**Embedded 3st Device Driver & Interrupt**

**(설계 프로젝트 수행 결과)**

**과목명: 임베디드 시스템 프로그래밍**

**담당교수: 서강대학교 컴퓨터공학과 박 성 용**

**학번 및 이름: 20141544 안시현**

**개발기간: 2019. 05. 28. -2019. 05. 29.**

**최 종 보 고 서**

**I. 개발 목표**

Module programming, 디바이스 드라이버 구현, interrupt 등, 실습 시간 때 배운 내용을 활용하여 간단한 stopwatch 프로그램을 작성한다.

(1) fpga\_fnd device driver와, timer module, interrupt를 포함한 stopwatch 기능을 가진 module을 구현한다.

(2) module을 실행시키는 application 을 구현한다.

**II. 개발 범위 및 내용**

**가. 개발 범위 및 개발 내용**

1. Module을 구현한다.

* Stopwatch를 구현하기 위해 FND device를 control한다.
* Stopwatch 구현을 위해 1초마다 생성되는 timer를 만든다.
* Stopwatch를 끝내기 위해 필요한 timer를 구현한다.
* 4가지 각 button(home, back, vol+, vol-)에 대해 Interrupt handler를 구현한다.
* Process를 sleep 시키고, wait 시키는 기능을 구현한다.

1. User program을 구현한다

**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. 또한 각 연구원의 역할을 분명히 기술할 것.

**가. 추진 일정**

**5/28 : project 명세서 분석**

**5/29 : 분석 및 보고서 작성**

**나. 개발 방법**

1. Module을 구현한다.

* fpga\_fnd device driver와, timer module, interrupt를 포함한 stopwatch 기능을 가진 module을 구현한다.
* Homework 2에서 fpga\_fnd 디바이스를 컨트롤 했던 코드를 참고하였다.
* Stopwatch 구현을 위해 1초마다 생성되는 timer를 만든다.
* 실습시간 timer 예제코드를 참고했다.
* 4가지 각 button에 대한 interrupt handler를 구현한다.
* 실습시간에 받은 interrupt handler 코드를 참고하였다.
* Process를 sleep 시키고, wait 시키는 기능을 구현한다.

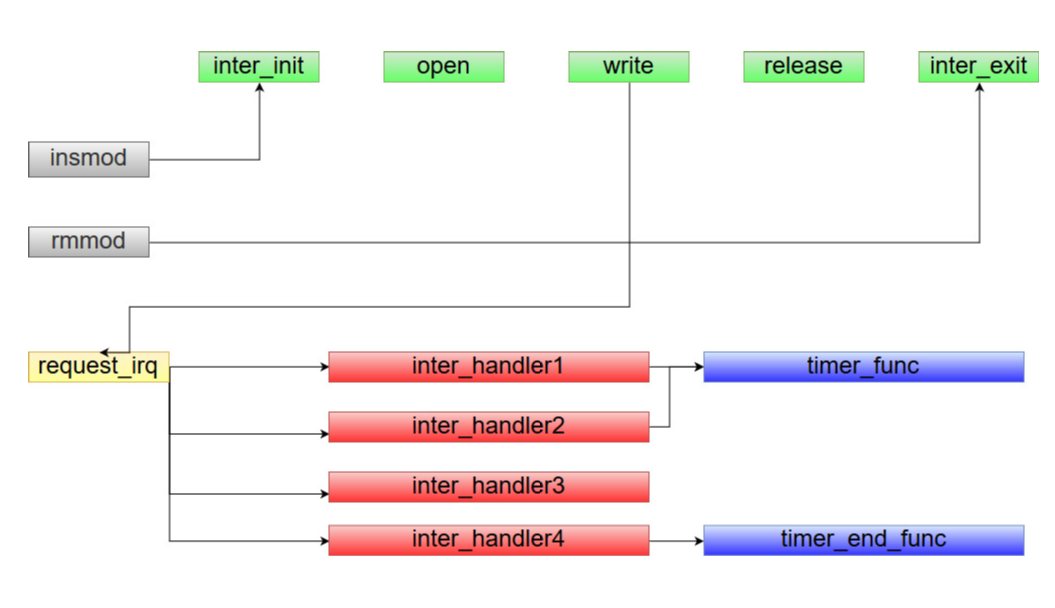
1. Module을 실행시키는 application을 구현한다.

* Device를 open후 write를 호출하고 device driver에서 write함수를 구현할 때 interrupt를 받을 수 있도록 하였다.

**IV. 연구 결과**

- 최종 연구 개발 결과를 자유롭게 기술할 것.

**1. 합성 내용:**



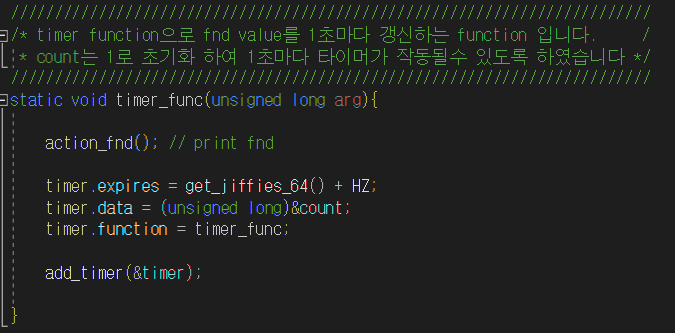
**2. 제작 내용:**

1. Module 개발

**static void timer\_func(unsigned long timeout)**

역할 : 1초마다 count를 1씩 증가 시킨다.

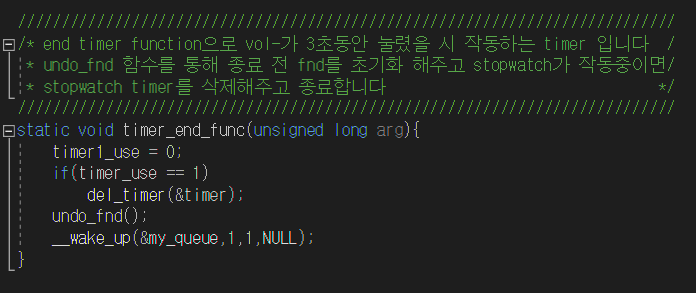
구현 방법 : timer module을 사용하여 1초마다 timer structure의 count variable을 1씩 증가 시킨다.



**static void timer\_end\_func(unsigned long timeout)**

역할 : program이 종료될 수 있도록 flag를 설정한다.

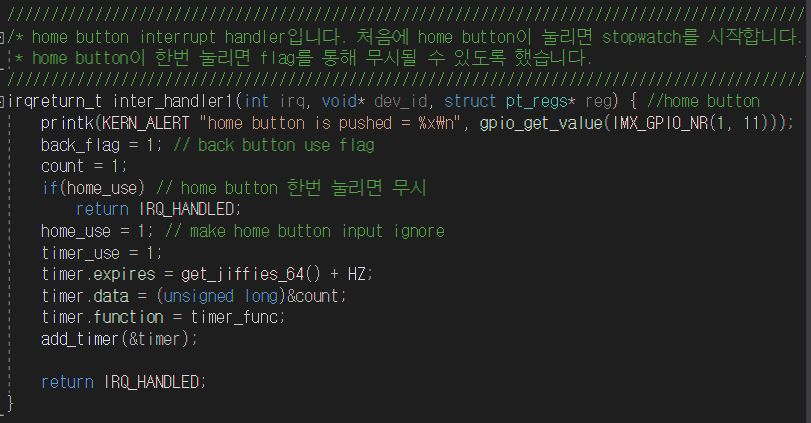
구현 방법 : timer1\_count를 통해 프로그램을 종료될 수 있도록 한다.



**irqreturn\_t inter\_handler1(int irq, void \*dev\_id, struct pt\_regs \*reg)**

역할 : timer를 추가하여 stopwatch를 시작한다.

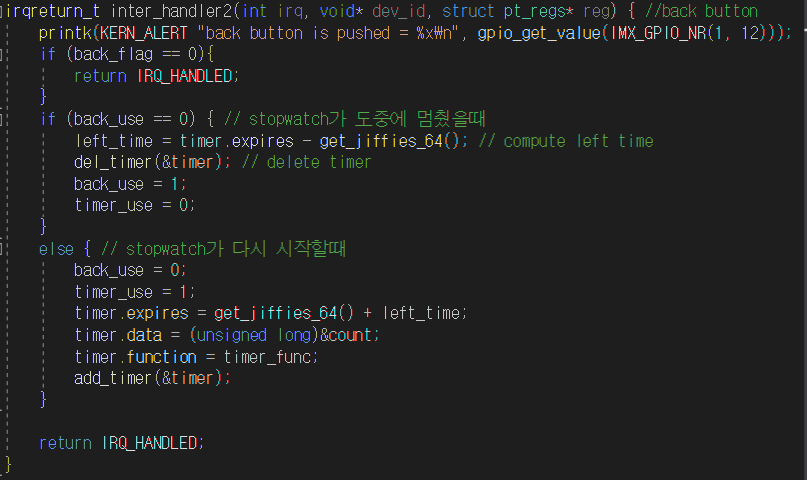
구현 방법 : interrupt가 발생하면 timer를 add하여 timer\_func가 수행되도록 한다.



**irqreturn\_t inter\_handler2(int irq, void \*dev\_id, struct pt\_regs \*reg)**

역할 : timer를 제거하여 stopwatch를 멈춘다.

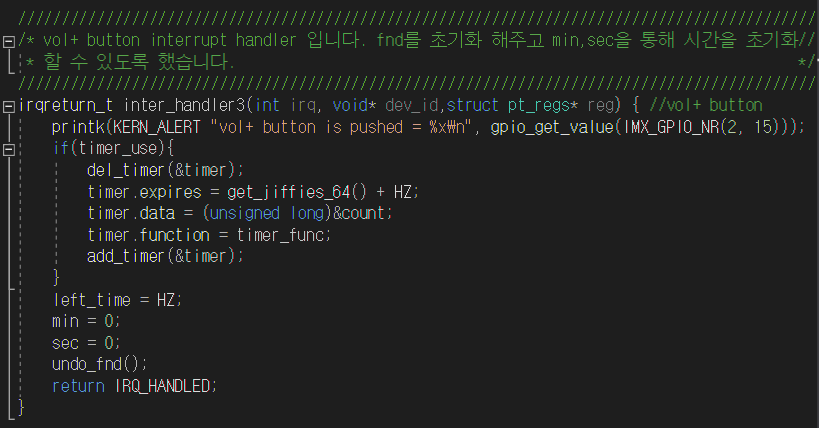
구현 방법 : interrupt가 발생하면 timer를 delete하여 더 이상 count가 올라가지 않도록 하고, 다시 실행시 남은 시간 만큼 timer를 add한다



**irqreturn\_t inter\_handler3(int irq, void \*dev\_id, struct pt\_regs \*reg)**

역할 : timer의 count를 0으로 초기화 한다.

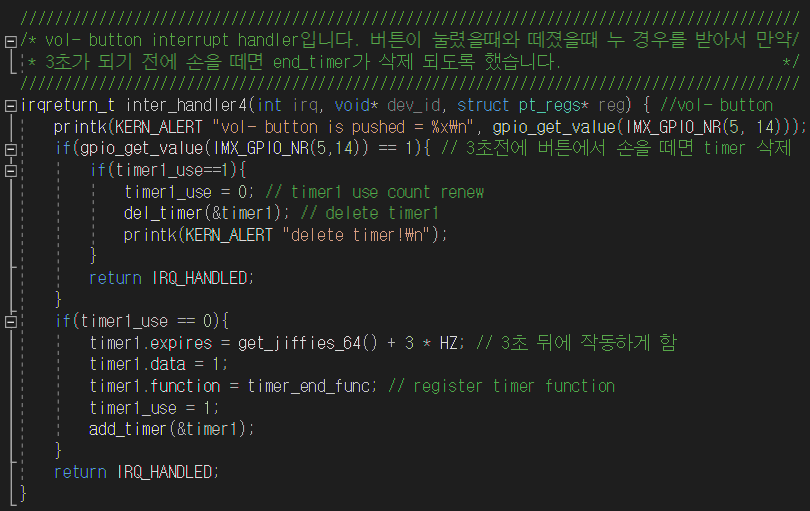
구현 방법 : interrupt가 발생하면 timer를 delete하여 더 이상 count가 올라가지 않도록 하고, count의 값을 0으로 설정하여 준다.



**irqreturn\_t inter\_handler4(int irq, void \*dev\_id, struct pt\_regs \*reg)**

역할 : switch가 3초 동안 눌렸는지 판단하여 프로그램을 종료 시킨다.

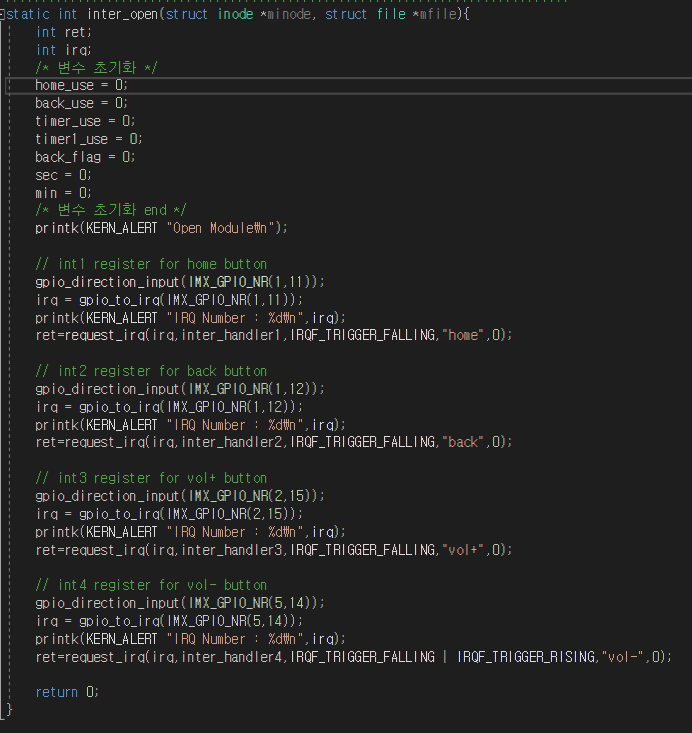
구현 방법 : interrupt가 발생하면 timer를 이용하여 3초 동안 interrupt가 발생 하였는지 판단하고, 3초 동안 interrupt가 발생한 경우 timer를 add하고, 3초 전에 종료 됐을 시에 timer를 delete하여 종료되지 않도록 한다.



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

역할 : 변수들을 초기화하고 interrupt를 등록한다..

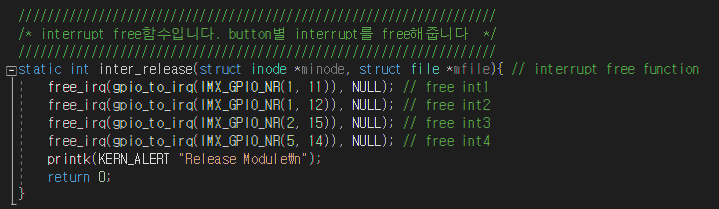
구현 방법 : 각각 변수들을 초기화 하고 request\_irq를 이용하여 각각의 switch의 interrupt 발생 조건을 설정한다.



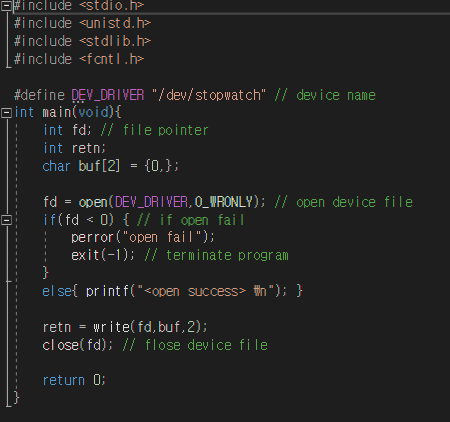
int inter\_release(struct inode \*minode, struct file \*mfile)

역할 : open했던 device들을 종료 시킨다.

구현 방법 : timer를 delete하고 interrupt를 free 시켜준다.



1. User program



**3. 시험 및 평가 내용:**

주어진 요구 사항에 대해 만족스러운 output을 만들어 냈다. 여러 차례 시도를 하였으며, 지속적인 업데이트를 통해 발견되는 문제에 대해 업데이트를 통해 보완해 나갈 수 있을 것으로 기대한다. 지난 2차 과제와 비슷한 유형의 과제였기 때문에 설계에 있어서 큰 어려움이 없었고, 그로 인해 프로그램의 완성도에 있어서 많은 향상을 얻을 수 있었다.

**V. 기타**

**1.** **연구 조원 기여도**:

20141544 안시현 : 100%

**2.** 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.

Interrupt를 등록하고 직접 제어해 봄으로써 interrupt에 대한 이해도를 높일 수 있었다. 또한 timer를 구현하는 과정에서 저번에 했던 project 내용을 참고할 수 있어서 수월 했고 timer에 대한 복습 효과도 있었던 것 같다.