

Infineon TC275 PWM (Pulse Width Modulation)

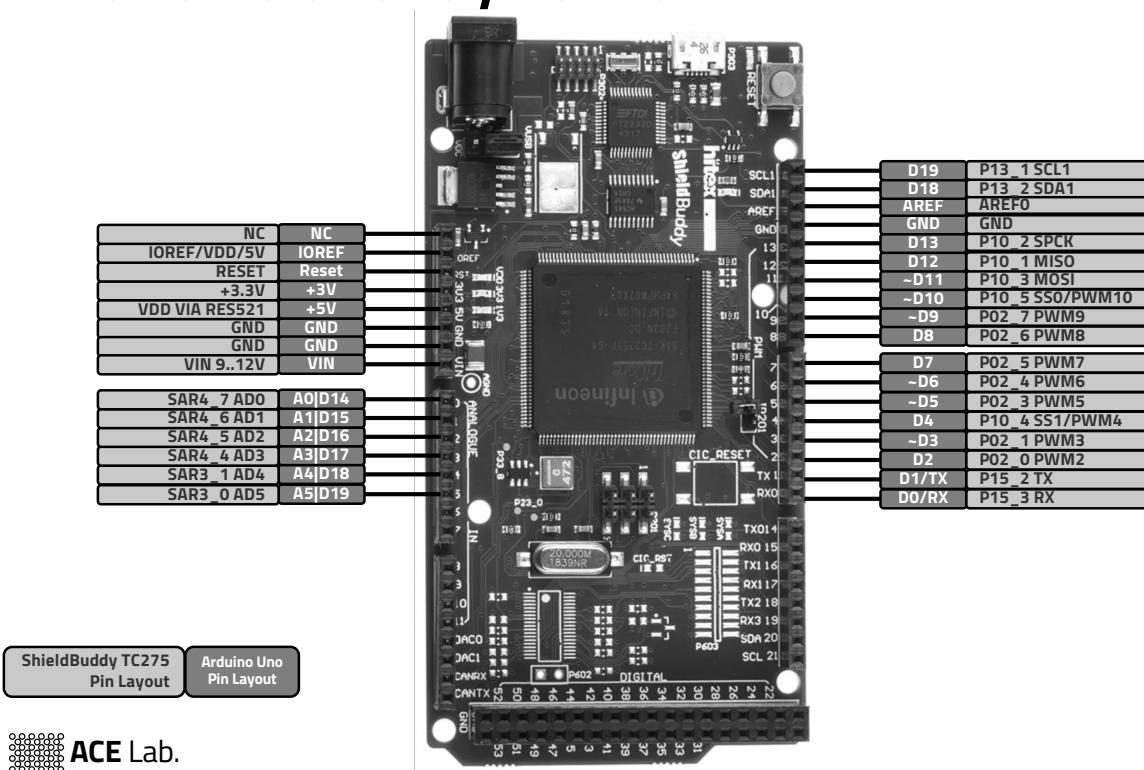
Hyeongrae Kim

Architecture and Compiler for Embedded system Lab.

School of Electronics Engineering, KNU, KOREA



Hitex ShieldBuddy TC275



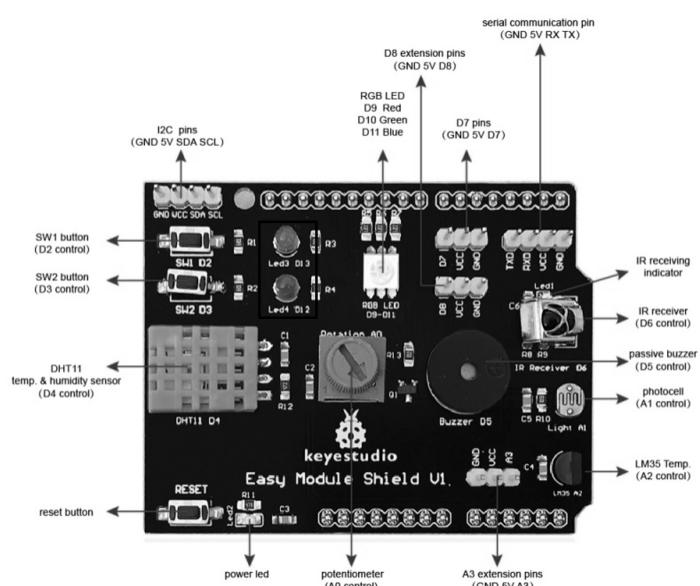
PWM Example

- PWM Duty Ratio에 따른 LED 밝기 변화
 1. 새로운 예제를 위한 프로젝트를 생성한다.
 2. 원하는 동작을 위해 레지스터와 메모리에 직접 접근해서 값을 써야한다.
 3. Board Schematic과 Datasheet를 통해 PWM 신호 출력에 대한 정보를 파악한다.
 4. PWM 신호 생성을 위해 사용할 GTM 모듈의 동작 원리를 파악하고 메모리 맵을 분석한다.
 5. 분석 결과를 활용해 임베디드 프로그래밍을 한다.

PWM Example

1. LED 연결 정보 파악

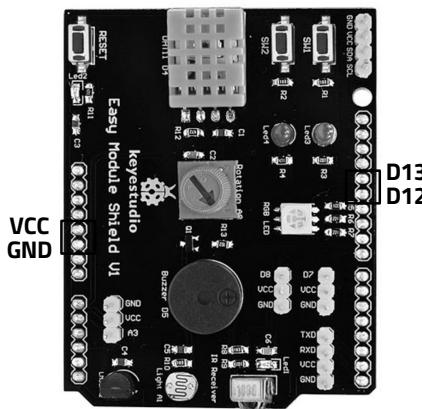
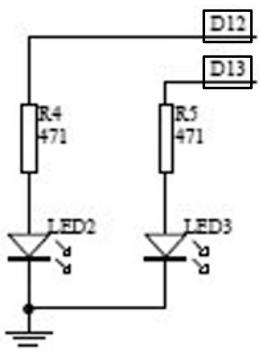
- ✓ 여러 LED를 사용하기 위해 Target Board가 아닌 **Easy Module Shield V1 확장 보드**의 LED를 사용한다.



PWM Example

1. LED 연결 정보 파악

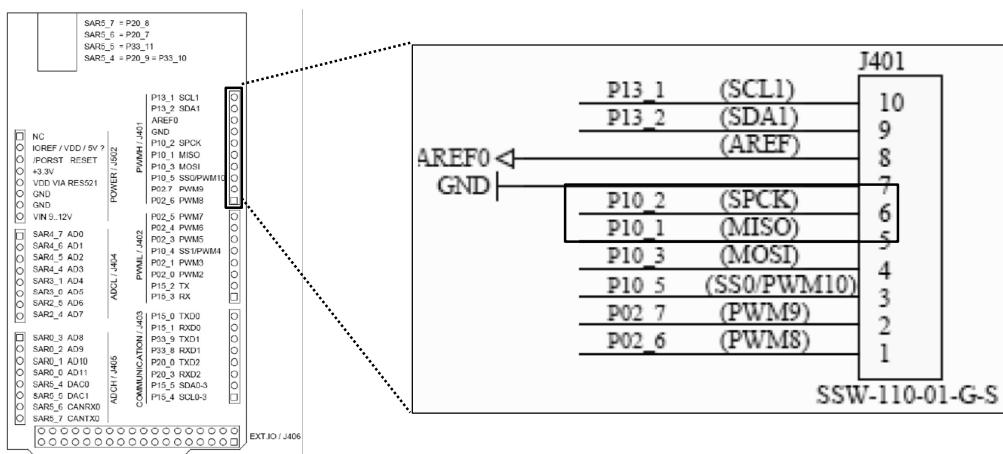
- ✓ LED는 Easy Module Shield V1 확장 보드의 Pin D12(RED)/D13(BLUE)과 연결되어 있다.
- ✓ 타겟 보드는 Easy Module Shield V1 확장 보드의 Pin D12/D13을 통해 LED 출력을 보낼 수 있다.



PWM Example

1. LED 연결 정보 파악

- ✓ TC275 보드의 Schematic과 Datasheet를 확인했을 때, Easy Module Shield V1 확장 보드의 Pin D12/D13과 연결되는 IO는 PORT10의 Pin 1-2다.
- ✓ 해당 Pin의 출력이 High-level 일 때 LED는 켜지고, Low-level 일 때 LED는 꺼진다.



PWM Example

1. PWM 신호 출력 정보 파악

- ✓ LED가 연결된 PORT10 Pin 1는 GTM 모듈의 TOUT103과 연결되어 있다.
- ✓ GTM 모듈의 TOUT103이 PWM 신호를 출력하면 PORT10 Pin 1을 통해 LED에 인가될 수 있다.
- ✓ PWM 신호를 통해 LED 밝기를 제어하기 위해 해당 Pin을 **GTM 모듈의 TOUT103 (O1)**으로 설정해야 한다.

Pin	Symbol	Ctrl	Type	Function
169	P10.1	I	MP+ / PU1 / VEXT	General-purpose input
	TIN103			GTM input
	MRST1A			QSPI1 input
	T5EUDB			GPT120 input
	P10.1			General-purpose output
	TOUT103		O	GTM output
	MTSR1			QSPI1 output
	MRST1			QSPI1 output
	EN01			MSC0 output
	VADCG6BFL1			VADC output
	END03			MSC0 output
	-			Reserved

PWM Example

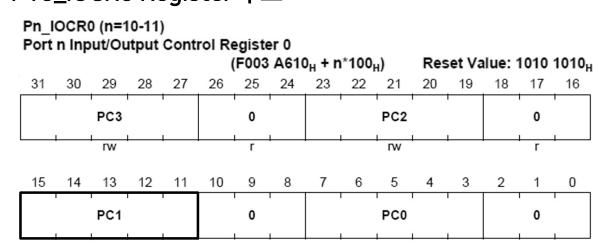
2. Data sheet 분석 : PORT 설정 (1)

- ✓ P10_IOC0 Register는 PORT10의 Input/Output을 설정한다.
- ✓ LED가 PORT10의 Pin 1에 연결되어 있기 때문에 **P10_IOC0 Register의 PC1 bits**를 설정한다.

Table 13-3 Registers Address Space			
Module	Base Address	End Address	Note
P00	F003 A000H	F003 A0FFH	13 pins
P01	F003 A100H	F003 A1FFH	5 pins
P02	F003 A200H	F003 A2FFH	12 pins
P10	F003 B000H	F003 B0FFH	9 pins
P11	F003 B100H	F003 B1FFH	16 pins
P12	F003 B200H	F003 B2FFH	2 pins
P13	F003 B300H	F003 B3FFH	4 pins
P14	F003 B400H	F003 B4FFH	11 pins
P15	F003 B500H	F003 B5FFH	9 pins

P10_IOC0 Register 주소: F003_B010h (F003B000h + 10h)

P10_IOC0 Register 구조:



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

PWM Example

2. Data sheet 분석 : PORT 설정 (2)

- ✓ PORT10의 Pin 1을 GTM 모듈의 TOUT103 (01)으로 설정하기 위해 PC1 bits를 10001b로 설정한다.

Table 13-5 PCx Coding

PCx[4:0]	I/O	Characteristics	Selected Pull-up / Pull-down / Selected Output Function
10000 _B	Output	Push-pull	General-purpose output
10001 _B			Alternate output function 1
10010 _B			Alternate output function 2
10011 _B			Alternate output function 3
10100 _B			Alternate output function 4
10101 _B			Alternate output function 5
10110 _B			Alternate output function 6
10111 _B			Alternate output function 7
11000 _B	Open-drain		General-purpose output
11001 _B			Alternate output function 1
11010 _B			Alternate output function 2
11011 _B			Alternate output function 3
11100 _B			Alternate output function 4
11101 _B			Alternate output function 5
11110 _B			Alternate output function 6
11111 _B			Alternate output function 7

PWM Example

2. Data sheet 분석 : GTM Enable 설정

- ✓ GTM_CLC Register는 GTM 모듈의 Enable 설정을 한다.
- ✓ GTM 모듈을 Enable 하기 위해 DISR bit를 0으로 설정한다.
- ✓ GTM 모듈이 Enable 되어 있는지 확인하기 위해 DISS bit가 0인지 확인한다.

GTM_CLC Register 주소: F019_FD00h (F0100000h + 9FD00h)

GTM_CLC Register 구조:

Table 25-63 Registers Address Space

Module	Base Address	End Address	Note
GTM	F010 0000 _H	F019 FFFF _H	
CLC			
Clock Control Register	(9FD00 _H)		Reset Value: 0000 0003 _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	EDIS DIS S DIS R		
	r r r rw		

Field	Bits	Type	Description
DISR	0	rw	Module Disable Request Bit Used for enable/disable control of the GTM module. 0 _B No disable requested 1 _B Disable requested
DISS	1	r	Module Disable Status Bit Bit indicates the current status of the GTM module. 0 _B GTM module is enabled 1 _B GTM module is disabled
EDIS	3	rw	Sleep Mode Enable Control Used for module sleep mode control.
0	2, [31:4]	r	Reserved Read as 0; should be written with 0.

PWM Example

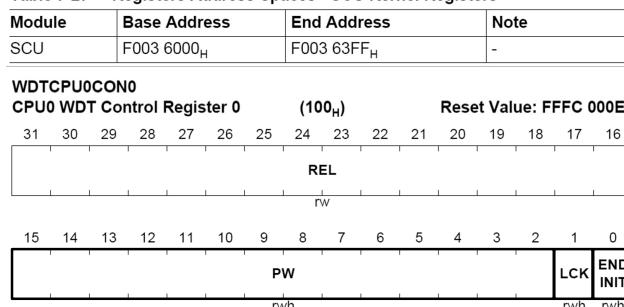
2. Data sheet 분석 : System Critical Register 설정 (1)

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

SCU_WDTCPUOCONO Register 주소: F003_6100h
(F0036000h + 100h)

SCU_WDTCPUOCONO Register 구조:

Table 7-27 Registers Address Spaces - SCU Kernel Registers



ACE Lab.

11/39

PWM 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_WDTCPUOCONO Register의 Lock 상태를 나타내며 해당 Register의 Lock 상태는 Password Access를 통해 Unlock 되고, Modify Access를 통해 Lock 된다.
- ✓ **PW bits**는 SCU_WDTCPUOCONO 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 00000000111100 _b A-step silicon: Bits [7:2] must be written with 111100 _b during Password Access and Modify Access. Read returns 00001100 _b for these bits.
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

12/39

PWM Example

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

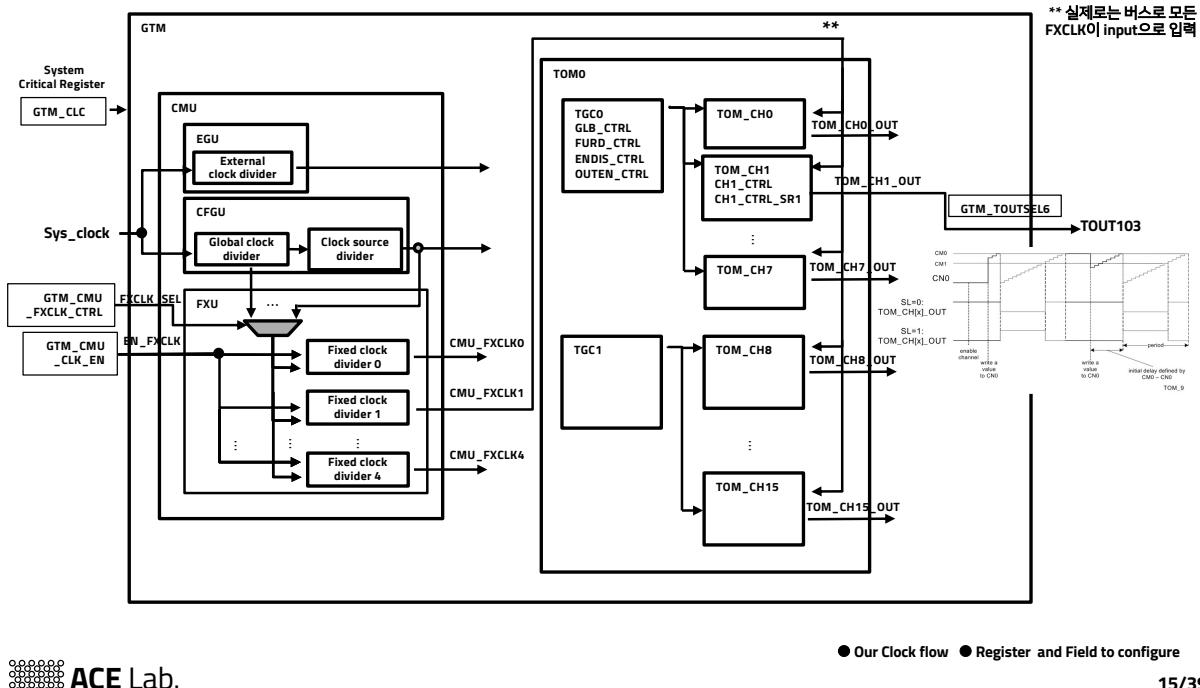
- ✓ SCU_WDTCPUOCONO Register에 적절한 값을 Write하여 **Password Access**를 수행한다.
- ✓ **Password Access**는 **SCU_WDTCPUOCONO Register**의 Lock 상태를 해제하며 과정은 다음과 같다.
 1. SCU_WDTCPUOCONO Register의 값을 읽어 REL bits, PW bits를 파악한다.
 2. Bits[7:2] (PW bits의 일부)가 반전되어 읽히기 때문에 이를 반전시켜 정확한 PW bits를 얻는다.
 3. Write 할 값의 bits[31:16]은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

PWM Example

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

- ✓ SCU_WDTCPUOCONO Register에 적절한 값을 Write하여 **Modify Access**를 수행한다.
- ✓ **Modify Access**는 **System ENDINIT**을 설정/해제하며 과정은 다음과 같다.
 1. SCU_WDTCPUOCONO 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_WDTCPUOCONO Register에 한번에 쓴다.
 6. SCU_WDTCPUOCONO 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.

GTM Overview



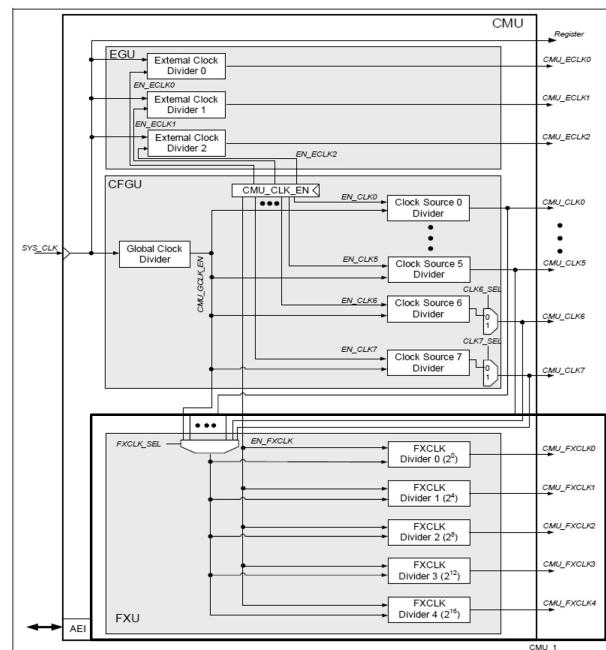
ACE Lab.

15/39

PWM Example

2. Data sheet 분석 : GTM 내부 Clock 설정 (1)

- ✓ GTM 모듈은 내부에 CMU (Clock Management Unit)을 포함하고 있다.
- ✓ CMU는 GTM 입력 클럭을 분주하여 다양한 내부 클럭을 생성하고, GTM 내부의 하위 모듈에 공급한다.
- ✓ 본 실습에서 PWM 신호 생성을 위해 사용할 하위 모듈인 **TOM (Timer Output Module)**은 CMU_FXCLK에 따라 동작한다.
- ✓ 따라서, CMU의 FXU에 대한 설정을 해야 한다.



ACE Lab.

16/39

PWM Example

2. Data sheet 분석 : GTM 내부 Clock 설정 (2)

- ✓ GTM_CMU_FXCLK_CTRL Register는 CMU_FXCLK의 소스 클럭을 설정한다.
- ✓ CMU_FXCLK의 소스 클럭으로 GTM 모듈의 입력 클럭인 CMU_GCLK_EN 또는 GTM 모듈 내부에서 생성된 CMU_CLKx가 사용될 수 있다.
- ✓ 소스 클럭을 CMU_GCLK_EN으로 설정하기 위해 **FXCLK_SEL bits**를 **0000b**로 설정한다.

GTM_CMU_FXCLK_CTRL Register 주소: F010_0344h
(F0100000h + 344h)

GTM_CMU_FXCLK_CTRL Register 구조:

Table 25-63 Registers Address Space			
Module	Base Address	End Address	Note
GTM			
GTM_CMU_FXCLK_CTRL CMU FXCLK Control Register (00344H) Reset Value: 00000000H			
31	30	29	28
27	26	25	24
23	22	21	20
19	18	17	16
Reserved			
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	0
Reserved			
			rw
			rw

Field	Bits	Type	Description
FXCLK_SEL	[3:0]	rw	Input clock selection for EN_FXCLK line 0000 _b : CMU_GCLK_EN selected 0001 _b : CMU_CLK0 selected 0010 _b : CMU_CLK1 selected 0011 _b : CMU_CLK2 selected 0100 _b : CMU_CLK3 selected 0101 _b : CMU_CLK4 selected 0110 _b : CMU_CLK5 selected 0111 _b : CMU_CLK6 selected 1000 _b : CMU_CLK7 selected <i>Note: This value can only be written, when the CMU_FXCLK generation is disabled. See bits 23..22 in register CMU_CLK_EN.</i> <i>Note: Other values for FXCLK_SEL are reserved and should not be used, but they behave like FXCLK_SEL = 0.</i>

PWM Example

2. Data sheet 분석 : GTM 내부 Clock 설정 (3)

- ✓ GTM_CMU_CLK_EN Register는 CMU 내부의 클럭에 대한 Enable 설정을 한다.
- ✓ GTM_CMU_CLK_EN Register는 CMU 내부에서 생성된 다양한 클럭에 대한 Enable을 설정할 수 있다.
- ✓ CMU_FXCLK을 Enable 하기 위해 **EN_FXCLK bits**를 **10b**로 설정한다.

GTM_CMU_CLK_EN Register 주소: F010_0300h
(F0100000h + 300h)

GTM_CMU_CLK_EN Register 구조:

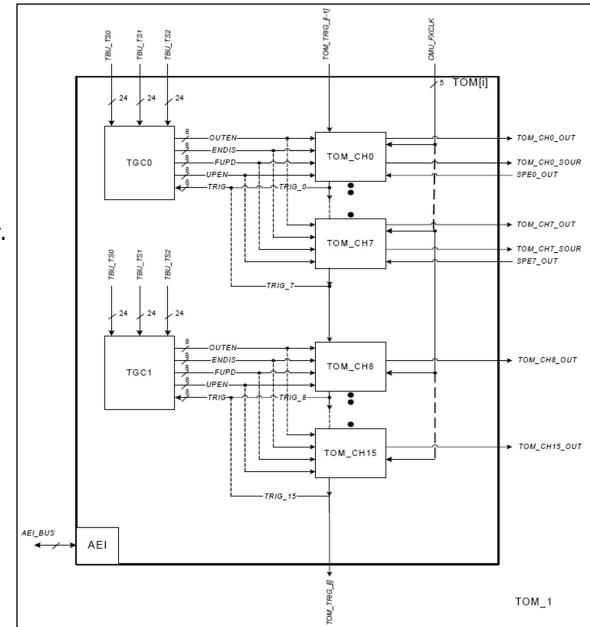
Table 25-63 Registers Address Space			
Module	Base Address	End Address	Note
GTM			
GTM_CMU_CLK_EN CMU Clock Enable Register (00300H) Reset Value: 00000000H			
31	30	29	28
27	26	25	24
23	22	21	20
19	18	17	16
Reserved			
		EN_FXCLK	EN_ECLK_K
		rw	rw
		EN_ECLK_2	EN_ECLK_1
		rw	rw
		EN_ECLK_0	0
		rw	rw
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	0
EN_CLK7	EN_CLK6	EN_CLK5	EN_CLK4
EN_CLK3	EN_CLK2	EN_CLK1	EN_CLK0
rw	rw	rw	rw
rw	rw	rw	rw
rw	rw	rw	rw

Field	Bits	Type	Description
EN_CLK4	[9:8]	rw	Enable clock source 4 see bits [1:0]
EN_CLK5	[11:10]	rw	Enable clock source 5 see bits [1:0]
EN_CLK6	[13:12]	rw	Enable clock source 6 see bits [1:0]
EN_CLK7	[15:14]	rw	Enable clock source 7 see bits [1:0]
EN_ECLK0	[17:16]	rw	Enable ECLK 0 generation subunit see bits [1:0]
EN_ECLK1	[19:18]	rw	Enable ECLK 1 generation subunit see bits [1:0]
EN_ECLK2	[21:20]	rw	Enable ECLK 2 generation subunit see bits [1:0]
EN_FXCLK	[23:22]	rw	Enable all CMU_FXCLK see bits [1:0]
			<i>Note: An enable value of 00_b clock source is disabled (ignore write access). 01_b disable clock signal and reset internal states 10_b enable clock signal 11_b clock signal enabled (ignore write access)</i>

PWM Example

2. Data sheet 분석 : TOM 구조 분석

- ✓ PWM 신호 생성을 위해 GTM 모듈 내부의 TOM을 사용한다.
- ✓ GTM 모듈은 3개의 TOM을 포함하고 있고, 각 TOM은 2개의 TGC (TOM Global Channel Control)와 16개의 TOM Channel을 가지고 있다.
- ✓ **TGC**는 8개의 TOM Channel과 연결되어 있으며 이를 통해 **TOM Channel을 제어할 수 있다.**
- ✓ **TOM Channel**은 TGC의 제어에 따라 동작을 수행하며 **출력 신호를 생성한다.**
- ✓ 본 실습에서는 **TOMO_CH1**를 사용한다. (TOUT103과 연결되어 있기 때문이다.)
- ✓ 따라서, **TOMO_CH1**를 사용하기 위한 설정을 수행한다.



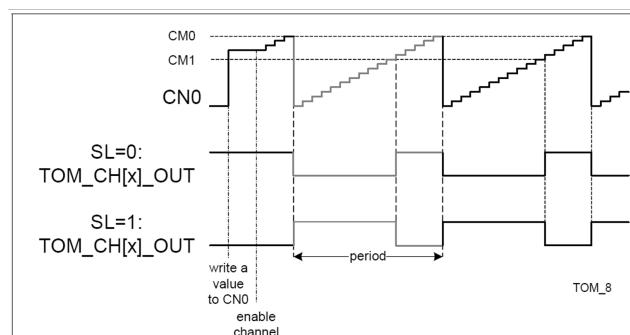
ACE Lab.

19/39

PWM Example

2. Data sheet 분석 : TOM 동작 분석

- ✓ TOM Channel은 **CNO / CMO / CM1**을 사용해 **PWM 신호를 생성한다.**
 - ✓ CNO : 동작 클럭에 따라 증가하는 Count 값을 저장한다.
 - ✓ CMO : PWM 신호의 주기를 결정하는 값을 저장한다.
 - ✓ CM1 : PWM 신호의 Duty Ratio를 결정하는 값을 저장한다.
- ✓ CNO는 동작 클럭에 따라 1씩 증가하며 CM0에 도달하면 0으로 초기화된다.
- ✓ CNO가 CM0에 도달했을 때, 출력 신호는 SL 값으로 설정된다.
- ✓ CNO가 CM1에 도달했을 때, 출력 신호는 SL 반전 값으로 설정된다.



ACE Lab.

Figure 25-39 PWM Output with respect to configuration bit SL in continuous mode

20/39

PWM Example

2. Data sheet 분석 : TOM0 – TGC0 설정 (1)

- ✓ GTM_TOM0_TGC0_GLB_CTRL Register는 Channel 0-7을 제어하는 TGC0에 대한 설정을 한다.
- ✓ Channel에 대한 Enable/Disable 설정 및 Output Enable 설정은 트리거 신호에 의해 일괄적으로 반영된다.
- ✓ **HOST_TRIGGER bit**를 1로 설정하여 사용자가 소프트웨어적으로 트리거 신호를 발생시킬 수 있다.

GTM_TOM0_TGC0_GLB_CTRL Register 주소: F010_8030h
(F0100000h + 8030h)

GTM_TOM0_TGC0_GLB_CTRL Register 구조:

Table 25-63 Registers Address Space

Module	Base Address	End Address	Note
GTM	F010 0000 _H	F019 FFFF _H	

GTM_TOM0_TGC0_GLB_CTRL (i=0-2)
TOMi TGC0 Global Control Register(08030_H+i*800_H) Reset Value: 00000000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPEN_CT RL7	UPEN_CT RL6	UPEN_CT RL5	UPEN_CT RL4	UPEN_CT RL3	UPEN_CT RL2	UPEN_CT RL1	UPEN_CT RL0								
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST -CH 7	RST -CH 6	RST -CH 5	RST -CH 4	RST -CH 3	RST -CH 2	RST -CH 1	RST -CH 0								
w	w	w	w	w	w	w	w								

Field	Bits	Type	Description
HOST_TRIGGER	0	w	Trigger request signal (see TGC0, TGC1) to update the register ENDIS_STAT and OUTEN_STAT 0 _B no trigger request 1 _B set trigger request Read as 0. Note: This flag is cleared automatically after triggering the update
Reserved	[7:1]	r	Reserved Read as zero, should be written as zero
RST_CH0	8	w	Software reset of channel 0 0 _B No action 1 _B Reset channel Read as 0. Note: This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-r FlipFlop SOUR is set to '1'.

21/39

PWM Example

2. Data sheet 분석 : TOM0 – TGC0 설정 (2)

- ✓ TOM 동작을 위한 CM0 / CM1 / CLK_SRC 값은 먼저 Shadow Register에 저장된다.
- ✓ 업데이트가 Enable 되어 있으면 업데이트를 할 때 Shadow Register에 저장되어 있는 값이 일괄적으로 반영되어 CM0 / CM1 / CLK_SRC가 설정된다.
- ✓ TOM Channel 1이 동작하기 위해서는 해당 Channel에 대한 CM0 / CM1 / CLK_SRC 값이 설정되어야 하며 이를 위해 **UPEN_CTRL1 bits**를 10b로 설정하여 업데이트를 Enable 한다.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPEN_CT RL7	UPEN_CT RL6	UPEN_CT RL5	UPEN_CT RL4	UPEN_CT RL3	UPEN_CT RL2	UPEN_CT RL1	UPEN_CT RL0								
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST -CH 7	RST -CH 6	RST -CH 5	RST -CH 4	RST -CH 3	RST -CH 2	RST -CH 1	RST -CH 0								
w	w	w	w	w	w	w	w								

UPEN_CT RL1	[19:18]	rw	TOM channel 1 enable update of register CM0, CM1 and CLK_SRC
See bits 17:			
Write of following double bit values is possible:			
00 _B don't care, bits 1:0 will not be changed			
01 _B update disabled: is read as 00 (see below)			
10 _B update enabled: is read as 11 (see below)			
11 _B don't care, bits 1:0 will not be changed			
Read of following double values means:			
00 _B channel disabled			
11 _B channel enabled			

PWM Example

2. Data sheet 분석 : TOM0 – TGCO 설정 (5)

- ✓ GTM_TOM0_TGCO_OUTEN_CTRL Register는 트리거 신호에 따른 Output Enable을 설정한다.
- ✓ 트리거 신호에 따라 각 Channel의 Output을 Enable 할지 Disable 할지 설정할 수 있다.
- ✓ 트리거 신호 발생 시, Channel 1의 Output0|Enable 되게 OUTEN_CTRL1 bits를 10b로 설정한다.

GTM_TOM0_TGCO_OUTEN_CTRL Register 주소: F010_8078h
(F0100000h + 8078h)

Field	Bits	Type	Description
OUTEN_C	[1:0]	rw	Output (A)TOM_OUT(0) enable/disable update value Write of following double bit values is possible: 00 _B don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger 01 _B disable channel output on an update trigger 10 _B enable channel output on an update trigger 11 _B don't change bits 1:0 of this register Note: if the channel is disabled (ENDS[0]=0) or the output is disabled (OUTEN[0]=0), the TOM channel 0 output TOM_OUT[0] is the inverted value of bit SL.
TRL0			
OUTEN_C	[3:2]	rw	Output (A)TOM_OUT(1)enable/disable update value See bits 1:0
TRL1			
OUTEN_C	[5:4]	rw	Output (A)TOM_OUT(2) enable/disable update value See bits 1:0
TRL2			
OUTEN_C	[7:6]	rw	Output (A)TOM_OUT(3) enable/disable update value See bits 1:0
TRL3			
OUTEN_C	[9:8]	rw	Output (A)TOM_OUT(4) enable/disable update value See bits 1:0
TRL4			
OUTEN_C	[11:10]	rw	Output (A)TOM_OUT(5) enable/disable update value See bits 1:0
TRL5			

GTM_TOM0_TGCO_OUTEN_CTRL Register 구조:

Table 25-63 Registers Address Space

Module	Base Address	End Address	Note
GTM	F010 0000 _H	F019 FFFF _H	

GTM_TOMi_TGCO_OUTEN_CTRL (i=0-2)
TOMi TGCO Output Enable Control Register
(08078_H+i*800_H) Reset Value: 00000000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTEN_C TRL7	OUTEN_C TRL6	OUTEN_C TRL5	OUTEN_C TRL4	OUTEN_C TRL3	OUTEN_C TRL2	OUTEN_C TRL1	OUTEN_C TRL0								
rw								rw							

PWM Example

2. Data sheet 분석 : TOM0 – Channel 1 설정 (1)

- ✓ GTM_TOM0_CHx_CTRL Register는 TOM0의 각 Channel에 대한 동작 설정을 한다.
- ✓ TOM Channel 1의 동작을 설정하기 위해 GTM_TOM0_CH1_CTRL Register를 설정한다.
- ✓ 출력 신호의 Duty Cycle에 대한 Signal level을 High로 설정하기 위해 SL bit를 1로 설정한다.

GTM_TOM0_CH1_CTRL Register 주소: F010_8040h
(F0100000h + 8040h)

GTM_TOM0_CH1_CTRL Register 구조:

Table 25-63 Registers Address Space

Module	Base Address	End Address	Note
GTM	F010 0000 _H	F019 FFFF _H	

GTM_TOM0_CHx_CTRL (x=0-14)
TOM0 Channel x Control Register
(08000_H+x*0040_H) Reset Value: 00000800_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	GCM	SPE M	Rese rved	OSM	Rese rved	TRIG OUT	Reserved	RST CC U0	Reserved						
r	rw	rw	r	rw	r	rw	r	rw	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rese rved	CLK_SRC_SR	SL					Reserved								
r	rw	rw					r								

Field Bits Type Description

SL	11	rw	Signal level for duty cycle 0 _B Low signal level 1 _B High signal level If the output is disabled, the output TOM_OUT[x] is set to inverse value of SL.
----	----	----	---

PWM Example

2. Data sheet 분석 : TOMO – Channel 1 설정 (2)

- ✓ TOM Channel 1의 동작 클럭을 CMU_FXCLK1로 설정하기 위해 **CLK_SRC_SR bits**를 **001b**로 설정한다.
 - ✓ CMU_FXCLK1의 주파수는 $100\text{MHz} / 16 = 6,250\text{kHz}$ 이다.
 - ✓ CLK_SRC_SR bits가 업데이트를 할 때 반영되기 때문에 TOM Channel 1의 동작 클럭 또한 업데이트를 할 때 반영된다.

GTM_TOM0_CHx_CTRL (x=0-14) TOM0 Channel x Control Register'															
(08000H+x*0040H)															
Reset Value: 00000800H															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	GCM	SPE M	Rese rvred	OSM	Rese rvred	TRIG OUT		Reserved	RST CC U0		Reserved				
r	fW	fW	r	fW	r	fW	r	fW	r	fW	r		r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rese rvred	CLK	SR	SR	SL							Reserved				
r	fW	fW									r				

CLK_SRC_SR	[14:12]	rw	<p>Clock source select for channel</p> <p>The register CLK_SRC is updated with the value of CLK_SRC_SR together with the update of register CM0 and CM1.</p> <p>The input of the FX clock divider depends on the value of FXCLK_SEL (see CMU).</p> <ul style="list-style-type: none"> 000_B CMU_FXCLK(0) selected: clock selected by FXCLKSEL 001_B CMU_FXCLK(1) selected: clock selected by FXCLKSEL / 2⁴ 010_B CMU_FXCLK(2) selected: clock selected by FXCLKSEL / 2⁸ 011_B CMU_FXCLK(3) selected: clock selected by FXCLKSEL / 2¹² 100_B CMU_FXCLK(4) selected: clock selected by FXCLKSEL / 2¹⁶ 101_B no CMU_FXCLK selected, clock of channel stopped 110_B no CMU_FXCLK selected, clock of channel stopped 111_B no CMU_FXCLK selected, clock of channel stopped <p>Note: if clock of channel is stopped (i.e. CLK_SRC = 101/110/111), the channel can only be restarted by resetting CLK_SRC_SR to a value of 000 to 100 and forcing an update via the force update mechanism.</p>
-------------------	---------	----	---



PWM Example

2. Data sheet 분석 : TOMO – Channel 1 설정 (3)

- ✓ GTM_TOMO_CHx_SRO Register는 CMO에 대한 Shadow Register이다.
 - ✓ TOM Channel 1의 CMO를 설정하기 위해 **GTM_TOMO_CH1_SRO Register**를 설정한다.
 - ✓ GTM_TOMO_CH1_SRO Register에 설정할 CMO 값을 저장하면 업데이트를 할 때 CMO에 반영된다.
 - ✓ 본 실습에서는 PWM 신호의 주기를 2ms로 설정하기 위해 해당 Register의 값을 **(12500 – 1)**로 설정한다.

GTM_TOM0_CH1_SR0 Register 주소: F010_8044h
(F0100000h + 8044h)

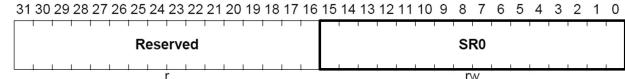
GTM_TOMO_CH1_SRO Register 구조:

Registers Address Space			
Module	Base Address	End Address	Note
GTM	E010_0000	E019_FFFF	

GTM_TOM0_CHx_SR0 (x=0-15)
TOM0 Channel x CCU0 Compare Shadow Register
(0x0004_tx*0040_) Reset Value: 00000000

$$(Period\ of\ PWM) = \frac{(Value\ of\ CM0) + 1}{(Freq.\ of\ CMU_FXCLK1)}$$

$$= \frac{12500}{6250\ kHz} = 0.002s$$



PWM Example

2. Data sheet 분석 : TOM0 – Channel 1 설정 (4)

- ✓ GTM_TOM0_CHx_SR1 Register는 CM1에 대한 Shadow Register이다.
- ✓ TOM Channel 1의 CM1을 설정하기 위해 **GTM_TOM0_CH1_SR1 Register**를 설정한다.
- ✓ GTM_TOM0_CH1_SR1 Register에 설정할 CM1 값을 저장하면 업데이트를 할 때 CM1에 반영된다.
- ✓ CM1에 의한 **Duty Ratio**는 $\left(\frac{CM1+1}{CM0+1} \times 100 \right) (\%)$ 이다.

GTM_TOM0_CH1_SR1 Register 주소: F010_8048h
(F0100000h + 8048h)

GTM_TOM0_CH1_SR1 Register 구조:

Table 25-63 Registers Address Space

Module	Base Address	End Address	Note
GTM	F010 0000 _H	F019 FFFF _H	

GTM_TOM0_CHx_SR1 (x=0-15)
TOM0 Channel x CCU1 Compare Shadow Register
(08008_H+x*0040_H) Reset Value: 00000000_H

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved SR1

r rw

PWM Example

2. Data sheet 분석 : TOUT 설정 (1)

- ✓ GTM 모듈 내 하위 모듈에서 생성한 출력 신호를 외부에 전달하기 위해서는 GTM 모듈의 출력 포트 (TOUT)와 연결 설정을 해야 한다.
- ✓ 하나의 출력 포트에는 하위 모듈에서 생성된 출력 신호 4개가 MUX를 통해 연결되어 있으며 MUX 제어를 통해 하나의 신호가 출력 포트와 연결된다.
- ✓ **GTM_TOUTSEL Register**는 MUX에 제어 신호를 입력하며 하나의 Register가 16개의 MUX를 제어한다.
- ✓ 따라서, LED가 연결된 TOUT103 (PORT10 Pin 1)은 **GTM_TOUTSEL6 Register**의 **SEL7 bits**를 통해 설정할 수 있다.

PWM Example

2. Data sheet 분석 : TOUT 설정 (2)

- ✓ GTM_TOUTSEL Register는 TOUT을 통해 출력될 신호를 설정한다.
- ✓ TOUT103에 대해 설정하기 위해 **GTM_TOUTSEL6 Register**의 **SEL7 bits**를 설정한다.
- ✓ TOM0 Channel 1을 통해 생성한 PWM 신호를 TOUT103로 출력하기 위해 **SEL7 bits**를 **00b**로 설정한다.

GTM_TOUTSEL6 Register 주소: F019_FD48h
(F0100000H + 9FD48H)

GTM_TOUTSEL6 Register 구조:

Table 25-63 Registers Address Space

Module	Base Address	End Address	Note
GTM	F010 0000H	F019 FFFFH	

TOUTSELn (n = 0-14)
Timer Output Select Register (9FD30H+n*4H) **Reset Value: 0000 0000H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15	SEL14	SEL13	SEL12	SEL11	SEL10	SEL9	SEL8								
RW	RW	RW	RW	RW	RW	RW	RW								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0								
RW	RW	RW	RW	RW	RW	RW	RW								



Field	Bits	Type	Description
SELx (x = 0-15)	[x*2+1: x*2]	rw	TOUT(n*16+x) Output Selection This bit defines which timer out is connected as TOUT(n*16+x). The mapping for each pin is defined by Table 25-67 / Table 25-68 . 00 _B Timer A form Table 25-67 / Table 25-68 is connected as TOUT(n*16+x) to the ports 01 _B Timer B form Table 25-67 / Table 25-68 is connected as TOUT(n*16+x) to the ports 10 _B Timer C form Table 25-67 / Table 25-68 is connected as TOUT(n*16+x) to the ports 11 _B Timer D form Table 25-67 / Table 25-68 is connected as TOUT(n*16+x) to the ports <small>Note: If TOUT(n*16+x) is not defined in Table 25-67/Table 25-68 this bit field has to be treated as reserved.</small>

Table 25-67 GTM to Port Mapping for QFP-176

UM 25-757

Port	Input	Output	Input Timer Mapped		Output Timer Mapped			
			A	B	A	B	C	D
P10.1	TIN103	TOUT103	TIM0_1	TIM1_1	TOM0_1	TOM2_9	ATOM1_1	ATOM4_1

PWM Example

3. 프로그래밍

- 1) LED가 연결된 PORT에 대한 설정을 수행하는 함수를 구현한다.

```

74 // PORT10 Registers
75 #define PORT10_BASE      (0xF003B000)
76 #define PORT10_IOCR0     (*volatile unsigned int*)(PORT10_BASE + 0x10)
77
78 #define PC1           11
    
```

PORT IO 설정관련 레지스터 주소 및 비트 필드 정의

```

109 void init_LED(void)
110 {
111     PORT10_IOCR0 &= ~((0x1F) << PC1);           // PORT10.1 : Alternate output function 1 (push-pull)
112     PORT10_IOCR0 |= ((0x11) << PC1);           // PORT10.1 : GTM_TOUT103
113 }
114 }
    
```

PORT IO 설정 함수



PWM Example

3. 프로그래밍

2) GTM을 설정하기 위한 함수를 구현한다.

- ① SCU_WDTCPU0CON0 Register를 통해 Password/Modify Access를 수행하여 System ENDINIT을 해제한다.
- ② GTM_CLC Register를 통해 GTM 모듈을 Enable 한다.
- ③ SCU_WDTCPU0CON0 Register를 통해 Password/Modify Access를 수행하여 System ENDINIT을 설정한다.
- ④ GTM_CMU_FXCLK_CTRL Register와 GTM_CMU_CLK_EN Register를 통해 CMU_FXCLK를 설정한다.
- ⑤ GTM_TOMO_TGCO_GLB_CTRL Register를 통해 CMO / CM1 / CLK_SRC에 대한 업데이트를 Enable 한다.
- ⑥ GTM_TOMO_TGCO_FUPD_CTRL Register를 통해 트리거 신호에 따른 동작 (Force update, Clear CNO)을 설정한다.
- ⑦ GTM_TOMO_TGCO_ENDIS_CTRL Register와 GTM_TOMO_TGCO_OUTEN_CTRL Register를 통해 트리거 신호에 따른 동작 (Channel enable, Output enable)을 설정한다.
- ⑧ GTM_TOMO_CH1_CTRL Register를 통해 Signal level을 설정한다.
- ⑨ GTM_TOMO_CH1_CTRL / GTM_TOMO_CH1_SR0 / GTM_TOMO_CH1_SR1 Register를 통해 CMO / CM1 / CLK_SRC에 대한 Shadow Register를 설정한다.
- ⑩ GTM_TOUTSEL6 Register를 통해 TOMO Channel 1의 PWM 신호가 TOUT103로 출력되도록 설정한다.
- ⑪ GTM_TOMO_TGCO_GLB_CTRL Register를 통해 모든 설정이 반영되도록 HOST TRIGGER를 발생시킨다.



33/39

PWM Example

3. 프로그래밍

2) GTM을 설정하기 위한 함수를 구현한다.

```
31/* Address of Registers */
32 // SCU Registers
33 #define SCU_BASE          (0xF0036000)
34 #define SCU_WDT_CPU0CON0 (*(volatile unsigned int*)(SCU_BASE + 0x100))
35
36 #define LCK               1
37 #define ENDINIT           0
38
39// // GTM Registers
40 // // CMU
41 #define GTM_BASE           (0xF1000000)
42 #define GTM_CMU_CLK_EN      (*(volatile unsigned int*)(GTM_BASE + 0x003000))
43 #define GTM_CMU_FXCLK_CTRL   (*(volatile unsigned int*)(GTM_BASE + 0x003400))
44
45 #define EN_FXCLK            22
46 #define FXCLK_SEL           0
47
48 // // GTM - TOM0
49 #define GTM_TOM0_TGCO_GLB_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x088030))
50 #define GTM_TOM0_TGCO_ENDIS_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x088070))
51 #define GTM_TOM0_TGCO_OUTEN_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x088078))
52 #define GTM_TOM0_TGCO_FUPD_CTRL   (*(volatile unsigned int*)(GTM_BASE + 0x088038))
53 #define GTM_TOM0_CH1_CTRL       (*(volatile unsigned int*)(GTM_BASE + 0x088040))
54 #define GTM_TOM0_CH1_SR0        (*(volatile unsigned int*)(GTM_BASE + 0x088044))
55 #define GTM_TOM0_CH1_SR1        (*(volatile unsigned int*)(GTM_BASE + 0x088048))
56
57 #define UPEN_CTRL1           18
58 #define HOST_TRIG             0
59 #define ENDIS_CTRL1           2
60 #define OUTEN_CTRL1           2
61 #define RSTCN0_CH1            18
62 #define FUPD_CTRL1            2
63 #define CLK_SRC_SR             12
64 #define SL                   11
65
66 // // GTM_CLC
67 #define GTM_CLC              (*(volatile unsigned int*)(GTM_BASE + 0x9FD000))
68 #define GTM_TOUTSEL6          (*(volatile unsigned int*)(GTM_BASE + 0x9FD48))
69
70 #define DISS                 1
71 #define DISR                 0
72 #define SEL7                  14
```



GTM 설정관련 레지스터 주소 및 비트 필드 정의

34/39

PWM Example

3. 프로그래밍

2) GTM을 설정하기 위한 함수를 구현한다.

```
116 void init_GTM_TOM0_PWM(void)
117 {
118     /* GTM Enable */
119     // Password Access to unlock WDTCPU0CON0
120     ① SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) & ~(1 << LCK)) | (1 << ENDINIT);
121     while((SCU_WDT_CPU0CON0 & (1 << LCK)) != 0);
122
123     // Modify Access to clear ENDINIT bit
124     SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) | (1 << LCK)) & ~(1 << ENDINIT);
125     while((SCU_WDT_CPU0CON0 & (1 << LCK)) == 0);
126
127     ② GTM_CLC &= ~(1 << DISR);           // Enable GTM Module
128
129     // Password Access to unlock WDTCPU0CON0
130     ③ SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) & ~(1 << LCK)) | (1 << ENDINIT);
131     while((SCU_WDT_CPU0CON0 & (1 << LCK)) != 0);
132
133     // Modify Access to set ENDINIT bit
134     SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) | (1 << LCK)) | (1 << ENDINIT);
135     while((SCU_WDT_CPU0CON0 & (1 << LCK)) == 0);
136
137     while((GTM_CLC & (1 << DISS)) != 0);      // Wait until module is enabled
138
139     /* GTM Clock Setting */
140     ④ GTM_CMU_FXCLK_CTRL &= ~((0xF) << FXCLK_SEL); // Input clock of CMU_FXCLK : CMU_GCLK_EN
141
142     GTM_CMU_CLK_EN &= ~((0x3) << EN_FXCLK);        // Enable all CMU_FXCLK
143     GTM_CMU_CLK_EN |= ((0x2) << EN_FXCLK);
```

ACE Lab.

GTM 설정 함수

35/39

PWM Example

3. 프로그래밍

2) GTM을 설정하기 위한 함수를 구현한다.

```
145     /* GTM TOM0 PWM Setting */
146     ⑤ GTM_TOM0_TGC0_GLB_CTRL &= ~((0x3) << UPEN_CTRL1); // TOM0 channel 1 enable update of
147     GTM_TOM0_TGC0_GLB_CTRL |= ((0x2) << UPEN_CTRL1); // register CM0, CM1, CLK_SRC
148
149     ⑥ GTM_TOM0_TGC0_FUPD_CTRL &= ~((0x3) << FUPD_CTRL1); // Enable force update of TOM0 channel 1
150     GTM_TOM0_TGC0_FUPD_CTRL |= ((0x2) << FUPD_CTRL1);
151     GTM_TOM0_TGC0_FUPD_CTRL &= ~((0x3) << RSTCN0_CH1); // Reset CN0 of TOM0 channel 1 on force update
152     GTM_TOM0_TGC0_FUPD_CTRL |= ((0x2) << RSTCN0_CH1);
153
154     ⑦ GTM_TOM0_TGC0_ENDIS_CTRL &= ~((0x3) << ENDIS_CTRL1); // Enable channel 1 on an update trigger
155     GTM_TOM0_TGC0_ENDIS_CTRL |= ((0x2) << ENDIS_CTRL1);
156     GTM_TOM0_TGC0_OUTEN_CTRL &= ~((0x3) << OUTEN_CTRL1); // Enable channel 1 output on an update trigger
157     GTM_TOM0_TGC0_OUTEN_CTRL |= ((0x2) << OUTEN_CTRL1);
158
158     ⑧ GTM_TOM0_CH1_CTRL |= (1 << SL);           // High signal level for duty cycle
159
160     ⑨ GTM_TOM0_CH1_CTRL &= ~((0x7) << CLK_SRC_SR); // Clock source : CMU_FXCLK(1) = 6250 kHz
161     GTM_TOM0_CH1_CTRL |= (1 << CLK_SRC_SR);
162     GTM_TOM0_CH1_SR0 = 12500 - 1;                  // PWM freq. = 6250 kHz / 12500 = 500 Hz
163
164     // GTM_TOM0_CH1_SR1 = 0;                         // Duty cycle = 0
165     // GTM_TOM0_CH1_SR1 = 6250-1;                    // Duty cycle = 50
166     GTM_TOM0_CH1_SR1 = 12500 - 1;                  // Duty cycle = 100
167
168     ⑩ GTM_TOUTSEL6 &= ~((0x3) << SEL7);        // TOUT103 : TOM0 channel 1
169
170     ⑪ GTM_TOM0_TGC0_GLB_CTRL |= (1 << HOST_TRIG); // Trigger request signal to update
171 }
```

GTM 설정 함수

ACE Lab.

36/39

PWM Example

3. 프로그래밍

3) 동작에 따라 'main' 함수를 구현한다. (앞서 구현한 함수들을 호출한다.)

```
31/* Address of Registers */
32 // SCU Registers
33 #define SCU_BASE          (0xF0036000)
34 #define SCU_WDT_CPU0CON0 (*(volatile unsigned int*)(SCU_BASE + 0x100))
35
36 #define LCK           1
37 #define ENDINIT      0
38
39// GTM Registers
40 // GTM - CMU
41 #define GTM_BASE          (0xF0100000)
42 #define GTM_CHU_CLK_EN    (*(volatile unsigned int*)(GTM_BASE + 0x00300))
43 #define GTM_CHU_FXCLK_CTRL (*(volatile unsigned int*)(GTM_BASE + 0x00344))
44
45 #define EN_FXCLK        22
46 #define FXCLK_SEL       0
47
48 // GTM - T0M0
49 #define GTM_T0M0_TGC0_GLB_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x00830))
50 #define GTM_T0M0_TGC0_ENDIS_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x00870))
51 #define GTM_T0M0_TGC0_OUTEN_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x00878))
52 #define GTM_T0M0_TGC0_FUPD_CTRL  (*(volatile unsigned int*)(GTM_BASE + 0x00858))
53 #define GTM_T0M0_CH1_CTRL    (*(volatile unsigned int*)(GTM_BASE + 0x00840))
54 #define GTM_T0M0_CH1_SR0    (*(volatile unsigned int*)(GTM_BASE + 0x00844))
55 #define GTM_T0M0_CH1_SR1    (*(volatile unsigned int*)(GTM_BASE + 0x00848))
56
57 #define UPEN_CTRL1      18
58 #define HOST_TRIG      0
59 #define ENDIS_CTRL1    2
60 #define OUTEN_CTRL1   2
61 #define RSTCN0_CH1     18
62 #define FUPD_CTRL1    2
63 #define CLK_SRC_SR     12
64 #define SL             11
65
66 // GTM
67 #define GTM_CLC        (*(volatile unsigned int*)(GTM_BASE + 0x9FD00))
68 #define GTM_TOUTSEL6  (*(volatile unsigned int*)(GTM_BASE + 0x9FD48))
69
70 #define DISS           1
71 #define DISR           0
72 #define SEL7          14
73
74 // PORT10 Registers
75 #define PORT10_BASE     (0xF003B000)
76 #define PORT10_ICR0    (*(volatile unsigned int*)(PORT10_BASE + 0x10))
77
78 #define PC1            11
79
80 /* Function Prototype */
81 void init_LED(void);
82 void init_GTM_T0M0_PWM(void);
83
84 IfxCpu_syncEvent g_cpuSyncEvent = 0;
85
86 int core0_main(void)
87 {
88     IfxCpu_enableInterrupts();
89
90     /* !WATCHDOG AND SAFETY WATCHDOG ARE DISABLED HERE!!
91      * Enable the watchdogs and service them periodically if it is required
92      */
93     IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
94     IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
95
96     /* Wait for CPU sync event */
97     IfxCpu_emitEvent(&g_cpuSyncEvent);
98     IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
99
100    init_LED();
101    init_GTM_T0M0_PWM();
102
103    while(1)
104    {
105    }
106
107    return (1);
108 }
```

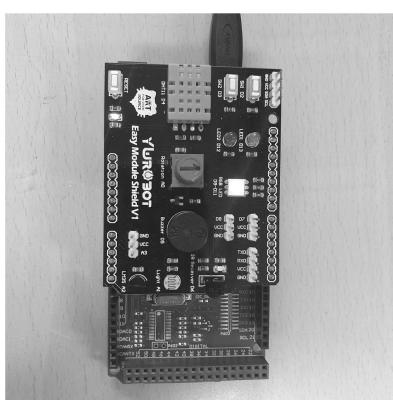
ACE Lab.

37/39

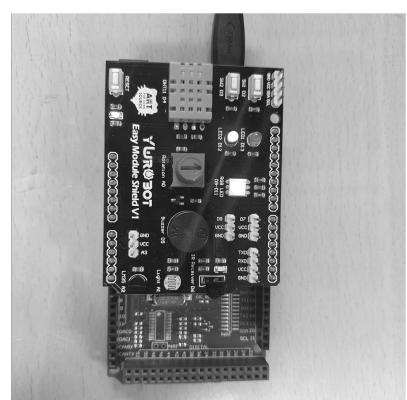
PWM Example

4. 동작 확인

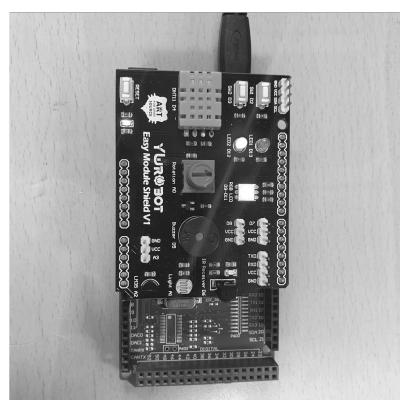
- ✓ Build 및 Debug 후 ('Resume' 버튼 클릭), CM1 값을 바꿔보며 Duty Ratio에 따른 LED 밝기를 확인한다.



Duty = 0%



Duty = 50%



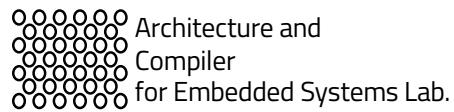
Duty = 100%

ACE Lab.

38/39

Q & A

Thank you for your attention



School of Electronics Engineering, KNU

ACE Lab (hn02301@gmail.com)