

AI Embedded Systems 과제 2

1. 과제 내용: LED 기억력 게임

- 개요

- 본 과제에서는 LED와 버튼을 활용하여 LED 기억력 게임을 진행해 주는 **디바이스 드라이버 (커널 모듈)**와 그 디바이스 드라이버를 이용하는 **사용자 애플리케이션**을 만드는 것을 목표로 한다.
- 이 게임에서는 **3개의 LED** (각각을 LED A~C로 지칭)와 **4개의 switch**를 이용한다. 각 LED마다 하나의 switch (각각을 Switch A~C로 지칭)에 대응되며, 남은 switch 하나 (Switch D로 지칭)는 입력이 끝난 것을 알려주는 용도로 사용된다.
- LED 기억력 게임은 다음과 같은 순서로 진행된다.
 1. 먼저, 사용자 애플리케이션이 실행되면 화면에 "Please focus on the LED lights..."라는 문자열을 출력하고, 디바이스 드라이버에게 "Game start"라는 문자열을 전달한다.
 2. 게임이 시작되면 디바이스 드라이버는 세 개의 LED를 임의의 시퀀스로 점등한다.
 - ✓ LED 점등의 시퀀스 길이는 3 ~ 7 사이에서 랜덤하게 결정된다.
 - ✓ 이때, 각 LED의 점등은 1초 동안 이루어지고, 다음 LED의 점등은 0.5초 뒤에 이루어진다.
 - ✓ 예시) 세 개의 LED A, B, C가 있고, LED 점등 시퀀스가 <B, A, C, A, B>로 랜덤하게 선택되었다고 하자. 이때, LED들은 <LED B 1초 점등> → <0.5초 후, LED A 1초 점등> → <0.5초 후, LED C 1초 점등> → <0.5초 후, LED A 1초 점등> → <0.5초 후, LED B 1초 점등> 순서로 이루어진다.
 3. 모든 LED의 점등이 끝나면 디바이스 드라이버는 이를 사용자 애플리케이션에게 알려주고, 사용자 애플리케이션은 "Please press the switches in the order you remember."이라는 문장을 화면에 출력한다.
 4. 이후, 사용자 애플리케이션은 디바이스 드라이버에게 "Ready for switch input"이라는 문자열을 전달하고, 디바이스 드라이버가 사용자의 스위치 입력을 받

기 시작한다.

- ✓ 디바이스 드라이버는 "Ready for switch input" 문자열을 받기 전 입력된 스위치 입력은 모두 무시한다.
- ✓ 스위치 입력이 발생하면 해당하는 LED를 0.5초 동안 점등한다.
- ✓ 0.5초 점등되는 동안 다른 버튼 입력은 받을 수 없다.

5. 사용자는 스위치 입력이 끝나면 완료 스위치 (Switch D)를 누른다. 이후, 디바이스 드라이버는 사용자 애플리케이션에게 입력 결과를 알려주고, 사용자 애플리케이션은 그 결과를 화면에 출력한다.

- ✓ 버튼 입력 순서가 올바르면 사용자 애플리케이션은 "Your input is correct.", 틀리면 "Your input is wrong."을 출력한다.

● 구현 요구 사항

- Step 1과 4에서 사용자 애플리케이션은 **write() system call**을 이용하여 디바이스 드라이버에게 문자열을 전달해야 한다.
- Step 3과 5에서 디바이스 드라이버가 사용자 애플리케이션에게 메시지를 전달하기 위해서는 **signal**을 이용해야 한다.
- Step 2와 4에서 LED 점등의 지속 시간을 제어하기 위해서 **timer interrupt handling**을 통해 구현해야 한다. 즉, delay() 또는 sleep() 같은 종류의 함수를 사용하거나 busy waiting시켜서는 안된다.
- Step 4에서 사용자의 스위치 입력은 **interrupt handling**을 통해 구현해야 한다.
- 스위치 입력 시, chattering 문제가 완화하기 위해 **debouncing**을 구현해야 한다.
- LED와 스위치에 대한 직접적인 제어는 모두 디바이스 드라이버 통해서만 이루어져야 한다.
- 3개의 LED와 4개의 스위치는 다음과 같은 GPIO number를 이용해야 한다.

1. LED A: GPIO 13
2. LED B: GPIO 19
3. LED C: GPIO 26
4. Switch A: GPIO 16
5. Switch B: GPIO 20
6. Switch C: GPIO 21
7. Switch D: GPIO 18

Raspberry Pi B+ B+ J8 GPIO Header			
		Pin No.	
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

2. 제출 형태

- 코드 제출 (파일 이름 지킬 것)

- 커널 모듈 코드 (game_module.c)와 Makefile
- 사용자 애플리케이션 코드 (led_game.c)

- 2페이지 분량 보고서 (학번_이름.pdf / ex) 202300001_홍길동.pdf)

- 코드 설명
 1. 구현 알고리즘 및 코드에 대해 간략히 설명
- 코드 실행결과
 1. 코드 실행 결과에 대한 스크린샷
 - ✓ 기능에 대한 실행 결과가 충분히 있어야 합니다.
 - ✓ 이에 대한 간략한 설명

- 실제 동작 영상 (학번_이름.mp4 / ex) 202300001_홍길동.mp4)

- 두 번의 게임을 수행하는 시나리오 (첫 게임에서는 정답을 맞추고, 두 번째 게임에서는 정답을 틀리는 시나리오)를 촬영
- 무편집 one take 영상으로 촬영할 것

- 과제를 구현하지 못한 경우 구현한 부분까지의 설명과 구현을 못한 이유를 분석하여 코드와 함께 제출할 것

- 코드와 보고서, 영상을 학번_이름.zip 형태로 압축하여 제출할 것

- 5월 15일 수요일 오후 11시59분까지 제출

- 제출 형태(ex)파일명, 파일 압축 등)를 지키지 않은 경우 예외 없이 감점 처리함.