임베디드 응용 및 실습 과제4

- 4. GPIO 실습

학과	전기공학과
학년	3학년
학번	2020161110
이름	전현서
과목명	임베디드 응용 및 실습
분반	01분반
담당교수님	우성민 교수님

```
import RPi.GPIO as GPIO
import time
SW =[0, 5, 6, 13, 19]


GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
for i in range(1,5):
    GPIO.setup(SW[i],GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
swtime =[0, 1, 1, 1, 1]
swstatus =[0, 0, 0, 0, 0]
swValue =[0, 0, 0, 0, 0]
i =1
try:
    while True:
        swValue[i] =GPIO.input(SW[i])
        if not swstatus[i]:
            if swValue[i]:
                swstatus[i] =1
                print("('SW",i, " click', ", swtime[i], ")")
                swtime[i] +=1
        else:
            if swValue[i] ==0:
                swstatus[i] =0
        i +=1
        if i ==5:
            i =1

        time.sleep(0.03)
except KeyboardInterrupt:
    pass
GPIO.cleanup()
```

```
import RPi.GPIO as GPIO
import time
SW =[0, 5, 6, 13, 19]
BUZZER =12
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)
for i in range(1, 5):
    GPIO.setup(SW[i], GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
swValue =[0, 0, 0, 0, 0]
swstatus =0
doremi =[0, 262, 294, 330, 349, 392, 440, 494, 523]
try:
    while True:
        swstatus =0
        for i in range(1, 5):
            swValue[i] =GPIO.input(SW[i])
            swstatus *=10
            swstatus +=swValue[i]
            time.sleep(0.03)

        if swstatus ==1000:
            p =GPIO.PWM(BUZZER, doremi[1])
        elif swstatus ==100:
            p =GPIO.PWM(BUZZER, doremi[2])
        elif swstatus ==10:
            p =GPIO.PWM(BUZZER, doremi[3])
        elif swstatus ==1:
            p =GPIO.PWM(BUZZER, doremi[4])
        elif swstatus ==1100:
            p =GPIO.PWM(BUZZER, doremi[5])
        elif swstatus ==1010:
            p =GPIO.PWM(BUZZER, doremi[6])
        elif swstatus ==1001:
            p =GPIO.PWM(BUZZER, doremi[7])
        elif swstatus ==110:
            p =GPIO.PWM(BUZZER, doremi[8])
        else:
            p =None
        if p:
            p.start(75)
            time.sleep(0.3)
            p.stop()
            p =None
except KeyboardInterrupt:
    pass
GPIO.cleanup()
```

```
import time
PWMA =18
PWMB =23
AIN1 =22
AIN2 =27
BIN1 =25
BIN2 =24
SW =[0, 5, 6, 13, 19]
SWstatus =[0, 0, 0, 0, 0]
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)
for i in range(1,5):
    GPIO.setup(SW[i],GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
L_Motor =GPIO.PWM(PWMA,500)
R_Motor =GPIO.PWM(PWMB,500)
L_Motor.start(0)
R_Motor.start(0)
Motor_Control =[0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0]
try:
    while True:
        for i in range(1,5):
            if GPIO.input(SW[i]) ==1:
                GPIO.output(AIN1,Motor_Control[4*i])
                GPIO.output(AIN2,Motor_Control[4*i+1])
                GPIO.output(BIN1,Motor_Control[4*i+2])
                GPIO.output(BIN2,Motor_Control[4*i+3])
                L_Motor.ChangeDutyCycle(100)
                R_Motor.ChangeDutyCycle(100)
                time.sleep(0.05)
                GPIO.output(AIN1,Motor_Control[4*i])
                GPIO.output(AIN2,Motor_Control[4*i+1])
                GPIO.output(BIN1,Motor_Control[4*i+2])
                GPIO.output(BIN2,Motor_Control[4*i+3])
                L_Motor.ChangeDutyCycle(0)
                R_Motor.ChangeDutyCycle(0)



except KeyboardInterrupt:
    pass
GPIO.cleanup()
```