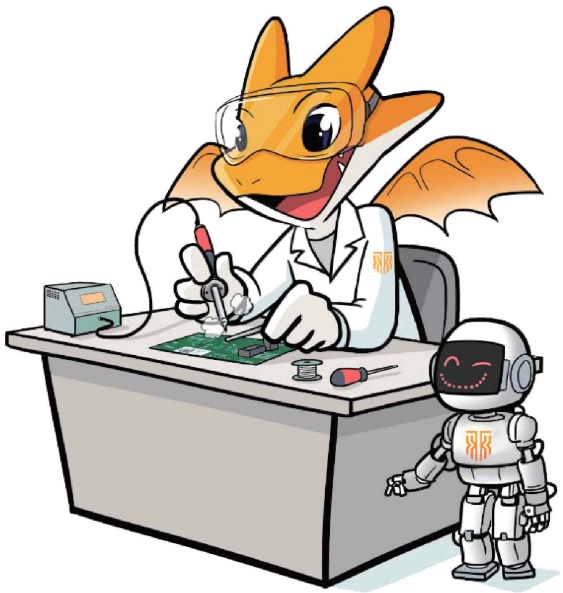
# homework_week7



| | |
|---|---|
| 제 출 일 | 24.10.21 |
| 과 목 명 | 임베디드 응용 및 실습 |
| 담당교수님 | 우성민 교수님 |
| 학 과 | 전자공학과 |
| 학 번 | 2020161060 |
| 이 름 | 신규철 |

## < 버튼 입력 받기 구현 >

### < 1 >

```
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 :
        swValue = GPIO.input(SW1)

        if swValue == 1 :
            print("\"click\"")

        time.sleep(0.1)

except KeyboardInterrupt :
    pass

GPIO.cleanup()
```

### < 2 >

```
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 :
        swValue = GPIO.input(SW1)

        if swValue == 1 :
            print("\"click SW1\"")

        time.sleep(0.1)

except KeyboardInterrupt :
    pass

GPIO.cleanup()
```

< 3 >

```
import RPi.GPIO as GPIO
import time

SW1 = 5
preSWValue = 0

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


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

        if preSWValue == 0 and swValue == 1 :
            print("\"click SW1\"")
            preSWValue = swValue

        elif preSWValue == 1 and swValue == 0:
            preSWValue = swValue
        time.sleep(0.1)

except KeyboardInterrupt :
    pass

GPIO.cleanup()
```

< 4 >

```
import RPi.GPIO as GPIO
import time

SW = [5, 6, 13, 19]
preSWValue = [0, 0, 0, 0]
count = [0, 0, 0, 0]

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

for i in SW:
    GPIO.setup(i, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

try :
    while True :
        swValue = [ GPIO.input(SW[0]), GPIO.input(SW[1]), GPIO.input(SW[2]),
GPIO.input(SW[3])]

        for i in range(0, 4) :
            if preSWValue[i] == 0 and swValue[i] == 1 :
                count[i] = count[i] + 1
                print("('SW", i+1, "click', ", count[i], ')')
                preSWValue[i] = swValue[i]

            elif preSWValue[i] == 1 and swValue[i] == 0:
                preSWValue[i] = swValue[i]
        time.sleep(0.1)

except KeyboardInterrupt :
    pass

GPIO.cleanup()
```

## < 부저 음계 출력 구현 >

### < 1 >

```
import RPi.GPIO as GPIO
import time

BUZZER = 12

'''       C4    D4    E4    F4    G4    A4    B4    C5       '''
freq = [ 262, 294, 330, 349, 392, 440, 494, 523 ]

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)

p = GPIO.PWM(BUZZER, freq[0])
p.start(50)

try :
    while True :
        p.start(50)
        for i in range(0, 8) :
            p.ChangeFrequency(freq[i])
            time.sleep(0.3)
        p.stop()
        time.sleep(0.1)
except KeyboardInterrupt :
    pass

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

### < 2 >

```
import RPi.GPIO as GPIO
import time

BUZZER = 12

'''      C4    D4    E4    F4    G4    A4    B4    C5'''
freq = [262, 294, 330, 349, 392, 440, 494, 523]
freq_time = [0.5, 0.125, 0.01]

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)

p = GPIO.PWM(BUZZER, freq[1])

try :
    while True :
        # C4 C4
        p.start(50)
        p.ChangeFrequency(262)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])

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

```
        time.sleep(freq_time[1])

        # E4 E4
        p.ChangeFrequency(330)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])

        p.start(50)
        p.ChangeFrequency(330)
        time.sleep(freq_time[1])

        # A3 A3
        p.ChangeFrequency(220)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])

        p.start(50)
        p.ChangeFrequency(220)
        time.sleep(freq_time[1])

        # C4 C4
        p.ChangeFrequency(262)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])

        p.start(50)
        p.ChangeFrequency(262)
        time.sleep(freq_time[1])

        # D4 D4
        p.ChangeFrequency(294)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])

        p.start(50)
        p.ChangeFrequency(294)
        time.sleep(freq_time[1])

        # F4 F4
        p.ChangeFrequency(349)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])

        p.start(50)
        p.ChangeFrequency(349)
        time.sleep(freq_time[1])

        # G3 G3
        p.ChangeFrequency(196)
        time.sleep(freq_time[0])
        p.stop()

        time.sleep(freq_time[2])
```

```
            p.start(50)
            p.ChangeFrequency(196)
            time.sleep(freq_time[1])

            # B3 B3
            p.ChangeFrequency(247)
            time.sleep(freq_time[0])
            p.stop()

            time.sleep(freq_time[2])

            p.start(50)
            p.ChangeFrequency(247)
            time.sleep(freq_time[1])
            p.stop()

except KeyboardInterrupt :
    pass

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

< 3 >

```
import RPi.GPIO as GPIO
import time

BUZZER = 12
SW1 = 5
preSWValue = 0

'''      C4    D4    E4    F4    G4    A4    B4    C5'''
freq = [262, 294, 330, 349, 392, 440, 494, 523]
freq_time = [0.5, 0.125, 0.01]

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, freq[1])

def honk() :
    # C4 C4
    p.start(50)
    p.ChangeFrequency(262)
    time.sleep(freq_time[0])
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(262)
    time.sleep(freq_time[1])

    # E4 E4
    p.ChangeFrequency(330)
    time.sleep(freq_time[0])
    p.stop()
```

```
    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(330)
    time.sleep(freq_time[1])

    # A3 A3
    p.ChangeFrequency(220)
    time.sleep(freq_time[0])
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(220)
    time.sleep(freq_time[1])

    # C4 C4
    p.ChangeFrequency(262)
    time.sleep(freq_time[0])
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(262)
    time.sleep(freq_time[1])

    # D4 D4
    p.ChangeFrequency(294)
    time.sleep(freq_time[0])
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(294)
    time.sleep(freq_time[1])

    # F4 F4
    p.ChangeFrequency(349)
    time.sleep(freq_time[0])
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(349)
    time.sleep(freq_time[1])

    # G3 G3
    p.ChangeFrequency(196)
    time.sleep(freq_time[0])
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(196)
    time.sleep(freq_time[1])

    # B3 B3
    p.ChangeFrequency(247)
    time.sleep(freq_time[0])
```

```
    p.stop()

    time.sleep(freq_time[2])

    p.start(50)
    p.ChangeFrequency(247)
    time.sleep(freq_time[1])
    p.stop()

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

        if preSWValue == 0 and swValue == 1:
            honk()
            preSWValue = swValue

        elif preSWValue == 1 and swValue == 0 :
            preSWValue = swValue

        time.sleep(0.1)
except KeyboardInterrupt :
    pass

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

< 4 >

```
import RPi.GPIO as GPIO
import time

BUZZER = 12
SW = [5, 6, 13, 19]

'''
        SW2
   SW1   ●   SW4
    ●         ●
          ●
        SW3
'''

preSWValue = [0, 0, 0, 0]

'''      C4    D4    E4    F4    G4    A4    A4#   B4   C5'''
freq = [262, 294, 330, 349, 392, 440, 466, 494, 523]

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUZZER, GPIO.OUT)

for i in SW:
    GPIO.setup(i, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

p = GPIO.PWM(BUZZER, freq[1])

try :
    while True :
        swValue = [ GPIO.input(SW[0]), GPIO.input(SW[1]), GPIO.input(SW[2]),
GPIO.input(SW[3])]

        if swValue[1] == 1 and swValue[3] == 1:
            p.start(50)
            p.ChangeFrequency(freq[8])

        elif swValue[2] == 1 and swValue[3] == 1 :
            p.start(50)
            p.ChangeFrequency(freq[6])

        elif swValue[0] == 1 and swValue[3] == 1 :
            p.start(50)
            p.ChangeFrequency(freq[7])

        elif swValue[0] == 1 and swValue[2] == 1 :
            p.start(50)
            p.ChangeFrequency(freq[5])

        elif swValue[0] == 1 and swValue[1] == 1 :
            p.start(50)
            p.ChangeFrequency(freq[4])

        elif swValue[3] == 1 :
            p.start(50)
            p.ChangeFrequency(freq[3])

        elif swValue[2] == 1 :
            p.start(50)
            p.ChangeFrequency(freq[2])

        elif swValue[1] == 1 :
```

```
                p.start(50)
                p.ChangeFrequency(freq[1])

            elif swValue[0] == 1 :
                p.start(50)
                p.ChangeFrequency(freq[0])

            else :
                p.stop()

except KeyboardInterrupt :
    pass

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

## < 자동차 움직이기 구현 >

### < 1 >

```
import RPi.GPIO as GPIO
import time

PWM = [ 18, 23 ]
AIN = [ 22, 27 ]
BIN = [ 25, 24 ]


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

for i in range(0, 2) :
    GPIO.setup(PWM[i], GPIO.OUT)
    GPIO.setup(AIN[i], GPIO.OUT)
    GPIO.setup(BIN[i], GPIO.OUT)


L_Motor = GPIO.PWM(PWM[0], 500)
L_Motor.start(0)
R_Motor = GPIO.PWM(PWM[1], 500)
R_Motor.start(0)

try :
    for i in range(0, 2) :
        GPIO.output(AIN[i], i)
        GPIO.output(BIN[i], i)

    while True :
        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()
```

< 2 >

```
import RPi.GPIO as GPIO
import time

PWM = [ 18, 23 ]
AIN = [ 22, 27 ]
BIN = [ 25, 24 ]

SW = [5, 6, 13, 19]
preSWValue = [0, 0, 0, 0]
direction = ['앞', '오른쪽', '왼쪽', '뒤']

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

for i in SW:
    GPIO.setup(i, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

for i in range(0, 2) :
    GPIO.setup(PWM[i], GPIO.OUT)
    GPIO.setup(AIN[i], GPIO.OUT)
    GPIO.setup(BIN[i], GPIO.OUT)


L_Motor = GPIO.PWM(PWM[0], 500)
L_Motor.start(0)
R_Motor = GPIO.PWM(PWM[1], 500)
R_Motor.start(0)

try :
    while True :
        swValue = [ GPIO.input(SW[0]), GPIO.input(SW[1]), GPIO.input(SW[2]),
GPIO.input(SW[3])]

        for i in range(0, 4) :
            if preSWValue[i] == 0 and swValue[i] == 1 :
                for j in range(0, 2) :
                    GPIO.output(AIN[j], j if i in [0, 1] else 1 - j)
                    GPIO.output(BIN[j], j if i in [0, 2] else 1 - j)

                L_Motor.ChangeDutyCycle(50)
                R_Motor.ChangeDutyCycle(50)

                print("\"SW", i+1, "click:", direction[i], "\"")
                preSWValue[i] = swValue[i]

            elif preSWValue[i] == 1 and swValue[i] == 0 :
                L_Motor.ChangeDutyCycle(0)
                R_Motor.ChangeDutyCycle(0)
                preSWValue[i] = swValue[i]

except KeyboardInterrupt :
    pass

GPIO.cleanup()
```