**Embedded System Software 과제 2**

**(과제 수행 결과 보고서)**

**과목명: [CSE4116] 임베디드시스템소프트웨어**

**담당교수: 서강대학교 컴퓨터공학과 박 성 용**

**학번 및 이름: 120230200, 김지섭**

**개발기간: 2023. 05. 12. - 2023. 05. 14.**

**최 종 보 고 서**

**I. 개발 목표**

- 각 과제마다 주어지는 주제를 바탕으로 본 과제에서 추구하는 개발 목표를 설정하고 그 내용을 기술할 것.

Timer Device Driver와 해당 Device driver를 사용하는 Example Application구현.

기존에 구현되어 있는 FPGA Device들의 driver를 참고하여, 해당 Driver들을 insmod하지 않아도 FPGA Devive를 control할 수 있는 Device Driver 개발.

Timer interval, Timer count, Timer init를 입력받아, ioctl명령어를 사용하여, init으로 device를 초기화하고, device해당 주기마다 1번씩 총 Timer Count만큼 반복하는 Device driver 구현

해당 Device driver의 기능을 활용할 수 있는 Test Application 개발

**II. 개발 범위 및 내용**

- 자신들이 설계한 개발 목표를 달성하기 위하여 어떠한 내용의 개발을 수행할 지 그 범위와 개발 내용을 기술할 것.

**가. 개발 범위**

Timer Device Driver 및 Test용 Example Application 개발

1. Example Application

* 사용자로부터 Argument로 Device 구동옵션을 받는다.
* ioctl을 사용하여 Device driver에 해당 데이터를 전달한다.
* Ioctl을 사용하여 Timer Device를 구동시킨다.
* 위의 과정에서 2번의 ioctl 명령어를 사용한다.

실행 예시

\*TIMER\_INTERVAL: HZ 값 1~100 (0.1~10초)

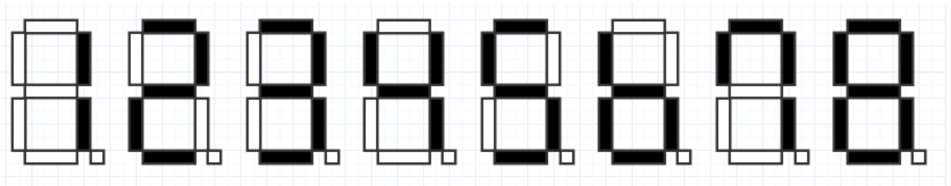
\*TIMER\_CNT: device출력 변경 횟수 (1~100)

\*TIMER\_INIT: FND 에 출력되는 초기 문양과 위치 (0001~8000)

1. Timer Device Driver

* 사용자 프로그램에서 ioctl 을 통하여 전달된 옵션을 기준으로 4 가지 device (FND, LED, DOT, TEXT\_LCD) 의 요구사항에 따라 device들에 동시 출력한다.
* 모든 device의 값들이 바뀌는 시간은 TIMER\_INTERVAL을 기준으로 한다. Device driver의 이름은 /dev/dev\_driver로 통일한다. (major number : 242)
  1. LED
* 현재 fpga\_fnd 에서 출력 중인 문양의 번호를 나타낸다.

\* D1:1번, D2:2번, D3:3번, D4:4번, D5:5번, D6:6번, D7:7번, D8:8번

* TIMER\_CNT 횟수만큼의 출력이 끝나면 fpga\_led 의 불을 꺼준다.
  1. Dot
* 현재 fpga\_fnd 에서 출력 중인 문양과 같은 모양의 문양을 출력한다. fpga\_fnd 의 문양이 바뀐다면, fpga\_dot 도 fpga\_fnd 와 같은 문양으로 함께 바뀐다.
* TIMER\_CNT 횟수만큼의 출력이 끝나면 dot 의 불을 꺼준다.
  1. FND
* 초기 상태: TIMER\_INIT 에서 전달된 문양과 위치에 따라 FND에 출력한다.
* 이후 상태: TIMER\_INTERVAL 마다 다음 문양를 지정된 위치에 출력한다.
* TIMER\_CNT 횟수만큼의 출력이 끝나면 FND를 0000 으로 초기화한다.
* 다음 문양이란 현재 수에서 1 증가된 수를 의미한다. 단,8의 다음 문양은 1이다.
* 문양이 출력되는 위치는 한번의 로테이션이 끝날때 마다 우측으로 이동한다.
* 만약 (왼쪽에서부터) 4 번째 자리에서 로테이션이 끝났다면, 1번째 자리로 이동한다  
  Ex2) 0008à0001 (TIMER\_INIT: 0008, TIMER\_CNT: 2)  
  Ex3) 0007à0008à0001à...à0006à7000 (TIMER\_INIT: 0007, TIMER\_CNT: 9)
  1. Text LCD
* 초기 상태: 첫 번째 줄에는 자신의 학번을, 다음줄에는 이름을 영문으로 출력한다.
* 이때 두 문장 모두 left align 시킨다. (가장 왼쪽 글자가 가장 왼쪽에 위치)
* 텍스트가 없는 나머지 칸은 “ “이 출력되도록 한다.
* 이후 상태: TIMER\_INTERVAL 마다 오른쪽으로 한 칸씩 shift 이동을 하고, 가장 우측 글자가 가장 오른 쪽 칸에 도달한 경우, 다시 왼쪽으로 shift 이동을 시작한다. 왼쪽 shift 이동도 마찬가지로, 왼쪽 글자가 가장 왼쪽 칸에 도달한 경우, 다시 오른쪽으로 shift 이동을 시작한다. (각 줄은 다른 형태로 이동을 하게 된다.).
* TIMER\_CNT 횟수만큼의 출력이 끝나면 LCD 의 불을 꺼준다.

**나. 개발 내용**

1. Flow Chart

텍스트, 도표, 기술 도면, 평면도이(가) 표시된 사진

자동 생성된 설명

1. Example Application 구현

Argument Error Check

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Test application 실행 시, 입력 받는 arguments의 값이 정해진 값과 일치하는지 확인 후, 일치하지 않는 경우 예외처리 하는 code입니다.

Run ioctl

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

ioctl을 사용하여 device를 control하는 code입니다.

SET\_OPTIONS는 value값을 device driver에 넘겨, initialize를 수행하도록 하는 option command 입니다.

START\_TIMER는 SET\_OPTIONS에서 setting된 값을 기반으로 timer기능을 수행하는 option command입니다.

1. Timer Device Driver 구현

Define & mapping function

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Device driver에 구현된 function을 file function과 mapping하기 위한 code입니다.

Set Options

텍스트, 소프트웨어, 스크린샷이(가) 표시된 사진

자동 생성된 설명

ioctl로 SET\_OPTIONS command를 요청하였을 때, 실행 code입니다.

Value값을 user memory 영역에서 copy한 후, 해당 data를 기반으로 device들을 setting합니다.

Start Timer

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

ioctl로 START\_TIMER command를 요청하였을 때, 실행 code입니다.

기존에 setting되어 있는 값을 기반으로 timer 기능을 수행합니다.

Set LED

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

LED를 입력받은 값으로 setting하는 code입니다.

Set Dot

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Dot를 입력받은 값으로 setting 하는 code입니다.

Update LED, Dot

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

본 코드에서 LED, Dot 의 set/write function에서 사용되는 값은 아래와 같은 미리 정의된 배열 값을 활용합니다.

텍스트, 폰트, 스크린샷, 친필이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

Set & Get FND

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

FND에 값을 setting하거나 FND에서 값을 가져오는 code입니다.

Update FND

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

FND에서 값을 읽어와 update하는 code입니다.

처음 입력된 setting 값을 사용하여, 현재 값이 1 cycle을 돌았다면, shift를 수행합니다.

Set Text LCD

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Text LCD에 지정된 값을 출력하는 code입니다.

Shift(move) Text LCD

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

각 line별로 현재 이동방향의 끝이 빈칸이 아니라면, 반대방향으로 shift하도록 Flag를 update하는 code입니다.

Device off

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Timer 종료 후, 초기화 code입니다.

**III. 추진 일정 및 개발 방법**

- 자신들이 설정한 개발 목표를 달성하기 위한 개발 일정을 설정하고, 각 요소 문제를 해결하기 위해서 어떤 방법을 사용할 지 기술할 것.

1. **추진 일정**

**텍스트, 스크린샷, 번호, 라인이(가) 표시된 사진

자동 생성된 설명**

1. **개발 방법**

개발해야 하는 기능들을 control해야 하는 device단위로 구분하여 waterfall 방식으로 개발하였습니다.

우선 device driver의 skeleton code로, led를 ioctl로 control하는 driver를 구현하였고, 이를 실행시킬 수 있는 test application prototype을 구현하였습니다.

이후 device별로 수행할 기능을 각각 순차적으로 구현하였습니다.

**IV. 연구 결과**

- 최종 연구 개발 결과를 자유롭게 기술할 것.

ioctl 을 활용하여, device driver를 비교적 자유롭게 control할 수 있었습니다. 단순한 read, write 기능만 존재한다면, user level에서 구현의 복잡도가 높았을 것이라 생각합니다.

어느정도 static한 동작을 수행해야하는 device라면, ioctl을 적극적으로 활용한 driver를 구현하여, user level 개발자들의 편의를 도모할 수 있을 것 같습니다.

**V. 기타**

- 본 설계 프로젝트를 수행하면서 느낀 점을 요약하여 기술하라. 내용은 어떤 것이든 상관이 없으며, 본 프로젝트에 대한 문제점 제시 및 제안을 포함하여 자유롭게 기술할 것.

강의를 통해 배운 Device Driver를 직접 개발해보는 경험을 통해, 강의에 대한 이해도가 높아진 것 같습니다. 처음엔 막막했지만, 배운 내용을 리마인드 하면서 문제를 해결할 수 있는 실마리들을 찾을 수 있었던 것 같습니다. 문제해결 능력, 개발 경험, 강의 이해 등 다양한 방면에서 도움이 많이 되는 프로젝트였던 것 같습니다.

 본 프로젝트의 학번, 이름이 device driver에 들어가 있도록 설계되어 있습니다. 과제 목적으로는 괜찮은 프로젝트일지 몰라도, device driver의 출력 값과 형태를 좀 더 자유롭게 할 수 있도록 application level에서 입력 받는 형태로 구현했으면 어땠을까 하는 생각이 들었습니다.