임베디드 응용 및 실습 7 주차 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()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

/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()
```

```
^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

```
5) import RPi.GPIO as GPIO
6) import time
7)
8) SW1 = 5
9) SW2 = 6
10)SW3 = 13
11)SW4 = 19
12)
13)GPIO.setwarnings(False)
14)GPIO.setmode(GPIO.BCM)
15)
16)GPIO.setup(SW1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
17)GPIO.setup(SW2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
18)GPIO.setup(SW3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
19)GPIO.setup(SW4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
20)
21)switch_names = ['SW1', 'SW2', 'SW3', 'SW4']
22)switch_pins = [SW1, SW2, SW3, SW4]
23)
24)prev_values = [0, 0, 0, 0]
25)click_counts = [0, 0, 0, 0]
26)
27)try:
28)    while True:
29)        for i in range(4):
30)            current_value = GPIO.input(switch_pins[i])
31)
32)            # 스위치가 눌렸을 때 (0 -> 1 으로 변할 때)
33)            if prev_values[i] == 0 and current_value == 1:
34)                click_counts[i] += 1  # 클릭 횟수 증가
35)                print("(", "'", switch_names[i], "click', ",
   click_counts[i], ")")
36)
37)            # 현재 값을 이전 값에 저장
38)            prev_values[i] = current_value
39)
40)        time.sleep(0.1)
41)
42)except KeyboardInterrupt:
43)    pass
44)
45)GPIO.cleanup()
```

```
^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

```
(2)import RPi.GPIO as GPIO
(3)import time
(4)
(5)BUZZER = 12
(6)
(7)GPIO.setwarnings(False)
(8)GPIO.setmode(GPIO.BCM)
(9)GPIO.setup(BUZZER, GPIO.OUT)
(10)
(11)    p = GPIO.PWM(BUZZER, 261)
(12)    p.start(50)
(13)
(14)    try:
(15)        while True:
(16)            p.start(50)
(17)            p.ChangeFrequency(262)
(18)            time.sleep(1.0)
(19)            p.ChangeFrequency(292)
(20)            time.sleep(1.0)
(21)            p.ChangeFrequency(330)
(22)            time.sleep(1.0)
(23)            p.ChangeFrequency(349)
(24)            time.sleep(1.0)
(25)            p.ChangeFrequency(394)
(26)            time.sleep(1.0)
(27)            p.ChangeFrequency(440)
(28)            time.sleep(1.0)
(29)            p.ChangeFrequency(494)
(30)            time.sleep(1.0)
(31)            p.ChangeFrequency(523)
```

```
(32)            time.sleep(1.0)
(33)
(34)            p.stop()
(35)            time.sleep(1.0)
(36)
(37)
(38)    except KeyboardInterrupt:
(39)        pass
(40)
(41)    p.stop()
(42)    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()
```

⑷ 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()
```