Embedded System Software

과제3 수행 결과 보고서

과목명: [CSE4116] 임베디드시스템 소프트웨어

담당교수: 서강대학교 컴퓨터공학과 박성용

학번 및 이름 : 20171696 채벼리

개발기간: 2020.05.31 ~ 2020.06.05

최종보고서

1. 개발 목표

* interrupt를 사용하여 디바이스 드라이버 구현 및 Module programming을 할 수 있다.
* 구현한 모듈을 테스트할 수 있는 어플리케이션을 구현할 수 있다.

1. 개발 범위 및 내용

* fpga\_fnd device driver와 interrupt를 포함한 스탑워치 기능을 가진 하나의 모듈 구현

1. fnd는 앞의 두 자리와 뒤의 두 자리를 사용하여 시간을 표시한다.

* 초기 상태는 0000
* 앞의 두 자리는 분(60분), 뒤의 두 자리는 초(60초)를 표시한다

1. Home 버튼

* 초기 상태에서 Home버튼을 누를 경우 0초에서 타이머가 시작하고, 1초 단위가 바뀔 때마다 fnd 정보를 갱신한다.
* 일시 정지상태에서 Home 버튼을 누를 경우 현재 시간에서 1초 단위가 바뀔 때마다 fnd 정보를 갱신한다.

1. Back 버튼

* 시간이 일시 정지한다.
* home버튼을 누르면 타이머가 다시 시작한다.

1. VOL+ 버튼

* 초기상태로 돌아간다.
* fnd가 0000이고 타이머가 구동되지 않는 상태가 된다.

1. VOL- 버튼

* 3초 이상 누르면 어플리케이션이 종료된다.
* 3초 이상 누르지 않으면 어플리케이션은 계속 실행되고 있다.

1. 키 입력은 interrupt를 이용하여 수행한다.
2. 디바이스 드라이버의 이름은 /dev/stopwatch, major number는 242이다.

* module을 실행시키는 application을 구현한다.

1. 추진 일정 및 개발 방법

가. 추진 일정

* 2020/05/31 : home, back, vol+ 버튼 함수 구현
* 2020/06/03 : vol-버튼 함수 구현
* 2020/06/04 : 소수점 단위의 초가 유지되도록 수정
* 2020/06/05 : 주석 추가, 보고서 작성

나. 개발 방법

1. Host PC : Linux(Ubuntu 16.04) System on x64 architecture
2. Target Board : Embedded linux system on ARM architecture
3. 연구 결과
4. interrupt를 사용하여 버튼 입력 수행
   1. request\_irq : 인터럽트 서비스 함수를 등록한다. 키가 입력될 때마다 인터럽트 요청을 하여 키 입력에 대한 함수를 수행하도록 한다.
   2. free\_irq : request\_irq 함수에서 등록한 인터럽트 서비스 함수를 제거한다.
5. kernel timer을 사용한다.
   1. Kernel Timer

Kernel에서 사용자 argument와 함께 사용자 함수를 사용자 설정 시간에 실행될 수 있도록 설정하는 자료구조이다.

* 1. Timer initialization

init\_timer : timer\_list structure를 초기화한다.

timer.expires 설정 : function이 실행될 시간을 설정한다.

timer.data 설정 : function에 전달할 argument를 설정한다.

timer.function 설정 : expires 시간 이상이 되면 해당 function이 실행된다.

* 1. add\_timer

timer\_structure를 timer list에 등록한다.

* 1. del\_timer

timer\_list에서 timer를 제거한다.

1. 모듈 프로그래밍을 사용하여 조건에 맞는 device driver 구현

* “app/intertest.c”

어플리케이션을 구동하여 device driver를 테스트한다.

* “module/driver\_header.h”

함수 정의, 구조체 정의, 전역 변수를 정의한다.

* + struct timer\_data mydata : fnd 출력 정보와 타이머를 저장하는 구조체
    - struct timer\_list timer : fnd 출력을 위한 타이머, 1초마다 구동되게 설정할 것이다.
    - int time : fnd출력을 위한 시간을 초단위로 저장한다.
    - int flag : -1일 경우 타이머 초기 상태, 1일 경우 타이머가 작동되는 상태, 0일 경우 일시 정지 상태
  + struct quit\_data quit\_timer : vol- 버튼을 눌렀을 경우 3초 후에 프로그램을 종료시키기 위한 타이머를 저장하는 구조체
    - struct timer\_list timer : 3초 후에 프로그램을 종료시킨다.
    - quit\_flag : vol-버튼이 눌려있는 상태일 경우 1, 그렇지 않을 경우 0
* “module/write.h”
  + void clear\_device(void) : fnd출력을 0000으로 초기화한다.
  + void fnd\_write(int n) : fnd 출력을 00분 00초 형태로 출력한다.

예) n = 61일 경우 0101로 출력

* “module/stopwatch.c”
  + inter\_handler1 : vol+ 버튼을 눌렀을 경우 실행되는 함수
    - 프로그램이 초기상태로 돌아간다.
    - 현재 실행되고 있는 timer함수를 없애고 fnd 출력 및 시간이 0으로 초기화 된다.
  + inter\_handler2 : back 버튼을 눌렀을 경우 실행되는 함수
    - 타이머가 흐르고 있을 경우, 타이머를 일시정지 상태로 바꾼다.
  + inter\_handler3 : home 버튼을 눌렀을 경우 실행되는 함수
    - 타이머가 시작된다. 1초마다 시간이 업데이트 된다. 일시정지 상태의 경우 구동상태로 바꾸고, 초기 상태의 경우 새로운 타이머를 등록하여 1초마다 timer\_func이 실행되도록 설정한다.
  + inter\_handler4 : vol- 버튼을 눌렀을 경우 실행되는 함수
    - 3초동안 vol-버튼을 누른 경우 quit\_timer.timer에 의해 quit\_func이 실행된다.
    - 3초미만 vol-버튼을 누른 경우 quit\_timer.timer를 제거하여 프로그램이 종료되지 않게 한다.
  + quit\_func : 프로그램을 종료한다.
  + timer\_func : 1초마다 타이머에 의해 실행되는 함수이다. 일시정지 상태일 경우에는 현재 시간이나 fnd 출력의 값을 업데이트 하지 않지만, 타이머 구동 상태일 경우에는 현재 시간과 fnd 출력을 증가하여 바꾸어준다.
  + inter\_register\_cdev : /dev/stopwatch를 major number 242로 등록

1. 실행 방법
   1. Host PC

>>./module/make

>>./app/make

* 1. Target Board

>>insmod stopwatch.ko

>>mknod /dev/stopwatch c 242 0

>>./app

1. 기타

없음