

마이크로프로세서 응용 설계  
(Microprocessor Application Design)  
Spring 2019

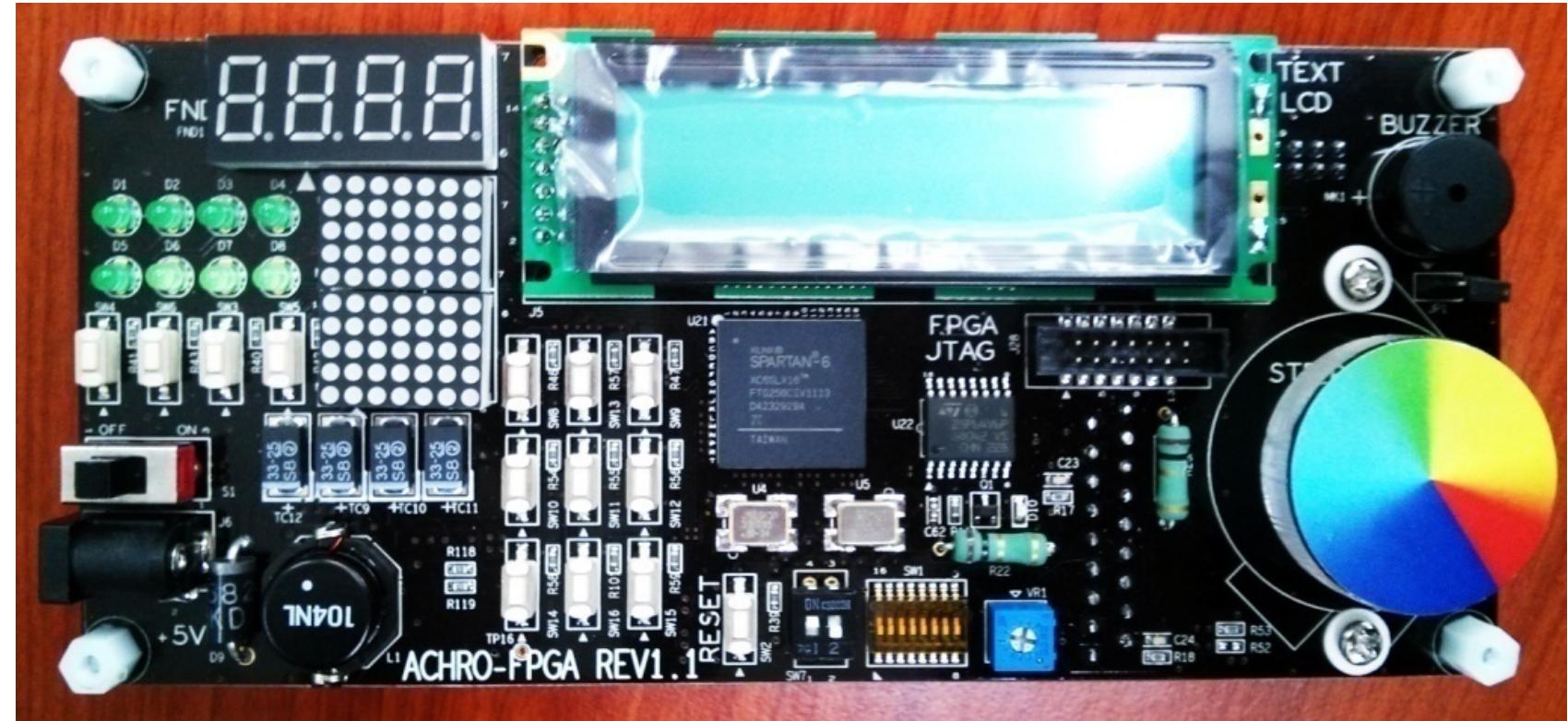
실습 2

LED 디바이스 드라이버 및 응용 프로그램 작성

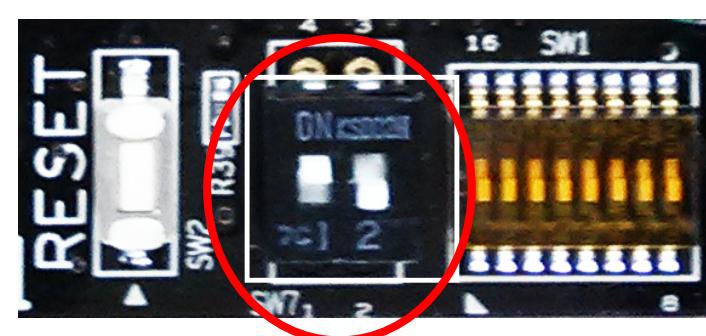
전북대학교 컴퓨터공학부

# I/O Board Overview

- I/O board 외관



- I/O board 스위치 설정



# I/O 장치들

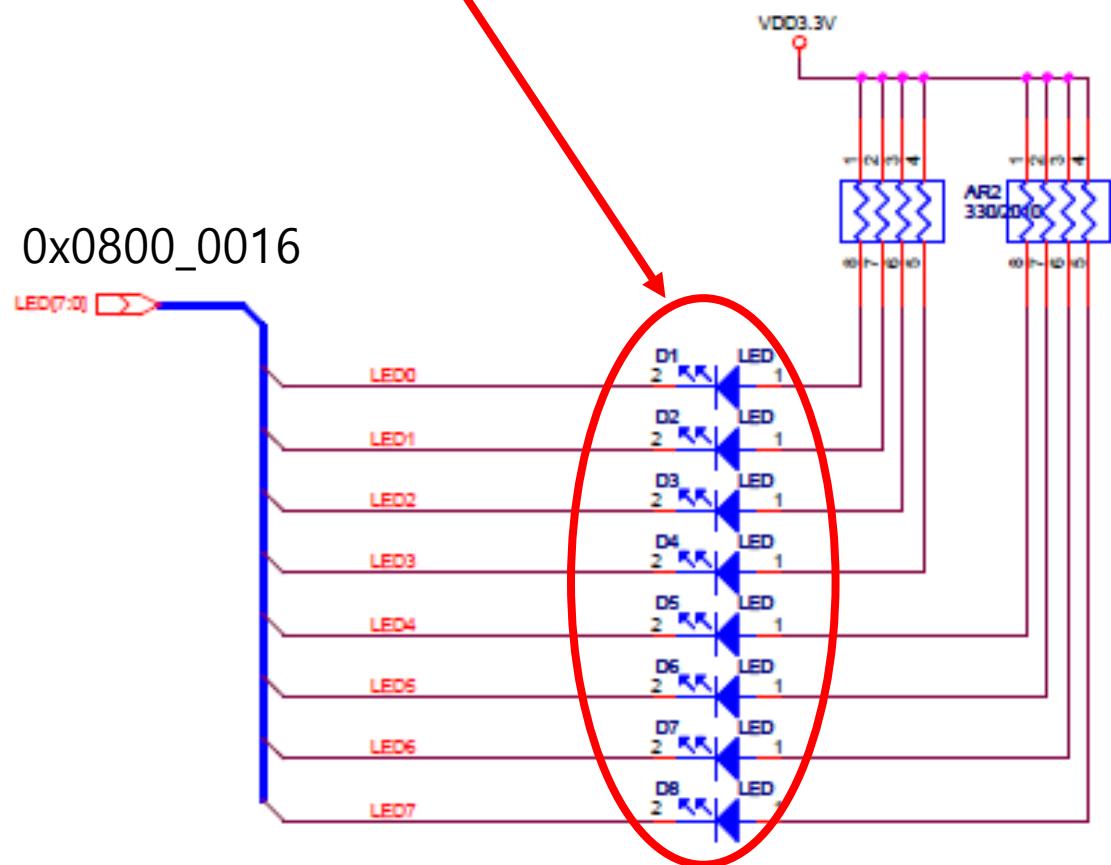
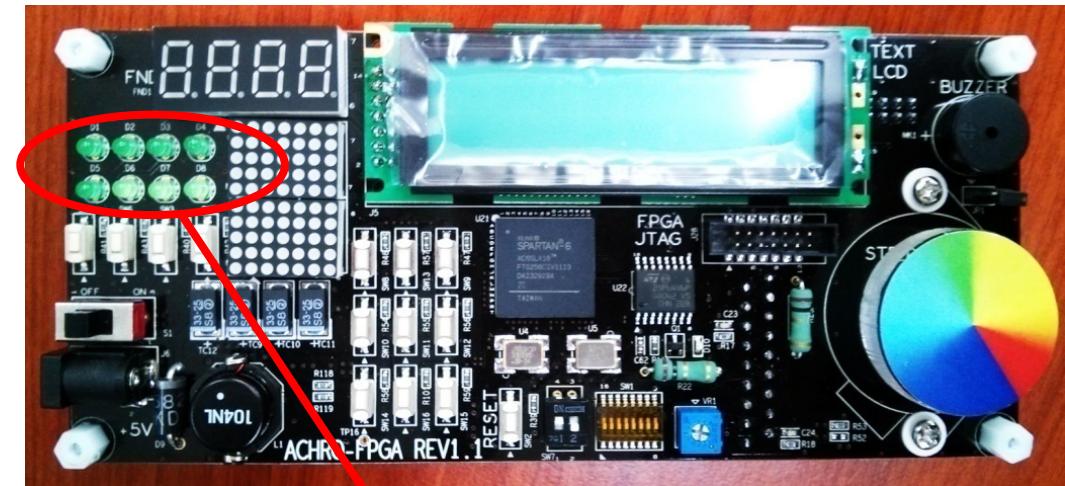
## I/O 장치들의 물리 주소, 디바이스 노드, 주번호

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

# LED 회로

- 물리 주소 0x0800\_0016에 쓰는 값에 따라 binary encoding으로 LED들이 켜짐

- 0x1: LED 0 ON
- 0x2: LED 1 ON
- 0x3: LED 1, 0 ON
- 0x4: LED 2 ON
- 0x5: LED 2, 0 ON
- ...



# LED 디바이스 드라이버 전체 구조

## led\_driver.c

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

// global variable definitions
...

// functions declarations
...

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

// initialize LED port
int __init csemad_led_init(void) { ... }

// release and exit LED port
void __exit csemad_led_exit(void) { ... }

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

// write to LED port
ssize_t csemad_led_write(struct file *inode, char *gdata, size_t length, loff_t *off_what) { ... }

module_init(csemad_led_init);
module_exit(csemad_led_exit);

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

# LED 디바이스 드라이버

## - header files and device definitions

```
#include <linux/module.h>
#include <linux/fs.h>
#include <linux/init.h>
#include <linux/slab.h>
#include <linux/platform_device.h>
#include <linux/delay.h>

#include <asm/io.h>
#include <asm/uaccess.h>
#include <linux/kernel.h>
#include <linux/ioport.h>
#include <linux/module.h>
#include <linux/fs.h>
#include <linux/init.h>
#include <linux/version.h>
#include <linux/types.h>

#define CSEMAAD_LED_MAJOR 260          // ioboard led device major number
#define CSEMAAD_LED_NAME "csemad_led"  // ioboard led device name

#define CSEMAAD_LED_ADDRESS 0x08000016 // physical address
```

# LED 디바이스 드라이버

## - device definitions and file operation

```
//Global variable
static int ledport_usage = 0;
static unsigned char *csemad_led_addr;

// define functions...
ssize_t csemad_led_write(struct file *inode, const char *gdata, size_t length, loff_t *off_what);
ssize_t csemad_led_read(struct file *inode, char *gdata, size_t length, loff_t *off_what);
int csemad_led_open(struct inode *minode, struct file *mfile);
int csemad_led_release(struct inode *minode, struct file *mfile);

// define file_operations structure
struct file_operations csemad_led_fops =
{
    .owner      = THIS_MODULE,
    .open       = csemad_led_open,
    .write     = csemad_led_write,
    .read      = csemad_led_read,
    .release   = csemad_led_release,
};
```

### ■ int ledport\_usage

- led\_device를 동시에 열지 못하도록 제어하기 위해 사용

### ■ csemad\_led\_addr

- led\_device의 물리주소에 대응하는 커널 공간에서의 가상 주소

# LED 디바이스 드라이버

## - initialize and release LED port

```
// initialize LED port
int __init csemad_led_init(void)
{
    LED port의 주번호, 디바이스 이름, file operation 등록; // register_chrdev() 사용
    LED port가 제대로 등록되지 않았을 경우 경고 메세지 출력하고 종료;
    csemad_led_addr ← LED port의 물리 주소에 대한 커널 공간 주소 // ioremap() 사용
    return 0;
}

// release LED port
void __exit csemad_led_exit(void)
{
    LED port에 대해 할당된 가상주소를 해제; // iounmap() 사용
    LED port 등록 해지; // unregister_chrdev() 사용
}
```

- int register\_chrdev(unsigned int major, const char\* name, struct file\_operations \*fops)
- int unregister\_chrdev(unsigned int major, const char \*name)
- void \*ioremap(unsigned long phys\_addr, unsigned long size);
  - 물리주소에 대한 가상주소 할당
  - phys\_addr: 변환할 물리 주소, size: 물리 주소 공간의 바이트 크기
- void iounmap(void \*addr);
  - 할당된 가상주소 해제
  - addr: 해제할 가상 주소

# LED 디바이스 드라이버

## - open and close LED port

```
// open LED port
int csemad_led_open(struct inode *minode, struct file *mfile)
{
    ledport_usage != 0 이면 -EBUSY 리턴;
    ledport_usage ← 1;
    return 0;
}

// close LED port
int csemad_led_release(struct inode *minode, struct file *mfile)
{
    ledport_usage ← 0;
}
```

# LED 디바이스 드라이버

```
// read from LED port
ssize_t csemad_led_read(struct file *inode, char *gdata, size_t length, loff_t *off_what)
{
    csemad_led_addr로부터 1바이트 데이터 읽음;
    읽은 데이터를 사용자 공간의 버퍼(gdata)로 복사; // copy_to_user() 사용
    if (copy_to_user()의 리턴값이 0이 아닐 경우) return -EFAULT;

    return length;
}

// write to LED port
ssize_t csemad_led_write(struct file *inode, const char *gdata, size_t length, loff_t *off_what)
{
    사용자 공간 버퍼(gdata)로부터 1바이트 데이터를 커널 공간 버퍼로 복사; // copy_from_user() 사용
    if (copy_from_user()의 리턴값이 0이 아닐 경우) return -EFAULT;

    csemad_led_addr에 사용자 공간에서 읽은 1바이트 데이터를 씀;
    return length;
}

■ unsigned long copy_to_user (void * to, const void * from, unsigned long n);
    • to: 복사될 사용자 공간의 주소, from: 원본 데이터의 커널 공간 주소, n: 복사할 데이터의 바이트 크기
■ unsigned long copy_from_user (void * to, const void * from, unsigned long n);
    • to: 복사될 커널 공간의 주소, from: 원본 데이터의 사용자 공간 주소, n: 복사할 데이터의 바이트 크기
```

## ■ LED 테스트 프로그램

- Lecture 4의 문자 디바이스 응용 프로그램 예를 이용해서 작성할 수 있음
- open()/close() 함수로 LED port를 열고 닫음
- read()를 통해 읽은 값은 8개 led들 중에서 현재 켜져 있는 led들을 이용해 8-bit binary encoding으로 해석
- write()를 이용해 8-bit binary encoding 형태의 값을 쓰는 동작으로 원하는 led(들)을 켜거나 끌 수 있음
- ioctl() 함수는 사용할 필요 없음

# ■ Makefile

```
obj-m = led_driver.o

PWD = $(shell pwd)

all: driver app

driver:
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

app:
    gcc -Wall -static -o test_led test_led.c

clean:
    rm -rf *.ko *.mod.* *.o test_led Module.symvers modules.order .led* .tmp*
```

## ■ driver

- 디바이스 드라이버 빌드 규칙

## ■ app

- 테스트 프로그램 빌드 규칙

## ■ clean

- 빌드에서 생성되는 파일들 제거

# LED 테스트 프로그램 빌드

- 아래의 절차로 빌드, 디바이스 드라이버 적재, 디바이스 노드 생성 수행

```
csemad@cbnucse:~/lab2_led$ ls
Makefile led_driver.c test_led.c
csemad@cbnucse:~/lab2_led$ make
make -C /mnt/nfs/kernel SUBDIRS=/home/csemad/lab2_led modules
make[1]: Entering directory `/mnt/nfs/kernel'
  CC [M] /home/csemad/lab2_led/led_driver.o
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/csemad/lab2_led/led_driver.mod.o
  LD [M] /home/csemad/lab2_led/led_driver.ko
make[1]: Leaving directory `/mnt/nfs/kernel'
gcc -Wall -static -o test_led test_led.c
csemad@cbnucse:~/lab2_led$ sudo insmod led_driver.ko
csemad@cbnucse:~/lab2_led$ lsmod
Module           Size  Used by
led_driver        1573   0
csemad@cbnucse:~/lab2_led$ sudo mknod /dev/csemad_led c 260 0
csemad@cbnucse:~/lab2_led$ ls -l /dev/csemad_led
crw-r--r-- 1 root root 260, 0 Apr  9 08:06 /dev/csemad_led
csemad@cbnucse:~/lab2_led$
```

# 실습 2

- 응용 프로그램 코드 – test\_led.c

- 동작

- 1-255 사이의 숫자를 명령어의 인자로 입력 받아 이진수에 해당하는 LED들을 점등
- LED port로부터 값을 읽어 출력 – 인자로 입력 받은 값과 같은 값이 출력되어야 함

```
csemad@cbnucse:~/lab2_led$ sudo ./test_led 1  
Current LED Value : 1
```

```
csemad@cbnucse:~/lab2_led$
```

