

Assignment #1

Subject: Microprocessor Laboratory [ICE3029]

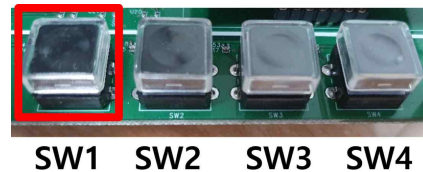
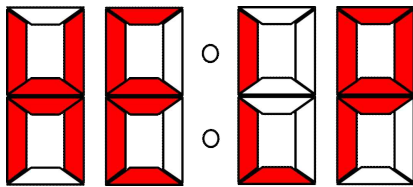
Deadline: Sunday, March 26, 2023, 11:59pm (Late submission is not allowed.)

Instruction

- 모든 문제는 반드시 하나의 프로그램에 구현할 것.
- 프로그램은 반드시 조교의 개발 환경(BSP Version 3.5.0)과 호환되어야만 함.
(조교의 개발 환경에서 해당 프로그램이 정상적으로 동작하지 않는 경우, 0점 처리)
- 결과물은 문항별로 주어진 모든 Condition을 만족하고, 제공된 Example Video와 유사한 형태로 동작하도록 설계할 것.
- “과제 진행 방법.pdf”를 참고하여 결과물을 제출할 것.
(①프로젝트 파일, ②동작 영상, ③‘hal_entry.c’ Code 파일)
- Assignment 1은 함께 제공된 ‘hal_entry.c’, ‘library.c’, 및 ‘library.h’ 파일을 이용할 것.
‘hal_entry.c’ 파일은 반드시 지정된 위치에서만 Code를 작성하고, ‘library.c’ 및 ‘library.h’ 파일은 절대 수정하지 말 것.
→ 조교가 제공한 ‘library.c’, ‘library.h’ 파일과 본인이 설계한 ‘hal_entry.c’ 파일이 호환되어야만 함.
- FSP Configuration은 반드시 제공된 ‘hal_entry.c’ 파일과 호환되도록 설정할 것.
- 별도의 지시사항이 없는 경우, Interrupt Trigger는 Falling Edge로 설정할 것.
- 제시한 Instruction을 만족하지 않을 경우, 해당 문항은 감점 혹은 0점 처리함.

Task (Total 100 points)

1-1. 특정 상황에서만 “HELP” 문자열을 7-Segment에 출력하는 기능을 설계하시오. (20points)



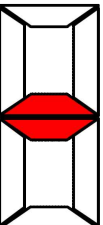
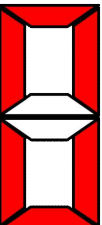



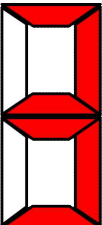

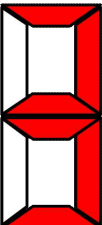

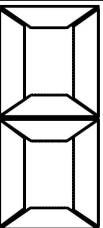
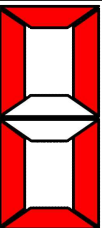



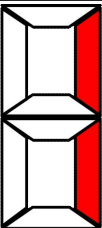

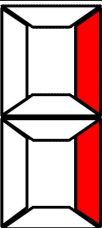

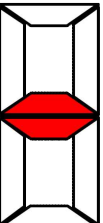
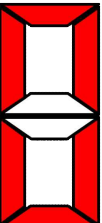



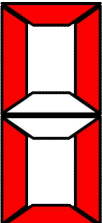



[Conditions]

- “HELP” 문자를 7-Segment에 출력하는 동작은 반드시 Switch 1번을 누르고 있을 때만 수행되어야 함.
- Rising/Falling Edge 둘 다에 대해 Interrupt가 동작하도록 FSP Configuration을 설정할 것.
- 해당 동작을 수행할 때만 LED 1번(PA08)이 ON 상태가 되도록 설계할 것.
- 해당 기능이 동작 중일 때는, 다른 Task(1-2, 1-3)에 대한 기능이 동작하지 않도록 설계할 것.

1-2. 다음 조건에 따라 실수 곱셈 기능을 설계하시오. (40points)

[Conditions]

- a. Switch 2번을 누를 경우, LED 2번(PAO9)이 ON 상태가 되며 다음 동작을 수행하도록 설계할 것.
→ 임의의 Float형 난수 n_1, n_2 를 생성하고, $n_1 \times n_2$ 에 대한 결과를 계산
(이때, 각각의 난수는 반드시 해당 범위를 만족해야만 함, $-1 < n_1 < 1$, $-1 < n_2 < 1$)
- b. LED 2번(PAO9)이 ON 상태일 때 Switch 3번을 누를 경우, 다음과 같은 Sequence에 따라 동작하도록 설계할 것.
 - 1) Float형 난수 n_1 을 7-Segment에 출력 (오른쪽 정렬), LED 3번(PA10)을 ON 상태로 설정
 - 2) 다시 Switch 3번을 누르면, Float형 난수 n_2 를 7-Segment에 출력 (오른쪽 정렬)
 - 3) 다시 Switch 3번을 누르면, $n_1 \times n_2$ 에 대한 결과를 7-Segment에 출력 (오른쪽 정렬)
 - 4) 다시 Switch 3번을 누르면, 7-Segment와 LED 2, 3번을 처음 상태(OFF)로 초기화
이때, 반드시 부호(-)와 소수점(.)도 함께 표시할 것. (부호는 음수일 경우만 표시)
- c. 7-Segment에 Float형 값을 출력할 때, 소수점 둘째 자리까지만 표시할 것. (셋째 자리 이하는 무시)
ex1) 0.245768 → 0.24 ex2) -0.79145863... → -0.79
- d. 해당 기능이 동작 중일 때는, 다른 Task(1-1, 1-3)에 대한 기능이 동작하지 않도록 설계할 것.
- e. 부동 소수점 오차를 감안하여, 결괏값에 대해 ±0.01의 오차를 허용함 (정답으로 인정)

RESULTS									
$n_1 = -0.33$				  					
$n_2 = 0.11$				  					
$n_1 \times n_2 = -0.03$				  					

1-3. 다음 조건에 따라 LED 전광판 기능을 설계하시오. (40points)

[Conditions]

- Switch 4번을 누를 때마다 다음 동작을 수행하도록 설계할 것.
 - 7-Segment에서 한 글자씩 Left 방향으로 Text가 이동
 - 한 번의 동작 Cycle이 끝나고 다시 Switch 4번을 누를 경우, 동일한 Sequence를 반복
- 7-Segment에 출력할 Text는 자신의 학번(20XXXXXXX)으로 설정할 것.
- 동작 과정에서, 7-Segment의 Initial State는 반드시 모든 Digit가 OFF인 상태로 설정할 것.
- 해당 문제와 관련된 Code에서는 특정 함수를 사용할 수 없음. (반드시 Register를 직접 설정할 것)
 - 사용 불가능한 함수는 다음과 같음. (관련 Code에서 해당 함수 사용 시, 최종 점수의 50% 감점)
 - R_IOPORT_PinWrite(), FND_Display_Data(), FND_Print_Data(), FND_Reset(), LED_Reset()
- 해당 기능이 동작 중일 때는, 다른 Task(1-1, 1-2)에 대한 기능이 동작하지 않도록 설계할 것.
- 아래 제공된 Example을 참고하여 기능을 구현할 것.

[Example (Student ID: 2023123456)]

