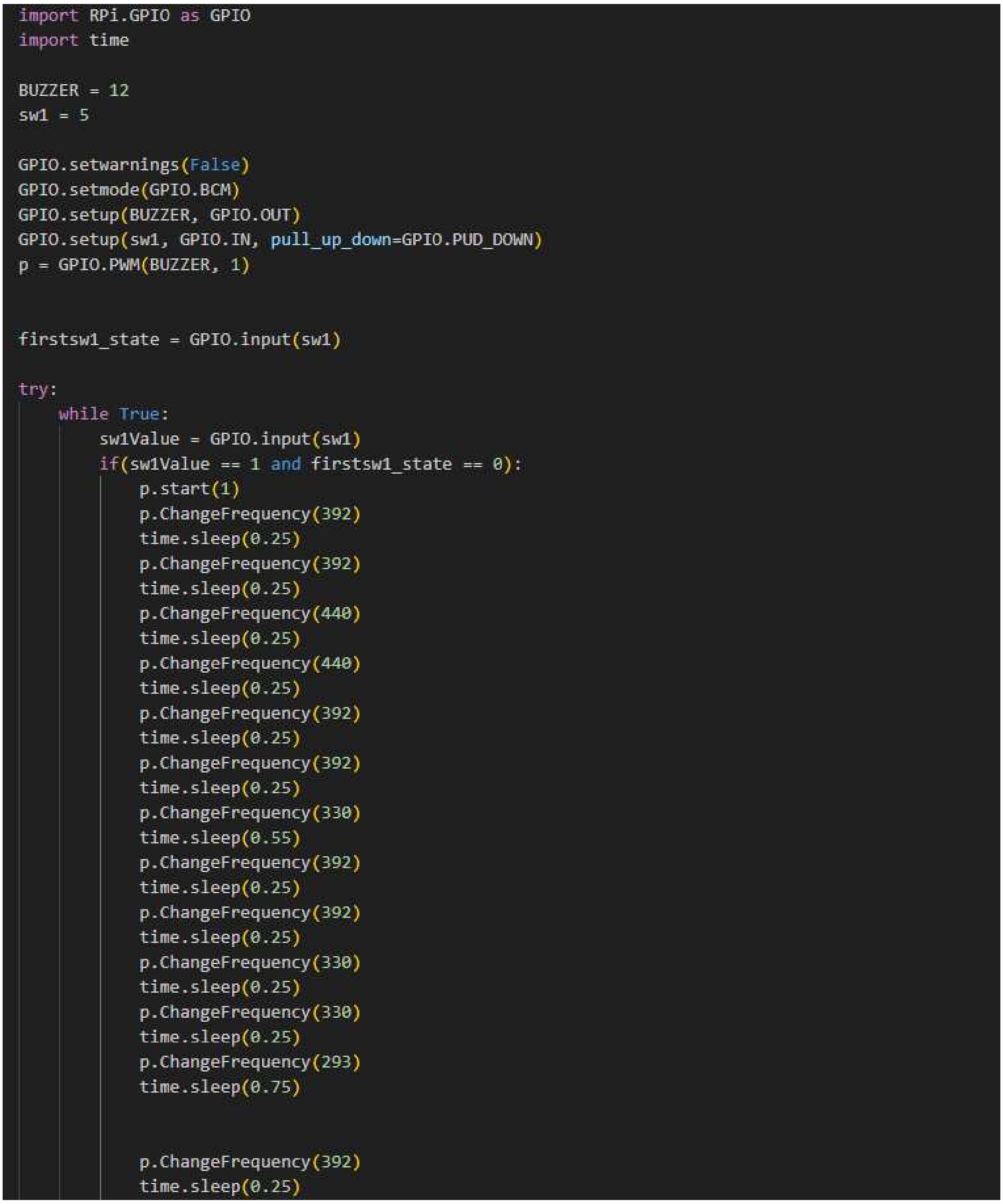


```
import RPi.GPIO as GPIO
import time

BUZZER = 12

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

p = GPIO.PWM(BUZZER, 1)

try:
    p.start(1)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(440)
    time.sleep(0.25)
    p.ChangeFrequency(440)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(330)
    time.sleep(0.55)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(330)
    time.sleep(0.25)
    p.ChangeFrequency(330)
    time.sleep(0.25)
    p.ChangeFrequency(293)
    time.sleep(0.75)
    p.ChangeFrequency(0)
    time.sleep(0.1)

    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
```

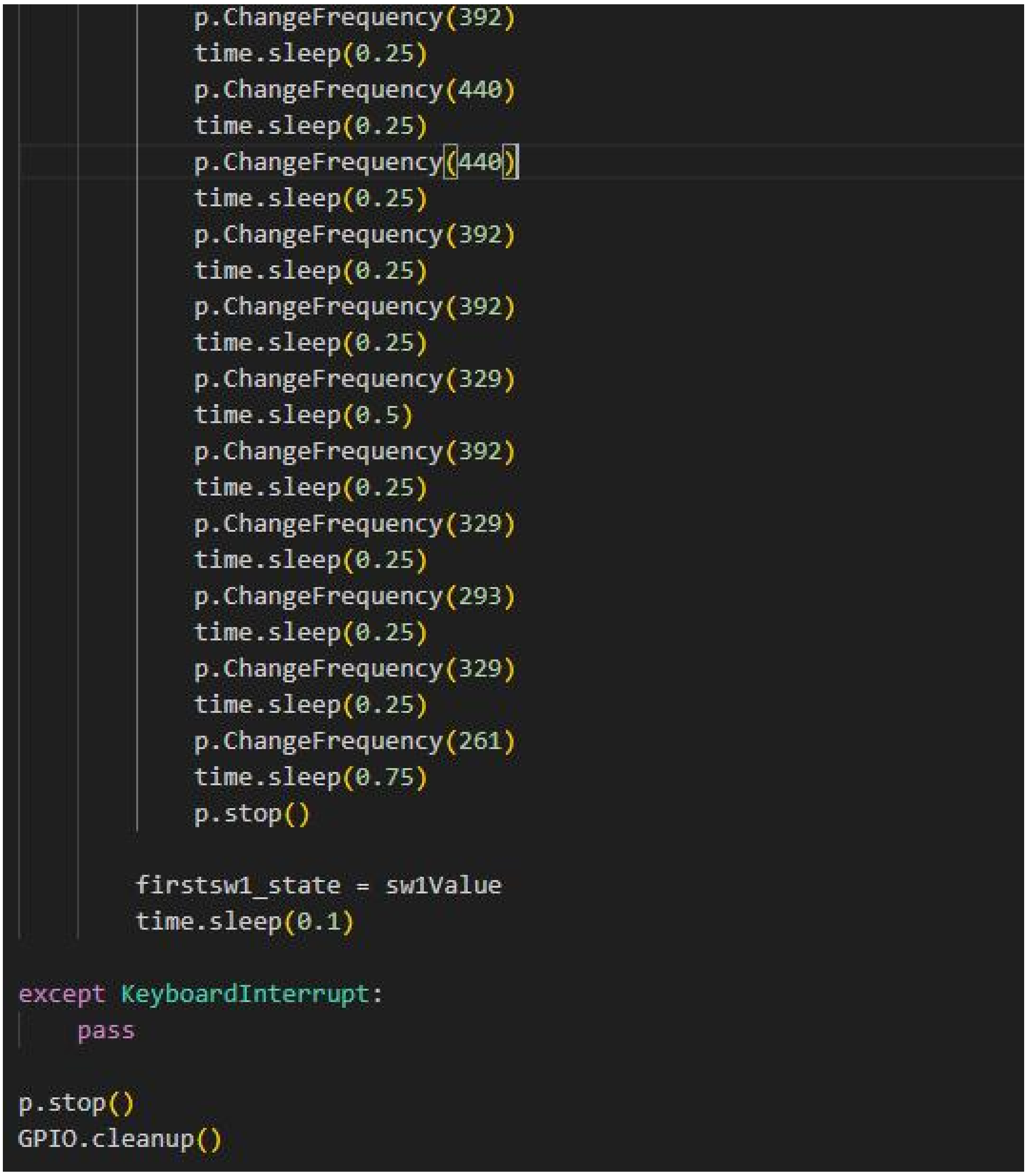


```
    p.ChangeFrequency(440)
    time.sleep(0.25)
    p.ChangeFrequency(440)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(329)
    time.sleep(0.5)
    p.ChangeFrequency(392)
    time.sleep(0.25)
    p.ChangeFrequency(329)
    time.sleep(0.25)
    p.ChangeFrequency(293)
    time.sleep(0.25)
    p.ChangeFrequency(329)
    time.sleep(0.25)
    p.ChangeFrequency(261)
    time.sleep(0.75)
    p.ChangeFrequency(0)
    time.sleep(0.1)

except KeyboardInterrupt:
    pass

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

3) 스위치를 한번 누르면 경적 소리가 나도록 구현

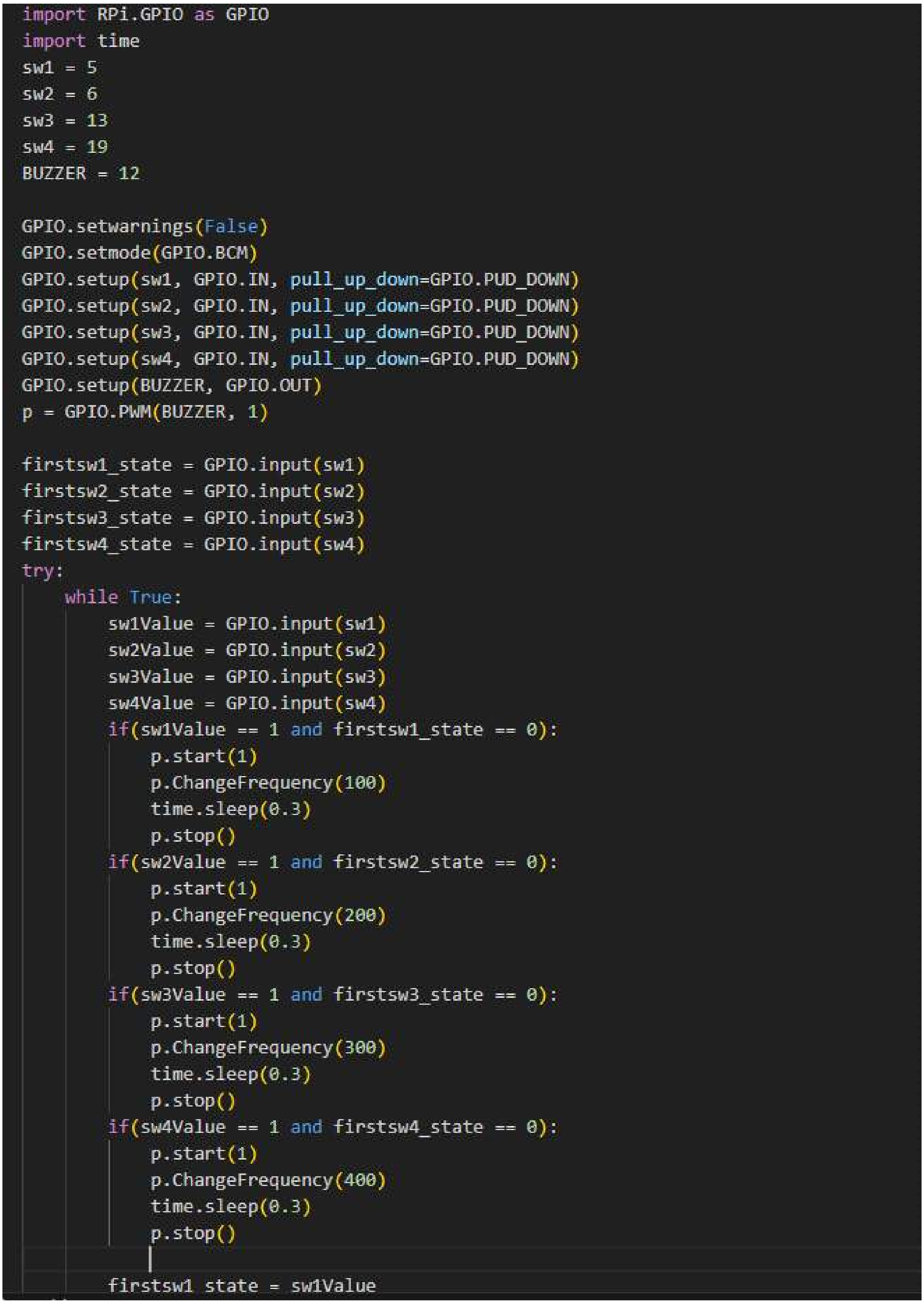


```
import RPi.GPIO as GPIO
import time

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, 1)


firstsw1_state = GPIO.input(sw1)

try:
    while True:
        sw1Value = GPIO.input(sw1)
        if(sw1Value == 1 and firstsw1_state == 0):
            p.start(1)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(440)
            time.sleep(0.25)
            p.ChangeFrequency(440)
            time.sleep(0.25)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(330)
            time.sleep(0.55)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(330)
            time.sleep(0.25)
            p.ChangeFrequency(330)
            time.sleep(0.25)
            p.ChangeFrequency(293)
            time.sleep(0.75)


            p.ChangeFrequency(392)
            time.sleep(0.25)
```

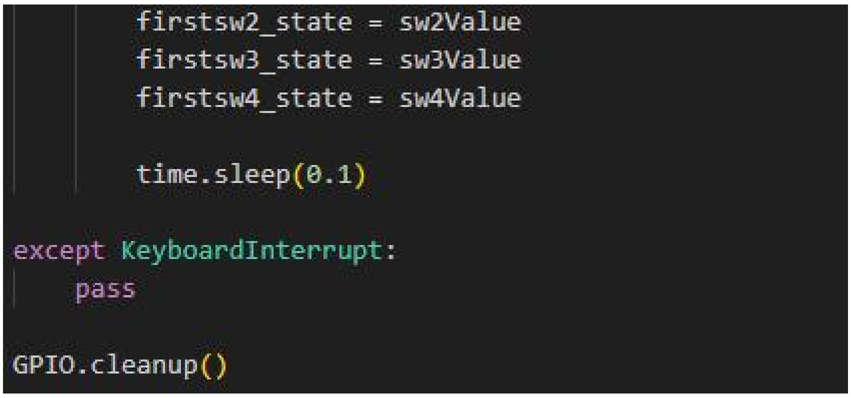


```
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(440)
            time.sleep(0.25)
            p.ChangeFrequency(440)
            time.sleep(0.25)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(329)
            time.sleep(0.5)
            p.ChangeFrequency(392)
            time.sleep(0.25)
            p.ChangeFrequency(329)
            time.sleep(0.25)
            p.ChangeFrequency(293)
            time.sleep(0.25)
            p.ChangeFrequency(329)
            time.sleep(0.25)
            p.ChangeFrequency(261)
            time.sleep(0.75)
            p.stop()

        firstsw1_state = sw1Value
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

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

4) 스위치 4개를 사용하여 나만의 음악을 연주

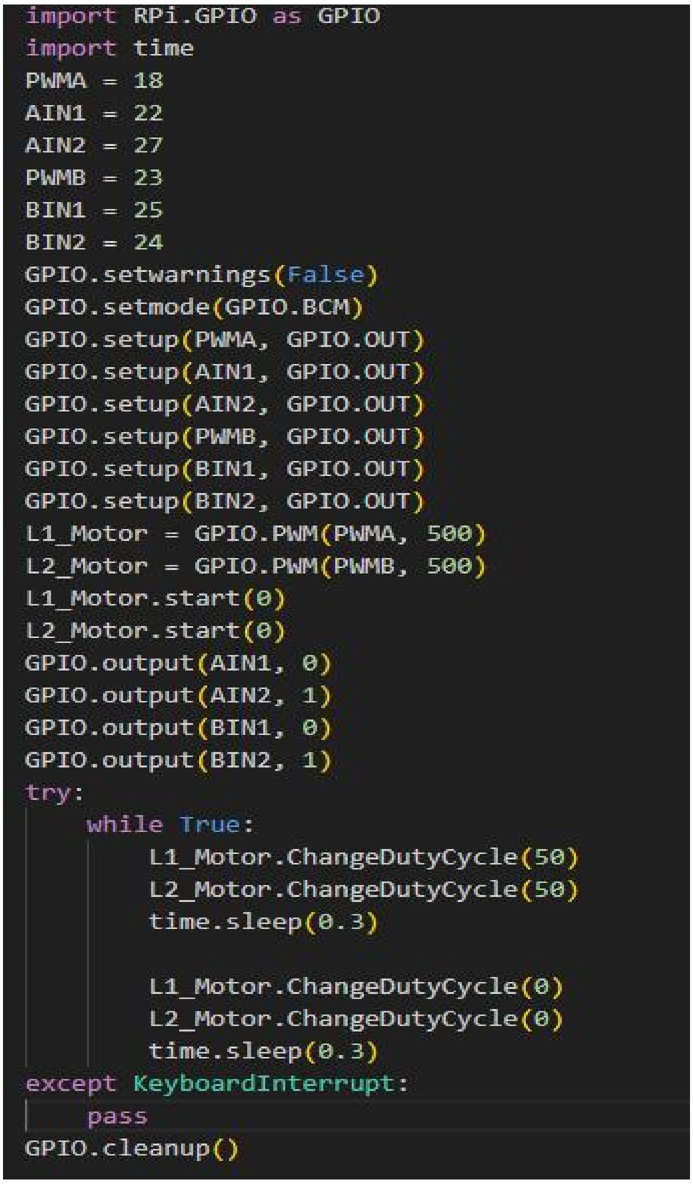


```python
import RPi.GPIO as GPIO
import time
sw1 = 5
sw2 = 6
sw3 = 13
sw4 = 19
BUZZER = 12

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)
GPIO.setup(BUZZER, GPIO.OUT)
p = GPIO.PWM(BUZZER, 1)

firstsw1_state = GPIO.input(sw1)
firstsw2_state = GPIO.input(sw2)
firstsw3_state = GPIO.input(sw3)
firstsw4_state = GPIO.input(sw4)
try:
    while True:
        sw1Value = GPIO.input(sw1)
        sw2Value = GPIO.input(sw2)
        sw3Value = GPIO.input(sw3)
        sw4Value = GPIO.input(sw4)
        if(sw1Value == 1 and firstsw1_state == 0):
            p.start(1)
            p.ChangeFrequency(100)
            time.sleep(0.3)
            p.stop()
        if(sw2Value == 1 and firstsw2_state == 0):
            p.start(1)
            p.ChangeFrequency(200)
            time.sleep(0.3)
            p.stop()
        if(sw3Value == 1 and firstsw3_state == 0):
            p.start(1)
            p.ChangeFrequency(300)
            time.sleep(0.3)
            p.stop()
        if(sw4Value == 1 and firstsw4_state == 0):
            p.start(1)
            p.ChangeFrequency(400)
            time.sleep(0.3)
            p.stop()

        firstsw1_state = sw1Value
```



```
        firstsw2_state = sw2Value
        firstsw3_state = sw3Value
        firstsw4_state = sw4Value

        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

1) 오른쪽 모터부분의 코드를 추가하여 정방향으로 50%로 동작->정지->동작->정지 …

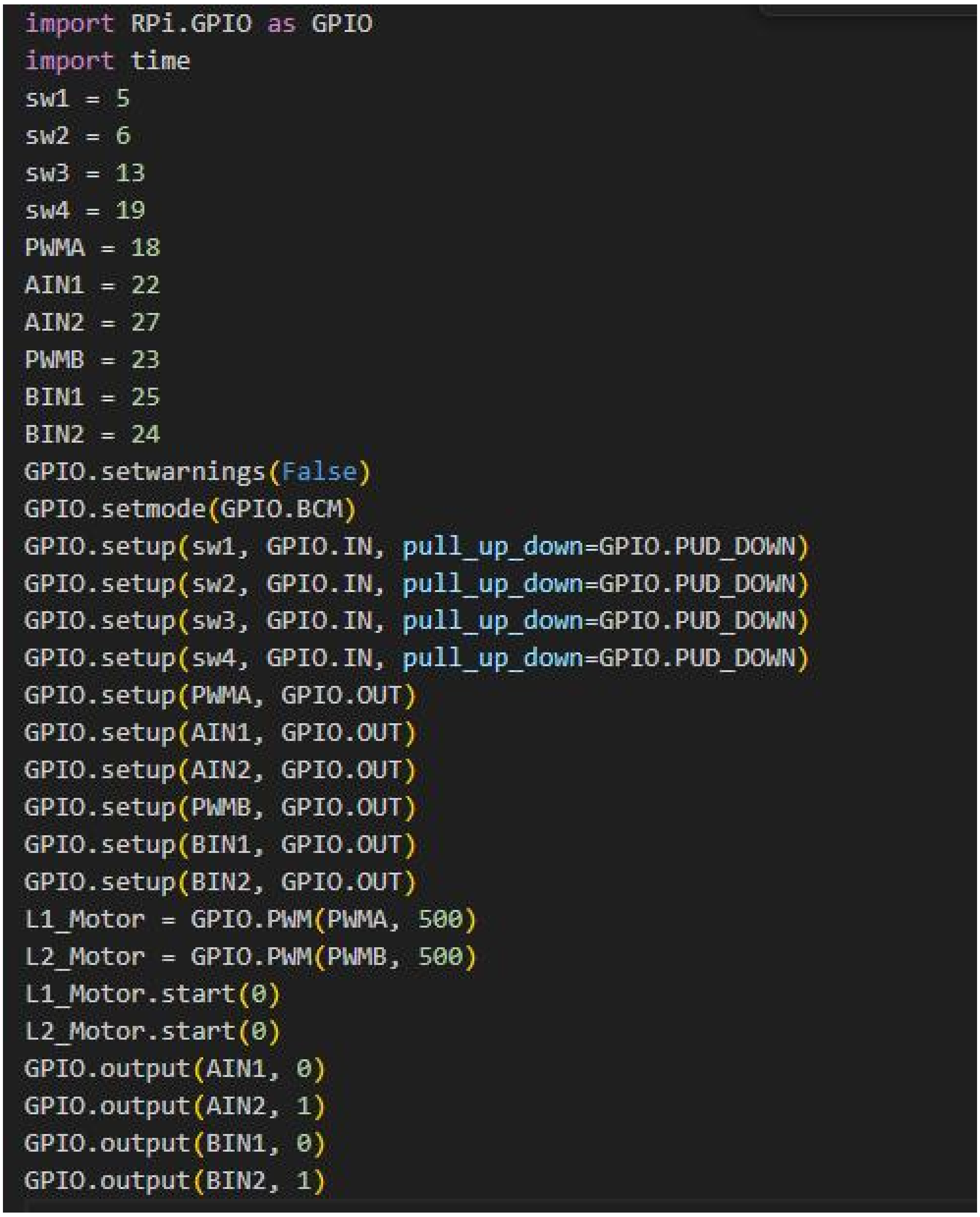


```
import RPi.GPIO as GPIO
import time
PWMA = 18
AIN1 = 22
AIN2 = 27
PWMB = 23
BIN1 = 25
BIN2 = 24
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(PWMA, GPIO.OUT)
GPIO.setup(AIN1, GPIO.OUT)
GPIO.setup(AIN2, GPIO.OUT)
GPIO.setup(PWMB, GPIO.OUT)
GPIO.setup(BIN1, GPIO.OUT)
GPIO.setup(BIN2, GPIO.OUT)
L1_Motor = GPIO.PWM(PWMA, 500)
L2_Motor = GPIO.PWM(PWMB, 500)
L1_Motor.start(0)
L2_Motor.start(0)
GPIO.output(AIN1, 0)
GPIO.output(AIN2, 1)
GPIO.output(BIN1, 0)
GPIO.output(BIN2, 1)
try:
    while True:
        L1_Motor.ChangeDutyCycle(50)
        L2_Motor.ChangeDutyCycle(50)
        time.sleep(0.3)

        L1_Motor.ChangeDutyCycle(0)
        L2_Motor.ChangeDutyCycle(0)
        time.sleep(0.3)
except KeyboardInterrupt:
    pass
GPIO.cleanup()
```

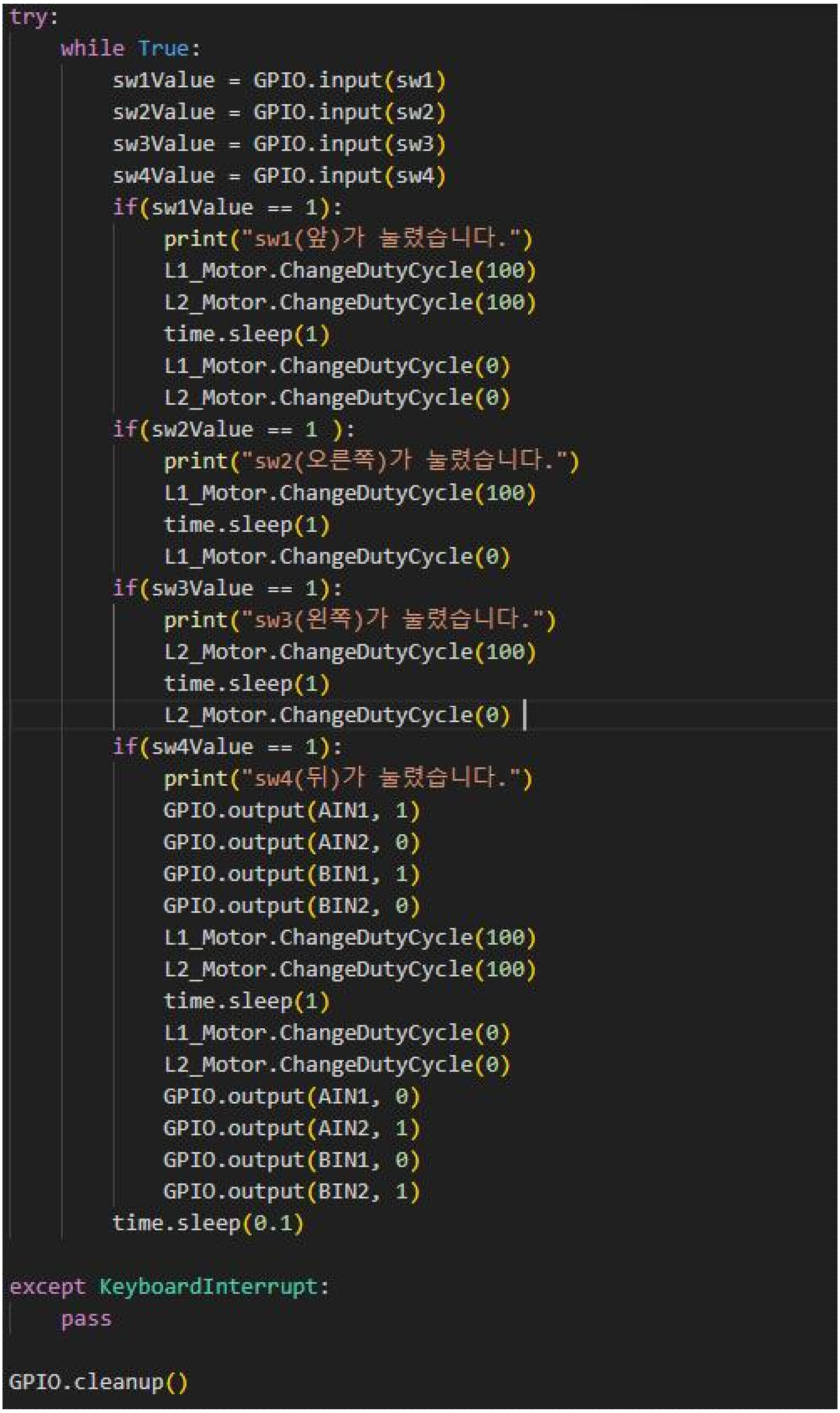
2) 스위치를 입력 받아 자동차 조종하기

SW1 : 앞
SW2 : 오른쪽
SW3 : 왼쪽
SW4 : 뒤

print문을 사용하여 어느 스위치가 눌렸는지 출력



```
import RPi.GPIO as GPIO
import time
sw1 = 5
sw2 = 6
sw3 = 13
sw4 = 19
PWMA = 18
AIN1 = 22
AIN2 = 27
PWMB = 23
BIN1 = 25
BIN2 = 24
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)
GPIO.setup(PWMA, GPIO.OUT)
GPIO.setup(AIN1, GPIO.OUT)
GPIO.setup(AIN2, GPIO.OUT)
GPIO.setup(PWMB, GPIO.OUT)
GPIO.setup(BIN1, GPIO.OUT)
GPIO.setup(BIN2, GPIO.OUT)
L1_Motor = GPIO.PWM(PWMA, 500)
L2_Motor = GPIO.PWM(PWMB, 500)
L1_Motor.start(0)
L2_Motor.start(0)
GPIO.output(AIN1, 0)
GPIO.output(AIN2, 1)
GPIO.output(BIN1, 0)
GPIO.output(BIN2, 1)
```



```
try:
    while True:
        sw1Value = GPIO.input(sw1)
        sw2Value = GPIO.input(sw2)
        sw3Value = GPIO.input(sw3)
        sw4Value = GPIO.input(sw4)
        if(sw1Value == 1):
            print("sw1(앞)가 눌렸습니다.")
            L1_Motor.ChangeDutyCycle(100)
            L2_Motor.ChangeDutyCycle(100)
            time.sleep(1)
            L1_Motor.ChangeDutyCycle(0)
            L2_Motor.ChangeDutyCycle(0)
        if(sw2Value == 1 ):
            print("sw2(오른쪽)가 눌렸습니다.")
            L1_Motor.ChangeDutyCycle(100)
            time.sleep(1)
            L1_Motor.ChangeDutyCycle(0)
        if(sw3Value == 1):
            print("sw3(왼쪽)가 눌렸습니다.")
            L2_Motor.ChangeDutyCycle(100)
            time.sleep(1)
            L2_Motor.ChangeDutyCycle(0)
        if(sw4Value == 1):
            print("sw4(뒤)가 눌렸습니다.")
            GPIO.output(AIN1, 1)
            GPIO.output(AIN2, 0)
            GPIO.output(BIN1, 1)
            GPIO.output(BIN2, 0)
            L1_Motor.ChangeDutyCycle(100)
            L2_Motor.ChangeDutyCycle(100)
            time.sleep(1)
            L1_Motor.ChangeDutyCycle(0)
            L2_Motor.ChangeDutyCycle(0)
            GPIO.output(AIN1, 0)
            GPIO.output(AIN2, 1)
            GPIO.output(BIN1, 0)
            GPIO.output(BIN2, 1)
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```