

# Infineon TC275 Interrupt

Hyeongrae Kim

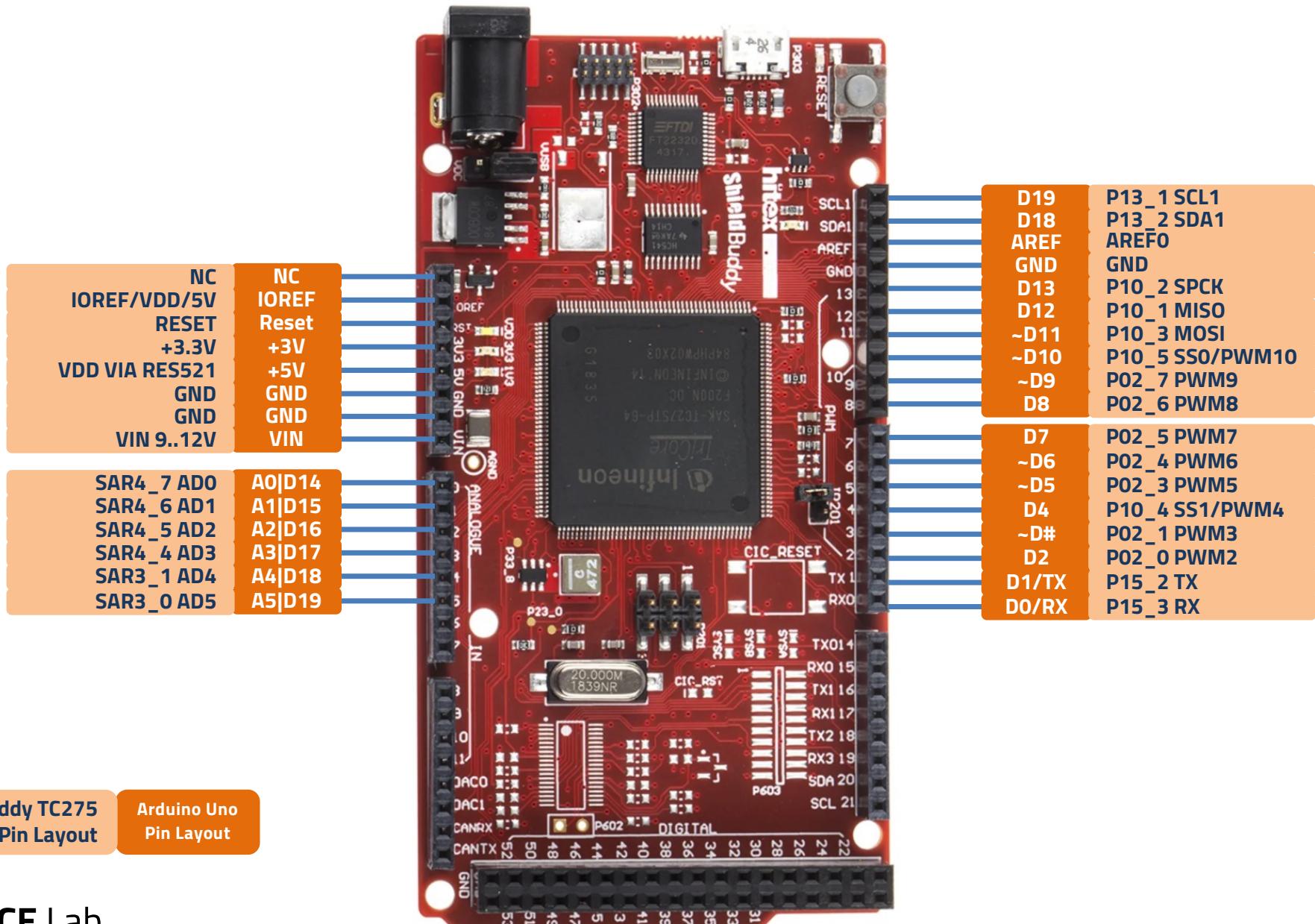
Architecture and Compiler for Embedded System LAB.

School of Electronics Engineering, KNU, KOREA

2021-05-11



# Hitex ShieldBuddy TC275



# Interrupt Flow in TC275

1. External Request Input을 External Request Unit (ERU)이 처리함

- ✓ 각 Input Channel에 있는 ERS에서 4개의 가능한 입력 중 하나의 입력 벡터를 선택함

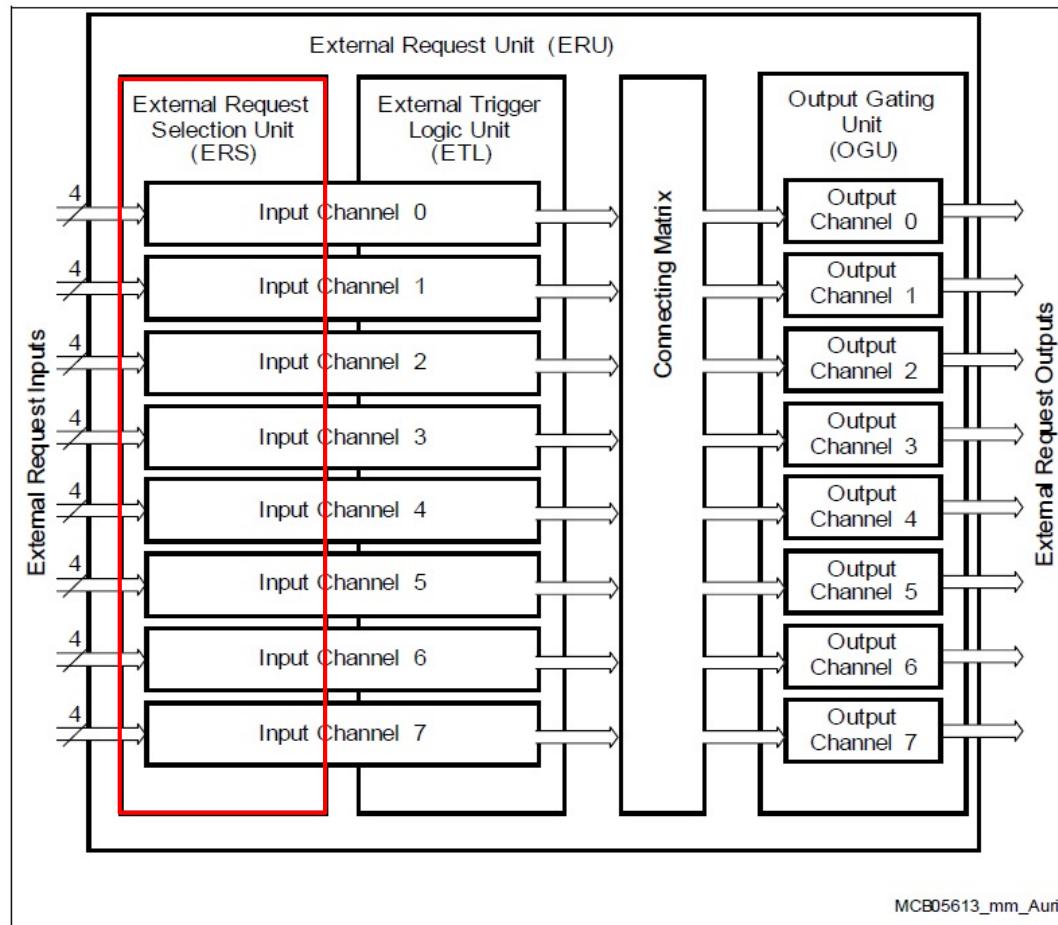


Figure 7-40 External Request Unit Overview

# Interrupt Flow in TC275

- External Request Input을 External Request Unit (ERU)이 처리함
  - 각 Input Channel에 있는 ETL이 지정된 엣지에서 입력을 트리거 이벤트로 전환함

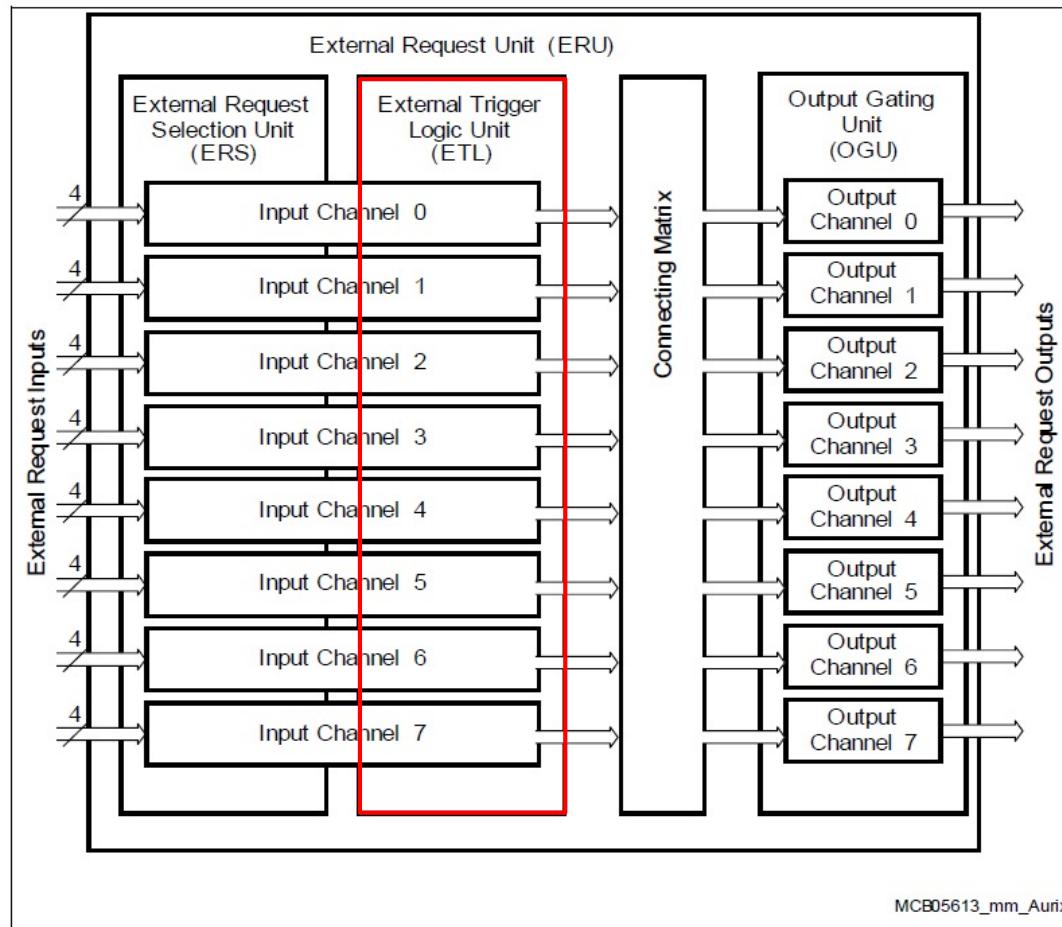


Figure 7-40 External Request Unit Overview

# Interrupt Flow in TC275

1. External Request Input을 External Request Unit (ERU)이 처리함

- ✓ Connecting Matrix는 입력 채널에서 생성된 이벤트를 출력 채널로 배포함

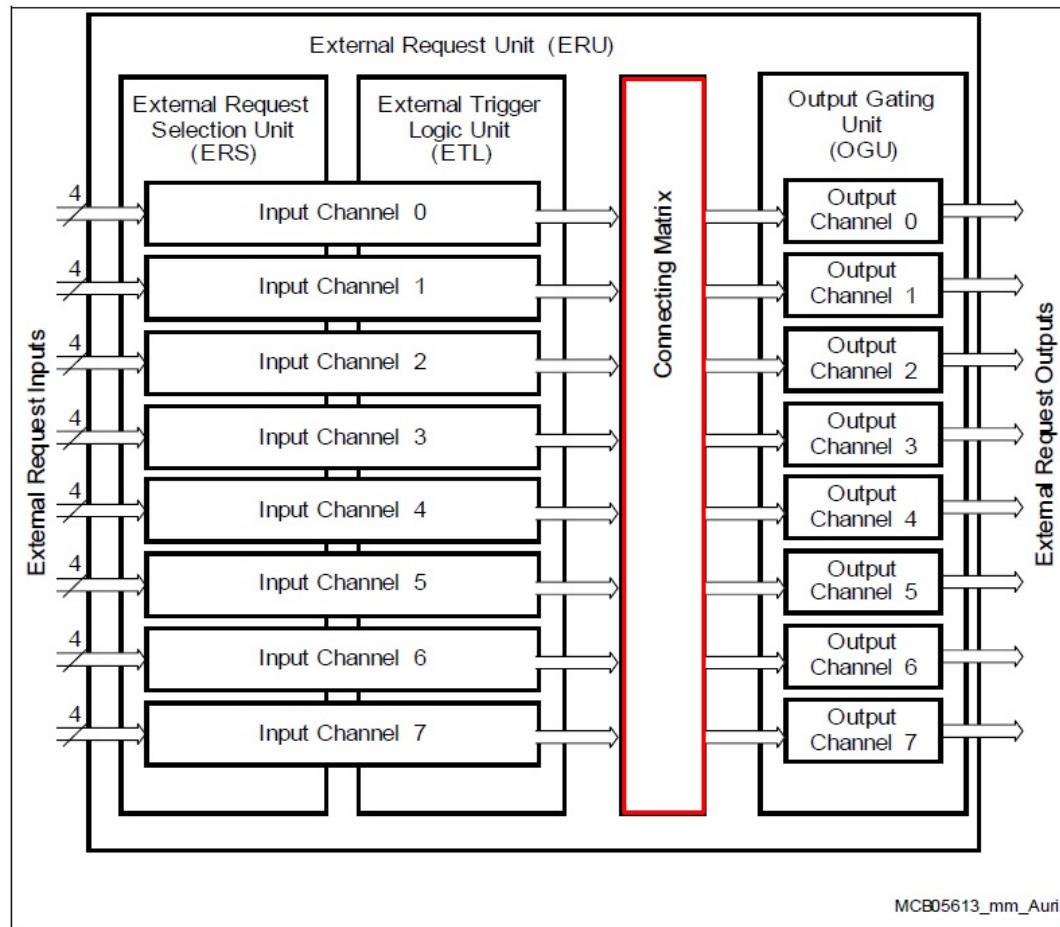


Figure 7-40 External Request Unit Overview

# Interrupt Flow in TC275

1. External Request Input을 External Request Unit (ERU)이 처리함

- ✓ Connecting Matrix는 입력 채널에서 생성된 이벤트를 출력 채널로 배포함

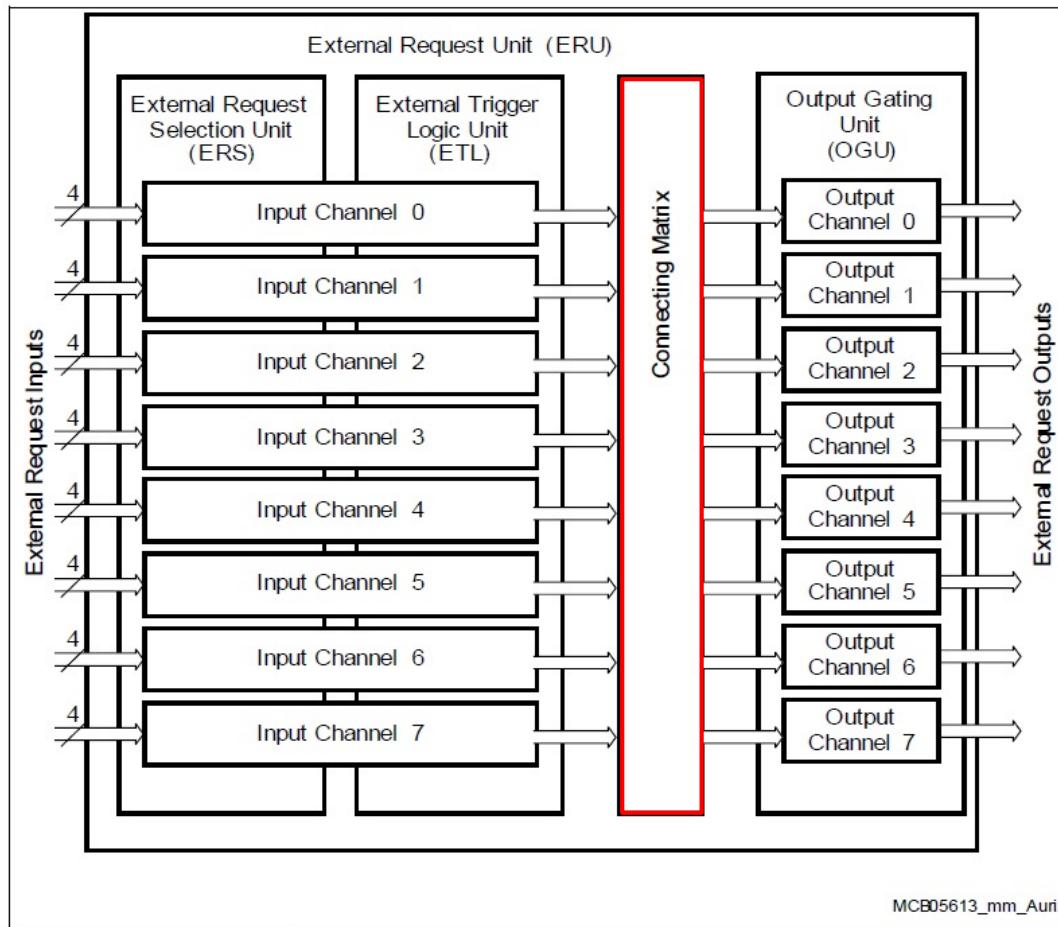
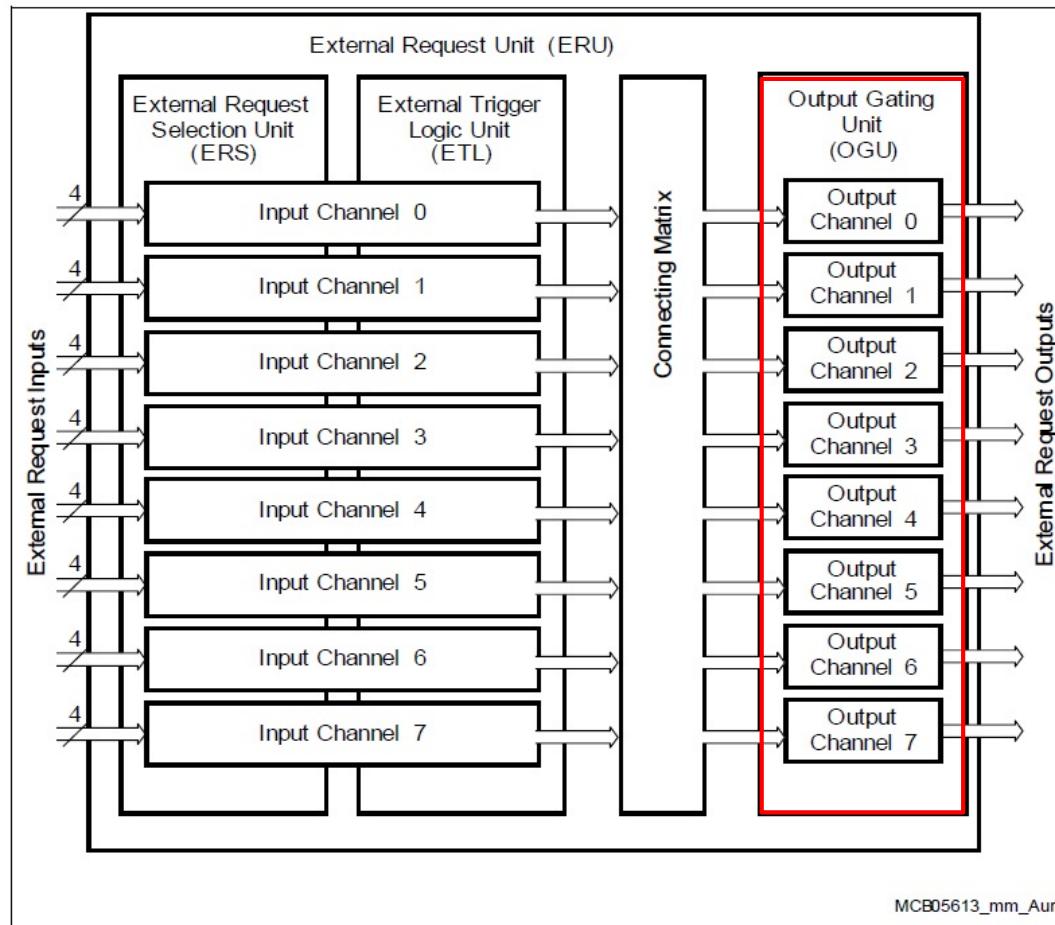


Figure 7-40 External Request Unit Overview

# Interrupt Flow in TC275

## 1. External Request Input을 External Request Unit (ERU)이 처리함

- ✓ Output Gating Unit은 Input Channel로 부터의 트리거 이벤트와 상태 정보를 조합하여 출력을 내보냄
- ✓ 하나의 이벤트가 여러 채널에 갈 수 있고 여러 이벤트가 하나의 채널에서 패턴을 만들 수 있음



# Interrupt Flow in TC275

## 2. ERU의 출력이 Interrupt Router (IR)의 입력이 됨

- ✓ OGU에서 나온 출력이 연결된 Service Request Nodes (SRN)의 입력이 됨
- ✓ SRN은 모든 Interrupt Control Units (ICU)에 연결되어 있고 Service Request Control Register (SRC) 설정을 통해 가능한 Service Provider (CPU0-2, DMA)에 매플링함
- ✓ 각 ICU는 ICU에 매플링된 SRN의 Service Request 간의 인터럽트 중재를 처리함

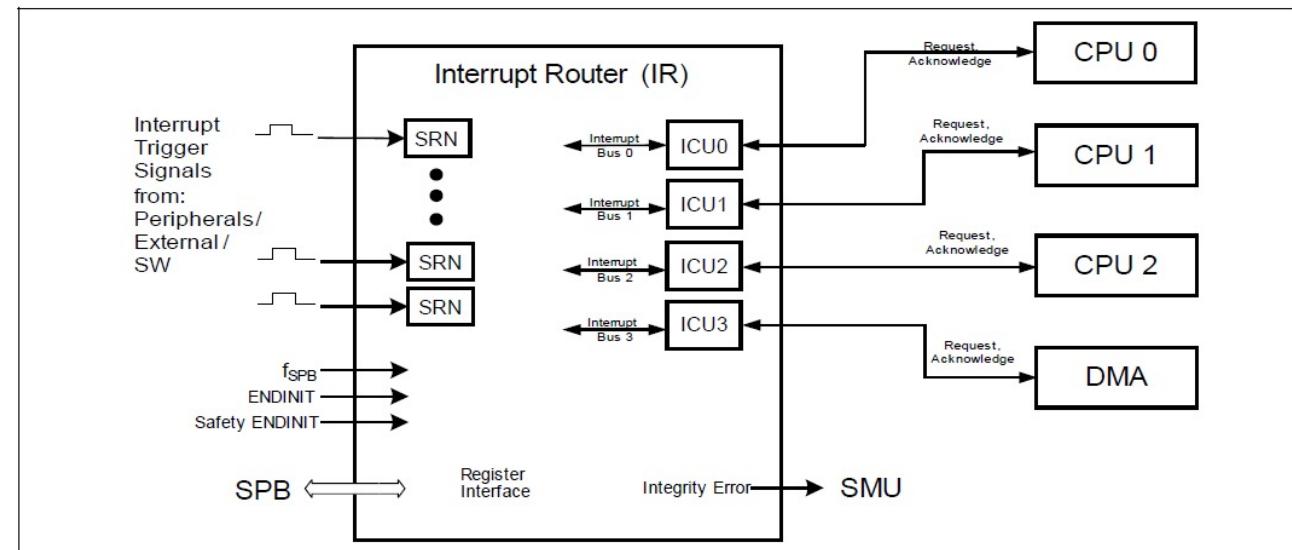
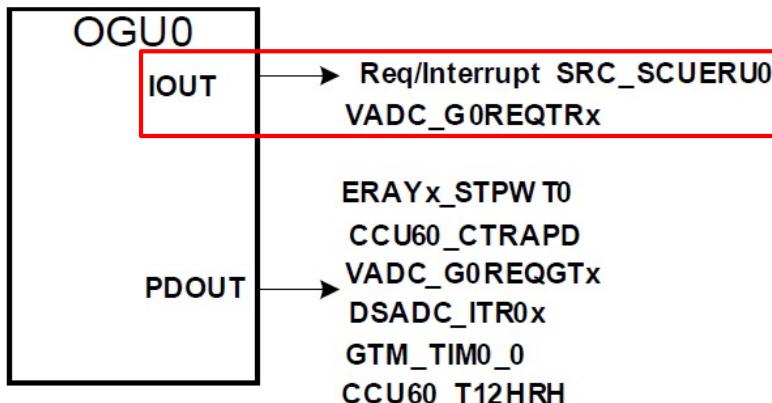


Figure 16-1 Block Diagram of the TC27x Interrupt System

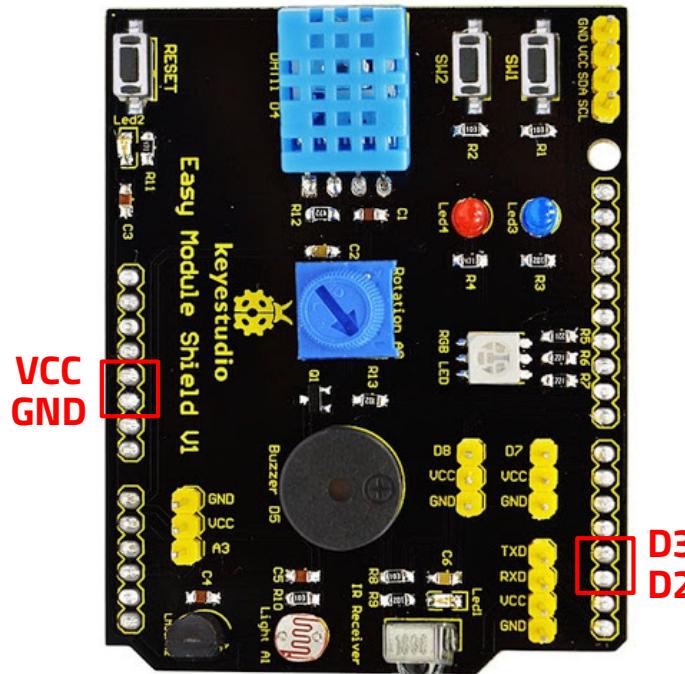
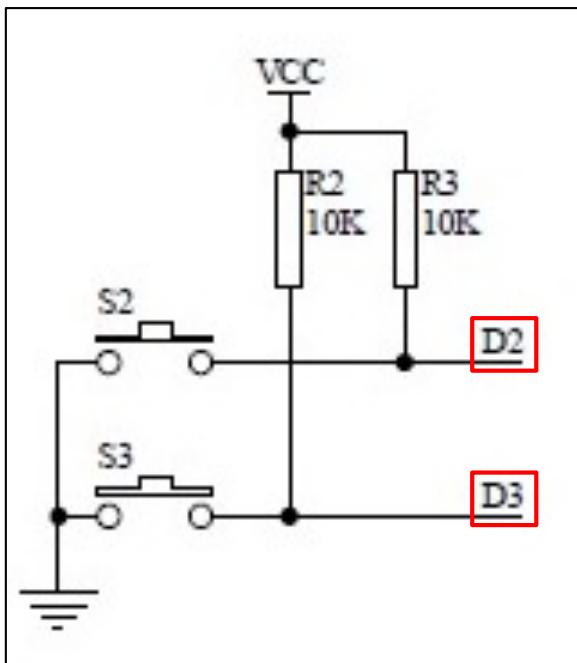
# External Interrupt Example

- External Interrupt를 사용하여 Switch를 눌렀을 때 LED Toggle
  1. 새로운 예제를 위한 프로젝트를 생성한다.
  2. 원하는 동작을 위해 레지스터와 메모리에 직접 접근해서 값을 써야한다.
  3. Switch 사용을 위해 Board Schematic과 Datasheet에서 Switch 연결 정보를 파악한다.
  4. External Interrupt를 사용하기 위해 Datasheet를 분석한다.
  5. 분석 결과를 활용해 임베디드 프로그래밍을 한다.

# External Interrupt Example

## 1. Switch 연결 정보 파악

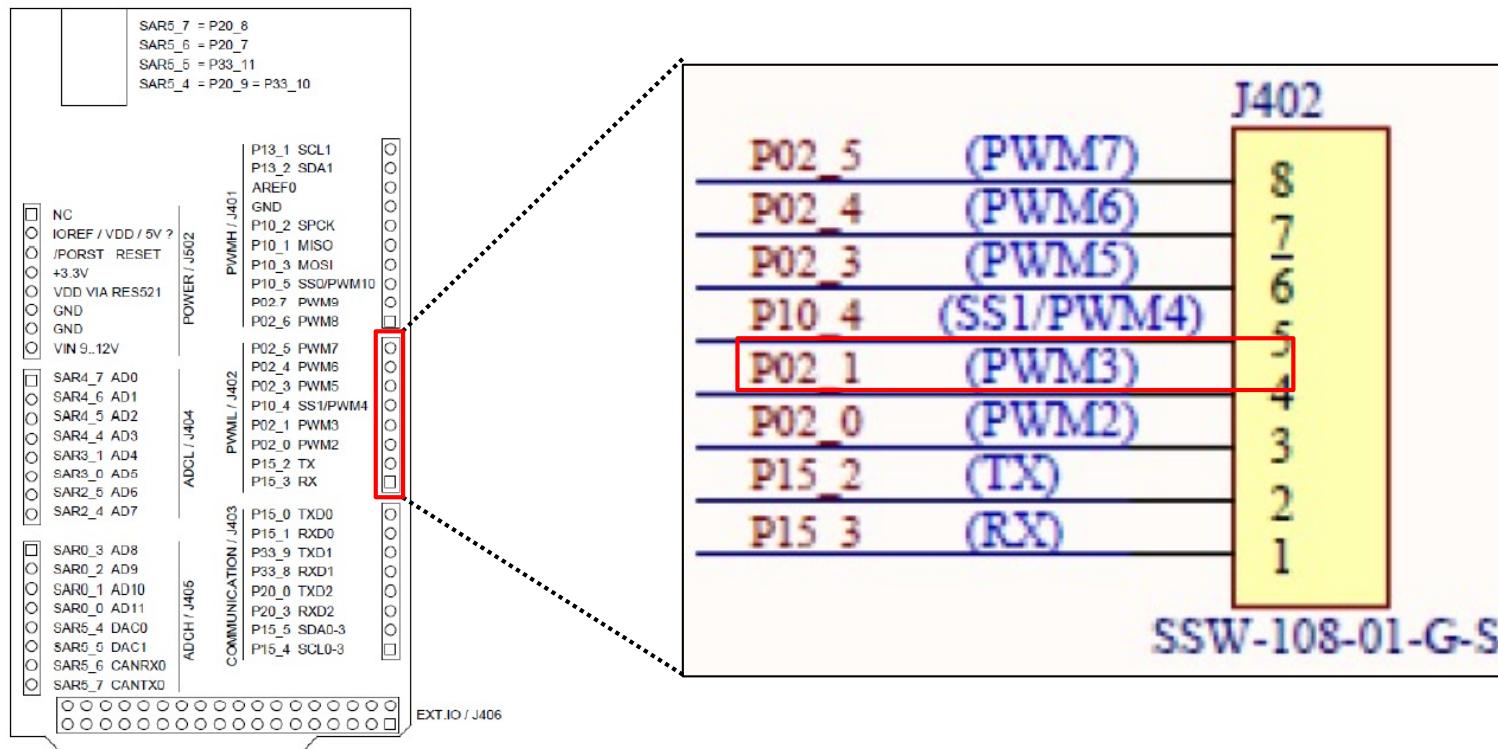
- ✓ Switch는 Easy Module Shield V1 확장 보드의 Pin D2/D3과 연결되어 있다.
- ✓ Switch가 눌리면 연결된 Pin은 Low-level이 되고, Switch가 눌리지 않으면 연결된 Pin은 High-level이 된다.
- ✓ 타겟 보드는 Easy Module Shield V1 확장 보드의 Pin D2/D3을 통해 Switch 입력을 받을 수 있다.  
(정상적인 Switch 동작을 위해 VCC 및 GND도 연결해야 한다.)



# External Interrupt Example

## 1. Switch 연결 정보 파악

- ✓ TC275 보드의 Schematic과 Datasheet를 확인했을 때, Easy Module Shield V1 확장 보드의 Pin D3와 연결되는 IO는 PORT02의 Pin 1이다.



# External Interrupt Example

## 2. Data sheet 분석 : IO 설정

- ✓ Switch에 의한 External Interrupt를 사용하기 위해 연결된 Pin의 IO 설정이 필요하다.
- ✓ Switch가 연결된 PORT02 Pin 1은 External Interrupt를 관리하는 **SCU (System Control Unit)** 내 **ERU (External Request Unit)**의 **REQ14**와 연결되어 있다.
- ✓ 따라서, PORT02 Pin 1을 **Input**으로 설정하여 Switch 신호를 **ERU의 입력**으로 설정해야 한다.

2	P02.1	I	LP / PU1 / VEXT	General-purpose input
	TIN1			GTM input
	REQ14			SCU input
	ARX2B			ASCLIN2 input
	RXDCAN0A			CAN node 0 input
	RXDA2			ERAY input
	CIFD1			CIF input
	P02.1	O0		General-purpose output
	TOUT1	O1		GTM output
	-	O2		Reserved
	SLSO32	O3		QSPI3 output
	DSCGPWMP	O4		DSADC output
	-	O5		Reserved
	-	O6		Reserved
	COUT60	O7		CCU60 output



# External Interrupt Example

## 2. Data sheet 분석 : PORT 설정 (1)

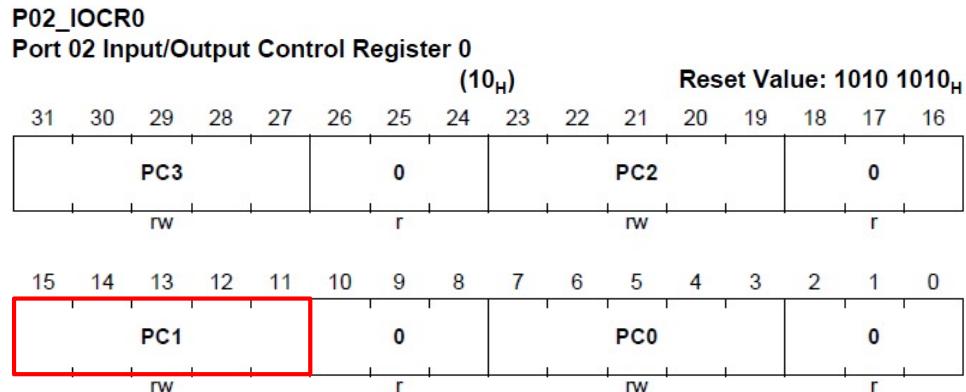
- ✓ P02\_IOC0 Register는 PORT02의 Input/Output을 설정한다.
- ✓ Switch가 PORT02의 Pin 1에 연결되어 있기 때문에 **P02\_IOC0 Register의 PC1 bits**를 설정한다.

Table 13-3 Registers Address Space

Module	Base Address	End Address	Note
P00	F003 A000 <sub>H</sub>	F003 A0FF <sub>H</sub>	13 pins
P01	F003 A100 <sub>H</sub>	F003 A1FF <sub>H</sub>	5 pins
<b>P02</b>	<b>F003 A200<sub>H</sub></b>	<b>F003 A2FF<sub>H</sub></b>	<b>12 pins</b>
P10	F003 B000 <sub>H</sub>	F003 B0FF <sub>H</sub>	9 pins
P11	F003 B100 <sub>H</sub>	F003 B1FF <sub>H</sub>	16 pins
P12	F003 B200 <sub>H</sub>	F003 B2FF <sub>H</sub>	2 pins
P13	F003 B300 <sub>H</sub>	F003 B3FF <sub>H</sub>	4 pins
P14	F003 B400 <sub>H</sub>	F003 B4FF <sub>H</sub>	11 pins
P15	F003 B500 <sub>H</sub>	F003 B5FF <sub>H</sub>	9 pins

P02\_IOC0 Register 주소: F003\_A210h (F003A200h + 10h)

P02\_IOC0 Register 구조:



Field	Bits	Type	Description
PC0, PC1, PC2, PC3	[7:3], [15:11], [23:19], [31:27]	rw	Port Control for Port n Pin 0 to 3 This bit field determines the Port n line x functionality (x = 0-3) according to the coding table (see <a href="#">Table 13-5</a> ).
0	[2:0], [10:8], [18:16], [26:24]	r	<b>Reserved</b> Read as 0; should be written with 0.

# External Interrupt Example

## 2. Data sheet 분석 : PORT 설정 (2)

- ✓ Easy Module Shield V1의 Switch는 pull-up device이다.
- ✓ 따라서, PORT02의 Pin 1을 Input으로 설정할 때 **PC1 bits**를 **0XX10b**로 설정한다.

Table 13-5 PCx Coding

PCx[4:0]	I/O	Characteristics	Selected Pull-up / Pull-down / Selected Output Function
0XX00 <sub>B</sub>	Input	–	No input pull device connected, tri-state mode
0XX01 <sub>B</sub>			Input pull-down device connected
0XX10 <sub>B</sub>			Input pull-up device connected <sup>1)</sup>
0XX11 <sub>B</sub>			No input pull device connected, tri-state mode



# External Interrupt Example

## 2. Data sheet 분석 : ERU External Input Channel 설정 (1)

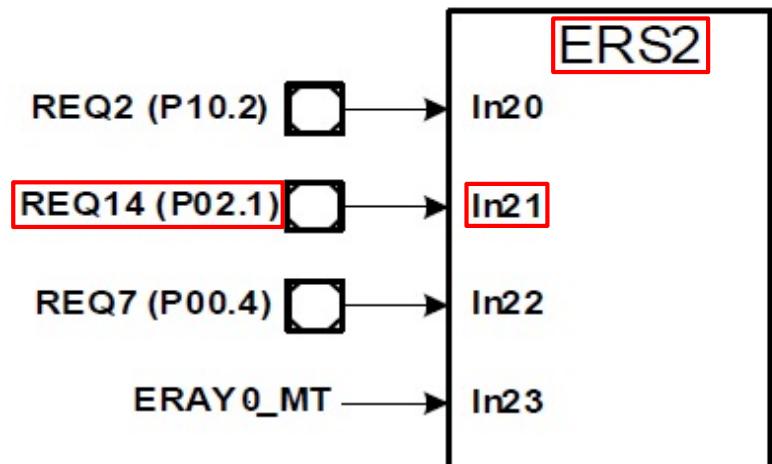
- ✓ SCU\_EICR Register는 ERU의 External input Channel 0-7에 대한 설정을 한다.
- ✓ 하나의 SCU\_EICR Register는 2개의 Channel에 대한 설정을 한다.  
(SCU\_EICR0: Channel 0-1, SCU\_EICR1: Channel 2-3, SCU\_EICR2: Channel 4-5, ...)
- ✓ PORT02 Pin 1과 연결된 REQ14가 ERU의 Channel 2 Input 1과 연결되어 있기 때문에 **SCU\_EICR1 Register**의 **INPO bits / EIENO bit / FENO bit / EXISO bits**를 설정한다.

SCU\_EICR1 Register 주소: F003\_6214h (F0036000h + 214h)

SCU\_EICR1 Register 구조:

Table 7-27 Registers Address Spaces - SCU Kernel Registers

Module	Base Address	End Address	Note
SCU	F003 6000 <sub>H</sub>	F003 63FF <sub>H</sub>	-
<b>EICR1</b> External Input Channel Register 1 (214 <sub>H</sub> )			
Reset Value: 0000 0000 <sub>H</sub>			
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16			
0 INP1 EI EN1 LD EN1 R EN1 F EN1 0 EXIS1 0	r rw rw rw rw r rw rw r		
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
0 INP0 EI EN0 LD EN0 R EN0 F EN0 0 EXIS0 0	r rw rw rw rw r rw rw r		



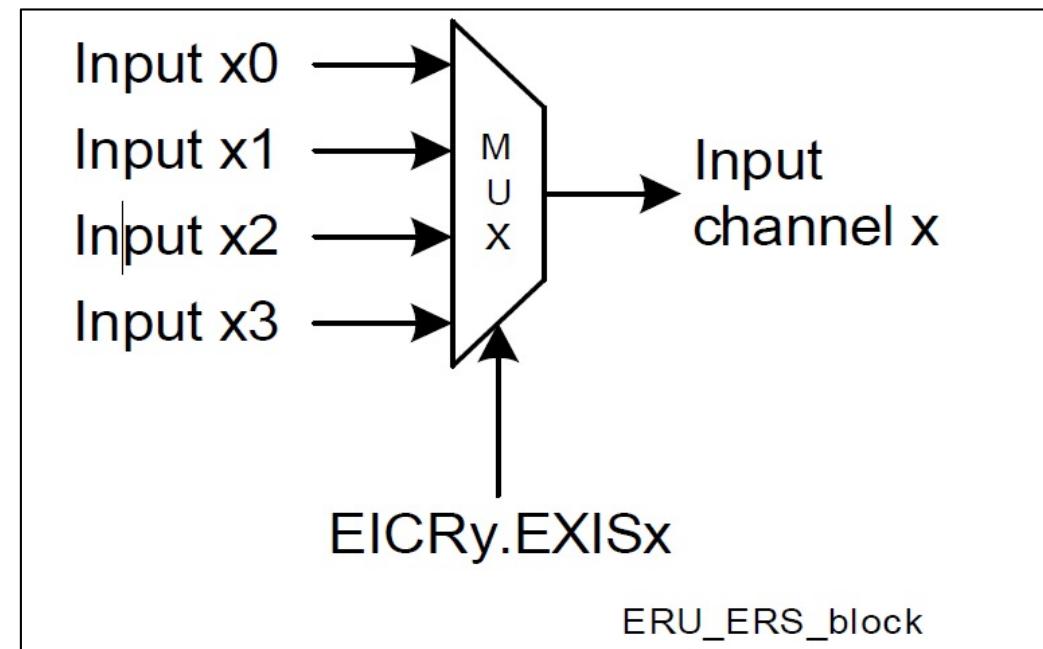
# External Interrupt Example

## 2. Data sheet 분석 : ERU External Input Channel 설정 (2)

- ✓ ERU의 각 External Input Channel은 여러 개의 External Request Input을 입력 받기 때문에 이들 중 하나의 입력을 결정해야 한다.
- ✓ PORT02 Pin 10| ERU의 Channel 2 Input 1과 연결되어 있기 때문에 **EXIS0 bits**를 **001b**로 설정한다.

EICR1 External Input Channel Register 1 (214H)																Reset Value: 0000 0000H			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
0		INP1		EI EN1	LD EN1	R EN1	F EN1	0		EXIS1			0			r			r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0		INP0		EI EN0	LD EN0	R EN0	F EN0	0		EXIS0			0			r			r

Field	Bits	Type	Description
EXIS0	[6:4]	rw	<b>External Input Selection 0</b> This bit field determines which input line is selected for Input Channel (2i). 000 <sub>B</sub> Input (2i) 0 is selected 001 <sub>B</sub> Input (2i) 1 is selected 010 <sub>B</sub> Input (2i) 2 is selected 011 <sub>B</sub> Input (2i) 3 is selected 100 <sub>B</sub> Reserved 101 <sub>B</sub> Reserved 110 <sub>B</sub> Reserved 111 <sub>B</sub> Reserved



# External Interrupt Example

## 2. Data sheet 분석 : ERU External Input Channel 설정 (3)

- ✓ Switch가 pull-up device이기 때문에 Switch가 눌렸을 때, 신호는 High-level에서 Low-level로 바뀌며 Falling edge가 발생한다.
- ✓ 따라서, Falling edge가 검출되었을 때 트리거 신호 (for External Interrupt)를 생성하기 위해 **FENO bit**를 1로 설정한다.
- ✓ 생성된 트리거 신호를 Enable 하기 위해 **EIENO bit**를 1로 설정한다.

EICR1 External Input Channel Register 1 (214H)												Reset Value: 0000 0000H								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
0			INP1		EI EN1	LD EN1	R EN1	F EN1	0		EXIS1		0			r	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0			INP0		EI EN0	LD EN0	R EN0	F EN0	0		EXIS0		0			r	rw	rw	rw	r

Field	Bits	Type	Description
<b>FENO</b>	8	rw	<b>Falling Edge Enable 0</b> This bit determines if the falling edge of Input Channel (2i) is used to set bit INTF(2i). 0 <sub>B</sub> The falling edge is not used 1 <sub>B</sub> The detection of a falling edge of Input Channel 0 generates a trigger event. INTF(2i) becomes set.
<b>REN0</b>	9	rw	<b>Rising Edge Enable 0</b> This bit determines if the rising edge of Input Channel (2*i) is used to set bit INTF(2i). 0 <sub>B</sub> The rising edge is not used 1 <sub>B</sub> The detection of a rising edge of Input Channel (2*i) generates a trigger event. INTF(2*i) becomes set
<b>LDEN0</b>	10	rw	<b>Level Detection Enable 0</b> This bit determines if bit INTF(2i) is cleared automatically if an edge of the input Input Channel (2i) is detected, which has not been selected (rising edge with REN0 = 0 or falling edge with FENO = 0). 0 <sub>B</sub> Bit INTF(2i) will not be cleared 1 <sub>B</sub> Bit INTF(2i) will be cleared
<b>EIENO</b>	11	rw	<b>External Input Enable 0</b> This bit enables the generation of a trigger event for request channel (2i) (e.g. for interrupt generation) when a selected edge is detected. 0 <sub>B</sub> The trigger event is disabled 1 <sub>B</sub> The trigger event is enabled

# External Interrupt Example

## 2. Data sheet 분석 : ERU External Input Channel 설정 (4)

- ✓ ERU의 각 External Input Channel에서 생성된 트리거 신호는 Connecting Matrix를 통해 Output Channel에 전달된다.
- ✓ Output Channel은 입력 받은 트리거 신호를 Interrupt 신호로 전달할 수 있다.
- ✓ 생성된 트리거 신호를 Output Channel 0에 전달하기 위해 **INP0 bits**를 **000b**로 설정한다.

**EICR1**  
**External Input Channel Register 1 (214H)**  
 Reset Value: 0000 0000H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		INP1		EI EN1	LD EN1	R EN1	F EN1	0		EXIS1		0			
r		rw		rw	rw	rw	rw	r		rw		r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		INP0		EI EN0	LD EN0	R EN0	F EN0	0		EXIS0		0			
r		rw		rw	rw	rw	rw	r		rw		r			

Field	Bits	Type	Description
INP0	[14:12]	rw	<b>Input Node Pointer</b> This bit field determines the destination (output channel) for trigger event (2i) (if enabled by EIEN(2i)). 000 <sub>B</sub> An event from input ETL 2i triggers output OGU0 (signal TR(2i) 0) 001 <sub>B</sub> An event from input ETL 2i triggers output OGU1 (signal TR(2i) 1) 010 <sub>B</sub> An event from input ETL 2i triggers output OGU2 (signal TR(2i) 2)

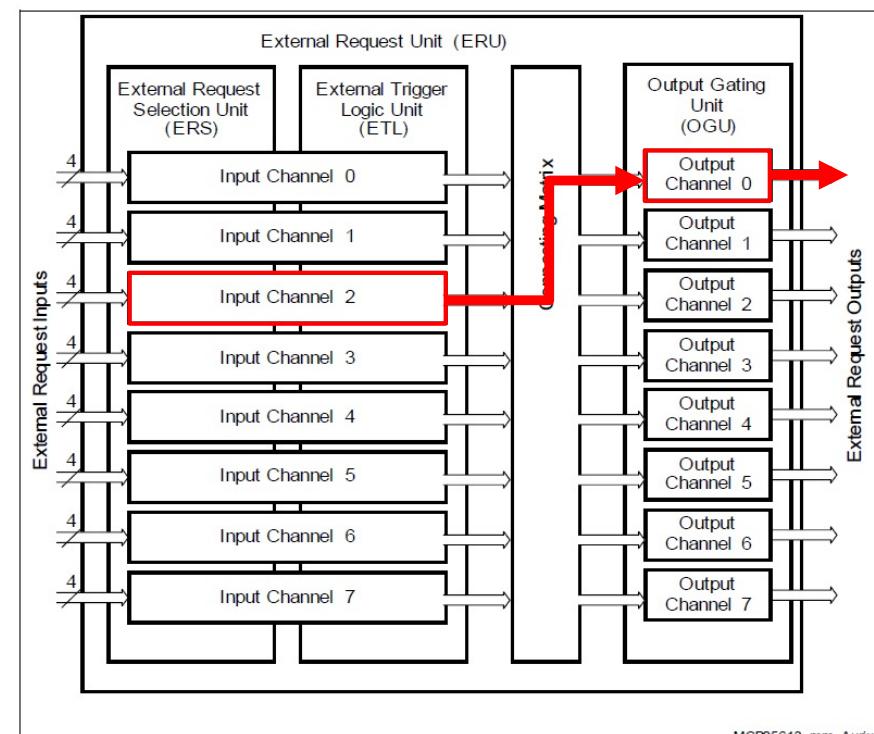


Figure 7-40 External Request Unit Overview

# External Interrupt Example

## 2. Data sheet 분석 : ERU Flag Gating 설정 (1)

- ✓ SCU\_IGCR Register는 ERU의 Output Channel 0-7에 대한 설정을 한다.
- ✓ 하나의 SCU\_IGCR Register는 2개의 Channel에 대한 설정을 한다.  
(SCU\_IGCRO: Channel 0-1, SCU\_IGCR1: Channel 2-3, SCU\_IGCR2: Channel 4-5, ...)
- ✓ External Input Channel 2에서 생성된 트리거 신호가 Output Channel 0에 전달되기 때문에 **SCU\_IGCRO Register**의 **IGP0 bits**를 설정한다.

SCU\_IGCRO Register 주소: F003\_622Ch (F0036000h + 22Ch)

SCU\_IGCRO Register 구조:

Table 7-27 Registers Address Spaces - SCU Kernel Registers

Module	Base Address	End Address	Note
SCU	F003 6000 <sub>H</sub>	F003 63FF <sub>H</sub>	-

IGCRO Flag Gating Register 0 (22C <sub>H</sub> ) Reset Value: 0000 0000 <sub>H</sub>															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IGP1	GE EN1			0			IPEN 17	IPEN 16	IPEN 15	IPEN 14	IPEN 13	IPEN 12	IPEN 11	IPEN 10	
r	w	r	w	r			r	w	r	r	w	r	w	r	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IGP0	GE EN0			0			IPEN 07	IPEN 06	IPEN 05	IPEN 04	IPEN 03	IPEN 02	IPEN 01	IPEN 00	
r	w	r	w	r			r	w	r	r	w	r	w	r	w

# External Interrupt Example

## 2. Data sheet 분석 : ERU Flag Gating 설정 (2)

- ✓ 트리거 신호를 IOUT (for External Interrupt)으로 출력하기 위해 **IGP0 bits**를 **01b**로 설정한다.

Field	Bits	Type	Description
IGP0	[15:14]	rw	<b>Interrupt Gating Pattern 0</b> In each register IGCRj, bit field IGP0 determines how the pattern detection influences the output lines GOUT(2j) and IOUT(2j). $00_B$ IOUT(2j) is inactive. The pattern is not considered. $01_B$ IOUT(2j) is activated in response to a trigger event. The pattern is not considered. $10_B$ The detected pattern is considered. IOUT(2j) is activated if a trigger event occurs while the pattern is present. $11_B$ The detected pattern is considered. IOUT(2j) is activated if a trigger event occurs while the pattern is not present.



# External Interrupt Example

## 2. Data sheet 분석 : Safety Critical Register 설정 (1)

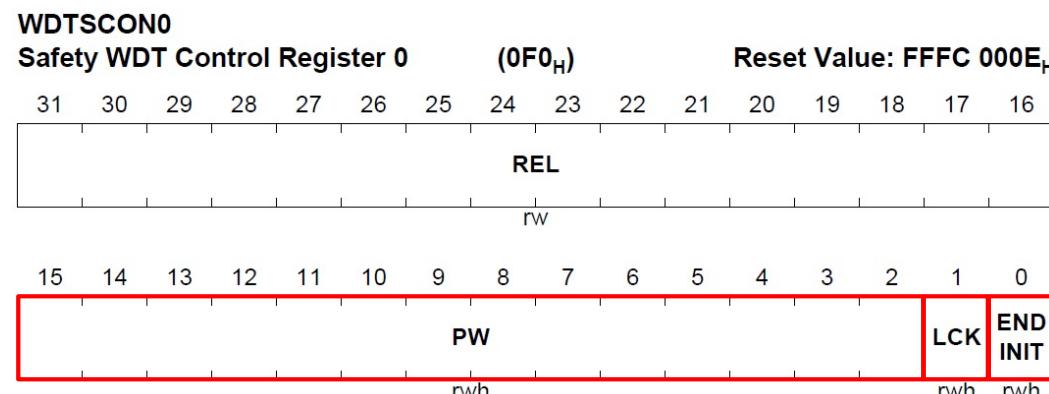
- ✓ 설정해야 하는 SCU\_EICR1 / SCU\_IGCRO Register는 Safety Critical Register이기 때문에 Write Protected (Safety ENDINIT, End-of-Initialization) 되어 있다.
- ✓ 해당 Register를 수정하기 위해서는 Safety ENDINIT을 해제해야 한다.
- ✓ SCU\_WDTSCONO Register는 **Safety Critical Register**에 대한 **Safety ENDINIT**을 설정/해제한다.

SCU\_WDTSCONO Register 주소: F003\_60F0h  
(F0036000h + 0F0h)

SCU\_WDTSCONO Register 구조:

Table 7-27 Registers Address Spaces - SCU Kernel Registers

Module	Base Address	End Address	Note
SCU	F003 6000 <sub>H</sub>	F003 63FF <sub>H</sub>	-



# External Interrupt Example

## 2. Data sheet 분석 : Safety Critical Register 설정 (2)

- ✓ **ENDINIT bit**는 Safety ENDINIT의 설정 상태를 나타내며 Modify Access를 통해서만 수정이 가능하다.
- ✓ **LCK bit**는 SCU\_WDTSCONO Register의 Lock 상태를 나타내며 해당 Register의 Lock 상태는 Password Access를 통해 Unlock 되고, Modify Access를 통해 Lock 된다.
- ✓ **PW bits**는 SCU\_WDTSCONO Register에 접근하기 위한 Password를 저장하며 해당 값을 읽으면 bits[7:2]가 반전되어 읽힌다.

Field	Bits	Type	Description	PW	[15:2]	rwh	User-Definable Password Field for Access to WDTxCON0
<b>ENDINIT</b>	0	rwh	<b>End-of-Initialization Control Bit</b> 0 <sub>B</sub> Access to Endinit-protected registers is permitted. 1 <sub>B</sub> Access to Endinit-protected registers is not permitted. This bit must be written with a '1' during a Password Access or Check Access (although this write is only used for the password-protection mechanism and is not stored). This bit must be written with the required ENDINIT update value during a Modify Access.				This bit field is written with an initial password value during a Modify Access. A read from this bitfield returns this initial password, but bits [7:2] are inverted (toggled) to ensure that a simple read/write is not sufficient to service the WDT.  If corresponding WDTxSR.PAS = 0 then this bit field must be written with its current contents during a Password Access or Check Access. If corresponding WDTxSR.PAS = 1 then this bit field must be written with the next password in the LFSR sequence during a Password Access or Check Access  The default password after Application Reset is 00000000111100 <sub>B</sub>
<b>LCK</b>	1	rwh	<b>Lock Bit to Control Access to WDTxCON0</b> 0 <sub>B</sub> Register WDTxCON0 is unlocked 1 <sub>B</sub> Register WDTxCON0 is locked (default after ApplicationReset) The current value of LCK is controlled by hardware. It is cleared after a valid Password Access to WDTxCON0 when WDTxSR.US is 0 (or when WDTxSR.US is 1 and the SMU is in RUN mode), and it is automatically set again after a valid Modify Access to WDTxCON0. During a write to WDTxCON0, the value written to this bit is only used for the password-protection mechanism and is not stored. This bit must be cleared during a Password Access to WDTxCON0, and set during a Modify Access to WDTxCON0. A Check Access does not clear LCK.				A-step silicon: Bits [7:2] must be written with 111100 <sub>B</sub> during Password Access and Modify Access. Read returns 000011 <sub>B</sub> for these bits.



A

# External Interrupt Example

## 2. Data sheet 분석 : Safety Critical Register 설정 (3)

- ✓ SCU\_WDTSCONO Register에 적절한 값을 Write하여 **Password Access**를 수행한다.
- ✓ **Password Access**는 **SCU\_WDTSCONO Register**의 **Lock** 상태를 해제하며 과정은 다음과 같다.
  1. SCU\_WDTSCONO Register의 값을 읽어 REL bits, PW bits를 파악한다.
  2. Bits[7:2] (PW bits의 일부)가 반전되어 읽히기 때문에 이를 반전시켜 정확한 PW bits를 얻는다.
  3. Write 할 값의 bits[31:16]은 읽혀진 REL bits 값으로 설정하고 bit[15:2]는 앞서 구한 정확한 PW bits 값으로 설정한다.
  4. Write 할 값의 bit[1]은 0으로 설정하고, bit[0]은 1로 설정한다.
  5. 설정된 값을 SCU\_WDTSCONO Register에 한번에 쓴다.
  6. SCU\_WDTSCONO Register의 LCK bit를 확인하여 Lock 상태가 해제되었는지 파악한다.  
(Password Access가 정상적으로 수행되면 Lock 상태가 해제되며 LCK bit가 0으로 설정된다.)
- ✓ Password Access를 통해 SCU\_WDTSCONO Register의 Lock 상태가 해제되면 Modify Access를 통해 Safety ENDINIT을 설정/해제할 수 있다.

# External Interrupt Example

## 2. Data sheet 분석 : Safety Critical Register 설정 (4)

- ✓ SCU\_WDTSCONO Register에 적절한 값을 Write하여 **Modify Access**를 수행한다.
- ✓ **Modify Access**는 **Safety ENDINIT**을 설정/해제하며 과정은 다음과 같다.
  1. SCU\_WDTSCONO Register의 값을 읽어 REL bits, PW bits를 파악한다.
  2. Bits[7:2] (PW bits의 일부)가 반전되어 읽히기 때문에 이를 반전시켜 정확한 PW bits를 얻는다.
  3. Write 할 값의 bits[31:16]은 읽혀진 REL bits 값으로 설정하고 bit[15:2]는 앞서 구한 정확한 PW bits 값으로 설정한다.
  4. Write 할 값의 bit[1]은 1로 설정하고, bit[0]은 적절한 값으로 설정한다.  
(Safety ENDINIT 설정: bit[0] = 1, Safety ENDINIT 해제 : bit[0] = 0)
  5. 설정된 값을 SCU\_WDTSCONO Register에 한번에 쓴다.
  6. SCU\_WDTSCONO Register의 LCK bit를 확인하여 Lock 상태가 다시 설정되었는지 파악한다.  
(Modify Access가 정상적으로 수행되면 Lock 상태가 설정되며 LCK bit가 1로 설정된다.)
- ✓ Modify Access를 통해 Safety ENDINIT을 해제하면 Safety Critical Register를 수정할 수 있으며 수정을 완료하면 Safety ENDINIT을 꼭 다시 설정해야 한다.



# External Interrupt Example

## 2. Data sheet 분석 : Interrupt Router 설정 (1)

- ✓ Interrupt Router는 Interrupt Trigger를 Service Providers (CPU 0-2, DMA)에 연결한다.
- ✓ Switch 신호에 따라 ERU에서 생성된 트리거 신호는 **Output Channel 0의 IOUT**으로 출력된다.
- ✓ 해당 출력은 Interrupt Router의 **SCUERU0 SRN (Service Request Node)**와 연결된다.
- ✓ 해당 노드에 대한 설정을 하기 위해 **SRC\_SCUERU0 Register**를 설정해야 한다.

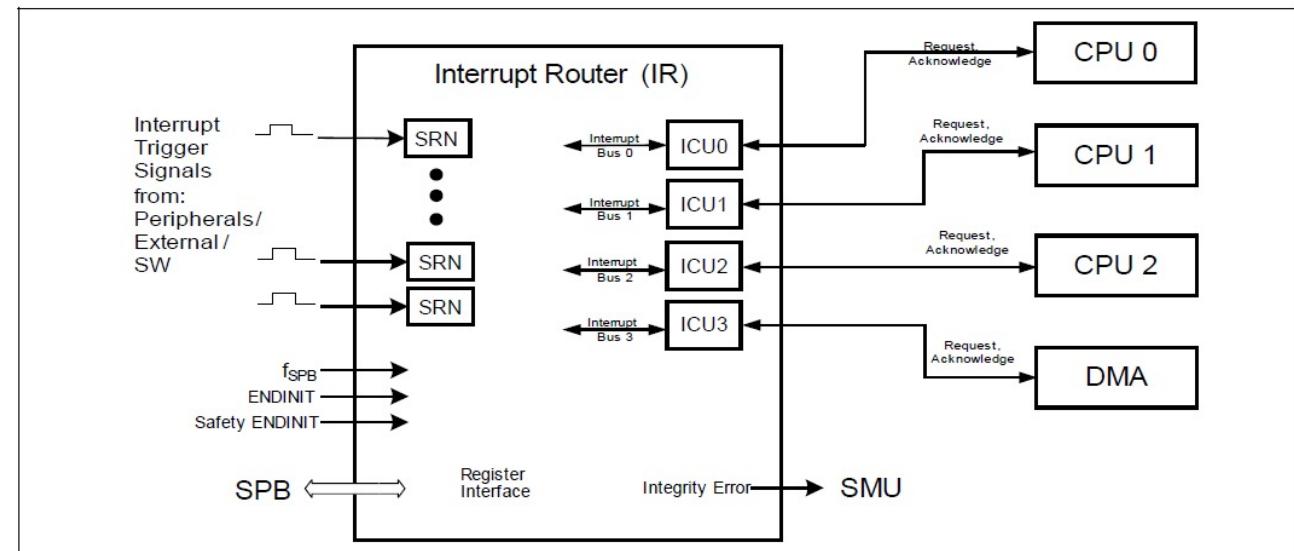
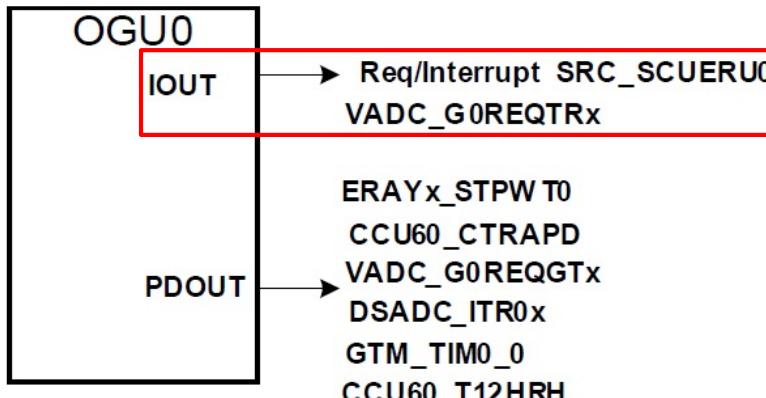


Figure 16-1 Block Diagram of the TC27x Interrupt System

# External Interrupt Example

## 2. Data sheet 분석 : Interrupt Router 설정 (2)

- ✓ SRC\_SCUERU0 Register는 SCUERU0 SRN에 대한 Interrupt 설정을 한다.
- ✓ 해당 Interrupt의 우선순위를 설정하기 위해 **SRPN bits**를 **Ah** (임의의 값)로 설정한다.  
(우선순위는 해당 Interrupt가 할당된 Service Provider에서 Interrupt Vector Table의 Index가 된다.)
- ✓ 해당 Interrupt가 CPU0에서 처리되도록 하기 위해 **TOS bits**를 **0h**로 설정한다.
- ✓ 해당 Interrupt를 Enable 하기 위해 **SRE bit**를 **1**로 설정한다.

SRC\_SCUERU0 Register 주소: F003\_8CD4h  
(F0038000h + CD4h)

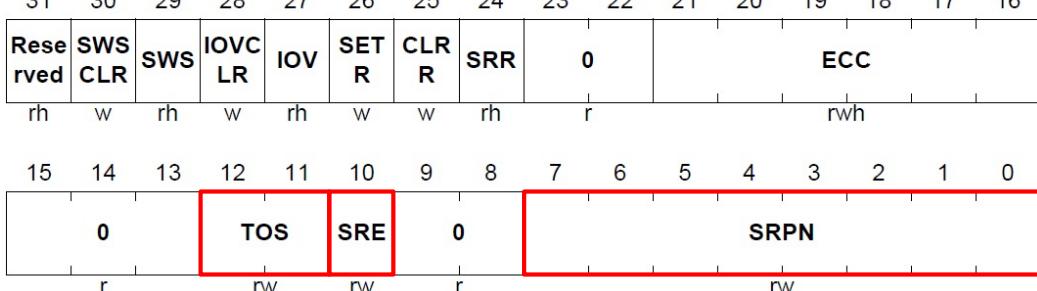
SRC\_SCUERU0 Register 구조:

Table 16-3 Registers Address Space - Service Request Control Registers (SRC)

Module	Base Address	End Address	Note
SRC	F003 8000 <sub>H</sub>	F003 9FFF <sub>H</sub>	

SRC\_SCUERUm (m=0-3)  
SCU ERU Service Request m (0CD4<sub>H</sub>+m\*4<sub>H</sub>) Reset Value: 0000 0000<sub>H</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rese rv ed	SWS CLR	SWS	IOVC LR	IOV	SET R	CLR R	SRR	0							ECC
rh	w	rh	w	rh	w	w	rh	r							rwh



Field	Bits	Type	Description
SRPN	[7:0]	rw	<b>Service Request Priority Number</b> 00 <sub>H</sub> Service request is on lowest priority 01 <sub>H</sub> Service request is one before lowest priority ... FF <sub>H</sub> Service request is on highest priority <small>Note: For a CPU 01H is the lowest priority as 00H is never serviced. For the DMA 00H triggers channel 0</small>
SRE	10	rw	<b>Service Request Enable</b> 0 <sub>B</sub> Service request is disabled 1 <sub>B</sub> Service request is enabled
TOS	[12:11]	rw	<b>Type of Service Control</b> 0 <sub>H</sub> CPU0 service is initiated 1 <sub>H</sub> CPU1 service is initiated 2 <sub>H</sub> CPU2 service is initiated 3 <sub>H</sub> DMA service is initiated

# External Interrupt Example

## 3. 프로그래밍

1) Switch 및 LED가 연결된 PORT에 대한 설정을 수행하는 함수를 구현한다.

- ✓ 자세한 내용은 이전 강의자료 (GPIO)를 참고한다.

```
31 #define PORT10_BASE      (0xF003B000)
32 #define PORT10_IOCR0     (*(volatile unsigned int*)(PORT10_BASE + 0x10))
33 #define PORT10_OMR       (*(volatile unsigned int*)(PORT10_BASE + 0x04))
34
35 #define PC1              11
36 #define PC2              19
37
38 #define PCL1             17
39 #define PCL2             18
40 #define PS1              1
41 #define PS2              2
```

PORT13 IO (LED RED) 설정관련 레지스터 주소 및 비트 필드 정의

```
45 /* Initialize LED (RED & BLUE) */
46 void init_LED(void)
47 {
48     /* Reset PC1 & PC2 in IOCR0*/
49     PORT10_IOCR0 &= ~((0x1F) << PC1);
50     PORT10_IOCR0 &= ~((0x1F) << PC2);
51
52     /* Set PC1 & PC2 with push-pull(2b10000) */
53     PORT10_IOCR0 |= ((0x10) << PC1);
54     PORT10_IOCR0 |= ((0x10) << PC2);
55 }
```

PORT13 IO (LED RED) 설정 초기화 코드



# External Interrupt Example

## 3. 프로그래밍

1) Switch 및 LED가 연결된 PORT에 대한 설정을 수행하는 함수를 구현한다.

- ✓ 자세한 내용은 이전 강의자료 (GPIO)를 참고한다.

```
40 /* Define PORT02 Registers for Switch2 */
41 #define PORT02_BASE      (0xF003A200)
42 #define PORT02_IOCR0     (*(volatile unsigned int*)(PORT02_BASE + 0x10))
43 #define PORT02_IN       (*(volatile unsigned int*)(PORT02_BASE + 0x24))
44
45 #define PC1             11
46 #define P1              1
```

PORT02 IO (Switch2) 설정관련 레지스터 주소 및 비트 필드 정의

```
60 /* Initialize Switch2 */
61 void init_Switch(void)
62 {
63     /* Reset PC1 in IOCR0*/
64     PORT02_IOCR0 &= ~((0x1F) << PC1);
65
66     /* Set PC1 with push-pull(2b0xx10) */
67     PORT02_IOCR0 |= ((0x2) << PC1);
68 }
```

PORT02 IO (Switch2) 설정 코드



# External Interrupt Example

## 3. 프로그래밍

2) ERU를 설정하기 위한 함수를 구현한다.

- ① SCU\_WDTSCONO Register를 통해 Password/Modify Access를 수행하여 Safety ENDINIT을 해제한다.
- ② SCU\_EICR1 Register를 통해 ERU의 Channel 2의 입력으로 Input 1을 설정한다.
- ③ SCU\_EICR1 Register를 통해 Falling edge가 트리거 신호를 생성하도록 설정하고 이를 Enable 한다.
- ④ SCU\_EICR1 Register를 통해 생성된 트리거 신호가 Output Channel 0에 전달되도록 한다.
- ⑤ SCU\_IGCRO Register를 통해 전달된 트리거 신호가 IOUT으로 출력되도록 설정한다.
- ⑥ SCU\_WDTSCONO Register를 통해 Password/Modify Access를 수행하여 Safety ENDINIT을 설정한다.
- ⑦ SRC\_SCUERU0 Register를 통해 SCUERU0 SRN의 우선순위를 설정한다.
- ⑧ SRC\_SCUERU0 Register를 통해 SCUERU0 SRN의 처리가 CPU0에서 수행되도록 설정한다.
- ⑨ SRC\_SCUERU0 Register를 통해 SCUERU0 SRN의 Interrupt를 Enable 한다.



# External Interrupt Example

## 3. 프로그래밍

- 2) ERU를 설정하기 위한 함수를 구현한다.

```
// SCU Registers
#define SCU_BASE          (0xF0036000)
#define SCU_WDTSCONO     (*(volatile unsigned int*) (SCU_BASE + 0x0F0))
#define SCU_EICR1         (*(volatile unsigned int*) (SCU_BASE + 0x214))
#define SCU_IGCR0         (*(volatile unsigned int*) (SCU_BASE + 0x22C))

#define LCK              1
#define ENDINIT          0
#define INPO             12
#define EIENO            11
#define FENO             8
#define EXISO            4
#define IGPO             14

// SRC Registers
#define SRC_BASE          (0xF0038000)
#define SRC_SCUERUO       (*(volatile unsigned int*) (SRC_BASE + 0xCD4))

#define TOS              11
#define SRE              10
#define SRPN             0
```

ERU 설정관련 레지스터 주소 및 비트 필드 정의

# External Interrupt Example

## 3. 프로그래밍

2) ERU를 설정하기 위한 함수를 구현한다.

```
93 /* Initialize External Request Unit (ERU) */
94 void init_ERU(void)
95 {
96     /* ERU Input Channel 2 Setting */
97     /* Password Access to unlock WDTSCON0 */
98     SCU_WDTSCON0 = ((SCU_WDTSCON0 ^ 0xFC) & ~(1 << LCK)) | (1 << ENDINIT);
99     while((SCU_WDTSCON0 & (1 << LCK)) != 0);
100
101    // Modify Access to clear ENDINIT bit
102    SCU_WDTSCON0 = ((SCU_WDTSCON0 ^ 0xFC) | (1 << LCK)) & ~ (1 << ENDINIT);
103    while((SCU_WDTSCON0 & (1 << LCK)) == 0);
104
105    (2) SCU_EICR1 &= ~((0x7) << EXIS0);           // External input 1 is selected
106    SCU_EICR1 |= ((0x1) << EXIS0);
107
108    (3) SCU_EICR1 |= (1 << FEN0);                  // Falling edge enable
109
110    SCU_EICR1 |= ((0x1) << EIEN0);                // The trigger event is enabled
111
112    (4) SCU_EICR1 &= ~((0x7) << INP0);            // An event from input ETL 2 triggers output OGU 0
113
114    (5) SCU_IGCR0 &= ~((0x3) << IGP0);          // IOUT(0) is activated in response to a trigger event
115    SCU_IGCR0 |= ((0x1) << IGP0);                // The pattern is not considered
116
117    /* Password Access to unlock WDTSCON0 */
118    (6) SCU_WDTSCON0 = ((SCU_WDTSCON0 ^ 0xFC) & ~(1 << LCK)) | (1 << ENDINIT);
119    while((SCU_WDTSCON0 & (1 << LCK)) != 0);
120
121    /* Modify Access to set ENDINIT bit */
122    SCU_WDTSCON0 = ((SCU_WDTSCON0 ^ 0xFC) | (1 << LCK)) | (1 << ENDINIT);
123    while((SCU_WDTSCON0 & (1 << LCK)) == 0);
124
125    (7) /* SRC Interrupt Setting For ECU */
126    SRC_SCUERU0 &= ~((0xFF) << SRPN);           // Set Priority : 0x0A
127    SRC_SCUERU0 |= ((0x0A) << SRPN);
128
129    (8) SRC_SCUERU0 &= ~((0x3) << TOS);          // CPU0 services
130
131    (9) SRC_SCUERU0 |= (1 << SRE);                // Service Request is enabled
132 }
```



# External Interrupt Example

## 3. 프로그래밍

### 3) ERU를 통한 External Interrupt에 대한 ISR를 구현한다.

- ✓ 해당 함수가 ISR 임을 나타내기 위해 컴파일러 지시자를 앞에 붙인다.
  - `__interrupt( PRIORITY )` : 꽤 안에는 ISR에 대응되는 Interrupt의 우선순위를 입력한다.
  - `__vector_table( CPU_NUM )` : 꽤 안에는 해당 ISR을 수행하는 CPU 번호를 입력한다.
- ✓ ISR이 수행된 후, 해당 Interrupt Flag가 자동으로 Clear 되기 때문에 이를 위한 코드가 필요하지 않다.

```
158 __interrupt(0x0A) __vector_table(0)
159 void ERU0_ISR(void)
160 {
161     PORT10_OMR |= ((1<<PCL1) | (1<<PS1));           // Toggle LED RED
162 }
```

ERU0 Interrupt Service Routine

# External Interrupt Example

## 3. 프로그래밍

### 4) 동작에 따라 'main' 함수를 구현한다.

```
32 /* Define PORT10 Registers for LED */
33 #define PORT10_BASE      (0xF003B000)
34 #define PORT10_IOCR0     (*(volatile unsigned int*)(PORT10_BASE + 0x10))
35 #define PORT10_OMR       (*(volatile unsigned int*)(PORT10_BASE + 0x04))
36
37 #define PC1              11
38 #define PCL1             17
39 #define PS1              1
40
41 /* Define PORT02 Registers for Switch2 */
42 #define PORT02_BASE      (0xF003A200)
43 #define PORT02_IOCR0     (*(volatile unsigned int*)(PORT02_BASE + 0x10))
44 #define PORT02_IN        (*(volatile unsigned int*)(PORT02_BASE + 0x24))
45
46 #define PC1              11
47 #define P1               1
48
49 /* Define SCU Registers for Interrupt */
50 #define SCU_BASE         (0xF0036000)
51 #define SCU_WDTSCON0    (*(volatile unsigned int*)(SCU_BASE + 0x0F0))
52 #define SCU_EICR1        (*(volatile unsigned int*)(SCU_BASE + 0x214))
53 #define SCU_IGCR0        (*(volatile unsigned int*)(SCU_BASE + 0x22C))
54
55 #define LCK              1
56 #define ENDINIT          0
57 #define INP0              12
58 #define EIENO             11
59 #define FEN0              8
60 #define EXISO             4
61 #define IGP0              14
62
63 /* Define SRC Registers for Interrupt */
64 #define SRC_BASE         (0xF0038000)
65 // #define SRC_SCUERU0    (*(volatile unsigned int*)(SRC_BASE + 0xCD4))
66
67 #define TOS              11
68 #define SRE              10
69 #define SRPN             0
70
71 IfxCpu_syncEvent g_cpuSyncEvent = 0;
```

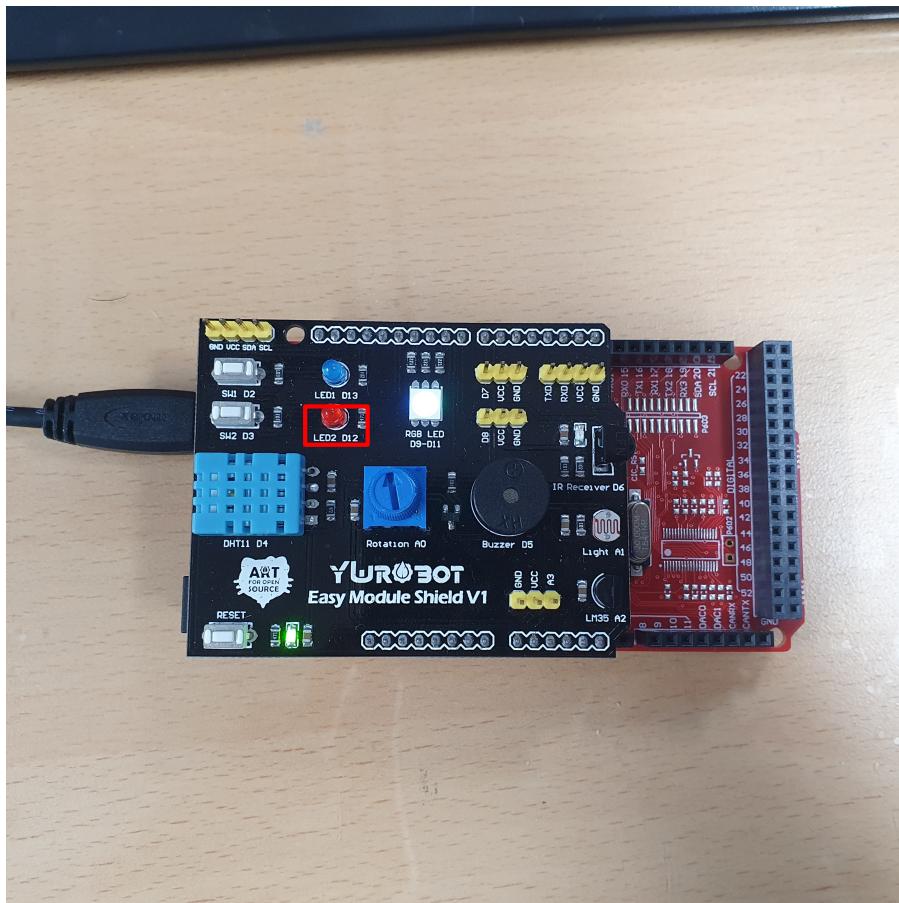
```
134 int core0_main(void)
135 {
136     IfxCpu_enableInterrupts();
137
138     /* !!WATCHDOG AND SAFETY WATCHDOG ARE DISABLED HERE!!
139      * Enable the watchdogs and service them periodically if it is required
140      */
141     IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
142     IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
143
144     /* Wait for CPU sync event */
145     IfxCpu_emitEvent(&g_cpuSyncEvent);
146     IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
147
148     init_ERU();                                // Initialize ERU
149     init_LED();                               // Initialize LED
150     init_Switch();                            // Initialize Switch
151
152     while(1)
153     {
154     }
155     return (1);
156 }
157
158 __interrupt(0x0A) __vector_table(0)
159 void ERU0_ISR(void)
160 {
161     PORT10_OMR |= ((1<<PCL1) | (1<<PS1));           // Toggle LED RED
162 }
163
```

- ✓ '\_\_enable()'을 통해 CPU의 Global Interrupt Enable을 수행한다.
- ✓ 앞서 구현한 함수들을 호출한다.

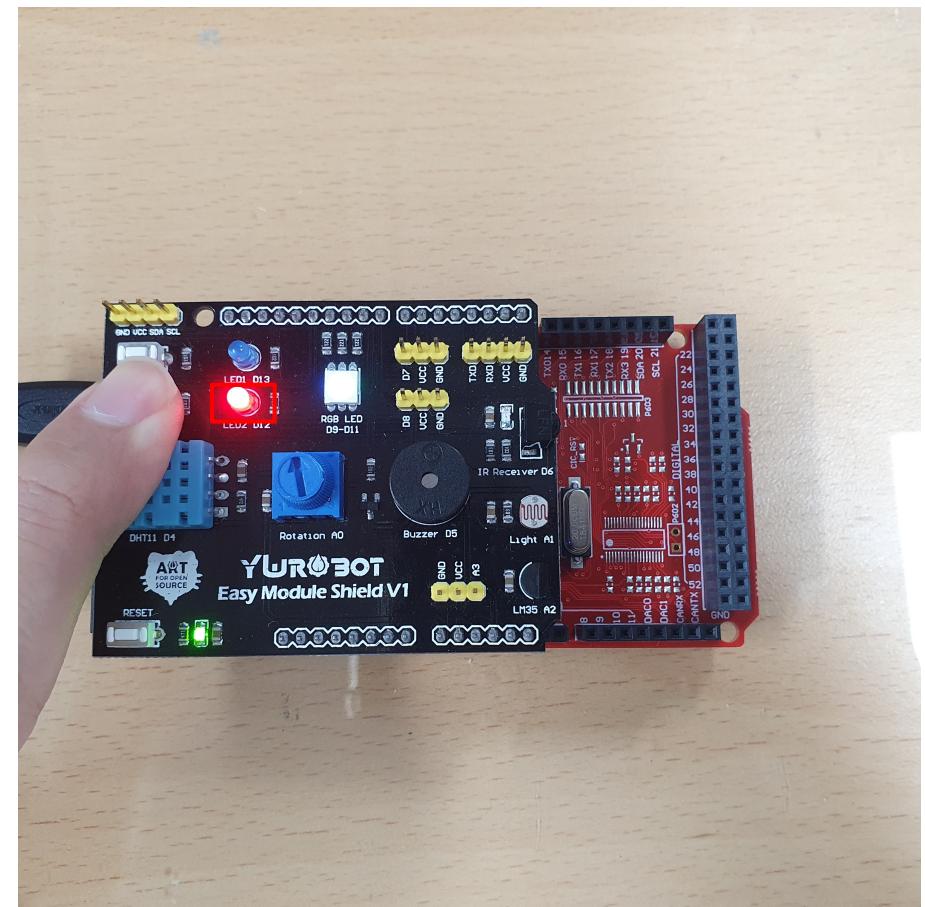
# External Interrupt Example

## 4. 동작 확인

- ✓ Build 및 Debug 후 ('Resume' 버튼 클릭), Switch를 누를 때마다 LED가 Toggle 되는 것을 확인한다.



Switch  
Click!  
↔



# Q & A

**Thank you for your attention**



**School of Electronics Engineering, KNU**

ACE Lab (hn02301@gmail.com)