**Embedded 3st Device Driver & Interrupt**

**(설계 프로젝트 수행 결과)**

**과목명: 임베디드 시스템 프로그래밍**

**담당교수: 서강대학교 컴퓨터공학과 박 성 용**

**학번 및 이름: 20141500, 권태국**

**개발기간: 2017. 05. 25. -2017. 05. 26.**

**최 종 보 고 서**

**I. 개발 목표**

이 프로젝트에서 목표하는 개발 목표는 크게 2가지이다. 첫째, fpga\_fnd device driver와, timer module, interrupt를 포함한 stopwatch 기능을 가진 module을 구현한다. 둘째, 구현된 module을 테스트하는 app을 구현한다. 나는 이러한 2가지 개발 목표를 가지고 개발을 하였다.

**II. 개발 범위 및 내용**

**가. 개발 범위 및 개발 내용**

1. Device Driver 개발.

* 프로젝트 설명서에 설명된 대로 fpga\_fnd 디바이스를 컨트롤한다.
* 주기적으로 특정 handler가 호출되는 타이머 기능을 구현한다.
* module내에서 stopwatch 디바이스를 register한다.
* Interrupt handler들을 등록하고, handler 내에서 프로젝트 설명서에 명시된 기능들을 수행한다.
* Process를 sleep 시키고, wait 시키는 기능을 구현한다.

1. Device driver와 추가된 system call을 활용하는 예제 응용프로그램을 제작한다.

**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**

**5/25 : 설계 및 구현.**

**5/26 : 결과 보고서 작성**

**나. 개발 방법**

위 개발 내용에 해당하는 프로그램을 개발하기 위해서 아래와 같이 개발 방법을 습득 및 적용했다.

1. Device Driver 개발.

* 프로젝트 설명서에 설명된 대로 fpga\_fnd 디바이스를 컨트롤한다.
* Homework 2에서 fpga\_fnd 디바이스를 컨트롤 했던 코드를 참고하였다.
* 주기적으로 특정 handler가 호출되는 타이머 기능을 구현한다.
* Homework 2에서 구현한 타이머 기능을 참고하였다. 단, interrupt context에서는 del\_timer\_sync 사용이 불가함으로, del\_timer를 사용하였다.
* module내에서 stopwatch 디바이스를 register한다.
* Homework 2와 실습 코드를 참고하였다.
* Interrupt handler들을 등록하고, handler 내에서 프로젝트 설명서에 명시된 기능들을 수행한다.
* Interrupt 실습 코드를 참고하였다.
* Process를 sleep 시키고, wait 시키는 기능을 구현한다.

1. Device driver와 추가된 system call을 활용하는 예제 응용프로그램을 제작한다.

* 별다른 참고사항이 없다.

**IV. 연구 결과**

- 최종 연구 개발 결과를 자유롭게 기술할 것.

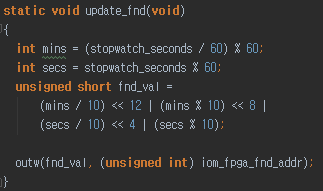
**1. 합성 내용:**

아래에 제작 내용을 참고.

**2. 제작 내용:**

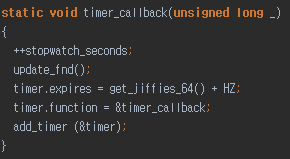
1. Device Driver 개발.

* 프로젝트 설명서에 설명된 대로 fpga\_fnd 디바이스를 컨트롤한다.

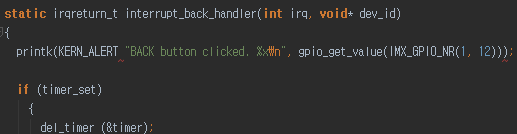


위와 같이 fpga\_fnd 디바이스에 현재 stopwatch 시간을 출력하는 기능을 함수로 만들어서, fnd 디바이스에 현재 시간을 업데이트해야 할 때 위 함수를 호출하도록 하였다.

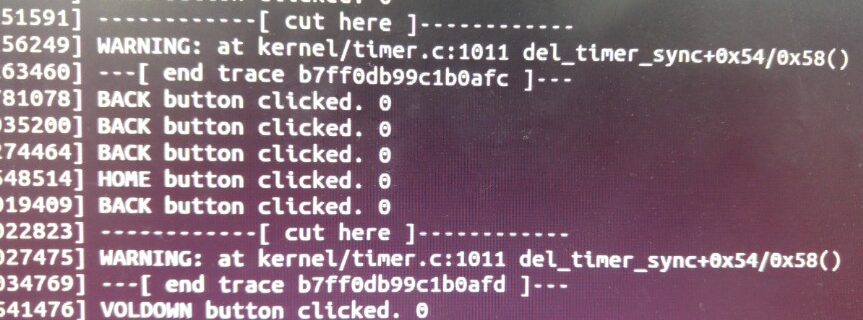
* 주기적으로 특정 handler가 호출되는 타이머 기능을 구현한다.



위와 같이 timer가 주기적으로 실행되도록 해서 stopwatch가 1초마다 시간을 업데이트하도록 하였다.



Interrupt handler 내에서 timer를 해제해야 할 때는 del\_timer를 사용하였다. 왜냐하면 del\_timer\_sync는 interrupt context에서 사용할 수 없기 때문이다.



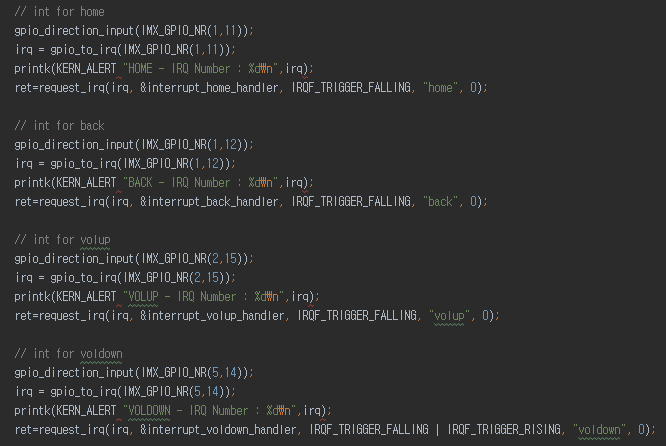
만약 del\_timer\_sync를 사용하면 위와 같은 경고가 뜬다. Del\_timer를 사용하면 아무런 경고도 뜨지 않는다.

* module내에서 stopwatch 디바이스를 register한다.



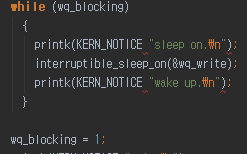
위와 같이 함으로서, 코드에 stopwatch\_major가 설정되어 있으면 설정된 값으로 device 등록을 시도하고, 만약 설정되어 있지 않으면 자동으로 device major number를 얻도록 코드를 작성하였다.

* Interrupt handler들을 등록하고, handler 내에서 프로젝트 설명서에 명시된 기능들을 수행한다.



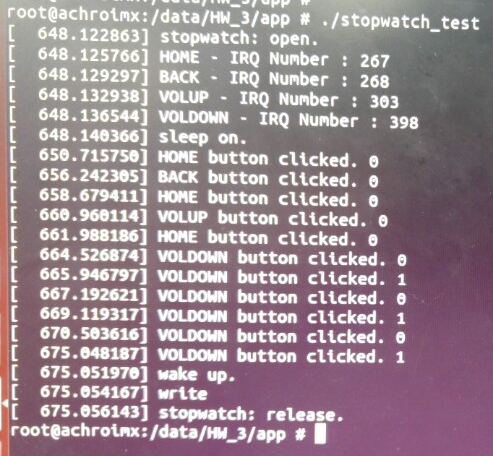
위와 같이 특정 인터럽트에 대해 irq number를 얻고, 그 number에 대해 interrupt handler를 등록하는 코드를 작성하였다.

* Process를 sleep 시키고, wait 시키는 기능을 구현한다.



나는 interruptible한 sleep을 사용하였다. 따라서 while을 돌면서 실제 조건을 만족했는지를 검사해야만 한다.

1. Device driver와 추가된 system call을 활용하는 예제 응용프로그램을 제작한다.



실행 결과는 위와 같다.

**3. 시험 및 평가 내용:**

평가방법은 프로젝트 설명서에 나와있는 흐름대로 평가하면 된다.

여러 가지 테스트를 진행함으로써 보건 및 안정, 내구성을 향상 시켰다.

**V. 기타**

**1.** **연구 조원 기여도**:

나 혼자 했다.

권태국 : 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.

참 재미있었다.