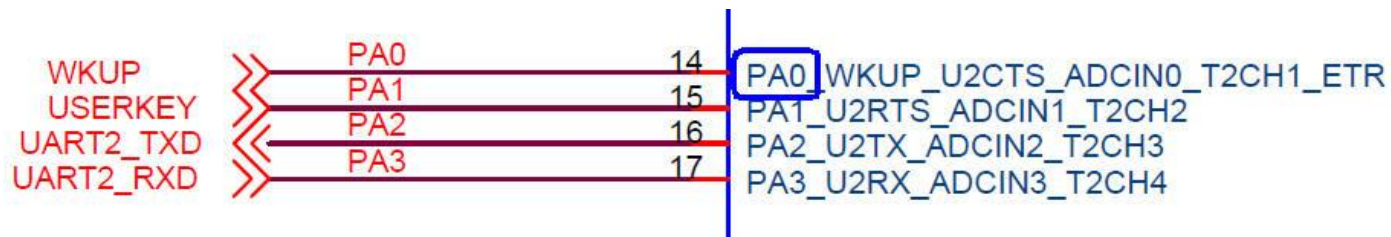


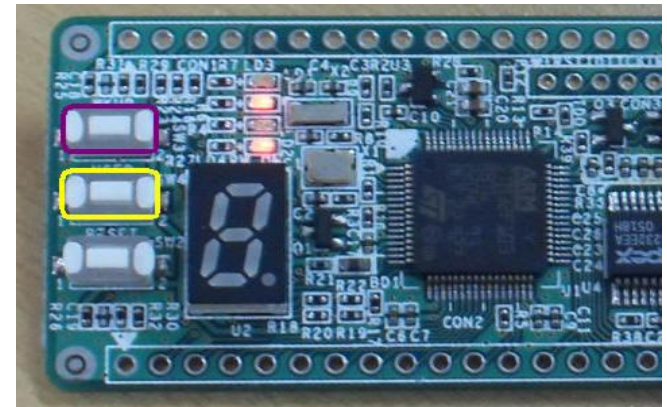
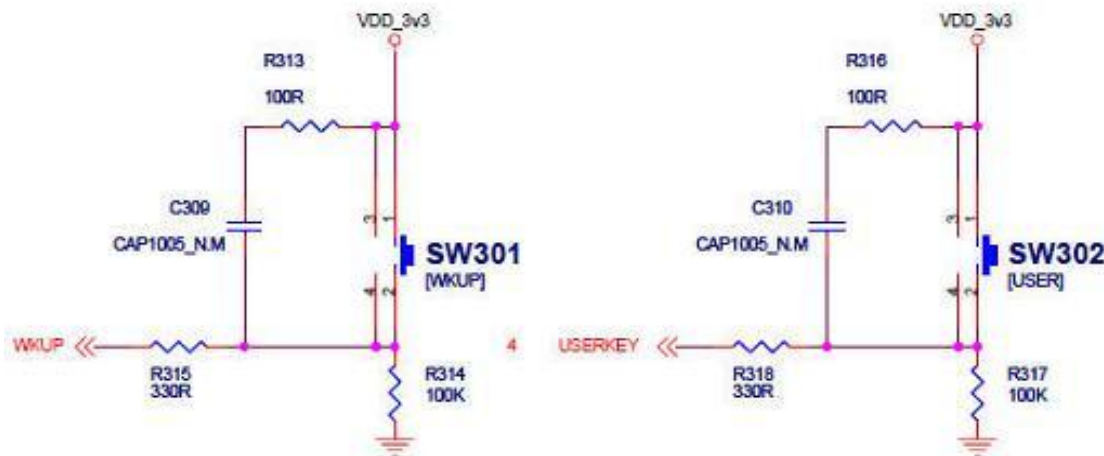
Key Interrupt 실습

2009.11.20

Key Button 구조



Keys



- 위 버튼 WKUP - SW301 Key #1에 할당
- 중간에 있는 버튼 USER - SW302 Key #2에 할당

No Interrupt version

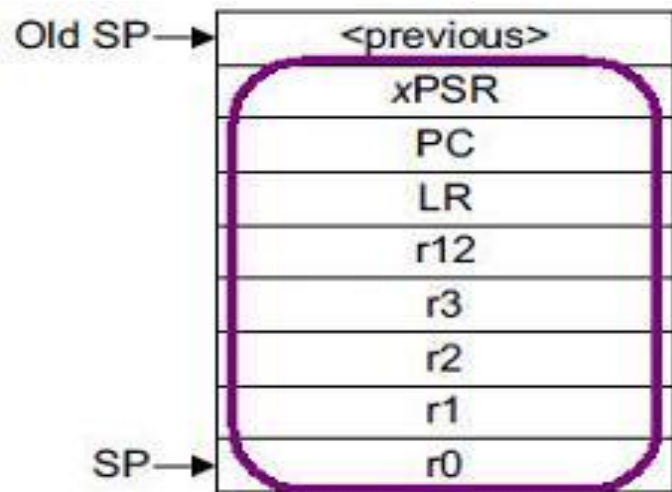
```
void KEY_Test (void)
{
    while(1)
    {
        ... ..
        if(GPIO_ReadInputDataBit(GPIO_KEY, GPIO_KEY1_PIN) == Bit_SET) {
            LED_On_Red();
        } else {
            LED_Off_Red();
        }

        if(GPIO_ReadInputDataBit(GPIO_KEY, GPIO_KEY2_PIN) == Bit_SET) {
            LED_On_Yellow();
        } else {
            LED_Off_Yellow();
        }
    }
}
```

Exception 종류

위치	Exception type	Priority	설명
0	-	-	리셋 시 vector table의 첫 entry
1	Reset	-3 (highest)	Power up과 Warm reset시.
2	Non-maskable Interrupt (NMI)	-2	pre-empted되지 않는다.
3	Hard Fault	-1	모든 종류의 Fault
4	Memory Management	Configurable	MPU mismatch
5	Bus Fault	Configurable	Pre-fetch, memory fault
6	Usage Fault	Configurable	Undefined instruction
7~10	-	-	Reserved
11	SVCall	Configurable	System service 호출
12	Debug Monitor	Configurable	not halting Debug monitor
13	-	-	Reserved
14	PendSV	Configurable	Pendable request
15	SysTick	Configurable	System tick timer.
16~255	External Interrupt	Configurable	core 밖 외부 interrupt (0~239)

pre-emption 상태 stack 변화

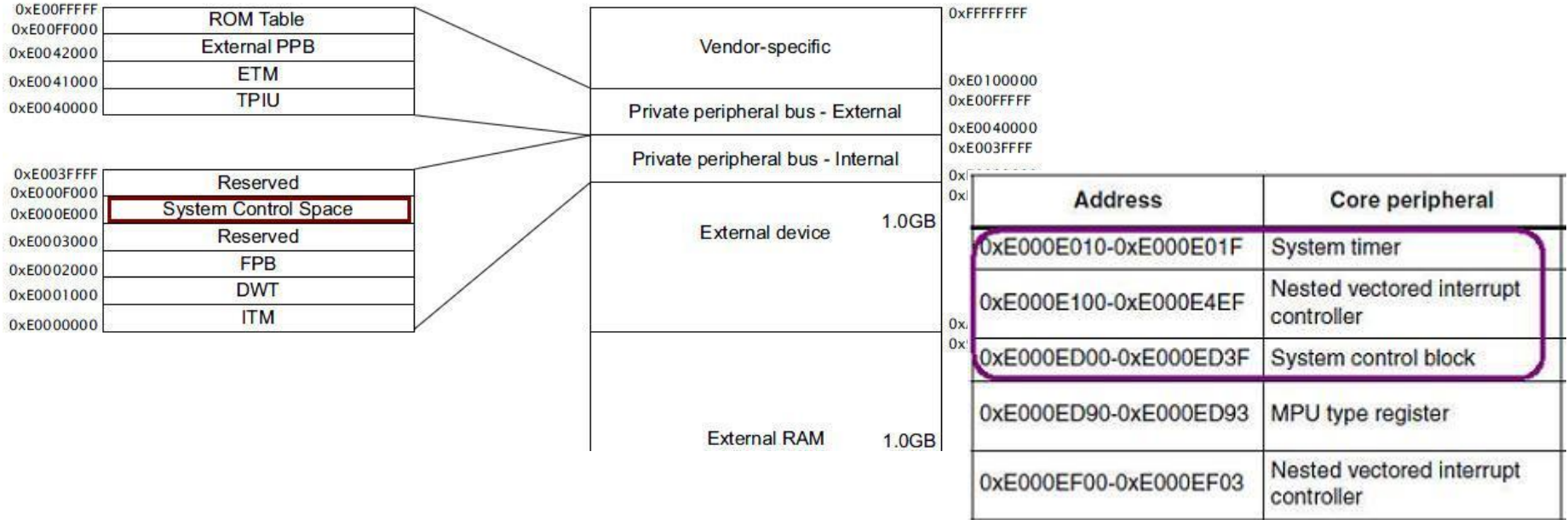


- interrupt 발생시 CPU는 자동으로 processor의 상태를 exception의 stack에 저장하고 빠져 나올 때 Interrupt Service Routine (ISR)의 끝에서 복구를 시켜준다.
- 여기에 저장되는 레지스터들은 **R0-3, R12, LR, PSR, PC** 등이다.
- 자동으로 이러한 부분을 처리해 줌으로 인해서 general한 C 함수를 interrupt handler로 사용할 수 있는 이점을 가져다 줌과 동시에 IRQ 지연 시간도 줄여주게 된다.
- pre-emption이 일어난 상태에서의 stack의 변화 모양은 위 그림. xPSR 부터 시작해서 8개의 레지스터가 차례로 저장되어 있는 모습

STM32 Vector table

Exception number	IRQ number	Offset	Vector
83	67	0x014C	IRQ67
.	.	.	.
.	.	.	.
.	.	.	.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5	0x002C	SVCall
10			Reserved
9			
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

System Control Space



```
#define SCS_BASE    (0xE000E000)    /*!< System Control Space Base Address */

#define SysTick_BASE (SCS_BASE + 0x0010)    /*!< SysTick Base Address */
#define NVIC_BASE   (SCS_BASE + 0x0100)    /*!< NVIC Base Address */
#define SCB_BASE    (SCS_BASE + 0x0D00)    /* System Control Block Base */

#define SysTick      ((SysTick_Type *) SysTick_BASE) /* SysTick configuration struct */
#define NVIC         ((NVIC_Type *)  NVIC_BASE)    /*!< NVIC configuration struct */
#define SCB          ((SCB_Type *)   SCB_BASE)     /*!< SCB configuration struct */
```

NVIC_Configuration

```
void NVIC_Configuration(void)
{
    NVIC_InitTypeDef NVIC_InitStructure;

    #ifdef VECT_TAB_RAM /* Set the Vector Table base 0x20000000 */
        NVIC_SetVectorTable(NVIC_VectTab_RAM, 0x0);
    #else /* VECT_TAB_FLASH */ /* Set the Vector Table base 0x08000000 */
        NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
    #endif

    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1); /* one bit for preemption priority */

    /* Enable the EXTI0 Interrupt */
    NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

    /* Enable the EXTI1 Interrupt */
    NVIC_InitStructure.NVIC_IRQChannel = EXTI1_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
```


NVIC Vector Table 설정

```
#define NVIC_VectTab_RAM      ((uint32_t)0x20000000)
#define NVIC_VectTab_FLASH    ((uint32_t)0x08000000)
```

```
void NVIC_SetVectorTable(uint32_t NVIC_VectTab, uint32_t Offset)
{
    SCB->VTOR = NVIC_VectTab
                | (Offset & (uint32_t)0x1FFFFFF80);
}
```

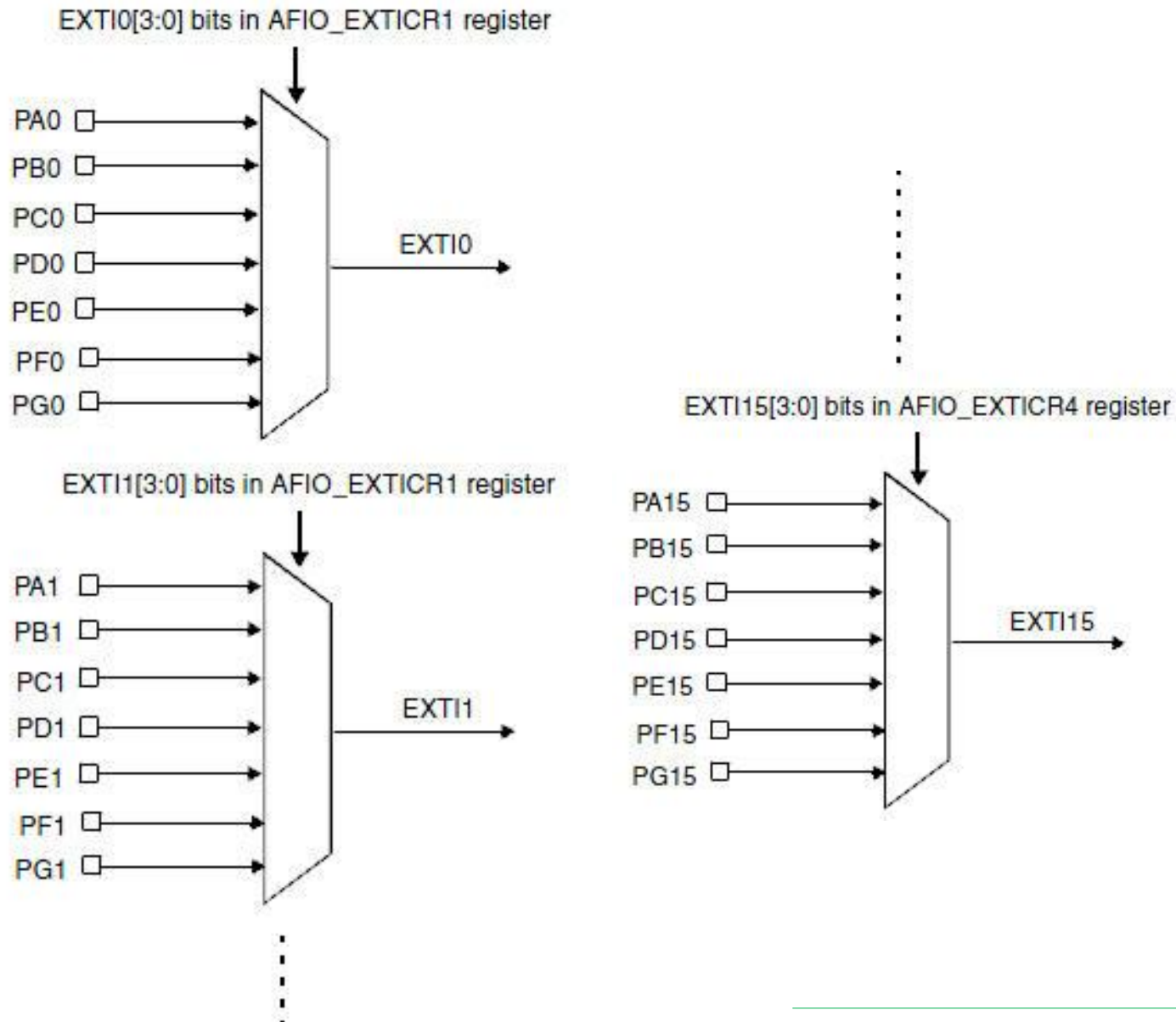
- SCB는 SCB_Type 포인터
- SCB 영역은 0xE000ED00 부분
- System Control Block의 기본적인 레지스터들
- SCB, NVIC, SysTick에 대한 부분은 PM0056. STM32F10xxx Cortex-M3 programming manual 부분을 참조해야 한다.
- **Vector Table Offset Register**를 **설정**하는 것임.

Exception priority

PRIGROUP[2:0]	Binary point position	Pre-emption field	Subpriority field	Number of pre-emption priorities	Number of subpriorities
b000	bxxxxxxx.y	[7:1]	[0]	128	2
b001	bxxxxxx.yy	[7:2]	[1:0]	64	4
b010	bxxxxx.yyy	[7:3]	[2:0]	32	8
b011	bxxxx.yyyy	[7:4]	[3:0]	16	16
b100	bxxx.yyyyy	[7:5]	[4:0]	8	32
b101	bxx.yyyyyy	[7:6]	[5:0]	4	64
b110	bx.yyyyyyy	[7]	[6:0]	2	128
b111	b.yyyyyyy	None	[7:0]	0	256

PRIGROUP [2:0]	Interrupt priority level value, PRI_M[7:4]			Number of	
	Binary point ⁽¹⁾	Group priority bits	Subpriority bits	Group priorities	Sub priorities
0b100	0bxxx.y	[7:5]	[4]	8	2
0b101	0bxx.yy	[7:6]	[5:4]	4	4
0b110	0bx.yyy	[7]	[6:4]	2	8
0b111	0b.yyyy	None	[7:4]	1	16

External interrupt GPIO mapping



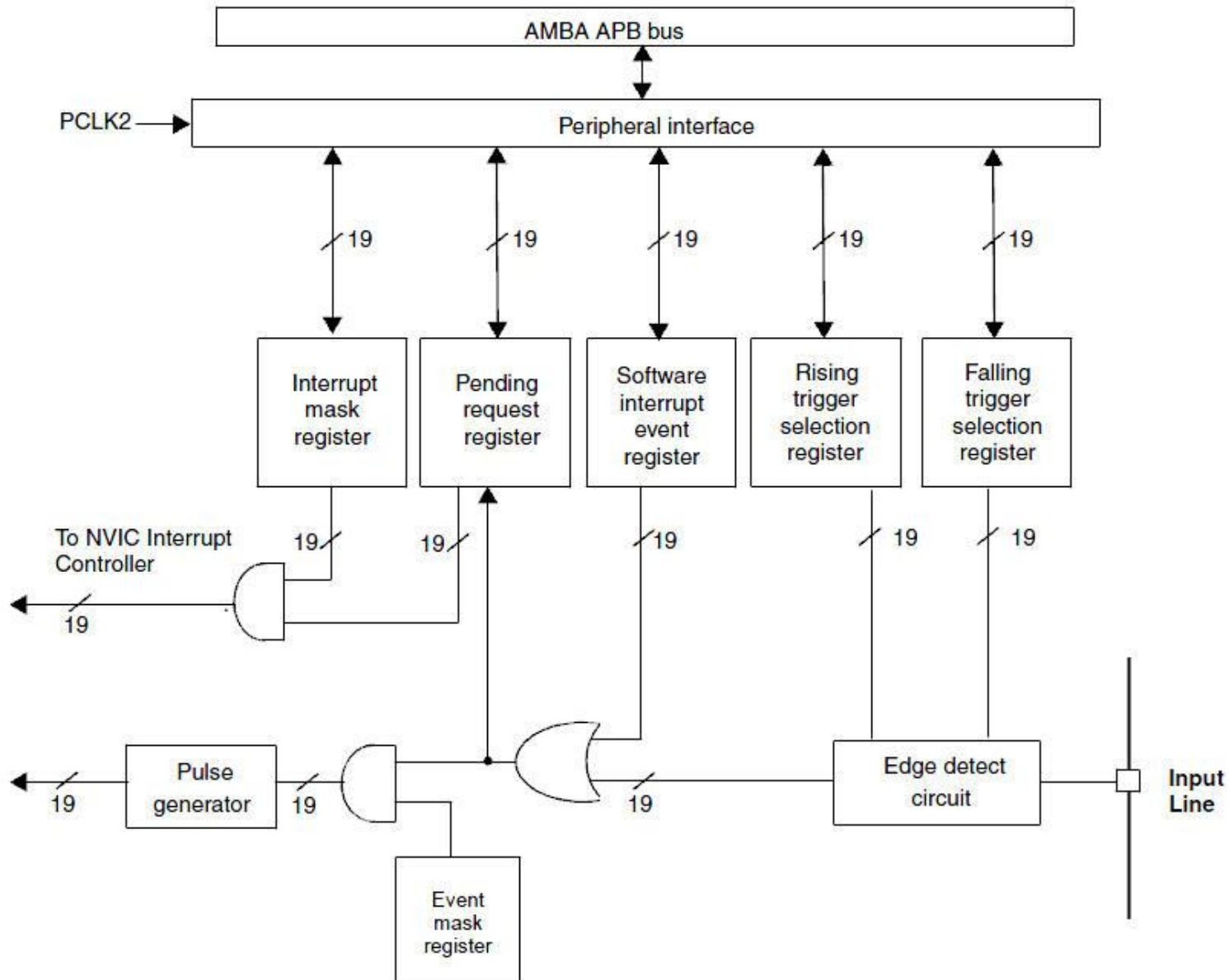
EXTI line 16, 17, 18

- 외부 핀과 연결된 EXTI는 0~15까지 총 16개
 - GPIO Line으로 보면 16 X 7 해서 112개
- 16,17,18은 외부에 핀이 나와있는 것이 아니라 내부적으로 event에 대한 처리를 하는 것.
 - 이것들도 External interrupt/event controller block을 함께 이용
- The three other EXTI lines are connected as follows:
 - EXTI line 16 is connected to the PVD (programmable voltage detector) output
 - EXTI line 17 is connected to the RTC Alarm event
 - EXTI line 18 is connected to the USB Wakeup event

External interrupt

1	8	settable	PVD	PVD through EXTI Line detection interrupt	0x0000_0044
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000_0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000_005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000_0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000_0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000_0068
23	30	settable	EXTI9_5	EXTI Line[9:5] interrupts	0x0000_009C
40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000_00E0
41	48	settable	RTCAlarm	RTC alarm through EXTI line interrupt	0x0000_00E4
42	49	settable	USBWakeup	USB wakeup from suspend through EXTI line interrupt	0x0000_00E8

External interrupt/event controller block diagram



IRQn_Type

```
typedef enum IRQn {
/***** Cortex-M3 Processor Exceptions Numbers *****/
    NonMaskableInt_IRQn    = -14,    /*!< 2 Non Maskable Interrupt */
    MemoryManagement_IRQn  = -12,    /*!< 4 Memory Management Interrupt */
    BusFault_IRQn           = -11,    /*!< 5 Cortex-M3 Bus Fault Interrupt */
    UsageFault_IRQn         = -10,    /*!< 6 Cortex-M3 Usage Fault Interrupt */
    SVCall_IRQn             = -5,     /*!< 11 Cortex-M3 SV Call Interrupt */
    DebugMonitor_IRQn       = -4,     /*!< 12 Cortex-M3 Debug Monitor Interrupt */
    PendSV_IRQn             = -2,     /*!< 14 Cortex-M3 Pend SV Interrupt */
    SysTick_IRQn           = -1,     /*!< 15 Cortex-M3 System Tick Interrupt */
/***** STM32 specific Interrupt Numbers *****/
    .....
    EXTI0_IRQn            = 6,      /*!< EXTI Line0 Interrupt      */
    EXTI1_IRQn            = 7,      /*!< EXTI Line1 Interrupt      */
    EXTI2_IRQn             = 8,       /*!< EXTI Line2 Interrupt      */
    EXTI3_IRQn             = 9,       /*!< EXTI Line3 Interrupt      */
    EXTI4_IRQn             = 10,      /*!< EXTI Line4 Interrupt      */
    .....
    EXTI9_5_IRQn           = 23,      /*!< External Line[9:5] Interrupts */
    .....
    EXTI15_10_IRQn         = 40,      /*!< External Line[15:10] Interrupts */
    .....
} IRQn_Type;
```


startup_stm32f10x_md.s

```
MODULE ?cstartup
;; Forward declaration of sections.
SECTION CSTACK:DATA:NOROOT(3)
SECTION .intvec:CODE:NOROOT(2)
```

```
EXTERN __iar_program_start
PUBLIC __vector_table
```

```
DATA
__vector_table
```

```
DCD sfe(CSTACK) ; section end ...
DCD __iar_program_start
```

```
DCD NMI_Handler ; NMI Handler
DCD HardFault_Handler ; Hard Fault
```

.....

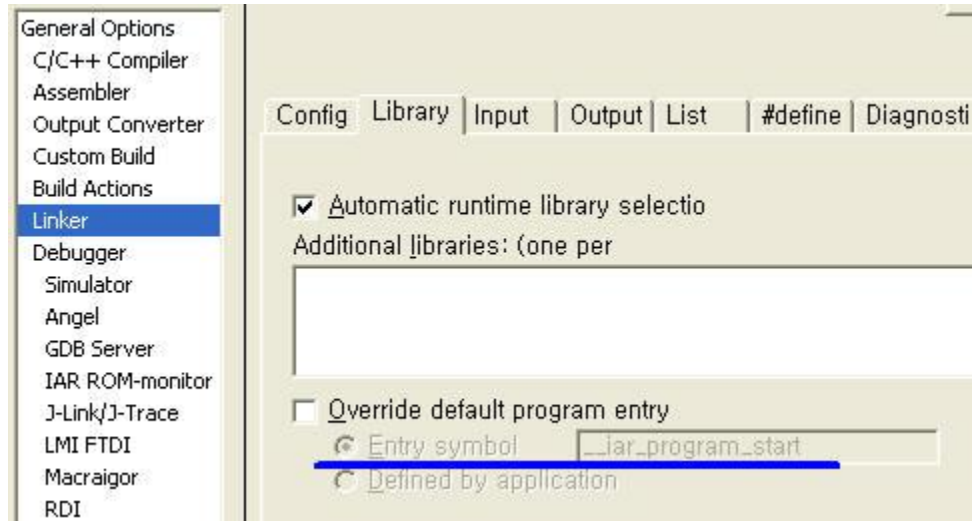
```
DCD SysTick_Handler ; SysTick
```

```
; External Interrupts
```

.....

```
DCD EXTIO_IRQHandler ; EXTI Line 0
DCD EXTI1_IRQHandler ; EXTI Line 1
```

.....



```
17 define memory mem with size = 4G;
18 define region ROM_region = mem:[from __ICFEDIT_region_start__ to __ICFEDIT_region_end__];
19 define region RAM_region = mem:[from __ICFEDIT_region_start__ to __ICFEDIT_region_end__];
20
21 define block CSTACK with alignment = 8, size = 1024;
22 define block HEAP with alignment = 8, size = 1024;
23
24 initialize by copy { readwrite };
25 do not initialize { section .noinit };
26
27 place at address mem:__ICFEDIT_intvec_start__ { readwrite, noinit }
28
29 place in ROM_region { readonly };
30 place in RAM_region { readwrite,
31 block CSTACK, block HEAP };
```


EXTI_Configuration

```
void EXTI_Configuration(void) {
    EXTI_InitTypeDef EXTI_InitStructure;

    /* Configure gpio as input : Button Left-WKUP */ /* Connect EXTI Line to gpio pin */
    GPIO_EXTILineConfig(GPIO_PORTSOURCE_KEY, GPIO_PINSOURCE_KEY1);

    /* Configure EXTI Line to generate an interrupt */
    EXTI_InitStructure.EXTI_Line    = GPIO_EXTI_Line_KEY1;
    EXTI_InitStructure.EXTI_Mode    = EXTI_Mode_Interrupt; // or Event
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling; // or Rising, Rising_Falling
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);

    /* Configure gpio as input : Button Right-USER */ /* Connect EXTI Line to gpio pin */
    GPIO_EXTILineConfig(GPIO_PORTSOURCE_KEY, GPIO_PINSOURCE_KEY2);

    /* Configure EXTI Line to generate an interrupt */
    EXTI_InitStructure.EXTI_Line    = GPIO_EXTI_Line_KEY2;
    EXTI_InitStructure.EXTI_Mode    = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);
}
```

EXTI0_IRQHandler, EXTI1_IRQHandler

```
void EXTI0_IRQHandler(void)
{
    if(EXTI_GetITStatus(GPIO_EXTI_Line_KEY1) != RESET) {
        printf("Left-WKUP Button Press\n");
        EXTI_ClearITPendingBit(GPIO_EXTI_Line_KEY1);
        OutPut_7_SEG_Number(0);
    }
}
```

```
void EXTI1_IRQHandler(void)
{
    if(EXTI_GetITStatus(GPIO_EXTI_Line_KEY2) != RESET) {
        printf("Right-USER Button Press\n");
        EXTI_ClearITPendingBit(GPIO_EXTI_Line_KEY2);
        OutPut_7_SEG_Number(1);
    }
}
```