

[강의 : 중간 프로젝트]

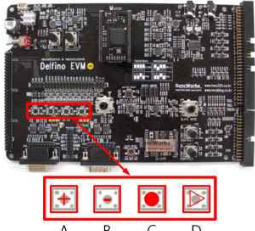
조 : 4

이름 : 이승복, 김희서

학번 : 201420820, 201623432

1. 프로젝트 내용 요약

GPIO 입력, 출력 예제와 Interrupt를 활용하여 LED의 점등 변화를 모드 별로 동작하도록 한다.

	<ul style="list-style-type: none">• 스위치 A: Mode 증가• 스위치 B: Mode 감소• 스위치 C: 정지• 스위치 D: 시작 (Mode 1 동작 시작)
---	--

GPIO 입력 44~47번을 D, C, B, A라고 할 때 스위치 입력에 따라 위와 같은 동작을 하도록 설정한다. 정지의 경우 LED 하나를 번갈아 점등하여 정지 상태를 표현하며, 시작의 경우 스위치 D를 누르면 Mode 1의 동작이 시작된다. 각 모드에 대한 설명은 아래와 같다.

모드	동작	시간 간격
Mode 1	점등되는 1개의 LED가 우측으로 이동한다.	1s
Mode 2	점등되는 1개의 LED가 우측으로 이동한 뒤, 돌아와서 비어있는 LED를 왼쪽부터 채운다. 모든 LED가 점등될 때까지 반복한다.	500ms
Mode 3	홀수 번째 LED 점등 → 짝수 번째 LED 점등 → 홀수 번째 LED 점등 ... 위 과정을 반복한다.	500ms
Mode 4	양 끝에서 점등된 LED가 중앙으로 이동하여 만난 뒤, 다시 양 끝으로 이동한다.	1s
Mode 5	점등되는 LED 개수가 1개부터 5개까지 증가하며 우측으로 이동	1s
Mode 6	점등되는 LED 개수가 3개까지 증가와 감소를 반복하며 우측으로 이동	1s

2. Quiz

- 중간 프로젝트에 사용하는 예제 코드 2가지(ex) @@@_@@@, @@@_@@@)

GPIO_Input_Qual에는 있는 c파일이 ADC_1ch_Timer_SOC_Interrupt에 없기 때문에
GPIO_Input_Qual 코드를 바탕으로 ADC_1ch_Timer_SOC_Interrupt 코드를 입힌다.

- 중간 프로젝트 코드 2가지의 차이(자세히)

1000줄을 이용해서 프로젝트를 구현한 코드는 GPCDAT 레지스터로 각각의 LED를 2진수로 표현했다. 총 16개의 LED가 있기 때문에 GPCDAT 값은 $2^0 \sim 2^{15}$ 의 덧셈으로 계산을 하여 해당 LED를 점등시켜주었다. mode B의 경우, i의 경우의 수는 $\sum_{i=1}^{16} i = 136$ 가지가 존재한다.

300줄을 이용해서 프로젝트를 구현한 코드는 반복문과 shift연산을 사용했다. Shift연산은 '변수 << 이동할 비트 수' 또는 '변수 >> 이동할 비트 수' 형식으로 사용하며 LED의 위치를 옮기는 역할을 한다.

- CCS 메모리 문제 해결 방법

메모리가 부족할 경우 빌드 오류가 나타난다.

이를 해결하기 위해서는 'Project Explorer'에서 'F28335_RAM.cmd'를 클릭한 뒤, 저장 공간을 늘려주면 해결된다. 코드 변경을 통해 저장 공간을 늘려줄 수 있으며 해당 코드는 아래와 같다.

```
9  RAML0      : origin = 0x008000, length = 0x003000 /* on-chip RAM block L0 */
10  RAML1      : origin = 0x00C000, length = 0x001000 /* on-chip RAM block L1 */
11  RAML2      : origin = 0x00D000, length = 0x001000 /* on-chip RAM block L2 */
12  RAML3      : origin = 0x00F000, length = 0x001000 /* on-chip RAM block L3 */
```