Lab. Fault Handler

Ref:

1. Cortex-M3 Technical Reference Manual
2. Keil Application Note 209, Using Cortex-M3 and Cortex-M4 Fault Exceptions
3. http://blog.feabhas.com/2013/02/developing-a-generic-hard-fault-handler-for-arm-cortex-m3cortex-m4/ (Developing a Generic Hard Fault handler for ARM Cortex-M3/Cortex-M4)

이 실험에서는 simulator를 이용하여 Fault가 발생하였을 때, ARM Cortex-M3 프로세서의 동작과 몇 가지 프로세서의 동작 특성에 대해서 살펴본다. (본 실험은 ref. 3번에서 기술하고 있는 내용을 수정 보완하여 구성하였다.)

1. 새로운 project를 생성한다.
2. 이 실험에서도 simulator를 사용할 것이므로 사용할 processor의 선택은 중요하지 않지만 일단 실험 키트에서 사용하는 TI 사의 LM3S9B92을 선택한다.
3. Startup.s 파일을 다음과 같이 수정한다.

```
;*******************************************************************
;
; The vector table.
;
;*******************************************************************
        EXPORT  __Vectors
__Vectors
        DCD     StackMem + Stack           ; Top of Stack
        DCD     Reset_Handler             ; Reset Handler
        DCD     NmiSR                    ; NMI Handler
        DCD     HardFaultHandler           ; Hard Fault Handler
        DCD     IntDefaultHandler          ; MPU Fault Handler
        DCD     IntDefaultHandler          ; Bus Fault Handler
        DCD     UsageFaultHandler          ; Usage Fault Handler
        DCD     0                        ; Reserved
        DCD     0                        ; Reserved
```

그리고 Starpup.s 파일에 다음과 같은 부분을 추가한다.

```
HardFaultHandler  PROC
        EXPORT HardFaultHandler           [WEAK]
        B      .
        ENDP

UsageFaultHandler  PROC
        EXPORT UsageFaultHandler          [WEAK]
        B      .
        ENDP
```

4. 다음과 같은 main.c 파일을 project에 new item으로 추가한다.

```
#include "ARMCM3.h"

int div(int lho, int rho)
{
   return lho/rho;
}

int a = 10;
int b = 0;
int c;

int main(void)
{
   SCB->CCR |= 0x18;         // enable trap on DIV_0 & UNALIGN

   SCB->SHCSR |= 0x60000;   //Usgae & Bus fault  exception enable

   c = div(a, b);

   while(1);
}
```

5. 주어진 fault_handler.c 파일을 project에 추가한다.

6. 컴파일을 하려면 "Options for Target 'Target1'" 의 C/C++ 부분에서 Include Paths에 다음과 같은 경로를 추가하여야 한다. 이것은 "ARMCM3.h" 파일이 위치한 경로이다.

   C:₩Keil₩ARM₩Device₩ARM₩ARMCM3₩Include

7. 이것을 빌드한 다음에 debugger를 실행시키면 된다.