

Fall 2004 Embedded Systems

실험 4. Interrupt and Timer

1. Purpose

Understand how to program the interrupt and timer in the Xscale processor with Linux.

2. Hardware connection

User I/O (keyboard and display) -- PC -- Serial/USB/Ethernet connections -- Embedded board.

3. Problem 4

Program the OS Timer 2 in the Xscale.
The OS Timer 2 should be initialized to interrupt just after 10 seconds. An interrupt service routine for the OS Timer 2 should manage the timer and print a suitable message using `printk`. This interrupt service routine should be executed just one time.

4. Hints

In the Xscale, there are 4 OS Timers. We use OS Timer 2 (OST2 hereafter).
Write a driver module named `ost2.o` for OST2 which performs

```
init_module
    Open the OST2 device
    Set interrupt and specify the OST2 interrupt service routine using
request_irq()
    Set OSMR2 and OIER (for interrupt after 10 seconds).
    Enable interrupt using enable_irq()
cleanup_module
    Disable OST2 irq using disable_irq()
    Free the IRQ number for IST2 using free_irq()
    Close the OST2 device
OST2 interrupt service routine
    save_flags_cli()
    cli()
    Print message using printk that the OST2 interrupt has happened.
    restore_flags()
```

OS Timer 에 대한 설명은 Xscale Developer Manual 을 참고한다.
`request_irq`, `enable_irq`, `disable_irq`, `free_irq` 에 대한 설명은 web 에서 볼 수 있다.
이 Module 의 전반적인 구조는 falinix.com 에서 제공하는 `gpio driver module` 을 참고한다.

5. Test

Application program 은 필요없다. 단지 EzBoard 에서 insmod ost2.o 라고 명령하면 이 module 이 설치될 것이고, 설치 후 10 초후에 interrupt service routine 에 의하여 해당 message 가 화면에 표시되면 성공!

Interrupt 가 1 회가 아니고 여러번 중복하여 발생할 경우 어떻게 program 을 수정해야 할까?

rmmod ost2 명령을 수행한 후, 다시 insmod ost2.o 명령을 수행해도 동일한 동작을 하는가 확인한다.

6. Demonstration

To TA, 11/8 Mon. 4 PM – 6 PM.

7. Report

To TA, Due 11/11 Thu 6 PM.

Each student should prepare his own report containing:

Purpose

Experiment sequence

Experimental results

Discussion: should be different even for each member of the same team.

References

8. 참고사항

Application program 이 주기적으로 수행되어야 할 경우, interrupt 및 module 의 도움없이도 다음과 같이 sleep function 을 활용하면 가능하다.

```
while(1) {  
    sleep(1);          // Sleep for 1 second. Other processes can be executed.  
    do_periodic_job(); // Perform your periodic job  
}
```