임베디드 응용 및 실습

- 7주차 과제 -

학번	2020161092
이름	이재윤

(과제) 버튼 입력 받기 구현

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

코드

```
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()
```

```
wodbs@wodbs:~/lec6 $ python 3_2_1.py
```

결과

누르기 전

```

```

누르기 후

```
click
click
click
```

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

코드

```
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:
        if(GPIO.input(SW1)==1):
            print("click 1")
        elif(GPIO.input(SW2)==1):
            print("click 2")
        elif(GPIO.input(SW3)==1):
            print("click 3")
        elif(GPIO.input(SW4)==1):
            print("click 4")
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

결과 (4 → 3 → 2 → 1 → 2 → 3 →4를 출력해보겠습니다.)

```
^Cwodbs@wodbs:~/lec6 $ python 3_2_2.py
click 4
click 3
click 2
click 1
click 2
click 3
click 4
```

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

코드

```
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)

count = 0
activation = False

try:
    while True:
        sw1Value=GPIO.input(SW1)
        if(sw1Value==GPIO.HIGH and not activation):
            count+=1
            print(count)
            activation = True
        elif(sw1Value==GPIO.LOW):
            activation=False
        if(count==0 and not activation):
            print(sw1Value)
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

결과 (스위치를 누르지 않았을 때 0 이 뜨다가 한번 누르면 계속 더해지는 구조)

```
^Cwodbs@wodbs:~/lec6 $ python 3_2_3.py
0
0
0
0
1
2
3
4
```

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

코드

```
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)

count = [0, 0, 0, 0]
activation = [False, False, False, False]

try:
    while True:
        swValue=[GPIO.input(SW1), GPIO.input(SW2), GPIO.input(SW3), GPIO.input(SW4)]
        for i in range(0,4):
            if(swValue[i]==GPIO.HIGH and not activation[i]):
                count[i]+=1
                print(('SW%d click' %(i+1), count[i]))
                activation[i]=True
            elif(swValue[i]==GPIO.LOW):
                activation[i]=False

        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

결과

```
wodbs@wodbs:~/lec6 $ python 3_2_5.py
('SW1 click', 1)
('SW1 click', 2)
('SW2 click', 1)
('SW4 click', 1)
('SW3 click', 1)
('SW1 click', 3)
```

(과제) 부저 음계 출력 구현

1) "도레미파솔라시도" 음계를 출력

코드

```
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, 262)
p.start(50)

try:
    while True:
        p.start(50)
        p.ChangeFrequency(262)
        time.sleep(1.0)
        p.ChangeFrequency(294)
        time.sleep(1.0)
        p.ChangeFrequency(330)
        time.sleep(1.0)
        p.ChangeFrequency(349)
        time.sleep(1.0)
        p.ChangeFrequency(392)
        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

코드

```
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, 262)
p.start(50)

try:
    while True:
        p.start(50)
        p.ChangeFrequency(262)
        time.sleep(0.8)
        p.ChangeFrequency(294)
        time.sleep(0.3)
        p.ChangeFrequency(330)
        time.sleep(0.7)
        p.ChangeFrequency(262)
        time.sleep(0.5)
        p.ChangeFrequency(330)
        time.sleep(0.5)
        p.ChangeFrequency(262)
        time.sleep(0.5)
        p.ChangeFrequency(330)
        time.sleep(0.5)
        p.stop()
        time.sleep(1.0)

except KeyboardInterrupt:
    pass

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

3) 스위치를 한번 누르면 경적 소리가 나도록 구현

코드

```
import RPi.GPIO as GPIO
import time

SW1=5
BUZZER = 12

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

p=GPIO.PWM(BUZZER, 262)

try:
    while True:
        sw1Value=GPIO.input(SW1)
        if(sw1Value==1):
            p.start(50)
            p.ChangeFrequency(262)
            time.sleep(1.0)
            p.stop()
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

4) 스위치 4개를 사용하여 나만의 음악을 연주

코드

```
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, 262)
```

```
try:
    while True:
        swValue=[GPIO.input(SW1), GPIO.input(SW2), GPIO.input(SW3), GPIO.input(SW4)]
        if(swValue[0]==GPIO.HIGH):
            p.start(50)
            p.ChangeFrequency(262)
            time.sleep(0.5)
            p.stop()
        elif(swValue[1]==GPIO.HIGH):
            p.start(50)
            p.ChangeFrequency(294)
            time.sleep(0.5)
            p.stop()
        elif(swValue[2]==GPIO.HIGH):
            p.start(50)
            p.ChangeFrequency(330)
            time.sleep(0.5)
            p.stop()
        elif(swValue[3]==GPIO.HIGH):
            p.start(50)
            p.ChangeFrequency(349)
            time.sleep(0.5)
            p.stop()
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

(과제) 자동차 움직이기 구현

1) 오른쪽 모터 부분의 코드를 추가하여 정방향으로 50%로 동작 → 정지 → 동작 → 정지

코드

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

```
try:
    while True:
        GPIO.output(AIN1,0)
        GPIO.output(AIN2,1)
        GPIO.output(BIN1,0)
        GPIO.output(BIN2,1)
        L_Motor.ChangeDutyCycle(100)
        R_Motor.ChangeDutyCycle(100)
        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
print 문을 사용하여 어느 스위치가 눌렸는지 출력

```
import RPi.GPIO as GPIO
import time

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

PWMA = 18
PWMB = 23
AIN1 = 22
AIN2 = 27
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(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)
R_Motor = GPIO.PWM(PWMB,500)
L_Motor.start(0)
R_Motor.start(0)
```

```
try:
    while True:
        swValue=[GPIO.input(SW1), GPIO.input(SW2), GPIO.input(SW3), GPIO.input(SW4)]
        if(swValue[0]==GPIO.HIGH):
            print('SW1 click')
            GPIO.output(AIN1,0)
            GPIO.output(AIN2,1)
            GPIO.output(BIN1,0)
            GPIO.output(BIN2,1)
            L_Motor.ChangeDutyCycle(50)
            R_Motor.ChangeDutyCycle(50)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
            R_Motor.ChangeDutyCycle(0)
        elif(swValue[1]==GPIO.HIGH):
            print('SW2 click')
            GPIO.output(BIN1,0)
            GPIO.output(BIN2,1)
            R_Motor.ChangeDutyCycle(50)
            time.sleep(1.0)
            R_Motor.ChangeDutyCycle(0)
        elif(swValue[2]==GPIO.HIGH):
            print('SW3 click')
            GPIO.output(AIN1,0)
            GPIO.output(AIN2,1)
            L_Motor.ChangeDutyCycle(50)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
        elif(swValue[3]==GPIO.HIGH):
            print('SW4 click')
            GPIO.output(AIN1,1)
            GPIO.output(AIN2,0)
            GPIO.output(BIN1,1)
            GPIO.output(BIN2,0)
            L_Motor.ChangeDutyCycle(50)
            R_Motor.ChangeDutyCycle(50)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
            R_Motor.ChangeDutyCycle(0)
        time.sleep(0.1)
```

```
except KeyboardInterrupt:
    pass

GPIO.cleanup()
```