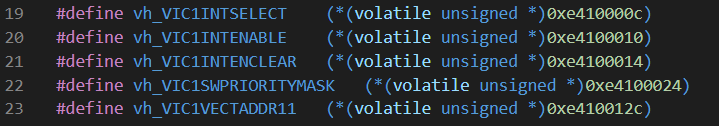
Timer Interrupt & GPIO setting

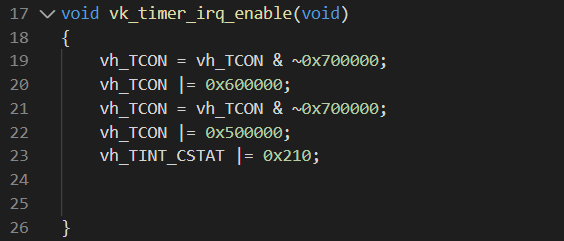
컴퓨터 소프트웨어학과

강정구

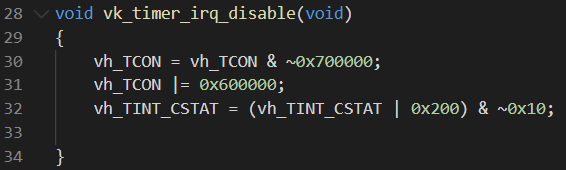
1**. Timer Interrupt**

Hal / include / vh\_io\_hal

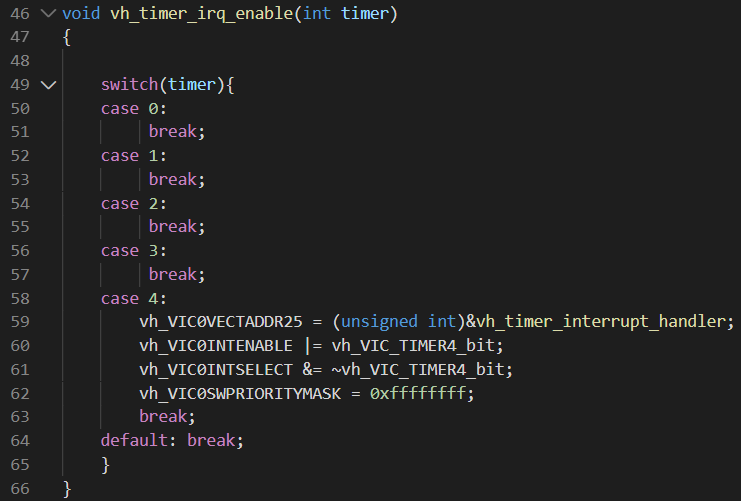
S5PC100 에서 TIMER4 에 관련된 register 들의 주소를 정의해준다.

Hal / io / timer.c / vk\_irq\_enable()

TCON register 의 TIMER4 을 시작하고 Interrupt 를 활성화하는 역할을 하는 함수이다. 위에서 부터 timer4 관련 비트를 0으로 clear, auto reload on & manual update, 다시 timer4 관련 비트를 0으로 clear, auto reload on & timer4 start, 그리고 마지막으로 TINT\_CSTAT 에서 timer4 interrupt enable & timer4 interrupt status bit 들을 clear 해준다.

Hal / io / timer.c / vk\_irq\_disable()

이 함수는 timer4 를 정지시키고 interrupt 를 비활성화한다. 이는 timer 을 통해 발생하는 이벤트를 block 해주는 효과가 있다. 이도 enable 과 마찬가지로 TCON 레지스터의 bit 을 이용하여 해당 함수를 수행할 수 있다.

Hal / io / vh\_irq\_enable

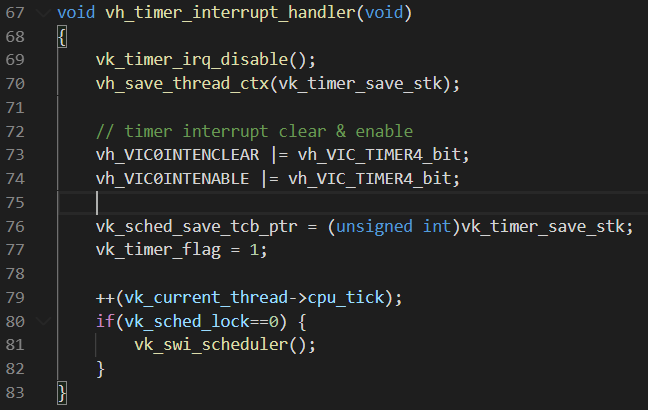
VIC 0 의 레지스터를 설정하여 timer4 interrupt 를 활성화시키는 함수이다. 우리가 사용하는 timer 는 timer4 이므로 case 4 만 정의해준다.

1) VIC0VECTADDR25 레지스터에 ISR 의 주소를 저장하고

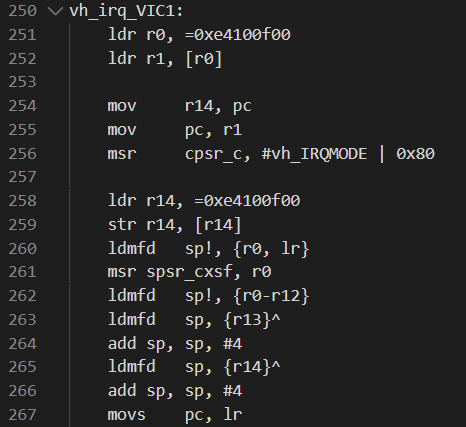
2) VIC0INTENABLE 레지스터에서 interrupt 를 활성화하고

3) VIC0INTSELECT 레지스터에서 timer4 interrupt 를 IRQ 로 설정하고

4) VIC0SWPRIORITYMASK 를 모두 1로 masking 해준다.

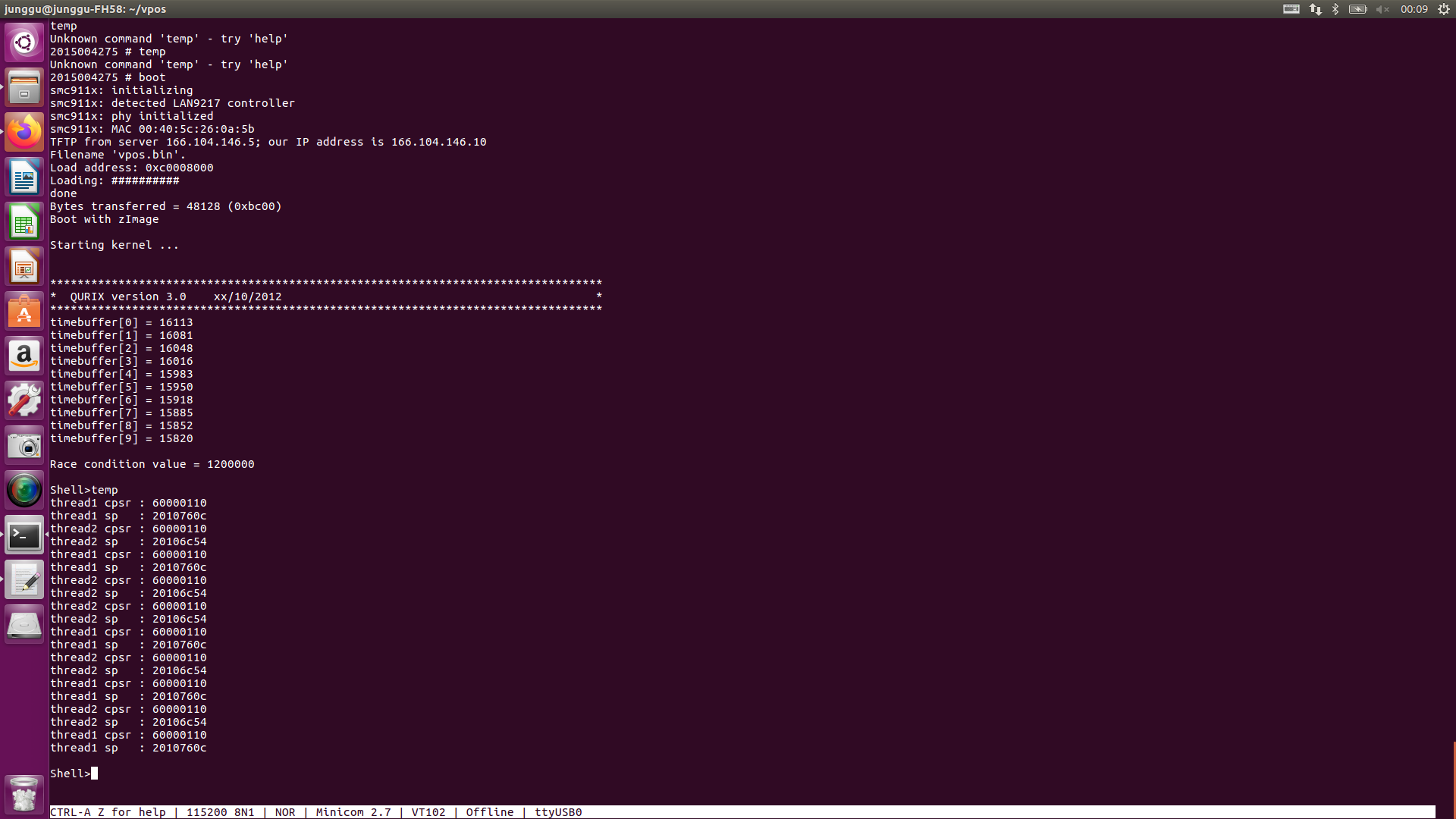
Hal / io / timer.c / vh\_timer\_lnterrupt\_handler

이 함수는 timer4 의 ISR 로 스케줄러를 1초의 주기로 호출한다. 이 함수가 호출하면 timer 의 interrupt 를 비활성화 하고 timer4 인터럽트를 pending clear 시킨 다음 sw interrupt 를 발생하여 모드를 privileged mode 로 바꾼 다음 스케줄러를 호출해준다.

Hal / cpu / HAL\_arch\_startup.S / vh\_irq\_VIC1

앞서 했던 다른 interrupt 와 마찬가지로 기존의 값을 레지스터에 저장하여 stack 에 push 해준 다음에 CPSR 을 IRQ 모드로 수정한 다음 SPSR, lr 을 복원해준다. 이 일이 끝나고 원래의 루틴으로 돌아가기 위해 레지스터와 pc 값을 stack 에서 복원한다.

결과화면



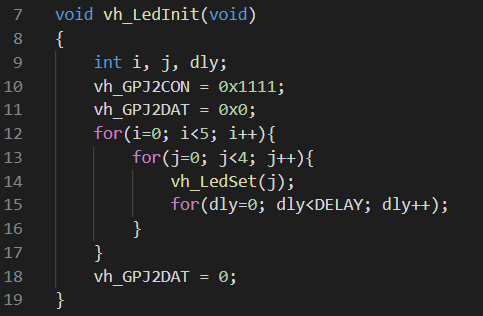
쉘에서 temp 명령어를 치면 context switching 이 수행되는 것을 확인할 수 있다.

**2. GPIO LED**

Hal / include / vh\_io\_hal

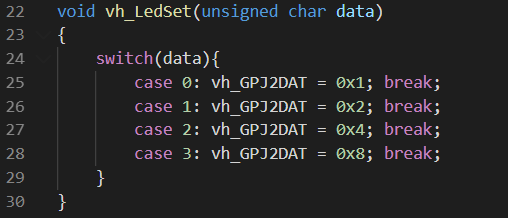
LED control 에서 쓰이는 두 register 의 address 를 정의해준다.

Hal / io / led.c / vh\_LedInit()



GPIO 설정을 하고 LED 초기화를 하는 함수이다. 이 때 4개의 LED 를 5번 깜빡인다.

Hal / io / led.c / vh\_LedSet()



위에서 LED 를 깜빡일 때 어떤 LED 를 깜빡일지 결정해주는 함수이다. 첨부된 동영상에서 볼 수 있듯이 첫번째, 두번째, 세번째, 네번째 LED 가 순서대로 5번 반짝인다.