임베디드 응용 및 실습

정보통신공학

박준석

2019161064

GPIOAS1_1.py

```
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()
```

GPIOAS1_2.py

```
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")
            time.sleep(0.5)

        if GPIO.input(SW2) == 1:
            print("click 2")
            time.sleep(0.5)

        if GPIO.input(SW3) == 1:
            print("click 3")
            time.sleep(0.5)

        if GPIO.input(SW4) == 1:
            print("click 4")
            time.sleep(0.5)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

GPIOAS1_3.py

```
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)

prev_state = 0

try:
    while True:
        current_state = GPIO.input(SW1)

        if current_state == 1 and prev_state == 0:
```

```
            print("1")
        elif current_state == 0 and prev_state == 1:
            print("0")

        prev_state = current_state
        time.sleep(0.1)

except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

GPIOAS1_4.py

```
import RPi.GPIO as GPIO
import time

pins = [5, 6, 13, 19]

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

GPIO.setup(pins[0], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pins[1], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pins[2], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(pins[3], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

counts = [0, 0, 0, 0]
prev_states = [0, 0, 0, 0]

try:
    while True:
        state1 = GPIO.input(pins[0])
        state2 = GPIO.input(pins[1])
        state3 = GPIO.input(pins[2])
        state4 = GPIO.input(pins[3])

        if state1 == 1 and prev_states[0] == 0:
            counts[0] += 1
            print(f"( 'SW1 click: {counts[0]}' )")

        if state2 == 1 and prev_states[1] == 0:
            counts[1] += 1
            print(f"( 'SW2 click: {counts[1]}' )")

        if state3 == 1 and prev_states[2] == 0:
```

```
            counts[2] += 1
            print(f"( 'SW3 click: {counts[2]}' )")

        if state4 == 1 and prev_states[3] == 0:
            counts[3] += 1
            print(f"( 'SW4 click: {counts[3]}' )")

        prev_states[0] = state1
        prev_states[1] = state2
        prev_states[2] = state3
        prev_states[3] = state4

        time.sleep(0.1)

except KeyboardInterrupt:
    pass
```

GPIOAS2_1.py

```
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(293)
        time.sleep(0.3)
        p.ChangeFrequency(330)
        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(494)
        time.sleep(0.3)
        p.ChangeFrequency(523)
        time.sleep(0.3)
        p.stop()
        time.sleep(0.3)


except KeyboardInterrupt:
    pass

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

GPIOAS2_2.py

```
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(0.1)
        p.ChangeFrequency(330)
        time.sleep(0.1)
        p.ChangeFrequency(293)
        time.sleep(0.1)
        p.ChangeFrequency(349)
        time.sleep(0.1)
        p.ChangeFrequency(440)
        time.sleep(0.1)
        p.ChangeFrequency(392)
        time.sleep(0.1)
        p.ChangeFrequency(523)
        time.sleep(0.1)
```

```
        p.ChangeFrequency(494)
        time.sleep(0.1)
        p.stop()
        time.sleep(1)


except KeyboardInterrupt:
    pass

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

GPIOAS2_3.py

```
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, 100)



try:
    while True:
        sw1Value = GPIO.input(SW1)
        if sw1Value == 1:  # 버튼이 눌리면
            p.start(50)
            p.ChangeFrequency(100)
            time.sleep(0.5)
            p.stop() # 소리 멈춤
            time.sleep(1)




except KeyboardInterrupt:
    pass
```

```
GPIO.cleanup()
```

GPIOAS2_4.py

```
       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, 100)


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

        if sw1Value == 1:  # 버튼이 눌리면
            p.start(50)   # 50% 듀티 사이클
            p.ChangeFrequency(100)
            time.sleep(0.5)  # 0.1초 동안 소리 유지
            p.stop() # 소리 멈춤
            time.sleep(1)

        if sw2Value == 1:  # 버튼이 눌리면
            p.start(50)   # 50% 듀티 사이클
            p.ChangeFrequency(200)
            time.sleep(0.5)  # 0.1초 동안 소리 유지
            p.stop() # 소리 멈춤
```

```
            time.sleep(1)

        if sw3Value == 1:  # 버튼이 눌리면
            p.start(50)   # 50% 듀티 사이클
            p.ChangeFrequency(300)
            time.sleep(0.5)  # 0.1초 동안 소리 유지
            p.stop() # 소리 멈춤
            time.sleep(1)

        if sw4Value == 1:  # 버튼이 눌리면
            p.start(50)   # 50% 듀티 사이클
            p.ChangeFrequency(400)
            time.sleep(0.5)  # 0.1초 동안 소리 유지
            p.stop() # 소리 멈춤
            time.sleep(1)




except KeyboardInterrupt:
    pass

GPIO.cleanup()
```

GPIOAS3_1.py

```
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(50)
       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()
```

GPIOAS3_2.py

```
       import RPi.GPIO as GPIO

import time

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

PWMA = 18
.000000.as_integer_ratio


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

        if sw1Value == 1:  # 버튼이 눌리면
            GPIO.output(AIN1,0)
            GPIO.output(AIN2,1) #오른쪽 앞
            GPIO.output(BIN1,0)
            GPIO.output(BIN2,1) #왼쪽 앞
            print("SW1 : 앞")
            L_Motor.ChangeDutyCycle(20)
            R_Motor.ChangeDutyCycle(20)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
            R_Motor.ChangeDutyCycle(0)

        if sw2Value == 1:  # 버튼이 눌리면
            GPIO.output(AIN1,0)
            GPIO.output(AIN2,1)
            GPIO.output(BIN1,1)
            GPIO.output(BIN2,0)
            print("SW2 : 오른쪽")
```

```
            L_Motor.ChangeDutyCycle(20)
            R_Motor.ChangeDutyCycle(20)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
            R_Motor.ChangeDutyCycle(0)

        if sw3Value == 1:  # 버튼이 눌리면
            GPIO.output(AIN1,1)
            GPIO.output(AIN2,0)
            GPIO.output(BIN1,0)
            GPIO.output(BIN2,1)
            print("SW3 : 왼쪽")
            L_Motor.ChangeDutyCycle(20)
            R_Motor.ChangeDutyCycle(20)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
            R_Motor.ChangeDutyCycle(0)

        if sw4Value == 1:  # 버튼이 눌리면
            GPIO.output(AIN1,1)
            GPIO.output(AIN2,0)
            GPIO.output(BIN1,1)
            GPIO.output(BIN2,0)
            print("SW4 : 뒤")
            L_Motor.ChangeDutyCycle(20)
            R_Motor.ChangeDutyCycle(20)
            time.sleep(1.0)
            L_Motor.ChangeDutyCycle(0)
            R_Motor.ChangeDutyCycle(0)




except KeyboardInterrupt:
    pass

GPIO.cleanup()
```