

Infineon TC275 Timer

Hyeongrae Kim

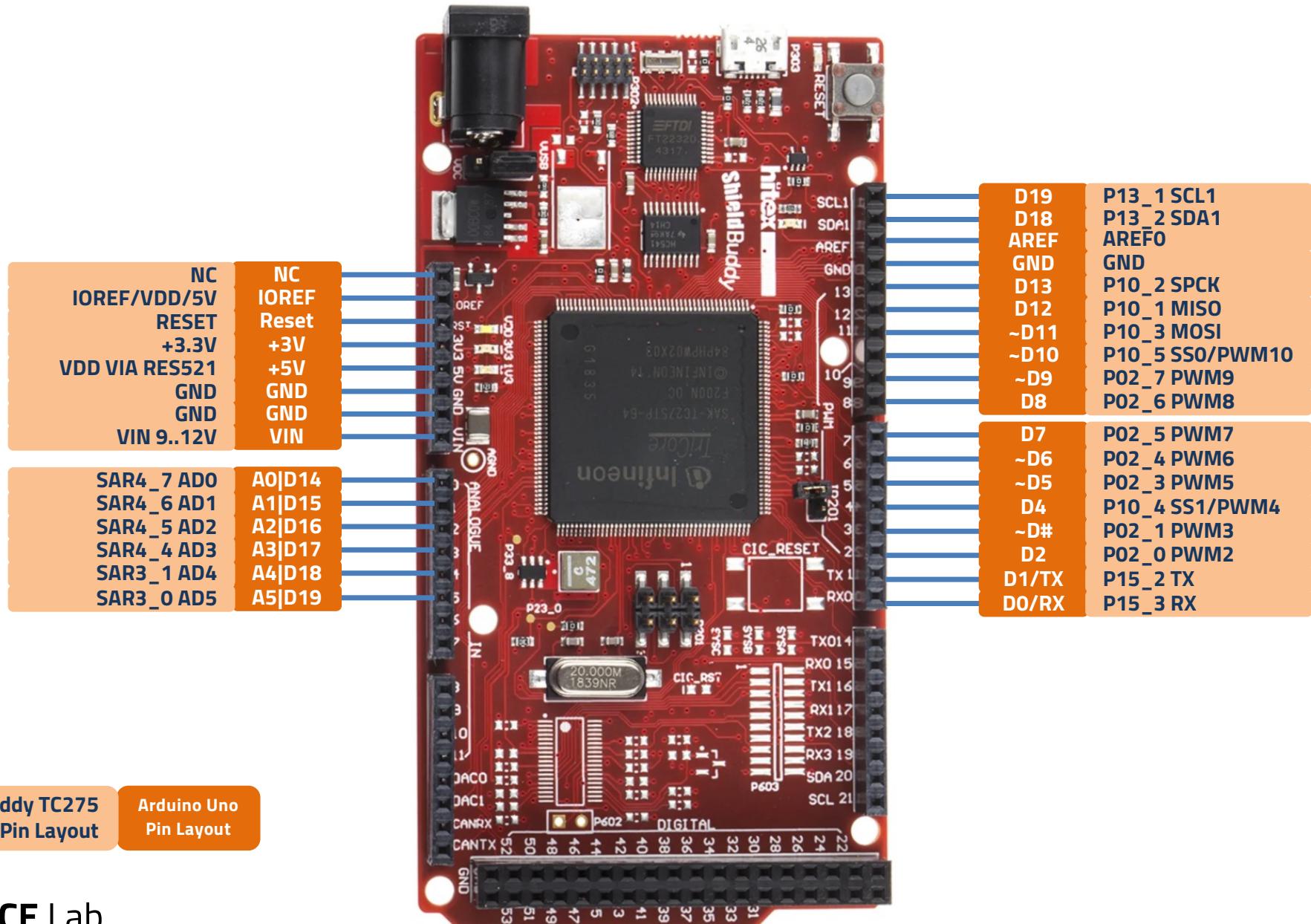
Architecture and Compiler for Embedded System LAB.

School of Electronics Engineering, KNU, KOREA

2021-05-11



Hitex ShieldBuddy TC275



Timer Interrupt Example

- Timer Interrupt를 사용한 일정 시간 간격의 LED Toggle
 1. 새로운 예제를 위한 프로젝트를 생성한다.
 2. 원하는 동작을 위해 레지스터와 메모리에 직접 접근해서 값을 써야한다.
 3. Timer 모듈의 동작 원리를 파악한다.
 4. Timer Interrupt를 사용하기 위해 Datasheet를 분석한다.
 5. 분석 결과를 활용해 임베디드 프로그래밍을 한다.

Timer Interrupt Example

1. Timer 모듈의 동작 원리 파악 (1)

- ✓ TC277은 여러 개의 Timer 모듈을 포함하고 있으며, 주기적인 Timer Interrupt를 발생시키기 위해 **CCU6 (Capture / Compare Unit 6)** 모듈을 사용할 수 있다.
- ✓ CCU6은 **Timer T12**를 포함하며 이는 다음과 같이 구성되어 있다.

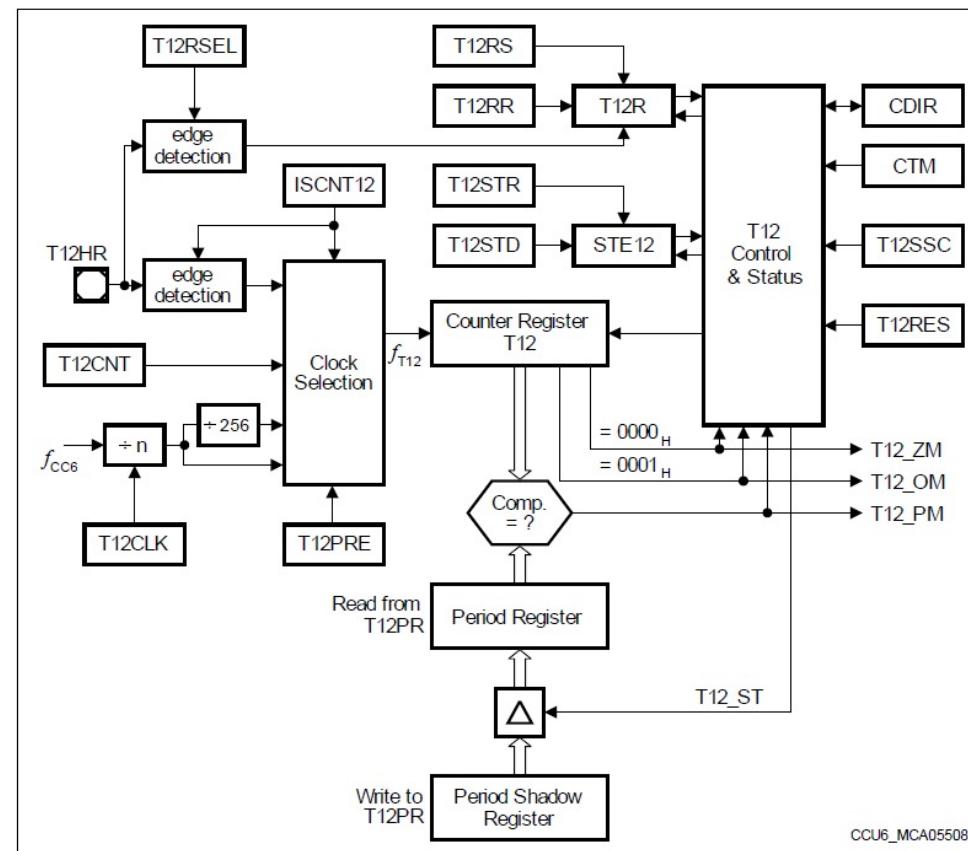


Figure 26-4 Timer T12 Logic and Period Comparators

Timer Interrupt Example

1. Timer 모듈의 동작 원리 파악 (2)

- ✓ Timer T12의 동작 클럭은 CCU6의 입력 클럭을 분주하여 생성할 수 있다.
- ✓ Timer T12의 **T12 Count**는 16-bit 크기를 가지며, 동작 클럭에 따라 1씩 증가한다.
- ✓ T12 Count가 Period Register의 값과 같아지면 **Period Match**가 발생하며 **T12 Count**의 값은 0으로 초기화된다.
- ✓ Timer T12의 동작 클럭과 Period Register의 값을 통해 Period Match의 발생 주기를 조절할 수 있으며, **Timer Interrupt**를 발생시킬 수 있다.

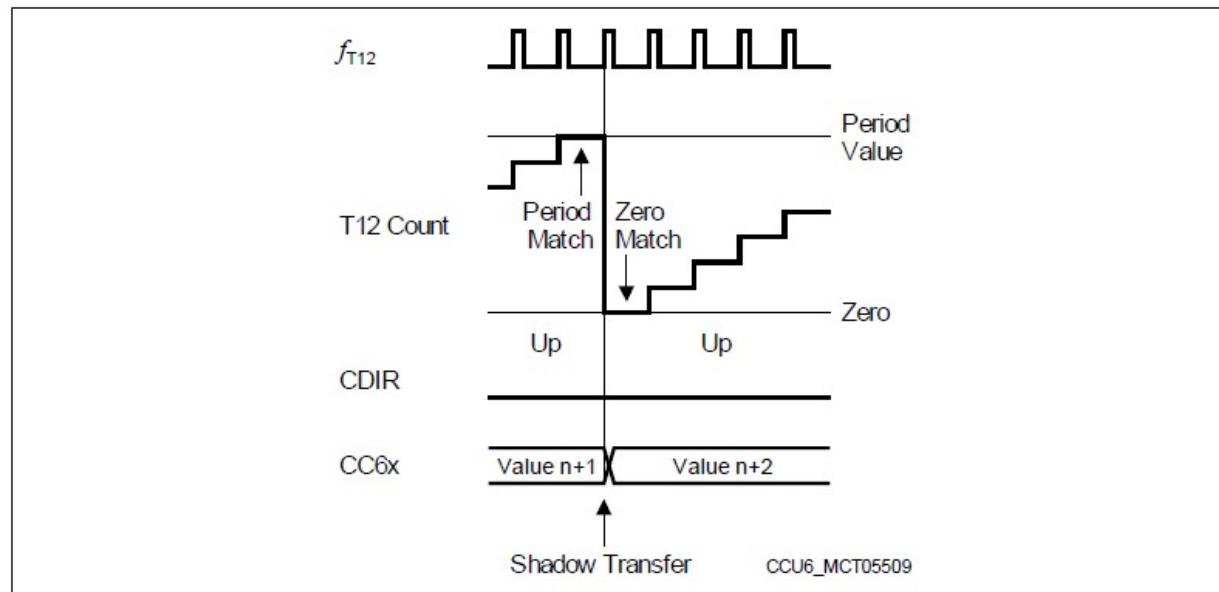


Figure 26-5 T12 Operation in Edge-Aligned Mode

Timer Interrupt Example

2. Data sheet 분석 : CCU6 Enable 설정

- ✓ CCU6_CLC Register는 CCU6 모듈의 Enable 설정을 한다.
- ✓ CCU6 모듈은 2개 (0-1)가 있으며 CCU60을 사용하기 위해 **CCU60_CLC Register**를 설정한다.
- ✓ CCU60 모듈을 Enable 하기 위해 **DISR bit**를 **0**으로 설정한다.
- ✓ CCU60 모듈이 Enable 되어 있는지 확인하기 위해 **DISS bit**가 **0**인지 확인한다.

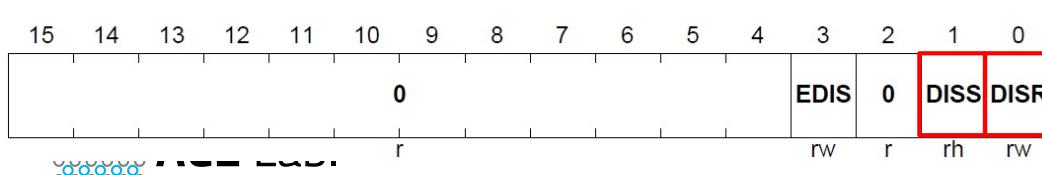
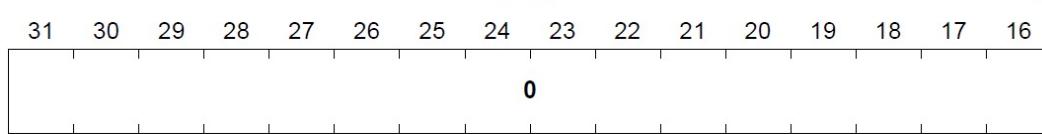
CCU60_CLC Register 주소: F000_2A00h (F0002A00h + 0h)

CCU60_CLC Register 구조:

Table 26-15 Registers Address Space

Module	Base Address	End Address	Note
CCU60	F000 2A00 _H	F000 2AFF _H	CCU6061 module includes CCU60 and CCU61 kernels

CLC
Clock Control Register (00_H) Reset Value: 0000 0003_H



Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the module. 0 _B Module disable is not requested. 1 _B Module disable is requested.
DISS	1	rh	Module Disable Status Bit Bit indicates the current status of the module. 0 _B Module is enabled. 1 _B Module is disabled.
EDIS	3	rw	Sleep Mode Enable Control Used to control module's sleep mode. 0 _B Sleep Mode request is regarded. Module is enabled to go into Sleep Mode. 1 _B Sleep Mode request is disregarded: Sleep Mode cannot be entered upon a request.



Timer Interrupt Example

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

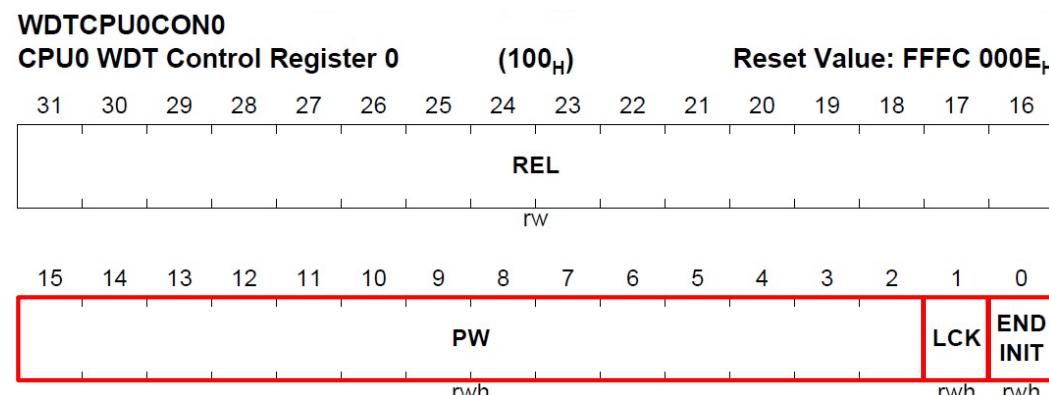
- ✓ 설정해야 하는 CCU60_CLC Register는 System Critical Register이기 때문에 Write Protected (System ENDINIT, End-of-Initialization) 되어 있다.
- ✓ 해당 Register를 수정하기 위해서는 System ENDINIT을 해제해야 한다.
- ✓ SCU_WDTCPU0CON0 Register는 **System Critical Register**으로 **System ENDINIT**을 설정/해제한다.

SCU_WDTCPU0CON0 Register 주소: F003_6100h
(F0036000h + 100h)

SCU_WDTCPU0CON0 Register 구조:

Table 7-27 Registers Address Spaces - SCU Kernel Registers

Module	Base Address	End Address	Note
SCU	F003 6000 _H	F003 63FF _H	-



Timer Interrupt Example

2. Data sheet 분석 : System Critical Register 설정 (2)

- ✓ **ENDINIT bit**는 System ENDINIT의 설정 상태를 나타내며 Modify Access를 통해서만 수정이 가능하다.
- ✓ **LCK bit**는 SCU_WDTCPUOCON0 Register의 Lock 상태를 나타내며 해당 Register의 Lock 상태는 Password Access를 통해 Unlock 되고, Modify Access를 통해 Lock 된다.
- ✓ **PW bits**는 SCU_WDTCPUOCON0 Register에 접근하기 위한 Password를 저장하며 해당 값을 읽으면 bits[7:2]가 반전되어 읽힌다.

Field	Bits	Type	Description	Field	Bits	Type	Description
ENDINIT	0	rwh	End-of-Initialization Control Bit 0 _B Access to Endinit-protected registers is permitted. 1 _B 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.	PW	[15:2]	rwh	User-Definable Password Field for Access to WDTxCON0 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 0000000111100 _B
LCK	1	rwh	Lock Bit to Control Access to WDTxCON0 0 _B Register WDTxCON0 is unlocked 1 _B 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 _B during Password Access and Modify Access. Read returns 000011 _B for these bits.



A

Timer Interrupt Example

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

- ✓ SCU_WDTCPUOCON0 Register에 적절한 값을 Write하여 **Password Access**를 수행한다.
- ✓ **Password Access**는 **SCU_WDTCPUOCON0 Register**의 **Lock** 상태를 해제하며 과정은 다음과 같다.
 1. SCU_WDTCPUOCON0 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_WDTCPUOCON0 Register에 한번에 쓴다.
 6. SCU_WDTCPUOCON0 Register의 LCK bit를 확인하여 Lock 상태가 해제되었는지 파악한다.
(Password Access가 정상적으로 수행되면 Lock 상태가 해제되며 LCK bit가 0으로 설정된다.)
- ✓ Password Access를 통해 SCU_WDTCPUOCON0 Register의 Lock 상태가 해제되면 Modify Access를 통해 System ENDINIT을 설정/해제할 수 있다.

Timer Interrupt Example

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

- ✓ SCU_WDTCPUOCON0 Register에 적절한 값을 Write하여 **Modify Access**를 수행한다.
- ✓ **Modify Access**는 **System ENDINIT**을 설정/해제하며 과정은 다음과 같다.
 1. SCU_WDTCPUOCON0 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.
(System ENDINIT 설정: bit[0] = 1, System ENDINIT 해제 : bit[0] = 0)
 5. 설정된 값을 SCU_WDTCPUOCON0 Register에 한번에 쓴다.
 6. SCU_WDTCPUOCON0 Register의 LCK bit를 확인하여 Lock 상태가 다시 설정되었는지 파악한다.
(Modify Access가 정상적으로 수행되면 Lock 상태가 설정되며 LCK bit가 1로 설정된다.)
- ✓ Modify Access를 통해 System ENDINIT을 해제하면 System Critical Register를 수정할 수 있으며 수정을 완료하면 System ENDINIT을 꼭 다시 설정해야 한다.



Timer Interrupt Example

2. Data sheet 분석 : Timer T12 Count 설정

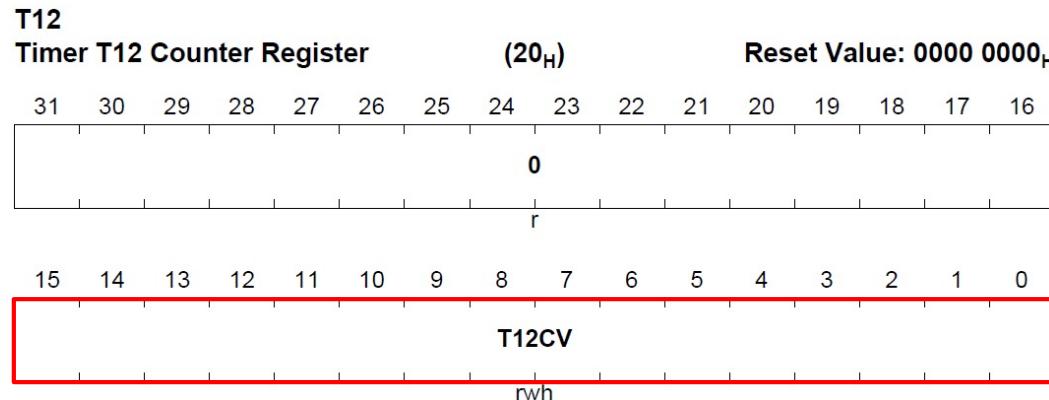
- ✓ CCU60_T12 Register는 Timer T12 Count 값을 저장한다.
- ✓ CCU60_T12 Register를 통해 현재 Timer T12 Count 값을 읽을 수 있다.
- ✓ Count 값을 초기화하기 위해 **CCU60_T12 Register**를 0으로 설정한다.

CCU60_T12 Register 주소: F000_2A20h (F0002A00h + 20h)

CCU60_T12 Register 구조:

Table 26-15 Registers Address Space

Module	Base Address	End Address	Note
CCU60	F000 2A00 _H	F000 2AFF _H	CCU6061 module includes CCU60 and CCU61 kernels



Field	Bits	Type	Description
T12CV	[15:0]	rwh	Timer 12 Counter Value This register represents the 16-bit counter value of Timer12.

Timer Interrupt Example

2. Data sheet 분석 : Timer Period Register 설정

- ✓ CCU60_T12PR Register는 Timer T12 Count 값과 비교될 Period 값을 저장한다.
 - ✓ CCU60_T12PR Register를 통해 Period 값을 설정할 수 있다.
 - ✓ Period Match의 발생 주기를 고려하여 **CCU60_T12PR Register**를 적절한 값으로 설정한다.

CCU60_T12PR Register 주소: F000_2A24h (F0002A00h + 24h)

CCU60_T12PR Register 구조:

Table 26-15 Registers Address Space

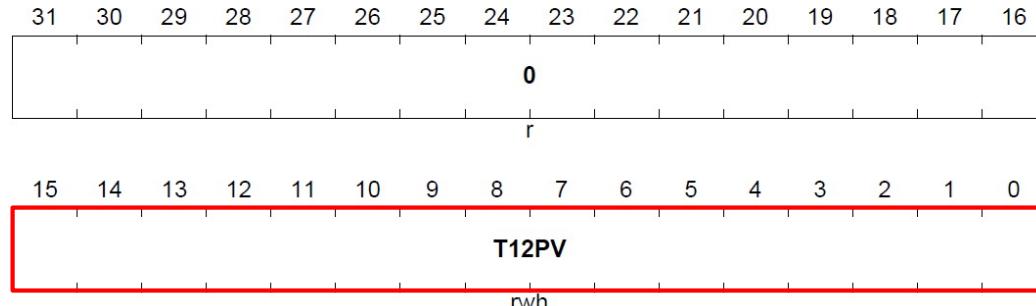
Module	Base Address	End Address	Note
CCU60	F000 2A00 _H	F000 2AFF _H	CCU60 module includes CCU60 and CCU61 kernels

T12PR

Timer 12 Period Register

(24_H)

Reset Value: 0000 0000



Field	Bits	Type	Description
T12PV	[15:0]	rwh	<p>T12 Period Value</p> <p>The value T12PV defines the counter value for T12 leading to a period-match. When reaching this value, the timerT12 is set to zero (edge-aligned mode) or changes its count direction to down counting (center-aligned mode).</p>



Timer Interrupt Example

2. Data sheet 분석 : Timer T12 Control 0 설정 (1)

- ✓ CCU60_TCTR0 Register는 Timer 동작에 대한 설정을 한다.
- ✓ CCU60_TCTR0 Register는 CCU60이 포함하고 있는 Timer T12 / T13에 대한 설정을 할 수 있다.
- ✓ Timer T12가 Period Match 이후 T12 Count를 초기화하고 다시 증가하도록 동작을 설정하기 위해 **CTM bit**를 **0**으로 설정한다.

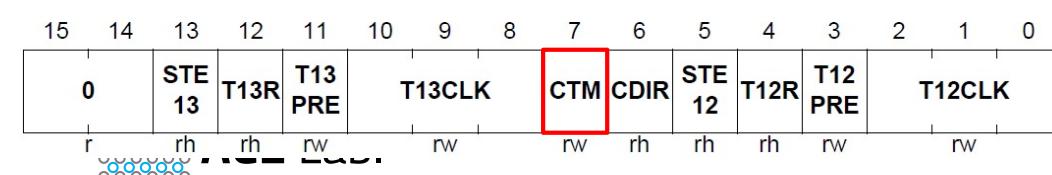
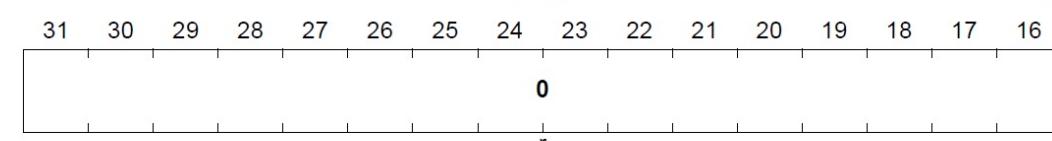
CCU60_TCTR0 Register 주소: F000_2A70h (F0002A00h + 70h)

CCU60_TCTR0 Register 구조:

Table 26-15 Registers Address Space

Module	Base Address	End Address	Note
CCU60	F000 2A00 _H	F000 2AFF _H	CCU6061 module includes CCU60 and CCU61 kernels

TCTR0
Timer Control Register 0 (70_H) Reset Value: 0000 0000_H



CDIR	6	rh	Count Direction of Timer T12 This bit is set/cleared according to the counting rules of T12. 0 _B T12 counts up. 1 _B T12 counts down.
CTM	7	rw	T12 Operating Mode 0 _B Edge-aligned Mode: T12 always counts up and continues counting from zero after reaching the period value. 1 _B Center-aligned Mode: T12 counts down after detecting a period-match and counts up after detecting a one-match.



Timer Interrupt Example

2. Data sheet 분석 : Timer T12 Control 0 설정 (2)

- ✓ Timer T12의 동작 클럭을 설정하기 위해 T12CLK bits와 T12PRE bit를 설정한다.
- ✓ CCU60 모듈 입력 클럭의 주파수는 50MHz이다.
- ✓ **T12CLK bits**를 **010b**로 설정하여 $50\text{MHz} / 4 = 12.5\text{MHz}$ 의 동작 클럭을 생성한다.
- ✓ **T12PRE bit**를 1로 설정하여 $12.5\text{MHz} / 256 = \textbf{48,828Hz}$ 의 동작 클럭을 생성한다.

TCTR0 Timer Control Register 0																(70 _H)				Reset Value: 0000 0000 _H																							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																												
0																																											
r																																											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	STE 13	T13R	T13 PRE	T13CLK	CTM	CDIR	STE 12	T12R	T12 PRE	T12CLK	r	rh	rh	rw	rw	rw	rh	rh	rh	rw							

Field	Bits	Type	Description
T12CLK	[2:0]	rw	Timer T12 Input Clock Select Selects the input clock for timer T12 that is derived from the peripheral clock according to the equation $f_{T12} = f_{CC6} / 2^{<T12CLK>}.$ 000 _B $f_{T12} = f_{CC6}$ 001 _B $f_{T12} = f_{CC6} / 2$ 010 _B $f_{T12} = f_{CC6} / 4$ 011 _B $f_{T12} = f_{CC6} / 8$ 100 _B $f_{T12} = f_{CC6} / 16$ 101 _B $f_{T12} = f_{CC6} / 32$ 110 _B $f_{T12} = f_{CC6} / 64$ 111 _B $f_{T12} = f_{CC6} / 128$
T12PRE	3	rw	Timer T12 Prescaler Bit In order to support higher clock frequencies, an additional prescaler factor of 1/256 can be enabled for the prescaler for T12. 0 _B The additional prescaler for T12 is disabled. 1 _B The additional prescaler for T12 is enabled.

Timer Interrupt Example

2. Data sheet 분석 : Period Match 주기 설정

- ✓ Period Match가 발생하는 빈도는 다음과 같다.

$$(Freq. \text{ of } Period \text{ Match}) = \frac{(Freq. \text{ of Timer T12 Clock})}{(Value \text{ of Period Register}) + 1}$$

- ✓ CCU60_TCTR0 Register를 통해 Timer T12 동작 클럭의 주파수가 48,828Hz로 설정된다.
- ✓ Period Match를 0.5초마다 발생시키기 위해 Period Match의 빈도를 2Hz로 설정해야 한다.
- ✓ 따라서, **CCU60_T12PR Register**를 **(24,414 – 1)**로 설정한다.

Timer Interrupt Example

2. Data sheet 분석 : Timer T12 Control 4 설정

- ✓ CCU60_TCTR4 Register는 Timer 제어에 대한 설정을 한다.
 - ✓ CCU60_TCTR4 Register는 CCU60이 포함하고 있는 Timer T12 / T13에 대한 제어를 할 수 있다.
 - ✓ Shadow Register에 저장되어 있는 Period 설정 값을 CCU60_T12PR Register에 적용하기 위해 **T12STR bit**를 1로 설정한다.
 - ✓ Timer T12의 동작을 시작하기 위해 **T12RS bit**를 1로 설정한다.

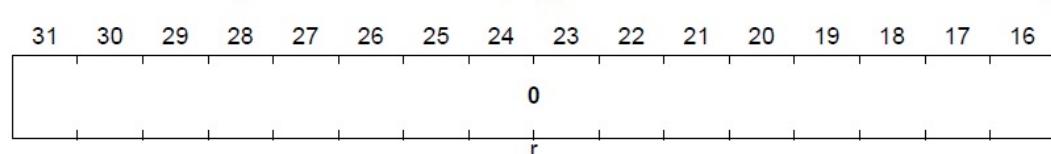
CCU60_TCTR4 Register 주소: F000_2A78h (F0002A00h + 78h)

CCU60_TCTR4 Register 구조:

Table 26-15 Registers Address Space

Module	Base Address	End Address	Note
CCU60	F000 2A00 _H	F000 2AFF _H	CCU6061 module includes CCU60 and CCU61 kernels

TCTR4 Timer Control Register 4 (78H) Reset Value: 0000 0000H



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13 STD	T13 STR	T13 CNT	0	T13 RES	T13 RS	T13 RR	T12 STD	T12 STR	T12 CNT	0	DT RES	T12 RES	T12 RS	T12 RR	
W	W	W	R	W	W	W	W	W	W	R	W	W	W	W	W

Field	Bits	Type	Description
T12RR	0	w	Timer T12 Run Reset Setting this bit clears the T12R bit. 0 _B T12R is not influenced. 1 _B T12R is cleared, T12 stops counting.

T12RS	1	w	Timer T12 Run Set Setting this bit sets the T12R bit. 0 _B T12R is not influenced. 1 _B T12R is set, T12 starts counting.
--------------	---	---	---

Field	Bits	Type	Description
T12STR	6	w	Timer T12 Shadow Transfer Request 0 _B No action 1 _B STE12 is set, enabling the shadow transfer.

Timer Interrupt Example

2. Data sheet 분석 : Timer T12 Interrupt Enable 설정

- ✓ CCU60_IEN Register는 Interrupt Enable 설정을 한다.
- ✓ CCU60_IEN Register는 CCU60에서 발생할 수 있는 여러 Interrupt에 대한 Enable을 각각 설정한다.
- ✓ Timer T12의 Period Match가 일어날 때마다 Interrupt가 발생하도록 하기 위해 **ENT12PM bit**를 1로 설정한다.

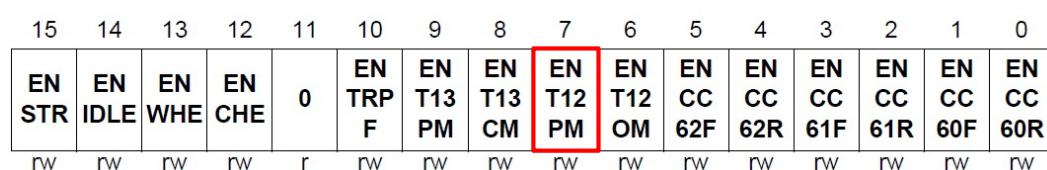
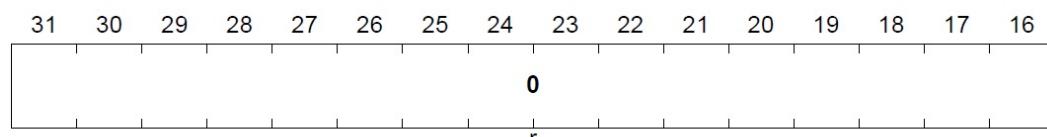
CCU60_IEN Register 주소: F000_2AB0h (F0002A00h + B0h)

CCU60_IEN Register 구조:

Table 26-15 Registers Address Space

Module	Base Address	End Address	Note
CCU60	F000 2A00 _H	F000 2AFF _H	CCU6061 module includes CCU60 and CCU61 kernels

IEN
Interrupt Enable Register (B0_H) Reset Value: 0000 0000_H



Field	Bits	Type	Description
ENT12OM	6	rw	Enable Interrupt for T12 One-Match 0 _B No interrupt will be generated if the set condition for bit T12OM in register IS occurs. 1 _B An interrupt will be generated if the set condition for bit T12OM in register IS occurs. The service request output that will be activated is selected by bit field INPT12.
ENT12PM	7	rw	Enable Interrupt for T12 Period-Match 0 _B No interrupt will be generated if the set condition for bit T12PM in register IS occurs. 1 _B An interrupt will be generated if the set condition for bit T12PM in register IS occurs. The service request output that will be activated is selected by bit field INPT12.
ENT13CM	8	rw	Enable Interrupt for T13 Compare-Match 0 _B No interrupt will be generated if the set condition for bit T13CM in register IS occurs. 1 _B An interrupt will be generated if the set condition for bit T13CM in register IS occurs. The service request output that will be activated is selected by bit field INPT13.

Timer Interrupt Example

2. Data sheet 분석 : Timer T12 Interrupt Node Pointer 설정

- ✓ CCU60_INP Register는 Interrupt가 전달될 Node Pointer 설정을 한다.
 - ✓ CCU60은 Interrupt Router로 연결되는 4개의 Node Pointer (SRO-3)을 가지며 Interrupt 신호를 Interrupt Router에 전달하기 위해 Node Pointer를 설정해야 한다.
 - ✓ Timer T12에서 발생한 Interrupt (Period Match)를 SR0에 전달하기 위해 **INPT12 bits**를 **00b**로 설정한다.

CCU60_INP Register 주소: F000_2AACh (F0002A00h + ACh)

CCU60_INP Register 구조:

Table 26-15 Registers Address Space

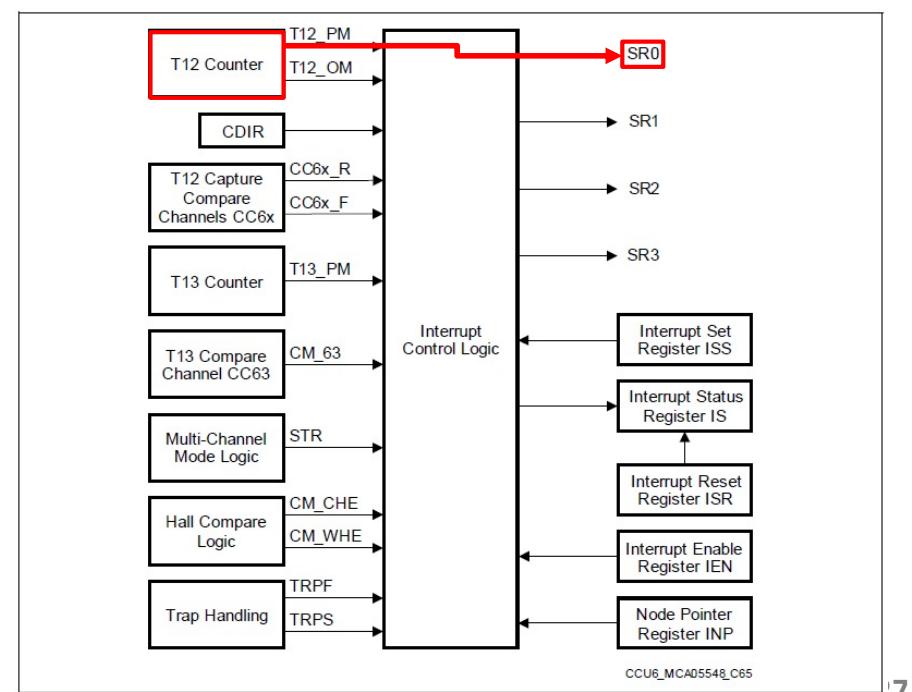


Figure 26-43 Interrupt Sources and Events

Timer Interrupt Example

2. Data sheet 분석 : Interrupt Router 설정

- ✓ SRC_CCU60SR0 Register는 CCU60SR0 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.

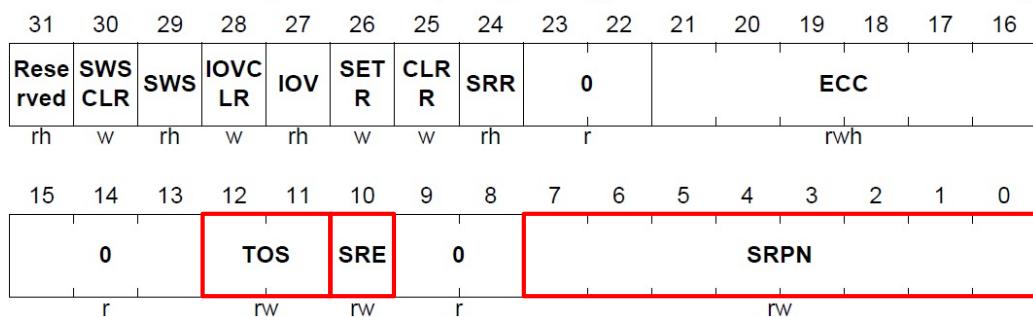
SRC_CCU60SR0 Register 주소: F003_8420h
(F0038000h + 420h)

SRC_CCU60SR0 Register 구조:

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

Module	Base Address	End Address	Note
SRC	F003 8000 _H	F003 9FFF _H	

SRC_CCU6mSR0 (m=0-1)
CCU6 m Service Request 0 (0420_H+m*10_H) Reset Value: 0000 0000_H



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

```
66 /* Define PORT10 Registers for LED */
67 #define PORT10_BASE      (0xF003B000)
68 #define PORT10_IOCR0     (*(volatile unsigned int*)(PORT10_BASE + 0x10))
69 #define PORT10_OMR       (*(volatile unsigned int*)(PORT10_BASE + 0x04))
70
71 #define PC1              11
72 #define PCL1             17
73 #define PS1              1
```

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

```
77 void init_LED(void)
78 {
79     /* Reset PC1 in IOCR0*/
80     PORT10_IOCR0 &= ~((0x1F) << PC1);
81
82     /* Set PC1 with push-pull(2b10000) */
83     PORT10_IOCR0 |= ((0x10) << PC1);
84 }
```

PORT13 IO (LED RED) 설정 초기화 코드

Timer Interrupt Example

3. 프로그래밍

2) CCU60을 설정하기 위한 함수를 구현한다.

- ① SCU_WDTCPU0CON0 Register를 통해 Password/Modify Access를 수행하여 System ENDINIT을 해제한다.
- ② CCU60_CLC Register를 통해 CCU60 모듈을 Enable 한다.
- ③ SCU_WDTCPU0CON0 Register를 통해 Password/Modify Access를 수행하여 System ENDINIT을 설정한다.
- ④ CCU60_TCTR0 Register를 통해 Timer T12의 동작 클럭의 주파수를 48,828Hz로 설정한다.
- ⑤ CCU60_TCTR0 Register를 통해 Timer T12의 동작 모드를 Up-count 모드로 설정한다.
- ⑥ CCU60_T12PR Register를 통해 Period Match가 1초에 2번 발생하도록 Period 값을 설정한다.
- ⑦ CCU60_TCTR4 Register를 통해 Shadow Register에 저장된 Period 값을 CCU60_T12PR Register에 적용한다.
- ⑧ CCU60_T12 Register를 통해 Timer T12 Count를 초기화한다.
- ⑨ CCU60_INP Register를 통해 Period Match에서 발생한 Interrupt 신호를 SR0에 전달한다.
- ⑩ CCU60_IEN Register를 통해 Period Match에서 발생한 Interrupt 신호를 Enable 한다.
- ⑪ SRC_CCU60SR0 Register를 통해 CCU60SR0 SRN의 우선순위를 설정한다.
- ⑫ SRC_CCU60SR0 Register를 통해 CCU60SR0 SRN의 처리가 CPU0에서 수행되도록 설정한다.
- ⑬ SRC_CCU60SR0 Register를 통해 CCU60SR0 SRN의 Interrupt를 Enable 한다.
- ⑭ CCU60_TCTR4 Register를 통해 Timer T12의 동작을 시작한다.



Timer Interrupt Example

3. 프로그래밍

2) CCU60을 설정하기 위한 함수를 구현한다.

```
31 /* SCU Registers */
32 #define SCU_BASE          (0xF0036000)
33 #define SCU_WDT_CPU0CON0   (*(volatile unsigned int*)(SCU_BASE + 0x100))
34
35 #define LCK                1
36 #define ENDINIT            0
37
38 /* SRC Registers */
39 #define SRC_BASE           (0xF0038000)
40 #define SRC_CCU60_SR0      (*(volatile unsigned int*)(SRC_BASE + 0x420))
41
42 #define TOS                11
43 #define SRE                10
44 #define SRPN               0
45
46 /* CCU60 Registers */
47 #define CCU60_BASE          (0xF0002A00)
48 #define CCU60_CLC           (*(volatile unsigned int*)(CCU60_BASE + 0x00))
49 #define CCU60_T12             (*(volatile unsigned int*)(CCU60_BASE + 0x20))
50 #define CCU60_T12PR          (*(volatile unsigned int*)(CCU60_BASE + 0x24))
51 #define CCU60_TCTR0           (*(volatile unsigned int*)(CCU60_BASE + 0x70))
52 #define CCU60_TCTR4           (*(volatile unsigned int*)(CCU60_BASE + 0x78))
53 #define CCU60_INP            (*(volatile unsigned int*)(CCU60_BASE + 0xAC))
54 #define CCU60_IEN            (*(volatile unsigned int*)(CCU60_BASE + 0xB0))
55
56 #define DISS                1
57 #define DISR                0
58 #define CTM                 7
59 #define T12PRE              3
60 #define T12CLK              0
61 #define T12STR              6
62 #define T12RS               1
63 #define INPT12              10
64 #define ENT12PM             7
```

Timer Interrupt Example

3. 프로그래밍

2) CCU60을 설정하기 위한 함수를 구현한다.

```
86① void init_CCU60(void)
87 {
88     /* CCU60 T12 Setting */
89
90     /* Password Access to unlock WDTSCON0 */
91     SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) & ~(1 << LCK)) | (1 << ENDINIT);
92     while((SCU_WDT_CPU0CON0 & (1 << LCK)) != 0);
93
94     /* Modify Access to clear ENDINIT bit */
95     SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) | (1 << LCK)) & ~ (1 << ENDINIT);
96     while((SCU_WDT_CPU0CON0 & (1 << LCK)) == 0);
97
98     ② CCU60_CLC &= ~(1 << DISR);           // Enable CCU60 Module
99
100
101    /* Password Access to unlock WDTSCON0 */
102    ③ SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) & ~(1 << LCK)) | (1 << ENDINIT);
103    while((SCU_WDT_CPU0CON0 & (1 << LCK)) != 0);
104
105    /* Modify Access to clear ENDINIT bit */
106    SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) | (1 << LCK)) & ~ (1 << ENDINIT);
107    while((SCU_WDT_CPU0CON0 & (1 << LCK)) == 0);
108
109    while((CCU60_CLC & (1 << DISS)) != 0); // Wait until module is enabled
110
111    ④ CCU60_TCTR0 &= ~((0x7) << T12CLK);      // f_T12 = f_CCU6 / prescaler
112    CCU60_TCTR0 |= ((0x2) << T12CLK);          // F_CCU6 = 50 MHz, prescaler = 1024
113    CCU60_TCTR0 |= (1 << T12PRE);              // f_T12 = 48828 Hz
114
115    ⑤ CCU60_TCTR0 &= ~(1 << CTM);            // T12 always counts up and continues counting
116                                            // from zero after reaching the period value
117
118    ⑥ CCU60_T12PR = 24414 - 1;                  // Interrupt freq. = f_T12 / (period value + 1)
119    CCU60_TCTR4 |= (1 << T12STR);             // Interrupt freq. = 2 Hz
120
121    ⑧ CCU60_T12 = 0;                          // Clear T12 counting value
```

Timer Interrupt Example

3. 프로그래밍

2) CCU60을 설정하기 위한 함수를 구현한다.

```
123 /* CCU60 T12 Interrupt Setting */
124 (9) CCU60_INP &= ~((0x3) << INPT12);           // Service Request output SR0 is selected
125
126 (A) CCU60_IEN |= (1 << ENT12PM);                // Enable Interrupt for T12 Period-Match
127
128 /* SRC Interrupt Setting For CCU60 */
129 (B) SRC_CCU60_SR0 &= ~((0xFF) << SRPN);        // Set Priority : 0x0A
130   SRC_CCU60_SR0 |= ((0x0A) << SRPN);
131
132 (C) SRC_CCU60_SR0 &= ~((0x3) << TOS);          // CPU0 services
133
134 (D) SRC_CCU60_SR0 |= (1 << SRE);                // Service Request is enabled
135
136 /* CCU60 T12 Start */
137 (E) CCU60_TCTR4 = (1 << T12RS);                 // T12 starts counting
138 }
```

CCU60 설정 함수



Timer Interrupt Example

3. 프로그래밍

3) CCU60의 Timer T12에 의한 Interrupt의 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.

```
163 __interrupt( 0x0A ) __vector_table( 0 )
164 void CCU60_T12_ISR(void)
165 {
166     PORT10_OMR |= ((1<<PCL1) | (1<<PS1));           // Toggle LED RED
167 }
```

CCU60 Interrupt Service Routine

Timer Interrupt Example

3. 프로그래밍

4) 동작에 따라 'main' 함수를 구현한다.

```
31 /* SCU Registers */
32 #define SCU_BASE          (0xF0036000)
33 #define SCU_WDT_CPU0CON0   (*(volatile unsigned int*)(SCU_BASE + 0x100))
34
35 #define LCK                1
36 #define ENDINIT             0
37
38 /* SRC Registers */
39 #define SRC_BASE           (0xF0038000)
40 #define SRC_CCU60_SR0      (*(volatile unsigned int*)(SRC_BASE + 0x420))
41
42 #define TOS                11
43 #define SRE                10
44 #define SRPN               0
45
46 /* CCU60 Registers */
47 #define CCU60_BASE          (0xF0002A00)
48 #define CCU60_CLC           (*(volatile unsigned int*)(CCU60_BASE + 0x00))
49 #define CCU60_T12            (*(volatile unsigned int*)(CCU60_BASE + 0x20))
50 #define CCU60_T12PR          (*(volatile unsigned int*)(CCU60_BASE + 0x24))
51 #define CCU60_TCTR0          (*(volatile unsigned int*)(CCU60_BASE + 0x70))
52 #define CCU60_TCTR4          (*(volatile unsigned int*)(CCU60_BASE + 0x78))
53 #define CCU60_INP            (*(volatile unsigned int*)(CCU60_BASE + 0xAC))
54 #define CCU60_IEN            (*(volatile unsigned int*)(CCU60_BASE + 0xB0))
55
56 #define DISS               1
57 #define DISR               0
58 #define CTM                7
59 #define T12PRE             3
60 #define T12CLK              0
61 #define T12STR              6
62 #define T12RS               1
63 #define INPT12             10
64 #define ENT12PM             7
65
66 /* Define PORT10 Registers for LED */
67 #define PORT10_BASE         (0xF003B000)
68 #define PORT10_IOCR0        (*(volatile unsigned int*)(PORT10_BASE + 0x10))
69 #define PORT10_OMR          (*(volatile unsigned int*)(PORT10_BASE + 0x04))
70
71 #define PC1                11
72 #define PCL1               17
73 #define PS1                1
74
75 IfxCpu_syncEvent g_cpuSyncEvent = 0;
```

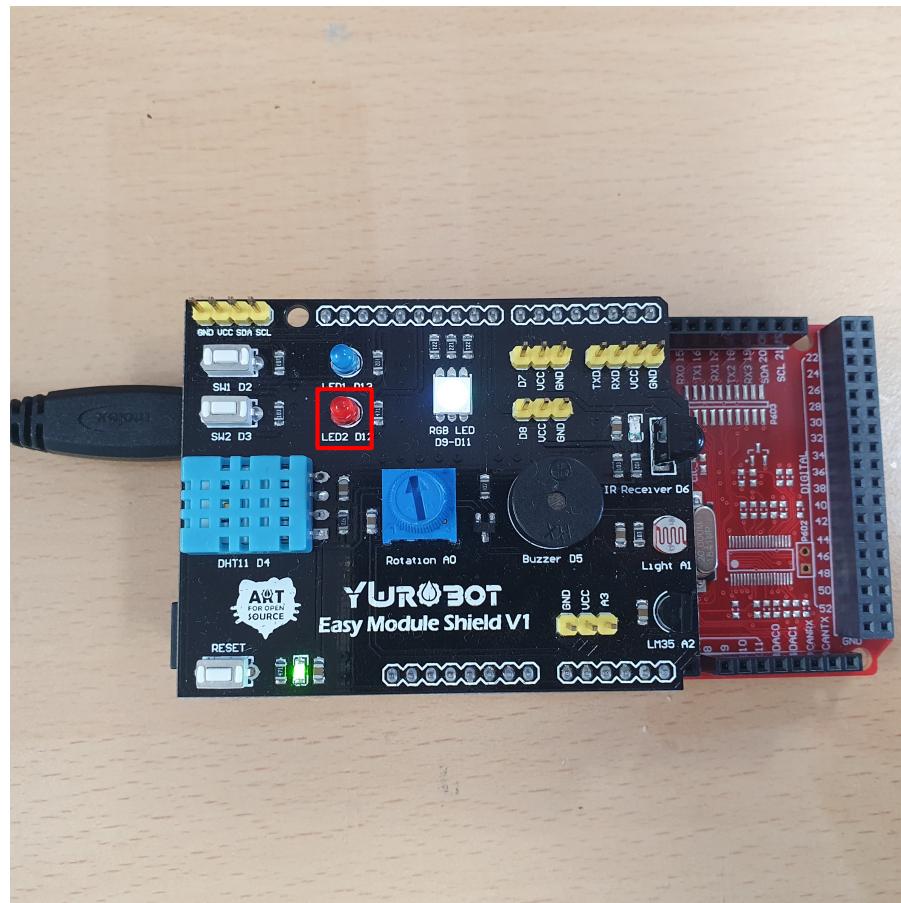
```
140 /* Function Prototype */
141 void init_LED(void);
142 void init_CCU60(void);
143
144 int core0_main(void)
145 {
146     IfxCpu_enableInterrupts();
147
148     /* !!!WATCHDOG AND SAFETY WATCHDOG ARE DISABLED HERE!!!
149     * Enable the watchdogs and service them periodically if it is required
150     */
151     IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
152     IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
153
154     /* Wait for CPU sync event */
155     IfxCpu_emitEvent(&g_cpuSyncEvent);
156     IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
157
158     init_LED();
159     init_CCU60();
160
161     while(1)
162     {
163     }
164
165     return (1);
166
167 __interrupt( 0x0A ) __vector_table( 0 )
168 void CCU60_T12_ISR(void)
169 {
170     PORT10_OMR |= ((1<<PCL1) | (1<<PS1));           // Toggle LED RED
171 }
```

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

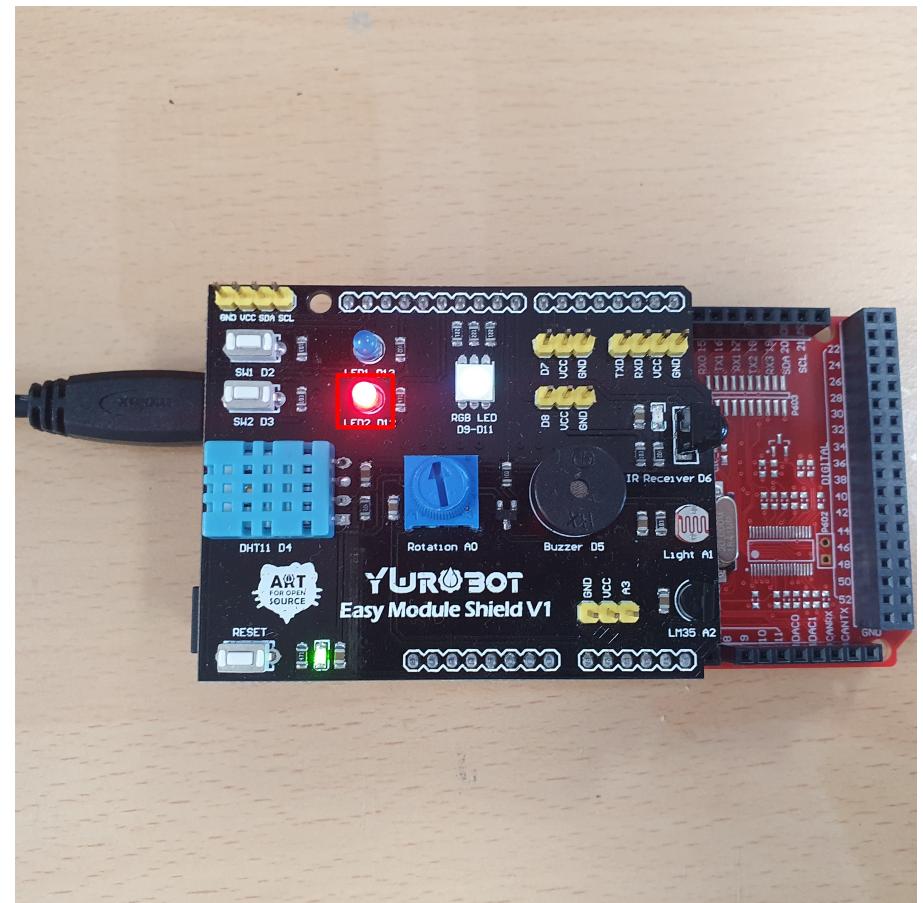
Timer Interrupt Example

4. 동작 확인

- ✓ Build 및 Debug 후 ('Resume' 버튼 클릭), 0.5초마다 LED가 깜빡이는 것을 확인한다.

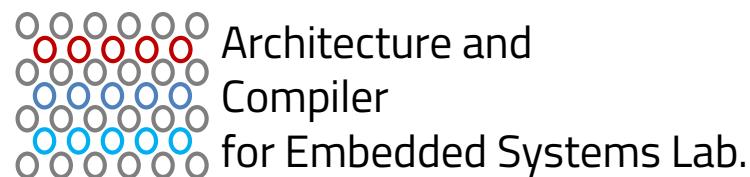


Every
0.5s



Q & A

Thank you for your attention



School of Electronics Engineering, KNU

ACE Lab (hn02301@gmail.com)