

Exercise #05 REPORT

Name: 송예린

Student ID: 2171023

1. Exercise Objective (10%)

Floating 모드는 High, Low voltage 범위의 중간에 값이 있어서 High 도 Low 도 아닌 상태를 의미한다. 이를 방지하기 위해서 저항이 있는 Pull down, Pull up 을 사용한다. Pull up 은 스위치가 열려있을 때 흘러가는 input 값이 High 값을 가지는 것을 의미하고 Pull down 은 스위치가 열려있을 때 흘러가는 input 값이 Low 값을 가지는 것을 의미한다. 2 개의 Switch 제어를 이용해 스위치의 상태를 체크하고 input 핀에 HIGH, GPIO 에서 input 값을 넣는다면 스위치로 LED on/off 를 할 수 있을 것이다.

2. Implementation (60%)

S1, S2 위치(p1 2 번째 5 번째)의 핀 번호 0 을 넣어서 input 으로 설정한다. Pull up, Pull down enable 할 핀번호에 1 을 넣어주고 Pull up 을 사용하기 때문에 p1 out 해당 핀 번호를 1 을 적어준다. P1 은 output 으로도 사용하기 때문에 p1->DIR 에서 빨간색에 해당하는 핀 번호를 1 로 설정해준다. P2 는 출력 Port 로 사용하므로 R,G,B 에 해당하는 핀 번호를 설정해준다. 반복문을 작성하고 만약 s1,s2 가 닫힌 상태라면 p1 -> out 값을 0x01 로 줘서 빨간 불이 켜지게 한다 이때 Pull up 인지 Pull down 인지가 변경되지 않도록 $P1 \rightarrow OUT = (1 < 1) | (1 < 4)$ 해준다. 닫힌 스위치가 무엇인지 체크 후 그 스위치 변수에 값을 변경한다. 코드가 반복되고 스위치가 열린 경우 빨간색 led 를 끄고 s1 스위치 변수가 변경됐다면 파란색(0x04) - 초록색(0x02) - 빨간색(0x01) 순으로 output address 의 값이 변경되도록 하고 s1 스위치 변수를 다른 값으로 변경해줘서 반복을 마치도록 한다. S2 스위치 변수가 변경됐다면 빨간색(0x01) - 초록색(0x02) - 파란색(0x04) output address 의 값이 변경되도록 하고 s2 스위치 변수를 다른 값으로 변경해줘서 반복을 마치도록 한다.

3. Discussion & Conclusion (10%)

GPIO 에 입력 값을 설정해 스위치 제어해보는 실험을 진행하였다. 저항이 높은 위치에 있는 Pull up 을 활용하여 Floating 방지와 스위치가 닫힌 지를 체크하였다. 체크 된 스위치는 다시 스위치가 열렸을 때 LED 색상이 한번만 변경되도록 반복문을 작성하였다. 자동으로 LED 색상이 변경되는 것이 아닌 사용자가 스위치를 이용해 색상을 변경할 수 있다는 점이 신기하였고 그 코드를 작성해볼 수 있어서 색다른 경험을 한 것 같다. 수업에 참여하지 못해서 작성한 코드를 임베디드 시스템에 확인을 하지 못하였는데 그 점이 아쉽고 어려웠다.

4. Reference(s)

5. Code (20%)

FILE NAME (ex: main.c)

```
#include "msp.h"

#define S1BUT ((P1->IN&(1<<1))==0X00)
#define S2BUT ((P1->IN&(1<<4))==0x00)

#define BS_INIT 1
#define BS_DOWN 2
#define BS_UP 3

void main(void)
{
    int pos=0;

    int s1but_bs;

    int s2but_bs;

    P1->DIR&=~(1<<4|1<<1);

    P1->DIR|=1<<0;

    P1->REN=(1<<4)|(1<<1);

    P1->OUT|=(1<<4)|(1<<1);

    P2->DIR|=(1<<2)|(1<<1)|(1<<0);

    P2->OUT=1<pos;

    WDT_A->CTL = WDT_A_CTL_PW | WDT_A_CTL_HOLD;// stop watchdog timer

    while(1){

        if(S1BUT|S2BUT){

            P1->OUT=(1<<1)|(1<<4)|(1<<0);

            if(S1BUT) s1but_bs=BS_DOWN;
```

```

        if(S2BUT) s2but_bs=BS_DOWN;

    }

    else{

        P1->OUT=(1<<1)|(1<<4)|(0x00);

        if(s1but_bs==BS_DOWN){

            P2->OUT=4>>pos;

            if(pos<3) pos++;

            else pos=0;

            s1but_bs=BS_UP;

        }

        if(s2but_bs==BS_DOWN){

            if(pos<3) pos++;

            else pos=0;

            P2->out=1<<pos;

            s2but_bs=BS_UP;

        }

    }

}

```

FILE NAME (ex: main2.c)