

마이크로프로세서응용설계, 2019년 1학기

실습 3

Push Switch 디바이스 드라이버 및
응용 프로그램 작성

컴퓨터공학부

I/O 장치들

- I/O 장치들의 물리 주소, 디바이스 노드, 주번호

Num	Device	Address	Major
1	LED	0x0800_0016	260
2	Seven Segment (FND)	0x0800_0004	261
3	Dot Matrix	0x0800_0210	262
4	Text LCD	0x0800_0100	263
5	Buzzer	0x0800_0020	264
6	Push Switch	0x0800_0050	265
7	3Dip Switch	0x0800_0000	266
8	Step Motor	0x0800_000C	267
EN	Demo Register	0x0800_0300	N/A

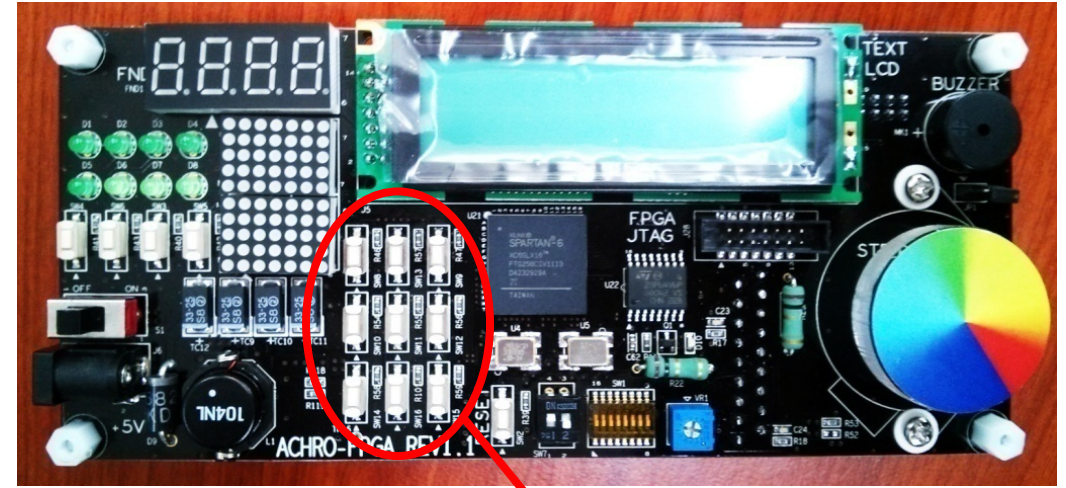
Push Switch

■ Address map

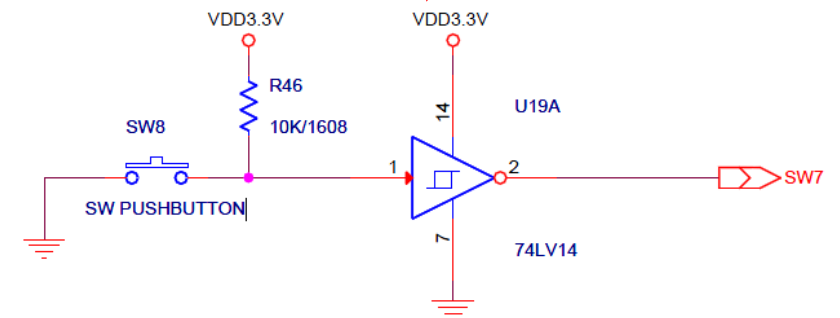
- 1번: 0x0800_0050
- 2번: 0x0800_0050+2
- 3번: 0x0800_0050+4
- ...

■ 각 스위치 주소에 대한 읽기 동작

- 스위치를 누르고 있는 때 1
- 그렇지 않으면 0



1	2	3
4	5	6
7	8	9



Push Switch 디바이스 드라이버 구조

- 각 함수들의 구현은 Lab2의 LED driver와 동일함

push_switch_driver.c

```
// header files and device definitions
#include ...
#define ...

// global variable definitions
...

// functions declarations
...

// device file operation
struct file_operations csemad_push_switch_fops = { ... };

// initialize push switch port
int __init csemad_push_switch_init(void) { ... }

// release and exit push switch port
void __exit csemad_push_switch_exit(void) { ... }

// read from push switch port
ssize_t csemad_push_switch_read(struct file *inode, char *gdata, size_t length, loff_t *off_what) { ... }

module_init(csemad_push_switch_init);
module_exit(csemad_push_switch_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("CSEMAD");
```

Push Switch 테스트 프로그램 구조

test_push_switch.c

```
// header files and device definitions
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <signal.h>

#define MAX_BUTTON 9
unsigned char quit = 0;

void user_signal1(int sig) { quit = 1; }

int main(void){
    int i;
    int dev;
    int buff_size;
    unsigned char push_sw_buff[MAX_BUTTON];
    // push switch device 오픈 및 에러 처리
    ....

    (void)signal(SIGINT, user_signal1);

    buff_size = sizeof(push_sw_buff);
    printf("Press <ctrl+c> to quit. \n");

    while(!quit) {
        usleep(400000); // 400ms(400*1000us) 동안 sleep하면서 대기
                        // 필요에 따라 적당히 값을 조절할 것
        // push switch 포트에서 9개의 스위치의 값을 읽어들이
        ....
    }

    // push switch device 닫음
    return 0;
}
```

■ ctrl+c

- 키 입력시 응용 프로그램이 user_signal1() 호출
- 변수 quit 값을 1로 만들고 main()의 while loop에서 빠져나올 수 있게 됨

실습 3

- 응용 프로그램 코드: test_push_switch.c
- 동작
 - Push switch 9개의 값을 주기적으로 읽어 들여 현재 어떤 스위치가 눌렸는지 저장하고 그 값을 화면에 출력
- 디바이스 드라이버 구조 및 Makefile은 LED 실습과 매우 유사하므로 이전 실습의 내용을 참고하여 만들것