임베디드응용및실습 6주차 과제

2020161132 최한나

1. 버튼 입력 받기 구현  
4번 : 4개의 스위치 입력을 받도록 해보자. 화면에 아래와 같이 출력되도록 한다.  
 단, 리스트를 최대한 활용하여 GPIO 전/후 값을 저장한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

import RPi.GPIO as GPIO

import time

SW=[5,6,13,19] # 스위치 핀 번호

state=[0,0,0,0] # 스위치의 상태

GPIO\_past\_List=[0,0,0,0] # 과거 스위치의 상태

num=[0,0,0,0] # 각 스위치가 눌리는 횟수

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

for i in range(len(SW)):

    GPIO.setup(SW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

try:

    while True:

        for i in range(len(SW)):

            state[i]=GPIO.input(SW[i]) # 각 스위치의 상태 저장

        for i in range(len(SW)):

# 현재 스위치의 상태가 1이고 과거 스위치 상태가 0일 때 = 스위치 누름

          if (state[i]==1 and GPIO\_past\_List[i]==0):

                num[i]+=1 # 각 스위치 눌린 횟수 저장

                print("'SW"+ str(i+1) + " click'", num[i]) # 출력 화면

                GPIO\_past\_List[i]=state[i]  # 현재 스위치 상태를 과거 상태에 저장

                if(GPIO\_past\_List[i]==1): # 과거 스위치 상태가 1이면 continue

                    continue

# 현재 스위치의 상태가 0이고 과거 스위치 상태가 1일 때 = 스위치에서 손 땜

            if (state[i]==0 and GPIO\_past\_List[i]==1):

                GPIO\_past\_List[i]=state[i]  # 현재 스위치 상태를 과거 상태에 저장

        time.sleep(0.1)

except KeyboardInterrupt:

    pass

GPIO.cleanup()

[ 출력 화면 ]

텍스트이(가) 표시된 사진

자동 생성된 설명

2. 부저 음계 출력 구현  
4번 : 스위치 4개를 사용하여 나만의 음악을 연주

import RPi.GPIO as GPIO

import time

BUZZER =12

SW=[5,6,13,19]

state=[0,0,0,0]

gye=[262,294,330,349] # 도, 레, 미, 파

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(BUZZER,GPIO.OUT)

for i in range(len(SW)):

    GPIO.setup(SW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

p=GPIO.PWM(BUZZER,261)

try:

    while True:

        for i in range(len(SW)):

            state[i]=GPIO.input(SW[i])

        for i in range(len(SW)):

            if state[i]:

                p.start(50)

                p.ChangeFrequency(gye[i])

                time.sleep(0.1)

            p.stop()

except KeyboardInterrupt:

    pass

p.stop()

GPIO.cleanup()

3. 자동차 움직이기 구현   
2번: 스위치를 입력 받아 자동차 조종하기  
- 앞, 뒤, 오른쪽, 왼쪽 을 함수를 통해 구현

import RPi.GPIO as GPIO

import time

# pin 번호

SW=[5,6,13,19]

state=[0,0,0,0]

PWMA=18

AIN1=22

AIN2=27

PWMB=23

BIN1=25

BIN2=24

moterSet=[PWMA,AIN1,AIN2,PWMB,BIN1,BIN2]

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

for i in range(len(SW)):   # 스위치 세팅

    GPIO.setup(SW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

for i in range(len(moterSet)): # 모터 세팅

    GPIO.setup(moterSet[i], GPIO.OUT)

L\_Motor=GPIO.PWM(PWMA,500)  # 왼쪽 모터 제어

L\_Motor.start(0)

R\_Motor=GPIO.PWM(PWMB,500)  # 오른쪽 모터 제어

R\_Motor.start(0)

def go(): # 전진 함수

    GPIO.output(AIN1,0)

    GPIO.output(AIN2,1)

    L\_Motor.ChangeDutyCycle(100) # 왼쪽에 있는 바퀴 전진

    GPIO.output(BIN1,0)

    GPIO.output(BIN2,1)

    R\_Motor.ChangeDutyCycle(100) # 오른쪽에 있는 바퀴 전진

def back(): # 후진 함수

    GPIO.output(AIN1,1)

    GPIO.output(AIN2,0)

    L\_Motor.ChangeDutyCycle(100) # 왼쪽에 있는 바퀴 후진

    GPIO.output(BIN1,1)

    GPIO.output(BIN2,0)

    R\_Motor.ChangeDutyCycle(100) # 오른쪽에 있는 바퀴 후진

def right(): # 우회전 함수

    GPIO.output(AIN1,0)

    GPIO.output(AIN2,1)

    L\_Motor.ChangeDutyCycle(100) # 왼쪽에 있는 바퀴 전진

    GPIO.output(BIN1,1)

    GPIO.output(BIN2,0)

    R\_Motor.ChangeDutyCycle(100) # 오른쪽에 있는 바퀴 후진

def left(): # 좌회전 함수

    GPIO.output(AIN1,1)

    GPIO.output(AIN2,0)

    L\_Motor.ChangeDutyCycle(100) # 왼쪽에 있는 바퀴 후진

    GPIO.output(BIN1,0)

    GPIO.output(BIN2,1)

    R\_Motor.ChangeDutyCycle(100) # 오른쪽에 있는 바퀴 전진

def stop(): # 정지 함수

    GPIO.output(AIN1,1)

    GPIO.output(AIN2,0)

    L\_Motor.ChangeDutyCycle(0) # 왼쪽에 있는 바퀴 정지

    GPIO.output(BIN1,1)

    GPIO.output(BIN2,0)

    R\_Motor.ChangeDutyCycle(0) # 오른쪽에 있는 바퀴 정지

try:

    while True:

            for i in range(len(SW)):

                state[i]=GPIO.input(SW[i])

            if state[0]: # sw1을 누르고 있으면 직진

                go()

            elif state[1]: # sw2를 누르고 있으면 우회전

                right()

            elif state[2]: # sw3을 누르고 있으면 좌회전

                left()

            elif state[3]: # sw4를 누르고 있으면 후진

                back()

            else: # sw를 때면 정지

                stop()

except KeyboardInterrupt:

    pass

GPIO.cleanup()