Embedded System Software 1st HW

과목명: [CSE4116] 임베디드시스템소프트웨어

담당교수: 서강대학교 컴퓨터공학과 박 성 용

20131574 오현석

**최 종 보 고 서**

**I. 개발 목표**

- 각 프로젝트마다 주어지는 주제를 바탕으로 본 프로젝트에서 추구하는 개발 목표를 설정하고 그 내용을 기술할 것. fork함수를 이용해 자식 프로세스를 2개 만들고 프로세스끼리 IPC(message queue)를 통해서 통신한다. output process에서 device driver 및 mmap 함수를 통해 process와 device 사이에 read, write를 수행한다.

**II. 개발 범위 및 내용**

- 자신들이 설계한 개발 목표를 달성하기 위하여 어떠한 내용의 개발을 수행할 지 그 범위와 개발 내용을 기술할 것.

**가. 개발 범위**

- 자식 프로세스 생성

- 주어진 명세서에 나타난 각각의 모드별로 개별적인 기능 구현

- message queue를 이용해 프로세스끼리 통신

- device driver 및 mmap 함수 사용

**나. 개발 내용**

- 자식 프로세스는 메인 프로세스에서 fork 후 exec 계열 함수를 통해 새로운 실행 파일로 메모리를 덮어쓴다. 이번 과제에서는 execve를 이용한다.

- Message queue를 이용하여 프로세스간 통신한다.

- Input process는 입력만 받고, Output process는 출력만 한다. 그 외 모든 처리는 Main process에서 한다.

- Device file과 mmap을 이용해 Device에서 입력을 받아오고 출력을 한다.

**III. 추진 일정 및 개발 방법**

- 자신들이 설정한 개발 목표를 달성하기 위한 개발 일정을 설정하고, 각 요소 문제를 해결하기 위해서 어떤 방법을 사용할 지 기술할 것. 또한 각 연구원의 역할을 분명히 기술할 것.

**가. 추진 일정**

|  |  |
| --- | --- |
| 일정 | 내용 |
| 2019.04.07. | 명세서 확인 및 자식 프로세스 생성 및 shared memory 구현 |
| 2019.04.08.~2019.04.12. | mode1 ~ mode5 구현 |
| 2019.04.13.~2019.04.15. | 주석 정리 및 보고서 작성 |

**나. 개발 방법**

- 백업을 함으로써 불의의 상황에 대비한다.

- 주어진 명세서와 예제 코드를 꼼꼼히 확인하고 개발을 시작한다.

- Message queue 를 사용하여 데이터를 주고 받는다.

- 개발 중 필요하거나 이해가 안가는 부분은 구글을 이용한다.

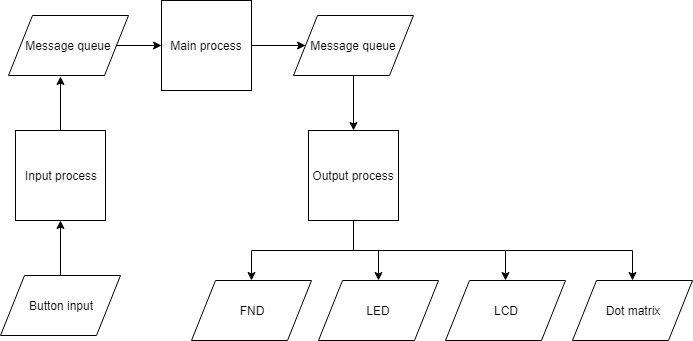
- Ubuntu환경에서 크로스 컴파일을 통해 타겟 보드인 휴인스 보드에서 작동하는 실행파일을 만든다. 보드에서 실행되는 과정은 Minicom을 통해 확인한다.

**IV. 연구 결과**

- 최종 연구 개발 결과를 자유롭게 기술할 것.

**1. 합성 내용:**

- 설계 목표에 필요한 내용을 조사 분석한 후 그들을 바탕으로 구성한 **전체 프로그램 구성도**.



**2. 제작 내용:** 개발 결과

- 구현한 컴포넌트들에 대한 역할 및 구현 방법에 관한 내용을 기술할 것. (예: 자료구조, 알고리즘).

- Message queue 구조

Message queue는 Main, Input, Output process들이 공유한다. 버튼이 눌렸을 때 Message queue 에 send하고 이 때 다른 프로세스에서 receive한다. Message queue에서 사용할 구조체는 main.h에 선언하였다.

- Main process

Main process에서는 각 모드들이 사용하는 변수들과 함수들로 구성 되어 있다.

- Input process, Output process

종료버튼이 들어오기 전 까지 무한 루프를 돌면서 입력 및 출력한다. Output process에서 각 device별로 함수를 구현했다.

**3. 시험 및 평가 내용:**

- 평가 방법에 대한 설명을 기술하라.(수행 sequence 등)

- 자신들의 결과물이 갖는 **보건 및 안정**, **생산성과 내구성**에 대하여 반드시 기술할 것.

- 프로그램 시작 시 기본 모드는 1이다. 모드가 바뀔 때 마다 init 함수를 호출하여 변수들을 초기화한다. 각 모듈별 초기 상태는 아래와 같다. 모드에서 특별히 정해진 초기 상태가 없으면 아래의 초기 상태를 유지한다.

**- mode1**

FND에 현재 시간을 출력한다. 24시간제로 표시한다. LED는 1에만 불이 들어온다.

초기상태

FND: 보드의 시간

LED: 1번 LED에만 불이 들어옴

기능

SW(1): 한번 누르면 보드의 시간을 변경할 수 있도록 할 수 있다. 보드의 시간을 변경하는 중에는 LED 3번, 4번이 1초에 한 번씩 번갈아 켜진다. 다시 한 번 누르면 LED 1번에만 불이 들어오고 바뀐 시간이 저장되며 시간이 자동으로 업데이트가 되지 않는다.

SW(2): 변경된 시간을 현재의 시간으로 reset시켜주는 버튼. 시간이 자동으로 업데이트 된다.

SW(3): 시간을 1시간 증가 시키는 버튼.

SW(4): 분을 1분 증가 시키는 버튼.

**- mode2**

FND에 카운팅된 숫자를 현재 선택된 진수에 맞춰 출력한다. 천단위는 생략한다. LED 1번은 2진수, 2번은 10진수, 3번은 8진수 4번은 4진수를 의미한다.

초기상태

FND: 0000

LED: 2번 LED에만 불이 들어옴

기능

SW(1): 진수 변환을 한다. 10진수->8진수->4진수->2진수->10진수 형태로 변환한다. LED 역시 2번, 3번, 4번, 1번, 2번 순서로 켜진다.

SW(2): 백의 자리 숫자를 1 증가시킨다.

SW(3): 십의 자리 숫자를 1 증가시킨다.

SW(4): 일의 자리 숫자를 1 증가시킨다.

**- mode3**

FND에 Switch button이 눌린 횟수를 출력한다. 만 단위는 생략한다. Text LCD에는 입력된 문자열을 출력한다.

초기상태

알파벳 입력 모드

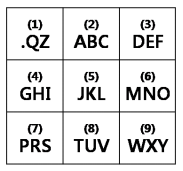
FND: 0000

Text LCD: 빈 문자열

Dot matrix: ‘A’

기능

알파벳 입력 모드에서 각 Switch button의 기능은 아래 그림과 같다. 같은 키를 연속해서 입력하면 그 횟수에 해당하는 알파벳으로 바뀐다. 숫자 입력 모드에서는 1~9까지 대응된다.



SW(2)+SW(3): Text LCD를 빈 상태로 만든다.

SW(5)+SW(6): 알파벳 입력 모드, 숫자 입력 모드를 전환한다. 숫자 입력 모드 시 Dot matrix에 ‘1’ 표시

SW(8)+SW(9): 공백 입력.

**- mode4**

FND에 Switch button이 눌린 횟수를 출력한다. 만 단위는 생략한다. Switch button을 통해 Dot matrix에 그림을 그린다.

초기상태

FND: 0000

Dot matrix: 좌 상단의 1개 커서가 깜빡거린다.

기능

Switch button의 각 기능은 아래 그림과 같다.



SW(1): 그려진 데이터를 지우고 커서를 좌 상단에 위치 시킨다.

SW(3): 커서 표시를 on/off 한다.

SW(5): 현재 커서 위치에 그림을 그리거나 지운다.

SW(7): 그려진 데이터는 지우고 커서 위치는 유지한다.

SW(9): 현재 그려진 그림을 반전 시킨다. 켜진 부분은 끄고 꺼진 부분은 켠다.

**- mode5**

똥 피하기, 위에서 똥이 내려오며, 똥을 피하면 점수가 올라간다. 똥을 맞으면 게임이 끝난다. FND에 점수를 출력한다. Dot matrix의 가장 밑 칸 중간에는 움직일 수 있는 캐릭터가 주어진다. Text LCD의 게임이 플레이 중이면 Playing, 게임이 끝나면 GAME OVER 가 출력된다. LED는 1번부터 8번까지 순서대로 껐다가 켜지는 것이 반복된다.

초기상태

FND: 0000

Dot matrix: 가장 밑 칸 중간에 불이 들어온다.

Text LCD: Playing

LED: 1~8 이 순서대로 껐다가 켜진다.

기능

Switch button 4와 6만 사용가능하다.

SW(4): 캐릭터를 왼쪽으로 움직인다.

SW(5): 캐릭터를 오른쪽으로 움직인다.

- 공통

각 모드에서 수행되는 동작들은 서로간에 간섭이 없다. 모드 변경시 필요한 변수들을 모두 초기화하여 안정성을 높혔다. 또한 버튼을 눌렀을 경우에만 message queue에 값을 넣음으로써 정확성을 높인다.

**V. 기타**

- 기타 관련 내용을 기술할 것.

**1.** **연구 조원 기여도**: 오현석 100%

**2.** 기타 본 설계 프로젝트를 수행하면서 느낀 점을 요약하여 기술하라. 내용은 어떤 것이든 상관이 없으며, 본 프로젝트에 대한 문제점 제시 및 제안을 포함하여 자유롭게 기술할 것.

- 처음 접하는 것이라 시작 단계에서 시간이 많이 들었다. 어려웠지만 내가 구현한 소스가 하드웨어를 제어한다는 것이 신기하였다.