



프로세서응용종합설계

4주차 실습 결과보고서

담당교수 : 황신환 교수
님

제출일자 : 20.11.04

학 과 : 전자공학과

학 년 : 3학년

이 름 : 배준성

학 번 :



한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

1. 코드

```
import RPi.GPIO as GPIO #GPIO 라이브러리 호출
import time             #time 함수 라이브러리 호출

GPIO.setmode(GPIO.BCM) #핀을 GPIO 핀 번호 기분으로 부름

GPIO.setup(17, GPIO.OUT) #17번 핀을 출력으로 사용
GPIO.setup(18, GPIO.IN) #18번 핀을 입력으로 사용

ledPWM = GPIO.PWM(17,100) #17번 핀을 PWM으로 사용, 100HZ
ledPWM.start(0) #듀티비를 0으로 세팅

i=1 기본 단계를 1단계로 세팅
try:
    while True :
        LED = 6*(i**2) #단계에 따라 LED의 값을 바꿔준다
        ledPWM.ChangeDutyCycle(LED) #듀티비를 바꿔준다

        if GPIO.wait_for_edge(18, GPIO.RISING, timeout=100): #Rising Edge가 발생
            하면 동작하며 발생하지 않는 경우 대기한다.
                i += 1 #단계를 증가
                if i > 4:i=1 #4단계이후 1단계로 초기화
                time.sleep(0.25) #한번에 버튼이 여러 번 눌러지는 것을 방지하기 위해
                서 delay를 준다.
except KeyboardInterrupt:
    pass
```

2. 결과 사진

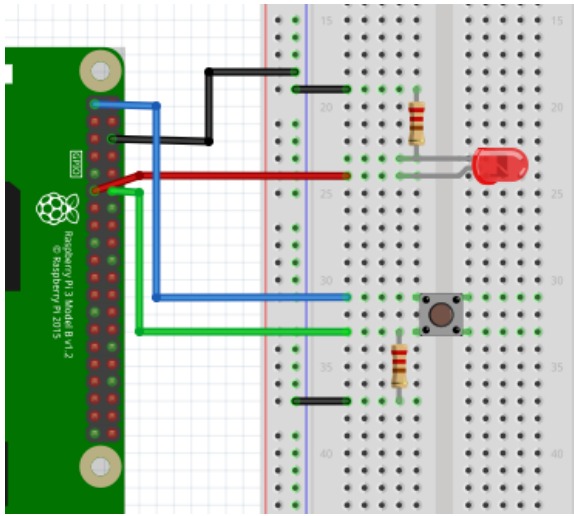


사진 1 - Fritzing으로 그려본 회로

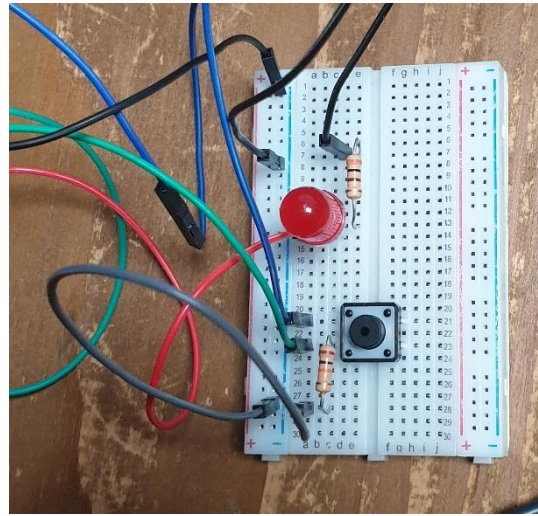


사진 2 - 실제로 구현한 회로

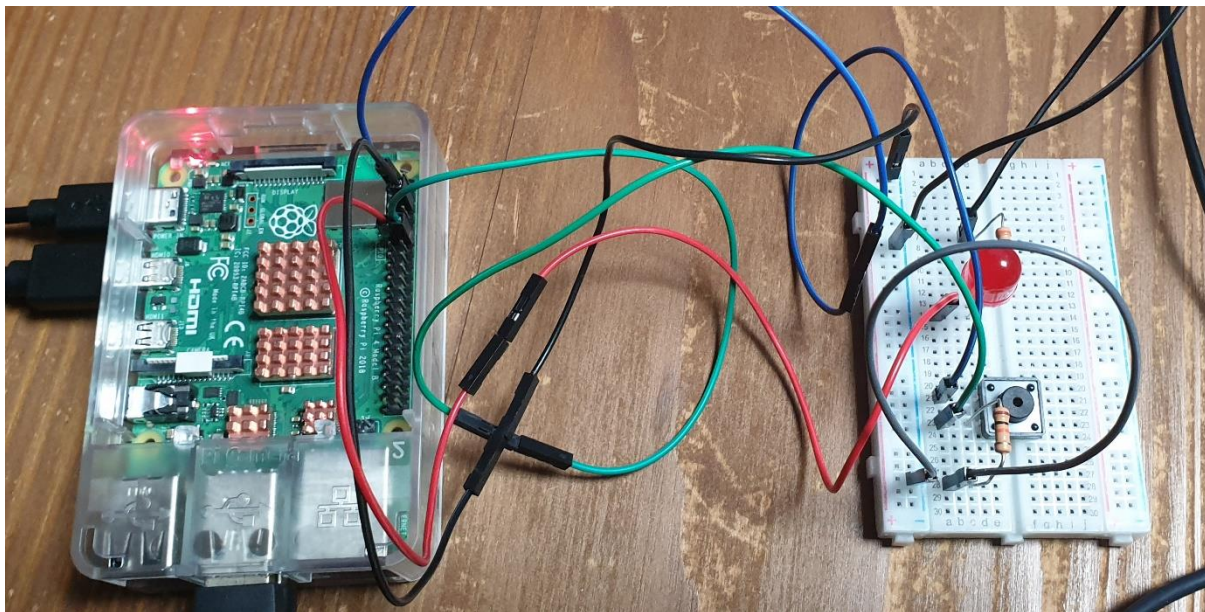


사진 3 - 라즈베리파이를 포함한 전체 회로

3. 결론 및 고찰

본 실습을 통하여 LED를 4단계로 구분되는 밝기로 제어하는 프로그램을 만들어 보았다. LED는 17번 핀을 이용하여 제어해 주었다. 제어에는 PWM을 이용하였으며 밝기는 `ledPWM.ChangeDutyCycle()`를 이용하여 듀티비를 바꾸어 주어 LED의 밝기를 조절하였다. 단계가 올라갈수록 더 밝아져야 하기 때문에 i값이 커지면 변수LED의 값이 증가하여 듀티비가 점점 커진다. 듀티비는 한 주기동안 HIGH신호가 차지하는 비율로서 이 비율을 늘리고 줄임으로써 총 전압대비 원하는 만큼의 전압을 입력 가능하다. 이번 실습처럼 LED의 밝기를 점점 늘려야 하는 경우 그 비율을 증가시킴으로써 그 밝기를 더 밝게 조절할 수 있었다.

18번 핀의 경우 스위치의 입력을 받아들여 LED를 제어하는데 사용하였다. 아래의 if문과 조합하여 Rising Edge가 발생하면 단계를 증가시키는 역할이다. 이 단계는 4단계 이후에는 다시 1단계로 초기화 하여 계속 반복되도록 만들어주었다. 다음의 `time.sleep()`의 경우 버튼이 짧은 시간내에 중복으로 입력되어 오류가 발생하는 것을 방지하기 위해서 추가해 주었다.

이번 실습을 통해서 PWM을 사용해 볼 수 있었다. 실습 과제의 경우 밝기가 즉각적으로 바뀌는 프로그램이었지만 예제의 경우 밝기를 천천히 Fade in/out하는 효과를 줄 수 있었다. 이를 통해서 우리는 PWM을 이용하면 아주 다양한 신호를 원하는 곳에 입력할 수 있음을 알 수 있었다. 듀티비를 조절하여 한 주기 동안 HIGH의 비율을 조절할 수 있기 때문에 원하는 시간동안 원하는 양만큼의 전압을 입력하는 효과를 줄 수 있었다.