Embedded System Software HW#2

Due Date : 2020. 5. 24.

1. 목표

타이머 디바이스 드라이버 및 관련 테스트 응용 프로그램 개발

2. 구현

(1) 테스트 응용 프로그램

사용자로부터 타이머 디바이스 구동 옵션을 받아 ioctl을 통하여 타이머 디바이스 드라이버에 전달하고 ioctl을 통해 타이머 디바이스를 구동한다. (두 개의 ioctl 명령어 사용)

(2) 타이머 디바이스 드라이버

디바이스 드라이버 (fpga_fnd, fpga_led, fpga_dot, fpga_text_lcd)와 타이머 기능을 포함한 하나의 module을 구현한다.

3. 세부 기능

(1) 응용 프로그램

① **사용자로부터 타이머 디바이스 구동 옵션을 받아 ioctl을 통하여 타이머 디바이스 드라이버에 전달하고 ioctl을 통해 타이머 디바이스를 구동한다. 두 개의 ioctl 명령어 사용: ioctl(fd, SET_OPTION, ...) / ioctl(fd, COMMAND, ...)**

② **실행 예)**

./app TIMER_INTERVAL[1-100] TIMER_CNT[1-100] TIMER_INIT[0001-8000]

TIMER_INTERVAL: HZ 값 1~100 (0.1~10초)

TIMER_CNT: 디바이스 출력 변경 횟수 (1~100)

TIMER_INIT: FND에 출력되는 초기 문양과 위치 (0001~8000)

단 4자리 중 한 자리만 1~8 숫자가 입력

예를 들어, 0040이라면 왼쪽에서 세번째 자리에 "4"를 출력

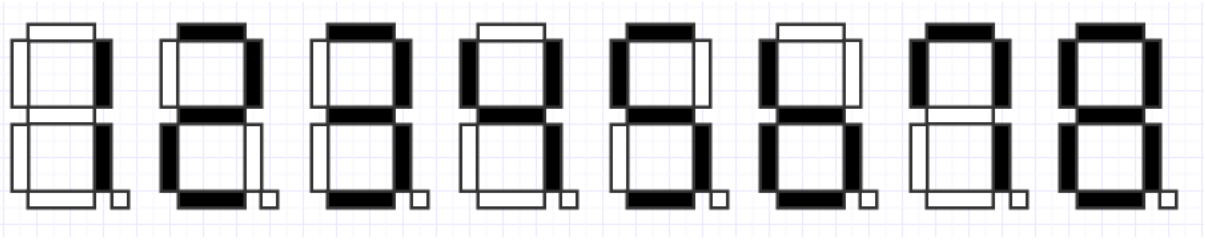
③ **TIMER_CNT 및 TIMER_INIT에 대한 사항은 (2) 타이머 디바이스 드라이버에서 상세히 설명한다**

(2) 타이머 디바이스 드라이버

① **사용자 프로그램에서 ioctl을 통하여 전달 된 옵션을 기준으로 4가지 디바이스 (FND, LED, DOT, TEXT_LCD)의 요구사항에 따라 디바이스에 동시에 출력한다.**

② **모든 디바이스의 값들이 바뀌는 시간은 TIMER_INTERVAL을 기준으로 한다.**

③ **타이머 디바이스 드라이버 이름은 /dev/dev_driver로 통일한다. (major number : 242)**



위의 문양은 왼쪽부터 1번~8번이다.

- fpga_fnd(FND)

초기 상태: **TIMER_INIT**에서 전달 된 문양과 위치에 따라 FND에 출력한다.

이후 상태: **TIMER_INTERVAL**마다 다음 문양을 **TIMER_CNT** 횟수만큼 지정된 위치에 출력한다. **TIMER_CNT** 횟수만큼의 출력이 끝나면 FND의 불을 꺼준다 (0000으로 초기화).

- 다음 문양이란 현재 수에서 1 증가된 수를 의미한다. 단, 8의 다음 문양은 1이다.
(예) 0040->0050->0060->0070->0080->0010->...
- 문양이 출력되는 위치는 한 번의 로테이션 (한 자리에서 모든 문양이 한 번씩 출력 되는 것)이 끝날 때마다 우측으로 이동한다. 또한 문양은 총 TIMER_CNT만큼 출력된다.
- 만약 (왼쪽에서부터) 4번째 자리에서 로테이션이 끝났다면, 1번째 자리로 이동한다
(예1) 0040->...->0080->0010->...->0030->0004 (**TIMER_INIT**: 0040, **TIMER_CNT**: 9)
(예2) 0008->0001 (**TIMER_INIT**: 0008, **TIMER_CNT**: 2)
(예3) 0007->0008->0001->...->0006->7000 (**TIMER_INIT**: 0007, **TIMER_CNT**: 9)

- fpga_led(LED)

: 현재 fpga_fnd에서 출력 중인 문양의 번호를 나타낸다. (D1: 1번, D2 : 2번, D3 : 3번, D4 : 4번, D5 : 5번, D6 : 6번, D7: 7번, D8 : 8번) **TIMER_CNT** 횟수만큼의 출력이 끝나면 fpga_led의 불을 꺼준다.

-fpga_dot(DOT)

: 현재 fpga_fnd에서 출력 중인 문양과 같은 모양의 문양을 출력한다. fpga_fnd의 문양이 바뀐다면, fpga_dot도 fpga_fnd와 같은 문양으로 함께 바뀐다. **TIMER_CNT** 횟수만큼의 출력이 끝나면 dot의 불을 꺼준다.

-fpga_text_lcd(TEXT_LCD)

초기 상태: 첫 번째 줄에는 자신의 학번을 출력하고, 두 번째 줄에는 자신의 이름을 영문으로 출력한다. 이때 두 문장 모두 left align 시킨다. (가장 왼쪽 글자가 가장 왼쪽에 위치)

이후 상태: **TIMER_INTERVAL** 마다 오른쪽으로 한 칸씩 shift 이동을 하고, 가장 우측 글자가 가장 오른쪽 칸에 도달한 경우, 다시 왼쪽으로 shift 이동을 시작한

다. 왼쪽 shift 이동도 마찬가지로, 왼쪽 글자가 가장 왼쪽 칸에 도달한 경우, 다시 오른쪽으로 shift 이동을 시작한다. (각 줄은 독립적이므로, 학번과 이름의 길이가 다를 경우, 각 줄은 다른 형태로 이동을 하게 된다)
TIMER_CNT 횟수만큼의 출력이 끝나면 LCD의 불을 꺼준다

4. 제출 방법

(1) 제출 파일

-> app 폴더 (소스코드, Makefile)

-> module 폴더 (소스코드, Makefile)

-> Document (ex. [HW2]20001234.docx or [HW2]20001234.hwp)

-> readme.txt (본인 driver의 name과 major number를 기술)

파일 압축 방법

학번 폴더(20001234)를 생성하고, 그 안에 보고서, 폴더들을 저장한 뒤, 학번 폴더가 있는 위치에서 **tar –cvzf [HW2]학번.tar.gz ./학번폴더**

5. 평가 기준

(1) 본 구현 80%, 문서 20%

구현한 모든 프로그램은 주석을 포함할 것

(2) Late, 하루에 10% 감점

(3) 제출 형식 틀린 경우 10% 감점

(4) Copy 적발 시 0점