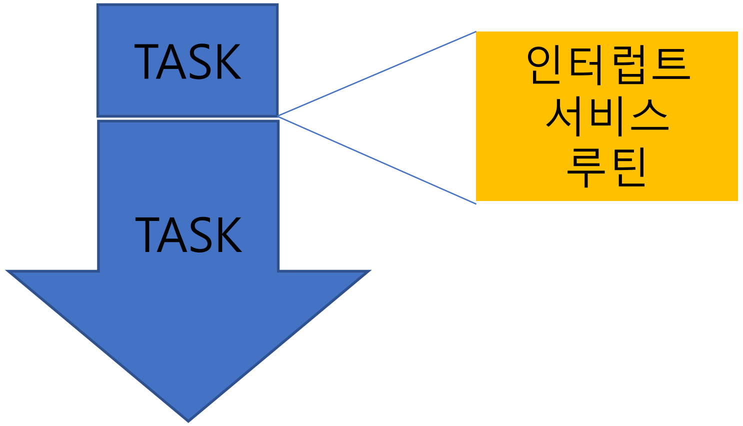
7주차 예비 실험내용

Pusan National University
Embedded Network System Lab.

인터럽트 방식을 활용한 GPIO제어 및 UART 통신

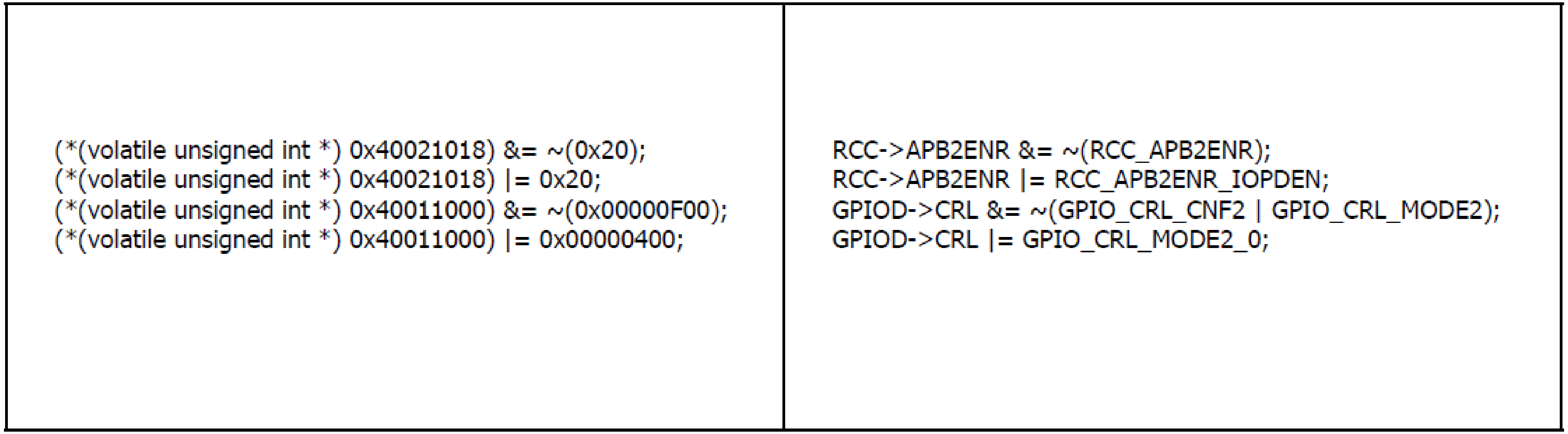
Interrupt



TASK가 동작하다가 인터럽트가 걸리면 하던 TASK
작업을 멈추고 해당 인터럽트 서비스 루틴 (ISR)
수행 후 다시 하던 TASK로 돌아감

- 해당 실험에서는 인터럽트 방식을 사용

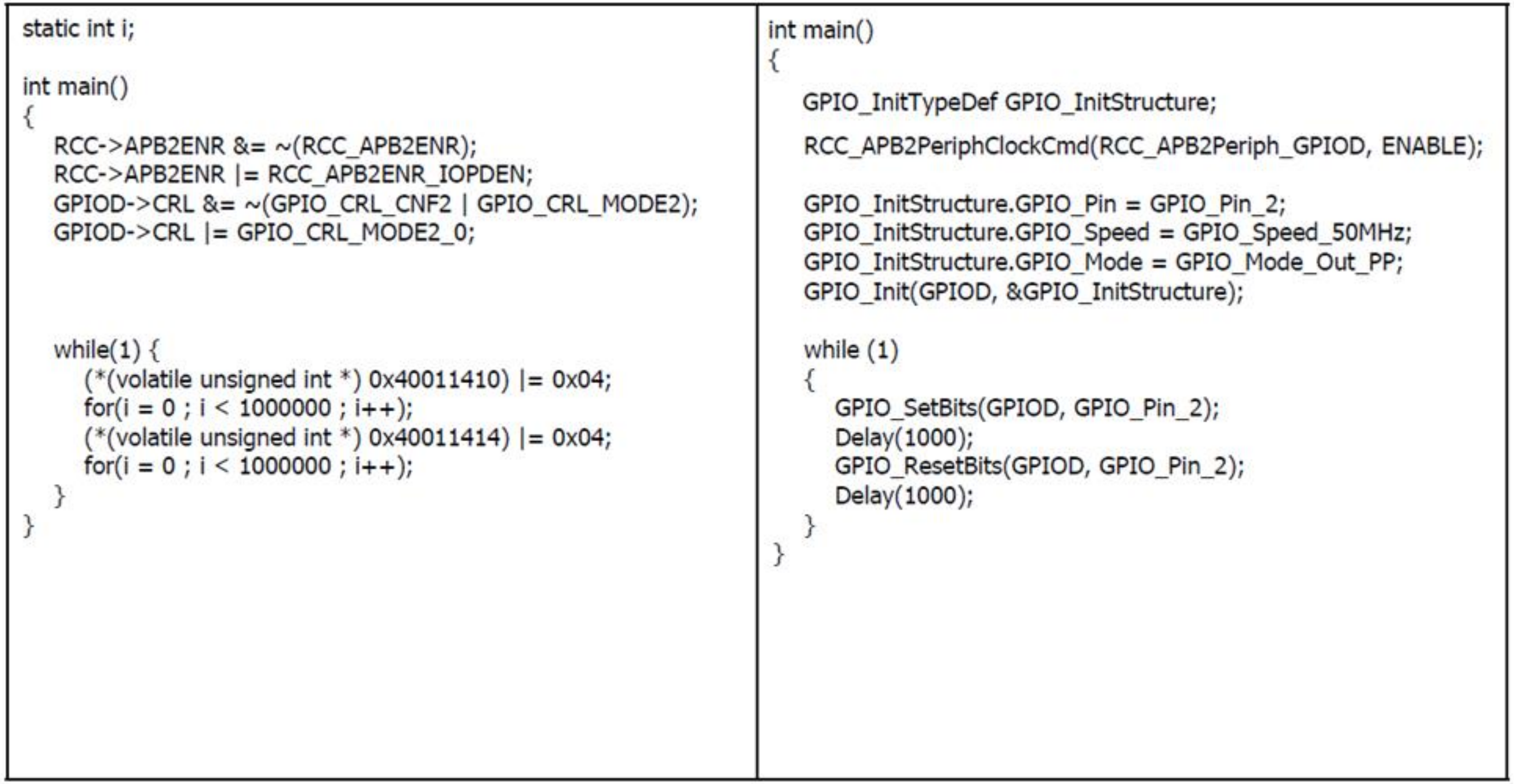
- 직접 주소로 접근 vs 정의된 주소 값 사용



```
(*(volatile unsigned int *) 0x40021018) &= ~(0x20);
(*(volatile unsigned int *) 0x40021018) |= 0x20;
(*(volatile unsigned int *) 0x40011000) &= ~(0x00000F00);
(*(volatile unsigned int *) 0x40011000) |= 0x00000400;
```

```
RCC->APB2ENR &= ~(RCC_APB2ENR);
RCC->APB2ENR |= RCC_APB2ENR_IOPDEN;
GPIOD->CRL &= ~(GPIO_CRL_CNF2 | GPIO_CRL_MODE2);
GPIOD->CRL |= GPIO_CRL_MODE2_0;
```

- 정의된 주소 값 사용 vs Structures, functions 사용



```
static int i;

int main()
{
   RCC->APB2ENR &= ~(RCC_APB2ENR);
   RCC->APB2ENR |= RCC_APB2ENR_IOPDEN;
   GPIOD->CRL &= ~(GPIO_CRL_CNF2 | GPIO_CRL_MODE2);
   GPIOD->CRL |= GPIO_CRL_MODE2_0;



   while(1) {
      (*(volatile unsigned int *) 0x40011410) |= 0x04;
      for(i = 0 ; i < 1000000 ; i++);
      (*(volatile unsigned int *) 0x40011414) |= 0x04;
      for(i = 0 ; i < 1000000 ; i++);
   }
}
```

```
int main()
{
    GPIO_InitTypeDef GPIO_InitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD, ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_Init(GPIOD, &GPIO_InitStructure);

    while (1)
    {
       GPIO_SetBits(GPIOD, GPIO_Pin_2);
       Delay(1000);
       GPIO_ResetBits(GPIOD, GPIO_Pin_2);
       Delay(1000);
    }
}
```

- EXTI(External interrupt/event controller block diagram)

Figure 21. External interrupt/event GPIO mapping

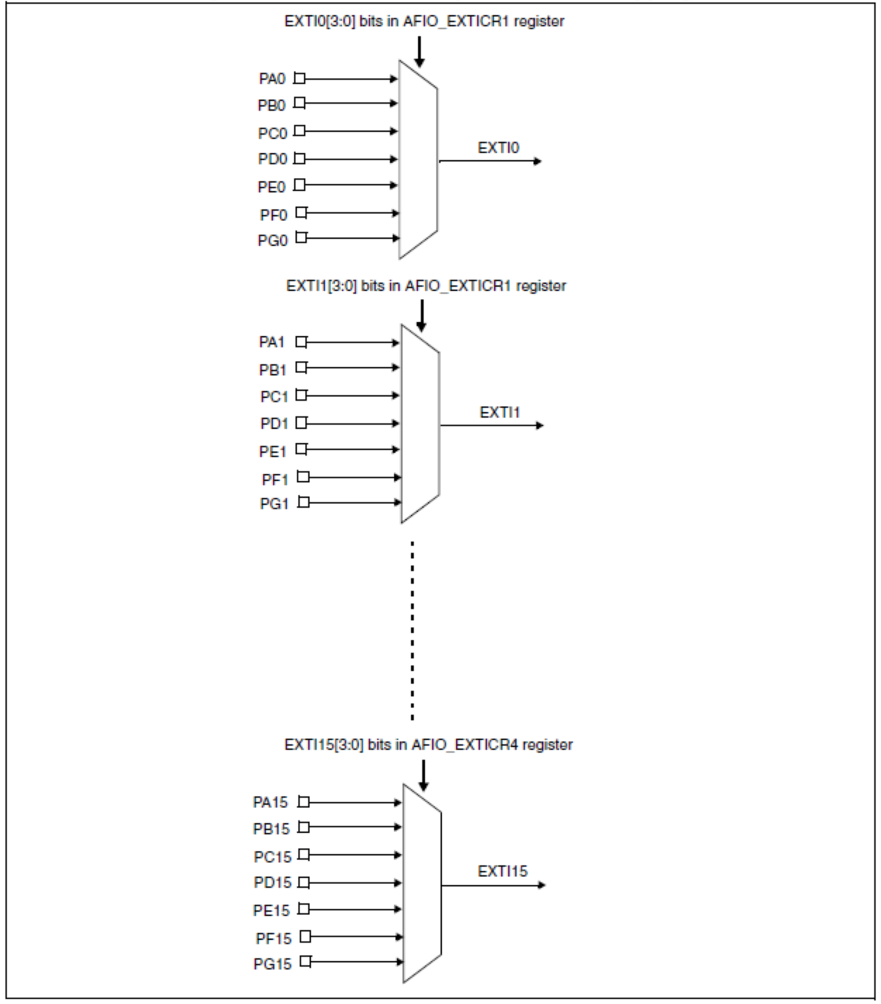
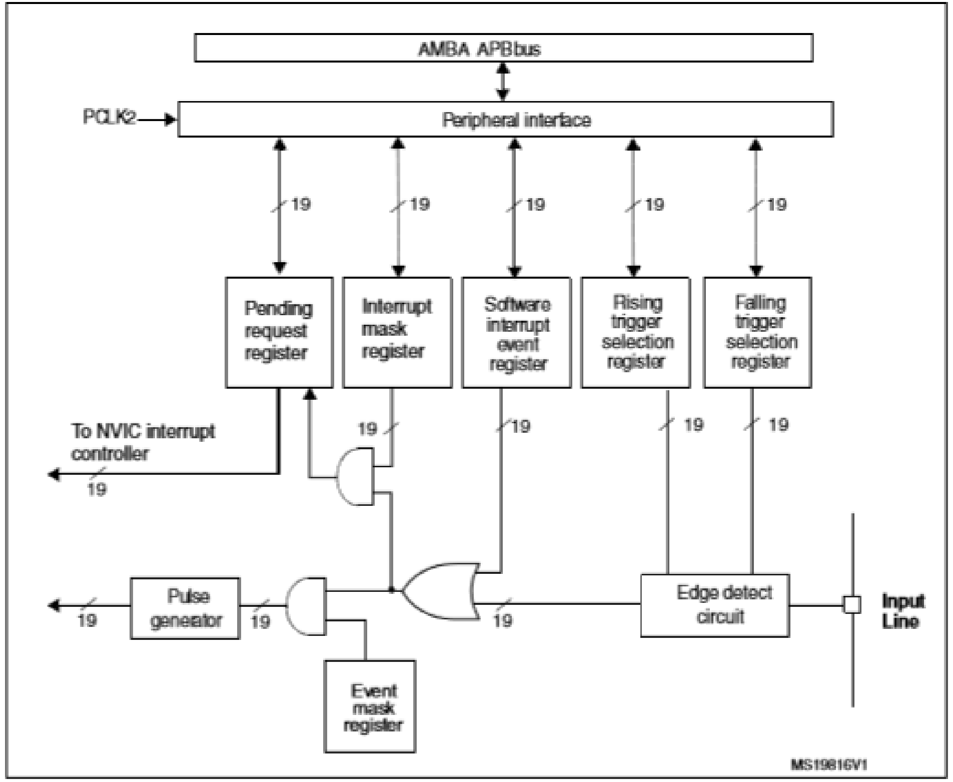
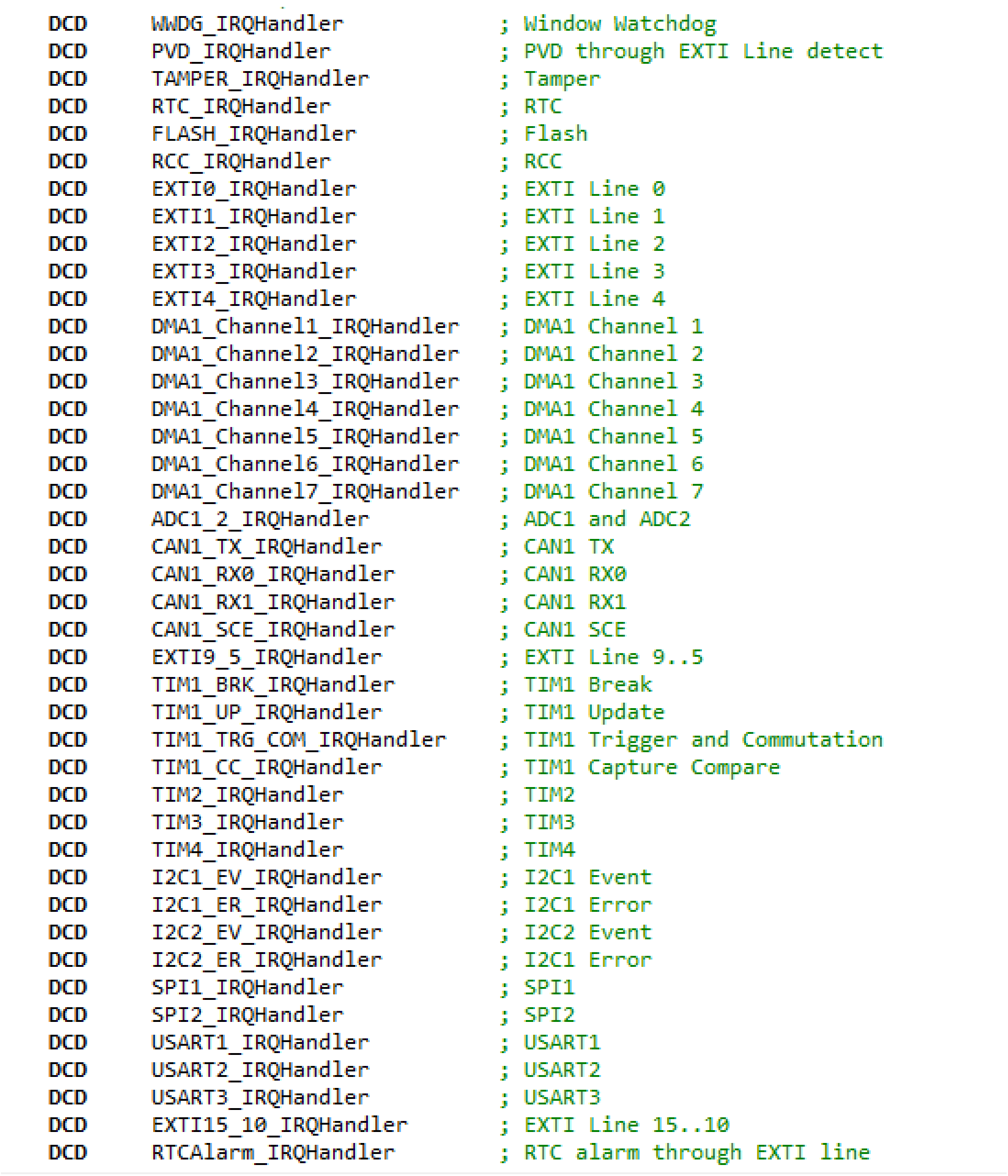

EXTI0[3:0] bits in AFIO_EXTICR1 register
PA0
PB0
PC0
PD0
PE0
PF0
PG0
EXTI0
EXTI1[3:0] bits in AFIO_EXTICR1 register
PA1
PB1
PC1
PD1
PE1
PF1
PG1
EXTI1
EXTI15[3:0] bits in AFIO_EXTICR4 register
PA15
PB15
PC15
PD15
PE15
PF15
PG15
EXTI15

Figure 20. External interrupt/event controller block diagram



- Interrupt Vector Table
  - (startup_stm32f10x_cl.s)

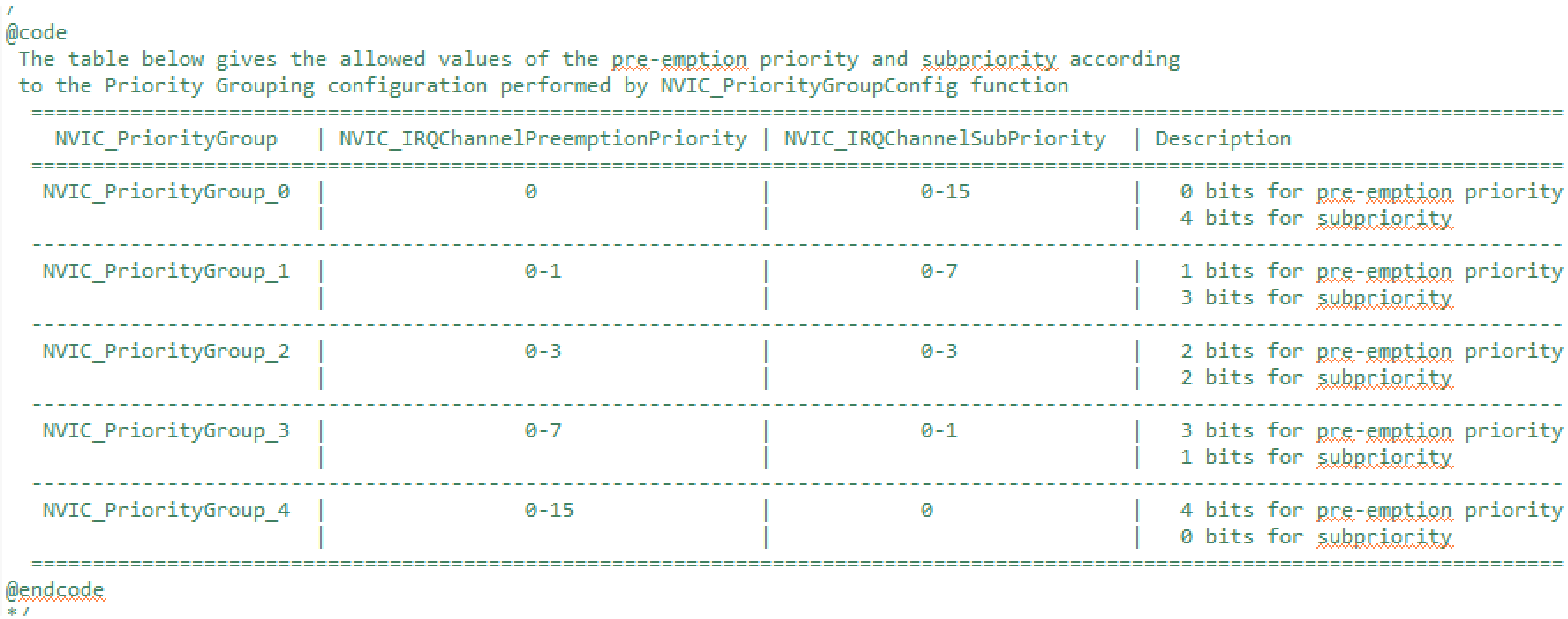
✓ 각 인터럽트 핸들러에서 호출되는 함수 프로토타입이 정의되어 있음



```
DCD     WWDG_IRQHandler            ; Window Watchdog
DCD     PVD_IRQHandler             ; PVD through EXTI Line detect
DCD     TAMPER_IRQHandler          ; Tamper
DCD     RTC_IRQHandler             ; RTC
DCD     FLASH_IRQHandler           ; Flash
DCD     RCC_IRQHandler             ; RCC
DCD     EXTI0_IRQHandler           ; EXTI Line 0
DCD     EXTI1_IRQHandler           ; EXTI Line 1
DCD     EXTI2_IRQHandler           ; EXTI Line 2
DCD     EXTI3_IRQHandler           ; EXTI Line 3
DCD     EXTI4_IRQHandler           ; EXTI Line 4
DCD     DMA1_Channel1_IRQHandler   ; DMA1 Channel 1
DCD     DMA1_Channel2_IRQHandler   ; DMA1 Channel 2
DCD     DMA1_Channel3_IRQHandler   ; DMA1 Channel 3
DCD     DMA1_Channel4_IRQHandler   ; DMA1 Channel 4
DCD     DMA1_Channel5_IRQHandler   ; DMA1 Channel 5
DCD     DMA1_Channel6_IRQHandler   ; DMA1 Channel 6
DCD     DMA1_Channel7_IRQHandler   ; DMA1 Channel 7
DCD     ADC1_2_IRQHandler          ; ADC1 and ADC2
DCD     CAN1_TX_IRQHandler         ; CAN1 TX
DCD     CAN1_RX0_IRQHandler        ; CAN1 RX0
DCD     CAN1_RX1_IRQHandler        ; CAN1 RX1
DCD     CAN1_SCE_IRQHandler        ; CAN1 SCE
DCD     EXTI9_5_IRQHandler         ; EXTI Line 9..5
DCD     TIM1_BRK_IRQHandler        ; TIM1 Break
DCD     TIM1_UP_IRQHandler         ; TIM1 Update
DCD     TIM1_TRG_COM_IRQHandler    ; TIM1 Trigger and Commutation
DCD     TIM1_CC_IRQHandler         ; TIM1 Capture Compare
DCD     TIM2_IRQHandler            ; TIM2
DCD     TIM3_IRQHandler            ; TIM3
DCD     TIM4_IRQHandler            ; TIM4
DCD     I2C1_EV_IRQHandler         ; I2C1 Event
DCD     I2C1_ER_IRQHandler         ; I2C1 Error
DCD     I2C2_EV_IRQHandler         ; I2C2 Event
DCD     I2C2_ER_IRQHandler         ; I2C1 Error
DCD     SPI1_IRQHandler            ; SPI1
DCD     SPI2_IRQHandler            ; SPI2
DCD     USART1_IRQHandler          ; USART1
DCD     USART2_IRQHandler          ; USART2
DCD     USART3_IRQHandler          ; USART3
DCD     EXTI15_10_IRQHandler       ; EXTI Line 15..10
DCD     RTCAlarm_IRQHandler        ; RTC alarm through EXTI line
```

- NVIC : 여러 인터럽트를 관리해주는 컨트롤러
  - ✓ 여러 인터럽트의 우선 순위를 정해줄 수 있음

- NVIC_Priority_Table
  - (misc.h)



```
/
@code
 The table below gives the allowed values of the pre-emption priority and subpriority according
 to the Priority Grouping configuration performed by NVIC_PriorityGroupConfig function
  ============================================================================================================================
    NVIC_PriorityGroup   | NVIC_IRQChannelPreemptionPriority | NVIC_IRQChannelSubPriority  | Description
  ============================================================================================================================
   NVIC_PriorityGroup_0  |                0                  |            0-15             |   0 bits for pre-emption priority
                         |                                   |                             |   4 bits for subpriority
  ----------------------------------------------------------------------------------------------------------------------------
   NVIC_PriorityGroup_1  |                0-1                |            0-7              |   1 bits for pre-emption priority
                         |                                   |                             |   3 bits for subpriority
  ----------------------------------------------------------------------------------------------------------------------------
   NVIC_PriorityGroup_2  |                0-3                |            0-3              |   2 bits for pre-emption priority
                         |                                   |                             |   2 bits for subpriority
  ----------------------------------------------------------------------------------------------------------------------------
   NVIC_PriorityGroup_3  |                0-7                |            0-1              |   3 bits for pre-emption priority
                         |                                   |                             |   1 bits for subpriority
  ----------------------------------------------------------------------------------------------------------------------------
   NVIC_PriorityGroup_4  |                0-15               |            0                |   4 bits for pre-emption priority
                         |                                   |                             |   0 bits for subpriority
  ============================================================================================================================
@endcode
*/
```

- 발표내용 준비
  - Polling vs Interrupt
  - H/W vs S/W Interrupt
  - EXTI
  - NVIC
  - 실험에 도움될 만한 것들