**Embedded System Software**

**HW2**

과목 명 : 임베디드 시스템 소프트웨어

담당 교수 : 박성용 교수님

20141544 안시현

개발 기간 : 5/8 ~ 5/13

**프로젝트 제목 : Embedded System Software HW#2**

**제출일 : 5/15**

1. **개발 목표**

**System call programming, module programming, 디바이스 드라이버 구현 등, 실습 시 간 때 배운 내용을 활용하여 프로그램을 작성한다**

**(1) Device driver(fpga\_fnd, fpga\_led, fpga\_dot, fpga\_text\_lcd)와 timer module을 포함 한 하나의 module을 구현한다.**

**(2) Parameter들을 받아서 하나의 변수로 만들어 return 해주는 system call을 구현한다.**

**(3) 구현된 디바이스 드라이버와 system call을 이용하여 간단한 출력을 해주는 응용프로 그램을 구현한다.**

1. **추진 일정**
   1. **추진 일정**

* **5/8~ 5/9 : requirement 분석 및 system call 구현**
* **5/10 ~ 5/12 : device drvier 구현**
* **5/13 : user program 구현**
* **5/14 : 보고서 작성**

1. **연구 결과**
   1. **구현한 함수 및 전역변수**

int my\_dev\_open(struct inode \*minode, struct file \*mfile)

int my\_dev\_release(struct inode \*minode, struct file \*mfile)

long my\_dev\_ioctl(struct file \*inode,unsigned int cmd,unsigned long arg)

static void timer\_func(unsigned long arg)

void action\_fnd()

void action\_led()

void action\_lcd()

void action\_dot()

void undo\_action()

int \_\_init my\_dev\_init

void \_\_exit my\_dev\_exit(void)

#define IOM\_FND\_ADDRESS 0x08000004

#define IOM\_LED\_ADDRESS 0x08000016

#define IOM\_LCD\_ADDRESS 0x08000090

#define IOM\_DOT\_ADDRESS 0x08000210

#define GET\_VALUE 0x0000000FF

static unsigned char \*iom\_fpga\_fnd\_addr;

static unsigned char \*iom\_fpga\_lcd\_addr;

static unsigned char \*iom\_fpga\_led\_addr;

static unsigned char \*iom\_fpga\_dot\_addr;

int dev\_usage;

struct timer\_list timer;

int count,fnd\_start,fnd\_value,interval,i;

int name\_offset,name\_dir,id\_offset,id\_dir,fnd\_count;

char val[4] = {0,0,0,0};

unsigned char blank[33];

unsigned char \*name = "AhnSiHyun";

unsigned char \*id = "20141544";

**함수 설명**

int my\_dev\_open(struct inode \*minode, struct file \*mfile)

- **open 함수로서 dev\_count가 0이 아닐시 -EBUSY를 return한다. 전역 변수의 초기화도 진행하였다**

int my\_dev\_release(struct inode \*minode, struct file \*mfile)

* **Release 함수로 dev\_count를 0으로 만들고 종료한다.**

long my\_dev\_ioctl(struct file \*inode,unsigned int cmd,unsigned long arg)

* **Ioctl 함수이다. Param 변수는 4byte stream을 받는다.**

static void timer\_func(unsigned long arg)

* **Timer 함수이다. 실제로 FND , LED , LCD , DOT 출력을 하며 count 변수가 0 되면 종료한다.**

void action\_fnd()

* **Fnd 출력 함수이다.**

void action\_led()

* **LED 출력 함수이다.**

void action\_lcd()

* **LCD 출력 함수이다.**

void action\_dot()

* **DOT 출력 함수이다.**

void undo\_action()

* **종료 전 초기화 함수로서 FND , LCD, LED, DOT 모두 초기 상태로 출력 시켜 놓고 종료한다.**

int \_\_init my\_dev\_init

* **Module initialize 함수이다.**

void \_\_exit my\_dev\_exit(void)

**Module exit 함수이다.**

* **전역변수 설명**

#define IOM\_FND\_ADDRESS 0x08000004 // FND ADDR

#define IOM\_LED\_ADDRESS 0x08000016 // LED ADDR

#define IOM\_LCD\_ADDRESS 0x08000090 // LCD ADDR

#define IOM\_DOT\_ADDRESS 0x08000210 // DOT ADDR

#define GET\_VALUE 0x0000000FF

static unsigned char \*iom\_fpga\_fnd\_addr;

static unsigned char \*iom\_fpga\_lcd\_addr;

static unsigned char \*iom\_fpga\_led\_addr;

static unsigned char \*iom\_fpga\_dot\_addr;

int dev\_usage; // device count

struct timer\_list timer;

int count,fnd\_start,fnd\_value,interval,i;

int name\_offset,name\_dir,id\_offset,id\_dir,fnd\_count;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

fnd\_start : fnd 시작 위치

fnd\_value : fnd 시작 값

interval : timer interval 값

count : timer 진행 횟수

name\_offset : LCD에서 name시작 위치.

name\_dir : ->, <- 진행 방향

id\_offset : LCD에서 id의 시작 위치

id\_dir : ->, <- 진행 방향

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

char val[4] = {0,0,0,0};

unsigned char blank[33];

unsigned char \*name = "AhnSiHyun"; // 이름을 저장

unsigned char \*id = "20141544"; // 학번을 저장

* 1. **추가 구현**

Write 대신 ioctl로 구현하였다. 우선, unsigned int type의 param 변수를 설정하여 4byte stream을 받는다. 4byte stream은 1byte씩 FND 시작 위치, FND 시작 값, time interval, count 순서로 저장하였다.

Copy\_from\_user 함수를 통해 param 변수에 4byte stream 값을 읽어 온 후, 0xFF와 & 연산 후 shift 연산으로 1byte씩 값을 꺼내 저장하였다. 그 후 timer를 setting하고 추가하여 구현하였다.

long my\_dev\_ioctl(struct file \*inode,unsigned int cmd,unsigned long arg)

{

unsigned int param; // 4byte stream을 받을 변수이다

if(copy\_from\_user(&param,(char \*)arg,sizeof(unsigned int))){

return -1;

}

printk("dev\_ioctl is called\n");

if(cmd!=0){ // wrong input

printk("Wrong cmd input\n");

return -1;

}

fnd\_start = (param >> 24) & GET\_VALUE; // get start pos

fnd\_value = (param >> 16) & GET\_VALUE; // get start value

interval = (param >> 8) & GET\_VALUE; // get interval

count = param & GET\_VALUE; // get count

timer.expires = get\_jiffies\_64() + (interval \* HZ / 10);

timer.data = (unsigned long)&count;

timer.function = timer\_func;

add\_timer(&timer);

return 0;

}

1. **기타**
   1. **소감**

* **이번 프로젝트는 첫 프로젝트에 비해서 수월하게 진행할 수 있었던 것 같다. 처음 시작할 때 추가구현인 ioctl이 조금 이해가 안가서 시간이 걸리기는 하였지만, 강의자료와 실습자료 그리고 google 검색을 통해 모르는 부분을 해결할 수 있었다. System call을 만드는 것은 생각보다 쉬웠다. Kernel을 계속 컴파일 해야 한다는 것이 조금 귀찮긴 했지만 그래도 난이도가 어렵지 않아 즐겁게 진행할 수 있었다. Device driver를 짜는 과정에서 또한 별다른 어려움은 없었고 오히려 수업시간에 배웠던 것들을 다시 한 번 코딩을 하며 되새길 수 있었던 것 같아 좋았다. 세번째 프로젝트가 기대된다.**