**Embedded System Software HW#3**

**(설계 프로젝트 수행 결과)**

과목 명 : [CSE4116] 임베디드 시스템 소프트웨어

담당 교수 : 박성용 교수님

개발 기간 : 2019. 05. 26 ~ 2019. 05. 29

학번 : 20141602

이름 : 황기덕

**프로젝트 제목 : Embedded System Software HW#3**

**제출일 : 2019년 5월 31일**

**참여 조원 : 황기덕**

1. **개발 목표**

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

1. **개발 범위 및 내용**
   1. **개발 범위, 내용**

* **Device driver**

보드에서 fpga\_fnd 디바이스와 interrupt를 이용한 stopwatch를 구현하고, dev driver를 통해 나타낼 수 있도록 한다.

* **Application program**

Device driver를 open하고 write하고 close해서 보드에 stopwatch를 나타낼 수 있게끔 간단한 응용 프로그램을 구현한다.

1. **추진 일정 및 개발 방법**
   1. **추진 일정**

|  |  |
| --- | --- |
| **일정** |  |
| **5.26** | **요구사항 분석** |
| **5.27 ~ 5.28** | **Dev driver, application program 작성** |
| **5.29** | **보고서 작성** |

* 1. **개발 방법**
* **Device driver**

내가 원하는 버튼이 그 버튼의 원래 기능으로 쓰이지 않아야 하므로 원래 있던 기능을 delete시키고 kernel을 재컴파일 하여야 한다.

stopwatch를 사용하기 위하여 원래의 context를 waitq에 넣고, sleep시킨 후 stopwatch를 돌린다.

매 초마다 초를 표시하는 device가 변경되어야 하므로, timer와 callback 함수를 만든다.

Interrupt가 들어올 시 각 버튼의 기능에 맞게 handling 함수를 구현해준다.

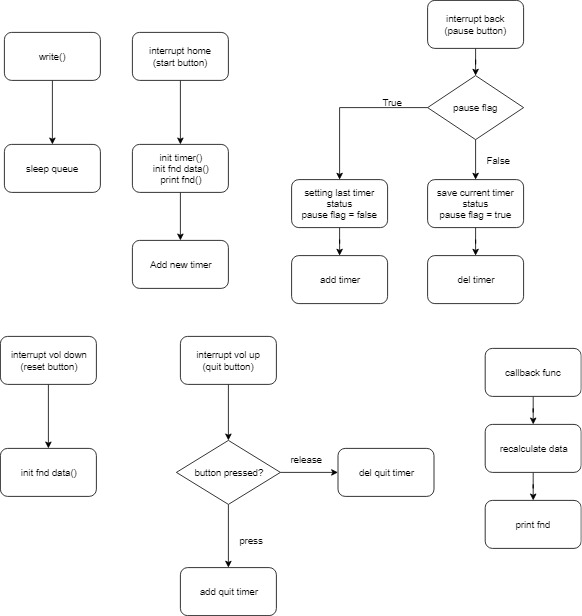
* **Application program**

실행을 해서 만든 dev driver를 open 하고, wirte로 stopwatch를 실행시켜 주며 그 후 close 시킨 후 종료한다.

* 1. **연구원 역할 분담**
* 개인 프로젝트이다.

1. **연구 결과**
   1. **합성 내용**

* **Dev driver**

****

* 1. **제작 내용**
* **Module**

Interrupt 를 사용하기 위하여 우선적으로 kernel을 재컴파일하고 시작해 주었다.

Inter open을 할 때 home, back, vol up, vol down 버튼을 모두 request irq에 등록해 주고 각각 interrupt가 불렸을 때 inter\_handler\_\*\*\* 함수가 불려지도록 해 주었다.

Write에서는 waitq에 새로 등록해주고 인터럽트를 기다리는 상태로 바꾸어 주었다.

먼저 home 버튼에서는 timer set을 setting해 주고 start timer를 해 주었다. 이 때 start timer에서는 pause에서 다시 풀려날 때와 맨 처음 시작할 때를 구분해 주는데, 만약 맨 처음 시작할 때에는 모든 것을 init해 주고 매 초마다 callback func이 불러지게 해 주고 add timer 해 주었다..

pause에서 시작할 경우에는 pause될 때 저장된 status를 다시 저장해주고 add timer 해 주었다.

Back 버튼에서는 timer가 도는 중이라면 중지시키고 del timer 하고 현재 timer status를 저장시켰으며, 중지된 상태라면 다시 돌 수 있게끔 해 주었다.

Vol up 버튼에서는 현재 상태 그대로 reset만 시키는 것이니 init fnd만 해 주었다.

Vol down 버튼에서는 맨 처음 눌리면 quit timer를 만들고, expires를 3초로 둔 후 add timer 시켜 주었다. 만약 3초 전에 release 되면 quit timer를 del timer 해 주었고, 만약 3초가 지나서 quit timer의 callback func이 불려지면 그 안에서 현재 도는 timer를 del 해주고 init fnd 해 준 다음 waitq를 깨워 주고 stopwatch 를 종료시켜 주었다.

* **Application program**

위에서 만든 dev driver를 사용할 수 있게끔 device를 open하고 write, close해 주었다.

1. **기타**
   1. **실행 순서**

* module과 app program을 make 후 target board에 push -> insmod and mknod (dev name = /dev/stopwatch, major number 242)-> app program run
* ./20141602
  1. **연구 조원 기여도**
* 개인 프로젝트이다.
  1. **소감**
* Timer와 interrupt를 사용해 보드가 현재 가지고 있는 기능을 정지시키고 내가 원하는 기능을 구현해 돌려 보았다. 보드 전체가 이런 식으로 코딩되어 있을 거라 생각하니 신기한 느낌이 들었다.