임베디드 과제

07주차. 과제 보고서

과 목 명	임베디드
학 번	2021161002
이 름	강민구
제 출 일	2025년10월17일

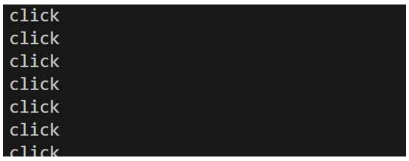
1-1 실습문제

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

1-2. 소스코드

```
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()
```

1-3. 실행결과



```
click
click
click
click
click
click
click
```

2-1 실습문제

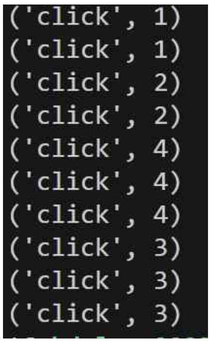
몇번 스위치가 눌렸는지 확인이 가능하도록 “click x” 등으로 화면 출력

2-2. 소스코드

```
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)
            time.sleep(0.1)
        elif(sw2Value == 1):
            print("click", 2)
            time.sleep(0.1)
        elif(sw3Value == 1):
            print("click", 3)
            time.sleep(0.1)
        elif(sw4Value == 1):
            print("click", 4)
            time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

2-3. 실행결과



```
('click', 1)
('click', 1)
('click', 2)
('click', 2)
('click', 4)
('click', 4)
('click', 4)
('click', 3)
('click', 3)
('click', 3)
```

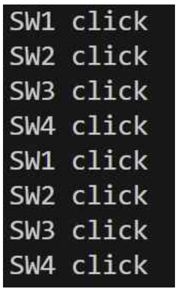
3-1 실습문제

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

3-2. 소스코드

```
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)

before_sw1_click_stat = 0
before_sw2_click_stat = 0
before_sw3_click_stat = 0
before_sw4_click_stat = 0

try:
    while True:
        sw1Value=GPIO.input(SW1)
        sw2Value=GPIO.input(SW2)
        sw3Value=GPIO.input(SW3)
        sw4Value=GPIO.input(SW4)
        if(before_sw1_click_stat==0 and sw1Value == 1):
            print("SW1 click")
            before_sw1_click_stat = sw1Value
            time.sleep(0.1)
        else: before_sw1_click_stat = sw1Value

        if(before_sw2_click_stat==0 and sw2Value == 1):

            print("SW2 click")
            before_sw2_click_stat = sw2Value
            time.sleep(0.1)
        else: before_sw2_click_stat = sw2Value

        if(before_sw3_click_stat==0 and sw3Value == 1):

            print("SW3 click")
            before_sw3_click_stat = sw3Value
            time.sleep(0.1)
        else: before_sw3_click_stat = sw3Value

        if(before_sw4_click_stat==0 and sw4Value == 1):

            print("SW4 click")
            before_sw4_click_stat = sw4Value
            time.sleep(0.1)
        else: before_sw4_click_stat = sw4Value


except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

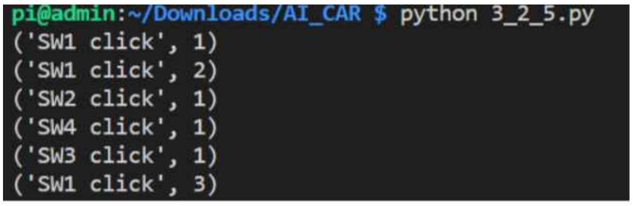
3-3. 실행결과



```
SW1 click
SW2 click
SW3 click
SW4 click
SW1 click
SW2 click
SW3 click
SW4 click
```

4-1 실습문제

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



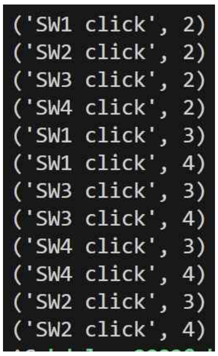
```
pi@admin:~/Downloads/AI_CAR $ python 3_2_5.py
('SW1 click', 1)
('SW1 click', 2)
('SW2 click', 1)
('SW4 click', 1)
('SW3 click', 1)
('SW1 click', 3)
```

4-2. 소스코드

```
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)

sw1_count = 0
sw2_count = 0
sw3_count = 0
```

```
sw4_count = 0

before_sw1_click_stat = 0
before_sw2_click_stat = 0
before_sw3_click_stat = 0
before_sw4_click_stat = 0




try:
    while True:
        sw1Value=GPIO.input(SW1)
        sw2Value=GPIO.input(SW2)
        sw3Value=GPIO.input(SW3)
        sw4Value=GPIO.input(SW4)
        if(before_sw1_click_stat==0 and sw1Value == 1):
            sw1_count += 1
            print("SW1 click", sw1_count)
            before_sw1_click_stat = sw1Value
            time.sleep(0.1)
        else: before_sw1_click_stat = sw1Value

        if(before_sw2_click_stat==0 and sw2Value == 1):
            sw2_count += 1
            print("SW2 click", sw2_count)
            before_sw2_click_stat = sw2Value
            time.sleep(0.1)
        else: before_sw2_click_stat = sw2Value

        if(before_sw3_click_stat==0 and sw3Value == 1):
            sw3_count += 1
            print("SW3 click", sw3_count)
            before_sw3_click_stat = sw3Value
            time.sleep(0.1)
        else: before_sw3_click_stat = sw3Value

        if(before_sw4_click_stat==0 and sw4Value == 1):
            sw4_count += 1
            print("SW4 click", sw4_count)
            before_sw4_click_stat = sw4Value
            time.sleep(0.1)
        else: before_sw4_click_stat = sw4Value


except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

4-3. 실행결과



```
('SW1 click', 2)
('SW2 click', 2)
('SW3 click', 2)
('SW4 click', 2)
('SW1 click', 3)
('SW1 click', 4)
('SW3 click', 3)
('SW3 click', 4)
('SW4 click', 3)
('SW4 click', 4)
('SW2 click', 3)
('SW2 click', 4)
```

5-1 실습문제

“도레미파솔라시도” 음계를 출력

5-2. 소스코드

```
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)
frequency = [262,294,330,349,392,440,494,523]

try:
    while True:
        for i in frequency:
            p.ChangeFrequency(i)
            p.start(50)
            time.sleep(1.0)
            p.stop()
            time.sleep(1.0)


except KeyboardInterrupt:
    pass

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

5-3. 실행결과

6-1 실습문제

나만의 경적 소리 구현

6-2. 소스코드

```
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)

music = [262,262,392,392,440,440,392,349,349,330,330,294,294,262]
try:
    while True:
        time.sleep(0.5)
        for i in music:
            p.ChangeFrequency(i)
            p.start(50)
            time.sleep(0.5)
            p.stop()


except KeyboardInterrupt:
    pass

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

6-3. 실행결과

7-1 실습문제

스위치를 한번 누르면 경적 소리가 나도록 구현

7-2. 소스코드

```
import RPi.GPIO as GPIO
import time

BUZZER = 12
SW1 = 5
befor_click = 0

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, 262)

music = [262,262,392,392,440,440,392,349,349,330,330,294,294,262]
try:
    while True:
        sw1Value=GPIO.input(SW1)
        if(befor_click == 0 and sw1Value == 1):
            for i in music:
                p.ChangeFrequency(i)
                p.start(50)
                time.sleep(0.5)
                p.stop()

        befor_click = sw1Value



except KeyboardInterrupt:
    pass

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

7-3. 실행결과

8-1 실습문제

스위치 4개를 사용하여 나만의 음악을 연주

8-2. 소스코드

```
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, 262)

school_song = [294, 330,392,440]
click1 = 0
click2 = 0
click3 = 0
click4 = 0

try:
    while True:
        sw1Value=GPIO.input(SW1)
        sw2Value=GPIO.input(SW2)
        sw3Value=GPIO.input(SW3)
        sw4Value=GPIO.input(SW4)
        if(click1 == 0 and sw1Value == 1):
            p.ChangeFrequency(school_song[0])
            p.start(50)
            time.sleep(0.2)
            p.stop()
        elif(click2 == 0 and sw2Value == 1):
            p.ChangeFrequency(school_song[1])
            p.start(50)
            time.sleep(0.2)
            p.stop()
        elif(click3 == 0 and sw3Value == 1):
            p.ChangeFrequency(school_song[2])
            p.start(50)
            time.sleep(0.2)
            p.stop()
        elif(click4 == 0 and sw4Value == 1):
            p.ChangeFrequency(school_song[3])
            p.start(50)
            time.sleep(0.2)
            p.stop()

        click1 = sw1Value
```

```
            click2 = sw2Value
            click3 = sw3Value
            click4 = sw4Value

except KeyboardInterrupt:
    pass

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

8-3. 실행결과

9-1 실습문제

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

9-2. 소스코드

```
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(AIN1, GPIO.OUT)
GPIO.setup(AIN2, GPIO.OUT)
GPIO.setup(PWMB, 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)
        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)
        time.sleep(1.0)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

9-3. 실행결과

10-1 실습문제

스위치를 입력 받아 자동차 조종하기

SW1 : 앞

SW2 : 오른쪽

SW3 : 왼쪽

SW4 : 뒤

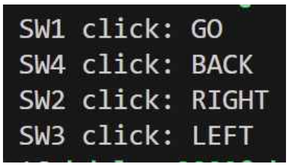
print문을 사용하여 어느 스위치가 눌렸는지 출력

10-2. 소스코드

```
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(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)

L_Motor = GPIO.PWM(PWMA, 500)
L_Motor.start(0)
R_Motor = GPIO.PWM(PWMB, 500)
R_Motor.start(0)
click1 = 0
click2 = 0
click3 = 0
click4 = 0
try:
    while True:
        sw1Value=GPIO.input(SW1)
        sw2Value=GPIO.input(SW2)
        sw3Value=GPIO.input(SW3)
        sw4Value=GPIO.input(SW4)
        if(click1 == 0 and sw1Value == 1):
            print("SW1 click: GO")
            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)

        elif(click4 == 0 and sw4Value == 1):
            print("SW4 click: BACK")
            GPIO.output(AIN1,1), GPIO.output(AIN2,0)
            GPIO.output(BIN1, 1), GPIO.output(BIN2, 0)
            R_Motor.ChangeDutyCycle(50)
            L_Motor.ChangeDutyCycle(50)
            time.sleep(1.0)

        elif(click2 == 0 and sw2Value == 1):
            print("SW2 click: RIGHT")
            GPIO.output(AIN1, 0), GPIO.output(AIN2, 1)
            GPIO.output(BIN1, 0), GPIO.output(BIN2, 0)
            R_Motor.ChangeDutyCycle(0)
            L_Motor.ChangeDutyCycle(50)
            time.sleep(1.0)

        elif(click3 == 0 and sw3Value == 1):
            print("SW3 click: LEFT")
            GPIO.output(AIN1, 0), GPIO.output(AIN2, 0)
            GPIO.output(BIN1, 0), GPIO.output(BIN2, 1)
            R_Motor.ChangeDutyCycle(50)
            L_Motor.ChangeDutyCycle(0)
            time.sleep(1.0)

        R_Motor.ChangeDutyCycle(0)
        L_Motor.ChangeDutyCycle(0)
        click1 = sw1Value
        click2 = sw2Value
        click3 = sw3Value
        click4 = sw4Value
except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

10-3. 실행결과



```
SW1 click: GO
SW4 click: BACK
SW2 click: RIGHT
SW3 click: LEFT
```