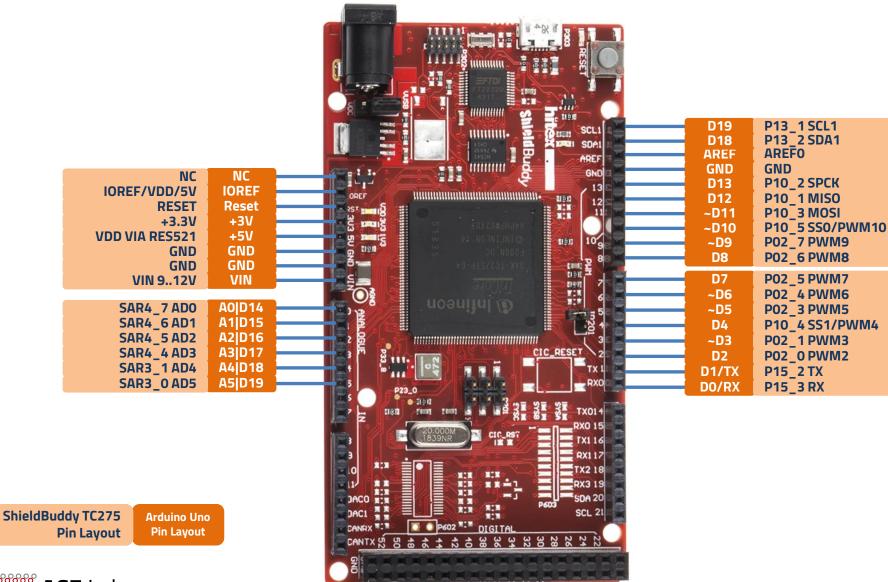
Infineon TC275 PWM (Pulse Width Modulation)

Hyeongrae Kim

Architecture and Compiler for Embedded System LAB.
School of Electronics Engineering, KNU, KOREA
2021-05-11



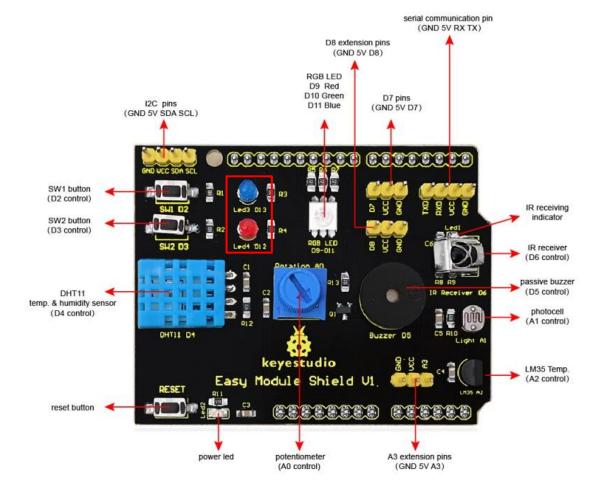
Hitex ShieldBuddy TC275



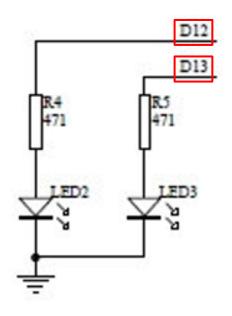


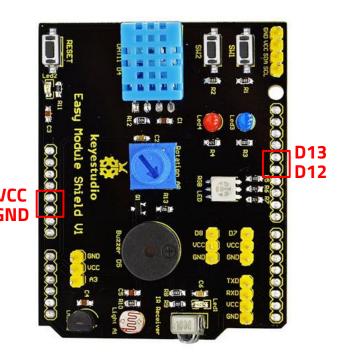
- PWM Duty Ratio에 따른 LED 밝기 변화
 - 1. 새로운 예제를 위한 프로젝트를 생성한다.
 - 2. 원하는 동작을 위해 레지스터와 메모리에 직접 접근해서 값을 써야한다.
 - 3. Board Schematic과 Datasheet를 통해 PWM 신호 출력에 대한 정보를 파악한다.
 - 4. PWM 신호 생성을 위해 사용할 GTM 모듈의 동작 원리를 파악하고 메모리 맵을 분석한다.
 - 5. 분석 결과를 활용해 임베디드 프로그래밍을 한다.

- 1. LED 연결 정보 파악
 - ✓ 여러 LED를 사용하기 위해 Target Board가 아닌 Easy Module Shield V1 확장 보드의 LED를 사용한다.

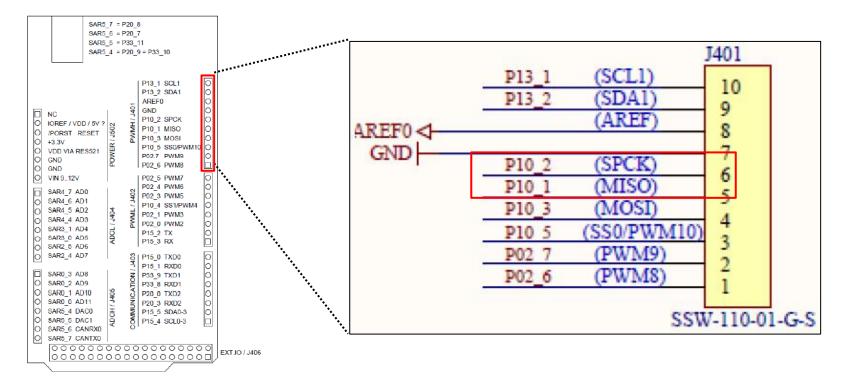


- 1. LED 연결 정보 파악
 - ✓ LED는 Easy Module Shield V1 확장 보드의 Pin D12(RED)/D13(BLUE)과 연결되어 있다.
 - ✓ 타겟 보드는 Easy Module Shield V1 확장 보드의 Pin D12/D13을 통해 LED 출력을 보낼 수 있다.





- 1. LED 연결 정보 파악
 - ✓ TC275 보드의 Schematic과 Datasheet를 확인했을 때, Easy Module Shield V1 확장 보드의 **Pin D12/D13**과 연결되는 IO는 PORT10**의 Pin 1-2**다.
 - ✔ 해당 Pin의 출력이 High-level 일 때 LED는 켜지고, Low-level 일 때 LED는 꺼진다.





- 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.

P10.2

P10 IN.P2

TIN104

SCLK1A

REQ2

SDI01

T6INB RXDCAN2E

P10 OUT.P2

TOUT104

SCLK1

General-purpose input

GTM input

SCU input

MSC0 input

GTM output

QSPI1 output

Reserved

GPT120 input

CAN node 2 input

General-purpose output

QSPI1 input

P10 IOCR0.

PC2

0XXXX_R

1X000_B

1X001_B

1X010_B

1X011_B

TOM2_10 ATOM ATOM

TOM0_2

TIN104 TOUT104 TIM0_2 TIM1_2

✓ PWM 신호를 통해 LED 밝기를 제어하기 위해 해당 Pin을 GTM 모듈의 TOUT103 (01)으로 설정해야 한다.

Pin	Symbol	Ctrl	Type	Function									
169	P10.1	1	MP+/	General-purpose input									
	TIN103		PU1/	GTM input									
	MRST1A		VEXT	QSPI1 input GPT120 input									
	T5EUDB												
	P10.1	00	1	General-purpose output									
	TOUT103	01		GTM output									
	MTSR1	02		QSPI1 output	Table Port	25-67 G1	M to Port I	Mapping Input Ti			mer Mapped		
	MRST1	O3	1	QSPI1 output		Mapped A B			A B C D				
	EN01	04	1	MSC0 output	P02.2			_	_	_	TOM1_10 TOM1_11	0_2	1_2
	VADCG6BFL1	O5	1	VADC output	P02.4		TOUT4	TIM0_4	TIM1_4	TOM0_12	TOM1_12	0_3 ATOM 0_4	1_3 ATOM 1_4
	END03	O6	1	MSC0 output							TOM1_14	0_5 ATOM	1_5
		07	1	Reserved					_	_	TOM1_15	0_7	1_7
 . CE La	h		1			TIN102	TOUT102	TIM0_4	TIM1_4	TOM0_4	TOM2_12	0_0 ATOM 1_4	1_0 ATOM 4_4
	17.				P10.1	1 TIN103	TOUT103	TIMO 1	TIM1 1	TOM0 1	TOM2 9	ATOM	ATOM



- 2. Data sheet 분석: PORT 설정 (1)
 - ✔ P10_IOCR Register는 PORT10의 Input/Output을 설정한다.
 - ✓ LED가 PORT10의 Pin 1에 연결되어 있기 때문에 P10_IOCRO Register의 PC1 bits를 설정한다.

Table 13-3 Registers Address Space

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

P10_IOCR0 Register 주소: F003_B010h (F003B000h + 10h) P10_IOCR0 Register 구조:

Pn_IOCR0 (n=10-11)
Port n Input/Output Control Register 0

Port	n inp	ut/Ot	ıtput	Cont	roi Re	egiste	er u								
					((F003	A610) _н + r	1*100 ₁	н)	Res	et Va	ilue:	1010	1010 _H
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		PC3				0				PC2				0	
		rw	-			r	-	-		rw		-		r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		PC1				0				PC0				0	
	•	гw				r	-	-	-	rw		-	-	г	

Field	Bits	Туре	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 Table 13-5).
0	[2:0], [10:8], [18:16], [26:24]	r	Reserved Read as 0; should be written with 0.

2. Data sheet 분석 : PORT 설정 (2)

Table 13-5

11110_B

11111_B

✓ PORT10의 Pin 1을 GTM 모듈의 TOUT103 (O1)으로 설정하기 위해 PC1 bits를 10001b로 설정한다.

PCx Coding Selected Pull-up / Pull-down / PCx[4:0] I/O Characteristics **Selected Output Function** Push-pull 10000_R Output General-purpose output 10001_B Alternate output function 1 10010_B Alternate output function 2 10011_B Alternate output function 3 10100_R Alternate output function 4 10101_B Alternate output function 5 10110_B Alternate output function 6 10111_B Alternate output function 7 $11000_{\rm B}$ Open-drain General-purpose output 11001_R Alternate output function 1 11010_{R} Alternate output function 2 11011_B Alternate output function 3 11100_{R} Alternate output function 4 11101_B Alternate output function 5

Alternate output function 6

Alternate output function 7

- 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 O인지 확인한다.

GTM_CLC Register 주소: F019_FD00h (F0100000h + 9FD00h)

GTM_CLC Register 구조:

Table 25-63 Registers Address Space

Modu	ıle		Base	Addr	ess		End	l Add	ress			Not	te		
GTM			F010 0000 _H F019 FFFF					F _H							
CLC Cloc	k Cor	ntrol	Regis	ter			(9FE)00 _H)			Res	et Va	lue: (0000	0003 _H
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								0							
	I	<u> </u>		I	<u>I</u>		1	r	I.	l			<u> </u>	1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1		1	1	()		1	1	1	1	EDIS		DIS S	DIS R
					·	-					1	rw	r	r	rw

Field	Bits	Туре	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	
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.



- 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_WDTCPU0CON0 Register는 System Critical Register의 System ENDINIT을 설정/해제한다.

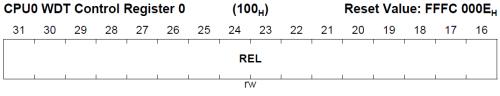
SCU_WDTCPU0CON0 Register 주소: F003_6100h (F0036000h + 100h)

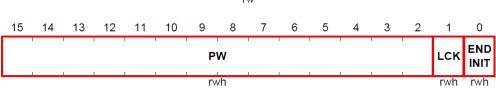
SCU_WDTCPU0CON0 Register 구조:

Table 7-27	Registers Address Spaces - SCU Kernel Registers
------------	---

			Note
SCU F003 60	000 _H	F003 63FF _H	-

WDTCPU0CON0







- 2. Data sheet 분석 : System Critical Register 설정 (2)
 - ✓ ENDINIT bit는 System ENDINIT의 설정 상태를 나타내며 Modify Access를 통해서만 수정이 가능하다.
 - ✓ **LCK bit**는 SCU_WDTCPU0CON0 Register의 Lock 상태를 나타내며 해당 Register의 Lock 상태는 Password Access를 통해 Unlock 되고, Modify Access를 통해 Lock 된다.
 - ✓ PW bits는 SCU_WDTCPU0CON0 Register에 접근하기 위한 Password를 저장하며 해당 값을

Field	Bits	Туре	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.
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.

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 000011 _B for these bits.

- 2. Data sheet 분석 : System Critical Register 설정 (3)
 - ✓ SCU_WDTCPU0CON0 Register에 적절한 값을 Write하여 Password Access를 수행한다.
 - ✓ Password Access는 SCU_WDTCPU0CON0 Register의 Lock 상태를 해제하며 과정은 다음과 같다.
 - 1. SCU_WDTCPU0CON0 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_WDTCPU0CON0 Register에 한번에 쓴다.
 - 6. SCU_WDTCPU0CON0 Register의 LCK bit를 확인하여 Lock 상태가 해제되었는지 파악한다. (Password Access가 정상적으로 수행되면 Lock 상태가 해제되며 LCK bit가 0으로 설정된다.)
 - ✓ Password Access를 통해 SCU_WDTCPU0CON0 Register의 Lock 상태가 해제되면 Modify
 Access를 통해 System ENDINIT을 설정/해제할 수 있다.

- 2. Data sheet 분석 : System Critical Register 설정 (4)
- ✓ SCU_WDTCPU0CON0 Register에 적절한 값을 Write하여 Modify Access를 수행한다.
- ✓ Modify Access는 System ENDINIT을 설정/해제하며 과정은 다음과 같다.
 - 1. SCU_WDTCPU0CON0 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_WDTCPU0CON0 Register에 한번에 쓴다.
 - 6. SCU_WDTCPU0CON0 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.

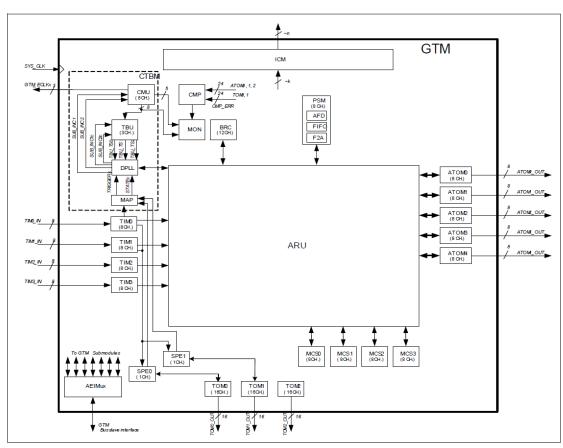


Figure 25-1 GTM Architecture Block Diagram



Table 25-1 Submodule groups										
Section	Submodule	Group								
Section 25.3	Advanced Routing Unit (ARU)	Infrastructural components								
Section 25.4	Broadcast Module (BRC)	Infrastructural components								
Section 25.5	First In First Out Module (FIFO)	Infrastructural components								
Section 25.6	AEI-to-FIFO Data Interface (AFD)	Infrastructural components								
Section 25.7	FIFO-to-ARU Interface (F2A)	Infrastructural components								
Section 25.8	Clock Management Unit (CMU)	Infrastructural components								
Section 25.9	Time Base Unit (TBU)	Infrastructural components								
Section 25.1	Timer Input Module (TIM)	IO Modules								
Section 25.1	Timer Output Module (TOM)	IO Modules								
Section 25.1 2	ARU-connected Timer Output Module (ATOM)	IO Modules								
Section 25.1 3	Multi Channel Sequencer (MCS)	Signal generation and processing								

User's Manual GTM, V1.9 25-2 V2.2, 2014-12



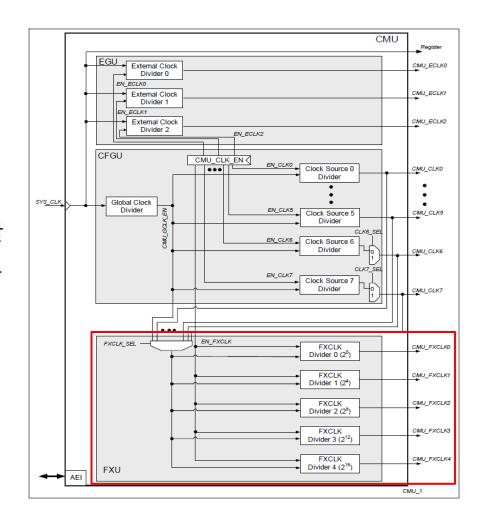
TC27x D-Step

Generic Timer Module (GTM)

Table 25-1 Submodule groups (cont'd)

Table 20 1	Cabinicatic groups (conta)	
Section	Submodule	Group
Section 25.1 4	Memory Configuration Module (MCFG)	Infrastructural component for MCS
Section 25.1 5	TIM0 Input Mapping Module (MAP)	Dedicated
Section 25.1 6	Digital PLL (DPLL)	Dedicated
Section 25.1 7	Sensor Pattern Evaluation Module (SPE)	BLDC support
Section 25.1 8	Interrupt Concentrator Module (ICM)	Interrupt services
Section 25.1 9	Output Compare Unit (CMP)	Safety features
Section 25.2 0	Monitoring Unit (MON)	Safety features

- 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.



- 2. Data sheet 분석 : GTM 내부 Clock 설정 (2)
 - ✓ GTM_CMU_FXCLK_CTRL Register는 CMU_FXCLK의 소스 클럭을 설정한다.

FXCLK_SEL

- ✓ 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 bits를 0000b로 설정한다.

GTM_CMU_FXCLK_CTRL Register 주소: F010_0344h

Reserved

JIIV	'I_C	IVIU	_' ^	CLI	,_Cı		neg	316	;ı Ţ	-⊥.	10	· U_\	JJ4.	711					
											(F0	100	000)h +	344	h)Field	Bits	Туре	Description
GTN	1_C	MU	_FX	CLK	CT_	RL	Regi	ist€	∙r 刁	^L 조:						FXCLK_S EL	[3:0]	rw	Input clock selection for EN_FXCLK line
Table	25-6	3 R	egist	ers A	ddres	ss Spa	ace												0000 _B CMU_GCLK_EN selected
Modu	ıle	I	Base	Addr	ess		End	Add	ress			No	te						0001 _B CMU_CLK0 selected
GTM		1	F010	0000	н		F019	FFF	FF _H										0010 _B CMU_CLK1 selected 0011 _B CMU_CLK2 selected
_	_	_	CLK_ ontro				(0034	14 _H)			Res	set V	alue:	0000	0000 _H				0100 _B CMU_CLK3 selected 0101 _B CMU_CLK4 selected 0110 _B CMU_CLK5 selected
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				0111 _B CMU_CLK6 selected 1000 _B CMU_CLK7 selected
		L	ı	ı	ı		Rese	ved		1		ı	1	1	<u> </u>				Note: This value can only be written, when the CMU FXCLK generation is disabled. See bits
							r												2322 in register CMU_CLK_EN.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				Note: Other values for FXCLK_SEL are reserved and

should not be used, but they behave like

FXCLK SEL = 0.

- 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 bits를 10b로 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	F010 0000 _H	F019 FFFF _H	

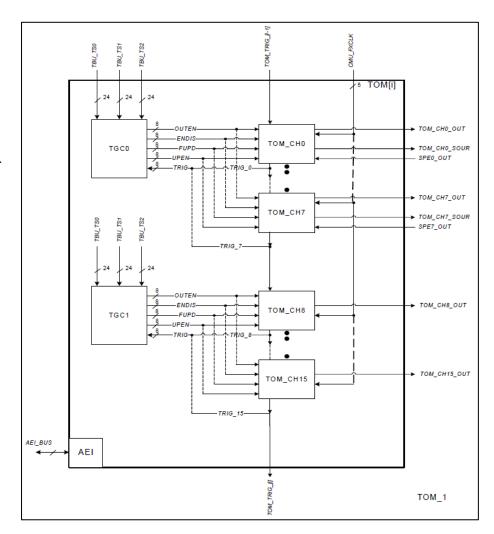
GTM_CMU_CLK_EN

CMU	Cloc	k Ena	able F	Regis	ter		(003	00 _H)			Res	set V	alue:	0000	0000 _H
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			Rese	erved	1		1		FXCL K	EN_E	CLK 2	EN_I	ECLK 1	_	ECLK 0
		•		r	•	•		r	W	r	W	r	w	ľ	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN_C	CLK7	EN_	CLK6	EN_	CLK5	EN_	CLK4	EN_	CLK3	EN_	CLK2	EN_	CLK1	EN_	CLK0

Field	Bits	Туре	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_ECLK 0	[17:16]	rw	Enable ECLK 0 generation subunit see bits [1:0]
EN_ECLK 1	[19:18]	rw	Enable ECLK 1 generation subunit see bits [1:0]
EN_ECLK 2	[21:20]	rw	Enable ECLK 2 generation subunit see bits [1:0]
EN_FXCL K	[23:22]	rw	Enable all CMU_FXCLK see bits [1:0] Note: An enable 00 _B clock source is disabled (ignore write access) disable clock signal and reset internal states enable clock signal



- 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.
- ✓ TGC는 8개의 TOM Channel과 연결되어 있으며 이를 통해 TOM Channel을 제어할 수 있다.
- ✓ TOM Channel은 TGC의 제어에 따라 동작을 수행하며 출력 신호를 생성한다.
- ✓ 본실습에서는 TOMO_CH1를 사용한다. (TOUT103과 연결되어 있기 때문이다.)
- ✓ 따라서, TOMO_CH1를 사용하기 위한 설정을 수행한다.



- 2. Data sheet 분석: TOM 동작 분석
 - ✓ TOM Channel은 CNO / CMO / CM1을 사용해 PWM 신호를 생성한다.
 - ✓ CNO: 동작 클럭에 따라 증가하는 Count 값을 저장한다.
 - ✓ CM0: PWM 신호의 주기를 결정하는 값을 저장한다.
 - ✓ CM1: PWM 신호의 Duty Ratio를 결정하는 값을 저장한다.
 - ✓ CNO는 동작 클럭에 따라 1씩 증가하며 CMO에 도달하면 0으로 초기화된다.
 - ✓ CNO가 CMO에 도달했을 때, 출력 신호는 SL 값으로 설정된다.
 - ✓ CNO가 CM1에 도달했을 때, 출력 신호는 SL 반전 값으로 설정된다.

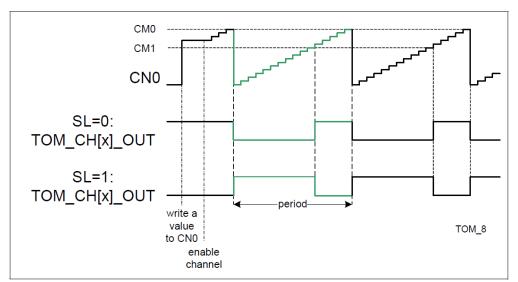




Figure 25-39 PWM Output with respect to configuration bit SL in continuos mode

- 2. Data sheet 분석 : TOMO TGCO 설정 (1), TOM Global Control
 - ✓ GTM_TOMO_TGCO_GLB_CTRL Register는 Channel 0-7을 제어하는 TGCO에 대한 설정을 한다.
 - ✓ Channel에 대한 Enable/Disable 설정 및 Output Enable 설정은 트리거 신호에 의해 일괄적으로 반영된다.
 - ✓ HOST_TRIG bit를 1로 설정하여 사용자가 소프트웨어적으로 트리거 신호를 발생시킬 수 있다.

RIG

GTM_TOM0_TGC0_GLB_CTRL Register 주소: F010_8030h (F0100000h + 8030h)

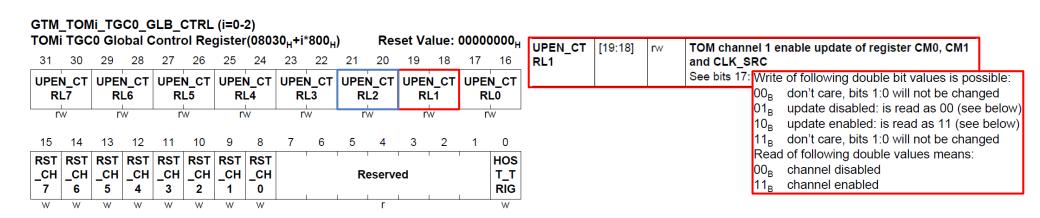
GTM_TOM0_TGC0_GLB_CTRL Register 구조:

Table 25-65	-63 Registers Address Space										
Module	Base Address	End Address	Note								

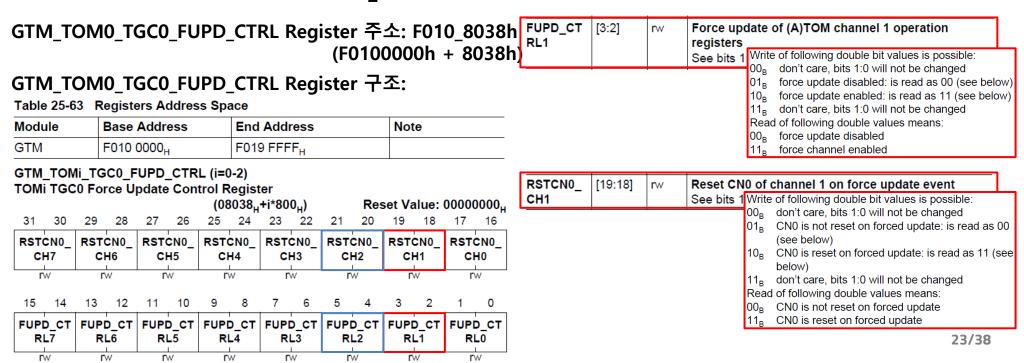
GIM		'	-010	UUUU _F	ł		FUI	9	ГН						
GTM TOM	_	_	_	_		•	•	30 _H +i	*800 _H)	Res	set Va	alue:	0000	0000 _H
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPEI RI	N_CT L7		N_CT L6		N_CT L5		N_CT _4	UPEI Ri	N_CT	UPEI RI	_	UPEI RI	N_CT		N_CT L0
r	W	r	W	r	N	r	N	r	N	r\	V	r	N	r	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST CH	RST CH			RST CH			RST CH			Re	eserve	ed			HOS T T

Field	Bits	Type	Description
HOST_TRI G	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'.

- 2. Data sheet 분석 : TOMO TGCO 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 bits를 10b로 설정하여 업데이트를 Enable 한다.



- 2. Data sheet 분석 : TOMO TGCO 설정 (3)
 - ✓ GTM_TOMO_TGCO_FUPD_CTRL Register는 트리거 신호에 따른 동작 설정을 한다.
 - ✓ FUPD_CTRLx bits는 CM0 / CM1 / CLK_SRC의 업데이트가 트리거 신호에 의해 실행되도록 설정하며 이를 Channel 1에 적용하기 위해 FUPD_CTRL1 bits를 10b로 설정한다.
 - ✓ RSTCNO_CHx bits는 CNO의 초기화가 트리거 신호에 의해 실행되도록 설정하며 이를 Channel
 1에 적용하기 위해 RSTCNO_CH1 bits를 10b로 설정한다.



- 2. Data sheet 분석 : TOMO TGCO 설정 (4)
 - ✔ GTM_TOMO_TGCO_ENDIS_CTRL Register는 트리거 신호에 따른 Enable/Disable을 설정한다.
 - ✓ 트리거 신호에 따라 각 Channel을 Enable 할지 Disable 할지 설정할 수 있다.
 - ✓ 트리거 신호 발생 시, Channel 1가 Enable 되게 ENDIS_CTRL1 bits를 10b로 설정한다.

GTM_TOM0_TGC0_ENDIS_CTRL Register 주소: F010_8070h (F0100000h + 8070h)

GTM TOMO TGCO ENDIS CTRL Register 구조:

Table 25-63	Registers Add	ress Snace
I able 25-05	Neuisieis Auu	icss Space

Modu	ıle		Base	Addr	ess		End	d Add	ress			No	te		
GTM			F010	0000	Н		F019 FFFF _H								
GTM_TOMi_TGC0_ENDIS_CTRL (i=0-2) TOMi TGC0 Enable/Disable Control Register (08070 _H +i*800 _H) Reset Value: 00000000 _H															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ı	1	1	1	1	1	Rese	erved	1				ı	1	
			•					r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	S_CT L7	1	S_CT		S_CT L5	1	S_CT L4		S_CT L3	ENDI:	_		S_CT L1		S_CT L0
r	W		w	r	W	r	W	r	W	r	N	r	W	r	W

ENDIS CT [3:2] (A)TOM channel 1 enable/disable update value RL1 See If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. Write of following double bit values is possible: 00_B don't care, bits 1:0 of register ENDIS STAT will not be changed on an update trigger 01_B disable channel on an update trigger 10_B enable channel on an update trigger 11_B don't change bits 1:0 of this register Note: if the channel is disabled (ENDIS[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.

- 2. Data sheet 분석 : TOMO TGCO 설정 (5)
 - GTM_TOMO_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 Output이 Enable 되게 OUTEN_CTRL1 bits를 10b로 설정한다.

GTM_TOM0_TGC0_OUTEN_CTRL Register 주소: F010_8078h Field $(F0100000h + 8078h_{T}^{O})$ GTM TOMO TGCO OUTEN CTRL Register 구조: Table 25-63 Registers Address Space Module **Base Address End Address** Note GTM F010 0000₁₁ F019 FFFF_H GTM_TOMi_TGC0_OUTEN_CTRL (i=0-2) TOMi TGC0 Output Enable Control Register $(08078_{H}+i*800_{H})$ Reset Value: 00000000_H Reserved OUTEN C TRL7 TRL6 TRL5 TRL4 TRL3 TRL2 TRL1 TRL0 rw rw rw rw

		7.	<u> </u>
OUTEN_C TRL0	[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 (ENDIS[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.
OUTEN_C TRL1	[3:2]	rw	Output (A)TOM_OUT(1)enable/disable update value See bits 1:0
OUTEN_C TRL2	[5:4]	rw	Output (A)TOM_OUT(2) enable/disable update value See bits 1:0
OUTEN_C TRL3	[7:6]	rw	Output (A)TOM_OUT(3) enable/disable update value See bits 1:0
OUTEN_C TRL4	[9:8]	rw	Output (A)TOM_OUT(4) enable/disable update value See bits 1:0
OUTEN_C TRL5	[11:10]	rw	Output (A)TOM_OUT(5) enable/disable update value See bits 1:0

Type Description

Bits

rw

25.11.8.9 Register TOMi_CHx_CTRL (x:0...14)

GTM_TOM0_CHx_CTRL (x=0-14)
TOM0 Channel x Control Register'

(08000_H+x*0040_H)
GTM_TOM1_CHx_CTRL (x=0-14)

TOM1 Channel x Control Register'
(08800 + x*0040 Reset Value: 00000800 Reset Value: 0000

(08800_H+x*0040_H)
GTM TOM2_CHx_CTRL (x=0-14)

TOM2 Channel x Control Register'

(09000_H+x*0040_H) Reset Value: 00000800_H

Reset Value: 00000800_H

- 2. Data sheet 분석 : TOMO Channel 1 설정 (1)
 - ✓ GTM_TOMO_CHx_CTRL Register는 TOMO의 각 Channel에 대한 동작 설정을 한다.
 - ✓ TOM Channel 1의 동작을 설정하기 위해 GTM_TOMO_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)

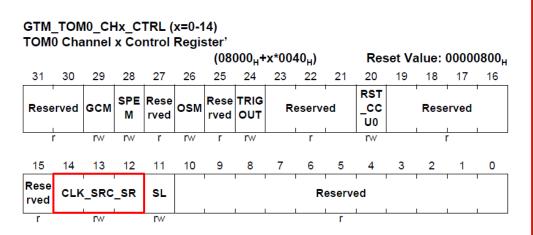
TOM0 Channel x Control Register'

					J	(080	000 _H +	x*004	40 _H)		Res	et V	alue:	00000	0800
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rese	rved	GCM	SPE M	Rese rved	OSM	Rese rved	TRIG OUT	R	eserve	ed	RST _CC U0		Rese	erved	
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					R	eserv	ed				
r		rw		rw						r					

Field	Bits	Туре	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.

- 2. Data sheet 분석 : TOMO Channel 1 설정 (2)
- ✔ TOM Channel 1의 동작 클럭을 CMU_FXCLK1로 설정하기 위해 CLK_SRC_SR bits를 001b로 설정한다.
- ✓ CMU_FXCLK1의 주파수는 100MHz / 16 = 6,250kHz 이다.

✓ CLK_SRC_SR bits가 업데이트를 할 때 반영되기 때문에 TOM Channel 1의 동작 클럭 또한 업데이트를 할 때 반영된다.



CLK_SRC	[14:12]	rw	Clock source select for channel
_SR			The register CLK_SRC is updated with the value of
			CLK_SRC_SR together with the update of register CM0
			and CM1.
			The input of the FX clock divider depends on the value of FXCLK SEL (see CMU).
			000 _B CMU_FXCLK(0) selected: clock selected by EXCLKSEL
			001 _B CMU_FXCLK(1) selected: clock selected by FXCLKSEL/ 2^4
			010 _B CMU_FXCLK(2) selected: clock selected by FXCLKSEL / 2^8
			011 _B CMU_FXCLK(3) selected: clock selected by FXCLKSEL / 2^12
			100 _B CMU_FXCLK(4) selected: clock selected by FXCLKSEL / 2^16
			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
			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.



- 2. Data sheet 분석 : TOMO Channel 1 설정 (3)
- ✔ GTM_TOMO_CHx_SRO Register는 CMO에 대한 Shadow Register이다.
- ✓ TOM Channel 1의 CM0를 설정하기 위해 GTM_TOM0_CH1_SR0 Register를 설정한다.
- ✓ GTM_TOMO_CH1_SR0 Register에 설정할 CM0 값을 저장하면 업데이트를 할 때 CM0에 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_TOM0_CH1_SR0 Register 구조:

Table 25-63	Registers Address Space
-------------	-------------------------

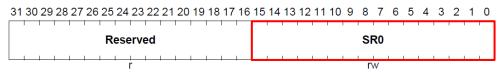
Module	Base Address	End Address	Note
GTM	F010 0000 _H	F019 FFFF _H	

GTM_TOM0_CHx_SR0 (x=0-15)

TOM0 Channel x CCU0 Compare Shadow Register

 $(08004_{H}+x*0040_{H})$

Reset Value: 00000000_H



$$(Period of PWM) = \frac{(Value of CM0) + 1}{(Freq. of CMU_FXCLK1)}$$
$$= \frac{12500}{6250kHz} = 0.002s$$



- 2. Data sheet 분석 : TOMO Channel 1 설정 (4)
 - ✓ GTM_TOMO_CHx_SR1 Register는 CM1에 대한 Shadow Register이다.
 - ✓ TOM Channel 1의 CM1을 설정하기 위해 GTM_TOMO_CH1_SR1 Register를 설정한다.
 - ✓ GTM_TOMO_CH1_SR1 Register에 설정할 CM1 값을 저장하면 업데이트를 할 때 CM1에 반영된다.
 - \checkmark 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

- 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 Register는 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.

- 2. Data sheet 분석: TOUT 설정 (2)
- ✓ GTM_TOUTSEL Register는 TOUT을 통해 출력될 신호를 설정한다.
- ✓ TOUT103에 대해 설정하기 위해 GTM_TOUTSEL6 Register의 SEL7 bits를 설정한다.
- ✓ TOM0 Channel 1를 통해 생성한 PWM 신호를 TOUT103로 출력하기 위해 SEL7 bits를 00b로 설정한다.

GTM_TOUTSEL6 Register 주소: F019_FD48h

(F0100000h + 9FD48h)

SEL9

rw

SEL8

GTM_TOUTSEL6 Register 구조:

SEL13

rw

Table 25-63 Registers Address Space

	Modu	ıle		Base	Addr	ess		End	I Add	ress			No	te		
GTM F010 0000 _H						F01	9 FFF	FFH								
TOUTSELn (n = 0-14) Timer Output Select Register						(9	FD30	_H +n*4	1 _H)		Res	et Va	lue: 0	0000	0000 _H	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		1		1		1		1		1		'		1		'

SEL11

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
,	SEL7	sı	EL6	SE	L 5	SE	L4	SE	L3	SE	L2	SE	L1	SE	L0

SEL12

rw



rw

SEL14

SEL15

rw

rw

103/16 = 6, 103%16 = 7

SEL10

104/16 = 6, 104%16=8

Field	Bits	Туре	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 25-67Table 25-68. OOB Timer A form Table 25-67Table 25-68 is connected as TOUT(n*16+x) to the ports O1B Timer B form Table 25-67Table 25-68 is connected as TOUT(n*16+x) to the ports Timer C form Table 25-67Table 25-68 is connected as TOUT(n*16+x) to the ports Timer D form Table 25-67Table 25-68 is connected as TOUT(n*16+x) to the ports Timer D form Table 25-67Table 25-68 is connected as TOUT(n*16+x) to the ports Note: If TOUT(n*16+x) is not defined in Table 25-67Table 25-68 this bit field has to be treated as reserved.							

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

Port	Input	Output	Input Timer Mapped		Output Timer Mapped					
			Α	В	Α	В	С	D		
P10.1	TIN103	TOUT103	TIM0_1	TIM1_1	TOM0_1	TOM2_9	ATOM 1_1	ATOM 4_1		

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

PORT IO 설정관련 레지스터 주소 및 비트 필드 정의

PORT IO 설정 함수

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.
 - ① 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_TOM0_CH1_CTRL Register를 통해 Signal level을 설정한다.
 - ⑨ GTM_TOM0_CH1_CTRL / GTM_TOM0_CH1_SR0 / GTM_TOM0_CH1_SR1 Register를 통해 CM0 / 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. ACE Lab.

- 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))
36 #define LCK
37 #define ENDINIT
39⊖ // GTM Registers
40 // GTM - CMU
41 #define GTM BASE
                                        (0xF0100000)
42 #define GTM CMU CLK EN
                                        (*(volatile unsigned int*)(GTM BASE + 0x00300))
43 #define GTM_CMU_FXCLK_CTRL
                                       (*(volatile unsigned int*)(GTM_BASE + 0x00344))
45 #define EN_FXCLK
                                       22
46 #define FXCLK SEL
48 // GTM - TOMØ
49 #define GTM_TOM0_TGC0_GLB_CTRL
                                        (*(volatile unsigned int*)(GTM BASE + 0x08030))
50 #define GTM TOM0 TGC0 ENDIS CTRL
                                        (*(volatile unsigned int*)(GTM BASE + 0x08070))
                                        (*(volatile unsigned int*)(GTM BASE + 0x08078))
51 #define GTM TOM0 TGC0 OUTEN CTRL
52 #define GTM_TOM0_TGC0_FUPD_CTRL
                                        (*(volatile unsigned int*)(GTM_BASE + 0x08038))
53 #define GTM TOM0 CH1 CTRL
                                        (*(volatile unsigned int*)(GTM_BASE + 0x08040))
54 #define GTM TOM0 CH1 SR0
                                        (*(volatile unsigned int*)(GTM BASE + 0x08044))
55 #define GTM TOM0 CH1 SR1
                                        (*(volatile unsigned int*)(GTM BASE + 0x08048))
57 #define UPEN CTRL1
                                       18
58 #define HOST TRIG
59 #define ENDIS CTRL1
                                       2
60 #define OUTEN_CTRL1
61 #define RSTCN0 CH1
62 #define FUPD CTRL1
63 #define CLK_SRC_SR
64 #define SL
65
67 #define GTM_CLC
                                        (*(volatile unsigned int*)(GTM_BASE + 0x9FD00))
68 #define GTM_TOUTSEL6
                                        (*(volatile unsigned int*)(GTM_BASE + 0x9FD48))
70 #define DISS
                                       1
   #define DISR
                                       0
   #define SEL7
```



GTM 설정관련 레지스터 주소 및 비트 필드 정의

- 3. 프로그래밍
 - 2) GTM을 설정하기 위한 함수를 구현한다.

```
116@ void init GTM TOMO PWM(void)
117 {
118
         /* GTM Enable */
119
       // Password Access to unlock WDTCPU0CON0
120
     (1) SCU WDT CPUØCONØ = ((SCU WDT CPUØCONØ ^ ØxFC) & ~(1 << LCK)) | (1 << ENDINIT);</p>
         while((SCU WDT CPU0CON0 & (1 << LCK)) != 0);
121
122
123
        // Modify Access to clear ENDINIT bit
124
        SCU_WDT_CPU0CON0 = ((SCU_WDT_CPU0CON0 ^ 0xFC) | (1 << LCK)) & ~(1 << ENDINIT);
         while((SCU WDT CPU0CON0 & (1 << LCK)) == 0);
125
126
127
                                                       // Enable GTM Module
     (2) GTM CLC &= ~(1 << DISR);
128
129
        // Password Access to unlock WDTCPU0CON0
130
     (3) SCU WDT CPUØCONØ = ((SCU WDT CPUØCONØ ^ ØxFC) & ~(1 << LCK)) | (1 << ENDINIT);</p>
131
         while((SCU WDT CPU0CON0 & (1 << LCK)) != 0);</pre>
132
133
        // Modify Access to set ENDINIT bit
        SCU WDT CPU0CON0 = ((SCU WDT CPU0CON0 ^{\circ} 0xFC) | (1 << LCK)) | (1 << ENDINIT);
134
        while((SCU WDT CPU0CON0 & (1 << LCK)) == 0);</pre>
135
136
        while((GTM CLC & (1 << DISS)) != 0);
                                               // Wait until module is enabled
137
138
139
         /* GTM Clock Setting */
140
     (4) GTM CMU FXCLK CTRL &= ~((0xF) << FXCLK SEL); // Input clock of CMU FXCLK : CMU GCLK EN
141
                                                         // Enable all CMU FXCLK
        GTM_CMU_CLK_EN |= ((0x2) << EN_FXCLK);
```

ACE Lab.

3. 프로그래밍

2) GTM을 설정하기 위한 함수를 구현한다.

```
/* GTM TOMO PWM Setting */
       GTM TOMO TGCO GLB CTRL = ((0x2) \ll UPEN CTRL1);
                                                           // TOMO channel 1 enable update of
                                                           // register CM0, CM1, CLK_SRC
146
147
    (6) GTM_TOMO_TGCO_FUPD_CTRL |= ((0x2) << FUPD_CTRL1); // Enable force update of TOMO channel 1
        GTM TOMO TGCO FUPD CTRL |= ((0x2) << RSTCNO CH1); // Reset CNO of TOMO channel 1 on force update
149
150
    (7) GTM_TOMO_TGCO_ENDIS_CTRL |= ((0x2) << ENDIS_CTRL1); // Enable channel 1 on an update trigger
151
152
        GTM TOMO TGCO OUTEN CTRL = ((0x2) << OUTEN CTRL1); // Enable channel 1 output on an update trigger
153
    8 GTM_TOM0_CH1_CTRL |= (1 << SL);
                                                          // High signal level for duty cycle
155
        GTM TOMO CH1 CTRL &= \sim((0x7) << CLK SRC SR);
156
                                                           // Clock source : CMU FXCLK(1) = 6250 kHz
157
        GTM TOMO CH1 CTRL |= (1 << CLK SRC SR);
        GTM TOM0 CH1 SR0 = 12500 - 1;
                                                           // PWM freq. = 6250 kHz / 12500 = 500 Hz
158
159⊖ //
          GTM TOMO CH1 SR1 = 0;
                                                            // Duty cycle = 0
                                                             // Duty cycle = 50
         GTM TOM0 CH1 SR1 = 6250-1;
160 //
        GTM TOM0 CH1 SR1 = 12500 - 1;
                                                           // Duty cycle = 100
161
162
        GTM TOUTSEL6 &= \sim((0x3) << SEL7);
                                                           // TOUT103 : TOM0 channel 1
163
164
    (B)
        GTM TOMO TGCO GLB CTRL = (1 << HOST TRIG);
                                                           // Trigger request signal to update
165
166
```

GTM 설정 함수



3. 프로그래밍

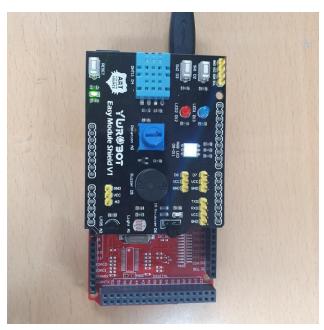
3) 동작에 따라 'main' 함수를 구현한다. (앞서 구현한 함수들을 호출한다.)

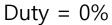
```
31⊖ /* Address of Registers */
32 // SCU Registers
33 #define SCU BASE
                           (0xF0036000)
34 #define SCU WDT CPU0CON0 (*(volatile unsigned int*)(SCU BASE + 0x100))
35
36 #define LCK
37 #define ENDINIT
39⊖ // GTM Registers
40 // GTM - CMU
41 #define GTM BASE
                                       (0xF0100000)
42 #define GTM CMU CLK EN
                                       (*(volatile unsigned int*)(GTM BASE + 0x00300))
43 #define GTM CMU FXCLK CTRL
                                       (*(volatile unsigned int*)(GTM BASE + 0x00344))
45 #define EN FXCLK
                                       22
46 #define FXCLK SEL
47
48 // GTM - TOMØ
49 #define GTM TOMO TGCO GLB CTRL
                                       (*(volatile unsigned int*)(GTM BASE + 0x08030))
50 #define GTM TOM0 TGC0 ENDIS CTRL
                                       (*(volatile unsigned int*)(GTM BASE + 0x08070))
51 #define GTM TOM0 TGC0 OUTEN CTRL
                                       (*(volatile unsigned int*)(GTM BASE + 0x08078))
52 #define GTM TOM0 TGC0 FUPD CTRL
                                       (*(volatile unsigned int*)(GTM BASE + 0x08038))
53 #define GTM TOM0 CH1 CTRL
                                       (*(volatile unsigned int*)(GTM BASE + 0x08040))
54 #define GTM TOM0 CH1 SR0
                                       (*(volatile unsigned int*)(GTM BASE + 0x08044))
55 #define GTM TOM0 CH1 SR1
                                       (*(volatile unsigned int*)(GTM BASE + 0x08048))
57 #define UPEN CTRL1
                                       18
58 #define HOST TRIG
59 #define ENDIS CTRL1
60 #define OUTEN_CTRL1
                                       2
61 #define RSTCN0 CH1
62 #define FUPD CTRL1
                                       2
63 #define CLK SRC SR
                                       12
64 #define SL
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
71 #define DISR
72 #define SEL7
                                       14
```

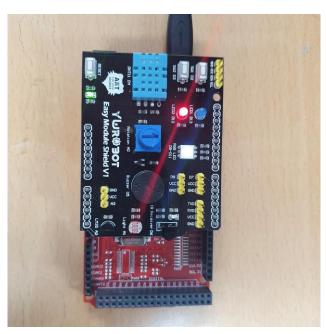
```
74 // PORT10 Registers
75 #define PORT10 BASE
                             (0xF003B000)
   #define PORT10 IOCR0
                             (*(volatile unsigned int*)(PORT10 BASE + 0x10))
77
78
   #define PC1
                            11
79
80
    /* Function Prototype */
   void init LED(void);
   void init GTM TOMO PWM(void);
83
84
   IfxCpu syncEvent g cpuSyncEvent = 0;
85
86⊖ int core0_main(void)
87
88
        IfxCpu enableInterrupts();
89
90⊝
        /* !!WATCHDOGO 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();
01
        init GTM TOM0 PWM();
02
103
        while(1)
104
.05
106
        return (1);
.07
```

4. 동작 확인

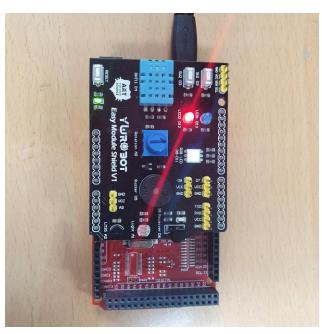
✓ Build 및 Debug 후 ('Resume' 버튼 클릭), CM1 값을 바꿔보며 Duty Ratio에 따른 LED 밝기를 확인한다.







Duty = 50%



Duty = 100%

- 1. import tc275_PWM
- 2. BLUE LED 추가 P10.2 > TOUT104





TC27x D-Step

Generic Timer Module (GTM)

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

Port	Input	Output	Input Timer Mapped		Output Tir	ner Mapped	d	
			Α	В	Α	В	С	D
P02.2	TIN2	TOUT2	TIM0_2	TIM1_2	TOM0_10	TOM1_10	ATOM 0_2	ATOM 1_2
P02.3	TIN3	TOUT3	TIM0_3	TIM1_3	TOM0_11	TOM1_11	ATOM 0_3	ATOM 1_3
P02.4	TIN4	TOUT4	TIM0_4	TIM1_4	TOM0_12	TOM1_12	ATOM 0_4	ATOM 1_4
P02.5	TIN5	TOUT5	TIM0_5	TIM1_5	TOM0_13	TOM1_13	ATOM 0_5	ATOM 1_5
P02.6	TIN6	TOUT6	TIM0_6	TIM1_6	TOM0_14	TOM1_14	ATOM 0_6	ATOM 1_6
P02.7	TIN7	TOUT7	TIM0_7	TIM1_7	TOM0_15	TOM1_15	ATOM 0_7	ATOM 1_7
P02.8	TIN8	TOUT8	TIM2_0	TIM3_0	TOM0_8	TOM1_0	ATOM 0_0	ATOM 1_0
P10.0	TIN102	TOUT102	TIM0_4	TIM1_4	TOM0_4	TOM2_12	ATOM 1_4	ATOM 4_4
P10.1	TIN103	TOUT103	TIM0_1	TIM1_1	TOM0_1	TOM2_9	ATOM 1_1	ATOM 4_1
P10.2	TIN104	TOUT104	TIM0_2	TIM1_2	TOM0_2	TOM2_10	ATOM 1_2	ATOM 4_2
P10.3	TIN105	TOUT105	TIM0_3	TIM1_3	TOM0_3	TOM2_11	ATOM 1_3	ATOM 4_3
P10.4	TIN106	TOUT106	TIM0_6	TIM1_6	TOM0_6	TOM2_6	ATOM 0_6	ATOM 4_6
P10.5	TIN107	TOUT107	TIM0_2	TIM1_2	TOM0_2	TOM2_10	ATOM 1_2	ATOM 4_2
P10.6	TIN108	TOUT108	TIM0_3	TIM1_3	TOM0_3	TOM2_11	ATOM 1_3	ATOM 4_3
P10.7	TIN109	TOUT109	TIM0_0	TIM1_0	TOM0_0	TOM2_8	ATOM 1_0	ATOM 4_0
P10.8	TIN110	TOUT110	TIM0_5	TIM1_5	TOM0_5	TOM2_13	ATOM 1_5	ATOM 4_5

- 1. import tc275_PWM
- 2. BLUE LED 추가

P10.2 → TOUT104

- 3. ADC4.7(Potential Meter) RED LED Duty
- 4. ADC4.6(Light Sensor) BLUE LED Duty

ADC → duty calculation

duty = ADC/4095*pwm period

https://github.com/swip6th/swip6th/mcu/tree/main/tc275_PWM

```
CpuO_Main_LAB0.c
```

my_lib.h

my_lib.c



- 1. import tc275_PWM
- 2. BLUE LED 추가

P10.2 → TOUT104

- 3. ADC4.7(Potential Meter) RED LED Duty
- 4. ADC4.6(Light Sensor) BLUE LED Duty

dynamic range scaling

```
light_sensor_duty = ((light_sensor - light_sensor_min)*4095)/(max-min) light_sensor_max, light_sensor_min 은 debugger break 기능 사용하여 측정 min 보다 작은 경우 0, max 보다 클 경우 4095 처리
```

while(1)

systick_curr = SYSTEM_TIMER_0_31_0;
systick = systick_curr - systick_prev;
if(systick > SYSTICK_100MHZ/10) // 100ms

systick_prev = systick_curr;
potential_meter = GetUADC4(7);
light_sensor = GetUADC4(6);

// FIXME: light sensor scaling

if(potential_meter_duty == 0)
 GTM_TOMO_CH1_SR1 = 0;

if(light_sensor_duty == 0)
 GTM_TOMO_CH2_SR1 = 0;

GTM_TOMO_CH1_SR1 = potential_meter_duty - 1;

GTM_TOM0_CH2_SR1 = light_sensor_duty - 1;

// FIXME: duty calculation

GTM_TOM0_CH1_SR0 = 12500 - 1;

GTM_TOM0_CH2_SR0 = 12500 - 1;

ADC → duty calculation

duty = ADC/4095*pwm period

https://github.com/swip6th/swip6th/mcu/tree/main/tc275_PWM

```
CpuO_Main_LAB1.c
my_lib.h
my_lib.c
```



RGB.PWM

RGB PWM

Red

TOUT7

Green

TOUT107

Blue

TOUT105

Blue

Red

SW1 누르면, Blue LED 켜고 RGB.Blue Duty 설정 SW2 누르면, Red LED 켜고 RGB.Red Duty 설정

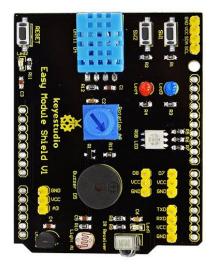
둘다 누르면, Blue/Red LED 켜고 RGB.Green Duty 설정

SW Debounce: 1초, SW state Debounce: 1초

모	듈	TC275 Shield Buddy Pin	TC275 Pin	TC275 관련기능
	Red	D9	P02_7	GPIO, PWM
RGB LED	Green	D10	P10_5	GPIO, PWM
	Blue	D11	P10_3	GPIO, PWM

Tc275_PWM_RGB project 생성 https://github.com/swip6th/mcu/tree/main/tc275_PWM_RGB CpuO_Main.c my lib.h my_lib.c





RGB.PWM

RGB PWM

Red TOUT7

Green TOUT107

Blue TOUT105

SW1 누르면, Blue LED 켜고 RGB.Blue Duty 설정 SW2 누르면, Red LED 켜고 RGB.Red Duty 설정 둘다 누르면, Blue/Red LED 켜고 RGB.Green Duty 설정 SW Debounce: 1초, SW state Debounce: 1초

모듈		TC275 Shield Buddy Pin	TC275 Pin	TC275 관련기능
RGB LED	Red	D9	P02_7	GPIO, PWM
	Green	D10	P10_5	GPIO, PWM
	Blue	D11	P10_3	GPIO, PWM

```
112
            // FIXME: 1s state debounce
113
            SW_state_prev = SW_state_curr;
114
            SW_state_curr = (SW2_debounce<<1) | (SW1_debounce<<0);</pre>
115
116
            if( SW_state_debounce != 0 )
117
118
                potential_meter = GetUADC4(7);
119
                potential_meter_duty = (potential_meter*12500)/4095;
120
121
                if( potential_meter_duty == 0 )
122
                    potential_meter_duty = 0;
123
124
                    potential_meter_duty -= 1;
                // FIXME
                // SW1 pushed
                                                                           TOMO_CH3
                                   : blue
                                              led on, RGB.blue duty
128
                // SW2 pushed
                                   : red
                                              led on, RGB.red dutu
                                                                           TOMO CH15
129
                // SW1&SW2 pushed : blue&red led on, RGB.green duty
                                                                           TOMO_CH2
130
131
                // End of FIXME
132
            }
133
            else
                                                                      // SW1 & SW2 are open
134
135
                    PORT10_OMR = (1<<PCL1) |
                                                                      // LED RED off
136
                                  (1<<PCL2) ;
                                                                      // LED BLUE off
137
138
139
        return (1);
140 }
141
142
143 // 100ms timer
    __interrupt( 0x0F ) __vector_table( 0 )
145 void CCU61_T12_ISR(void)
146 (
147
        SW1_prev = SW1_curr;
148
        SW1_curr = (PORT02_IN & (1<<P0)) == 0;
        SW2_prev = SW2_curr;
150
        SW2\_curr = (PORTO2\_IN & (1<<P1)) == 0;
152
        // FIXME: SW1, SW2 software debounce for 1 second
153
154
        irq_timer = 1;
155 }
```

Buzzer

 $GTM_CMU_CLK_EN &= \sim ((0x2) << EN_FXCLK); // disable$

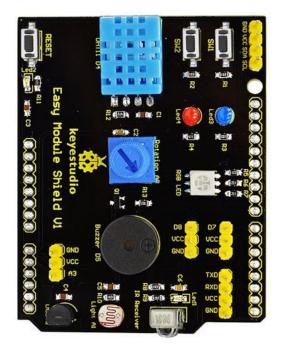
TOUT3 GTM_TOMO

Channel 11

delay 앞 Buzzer Enable delay 뒤 Buzzer Disable

int pwm_cnt = PWM_FREQ / tones[i]; break 시 Buzzer 들리지 않게!!!!!

모듈		TC275 Shield Buddy Pin	TC275 Pin	TC275 관련기능
LED1		D13	P10_2	GPIO, PWM
LED2		D12	P10_1	GPIO, PWM
SW1		D2	P02_0	GPI0
SW2		D3	P02_1	GPIO
RGB LED	Red	D9	P02_7	GPIO, PWM
	Green	D10	P10_5	GPIO, PWM
	Blue	D11	P10_3	GPIO, PWM
Rotation		AO	SAR4_7	ADC
Buzzer		D5	P02_3	PWM
Light		A1	SAR4_6	ADC
LM35		A2	SAR4_5	ADC



Ultrasonic

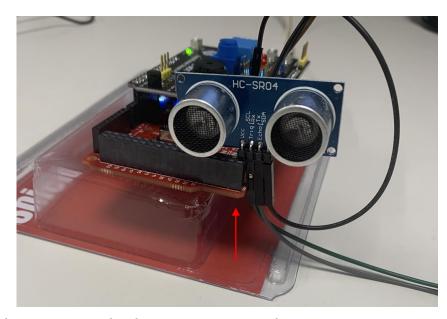
VDD(5V)

Trig P15_4

Echo P15_5

GND GND

거리(cm) Live Update



https://github.com/swip6th/mcu/tree/main/tc275_Ultrasonic Cpu0_Main.c



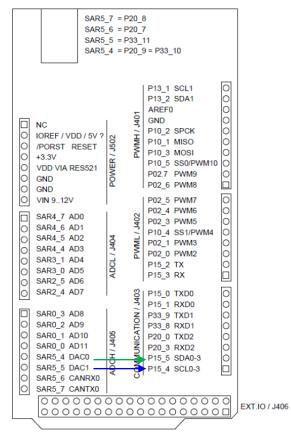


Figure 7 TC275 to Arduino Connector Mapping

```
GND
ASC2RX,ASC3TX P33_5 P15_8
                         0 0
                               CANTXIO P11_12, P20_10
PIN50 P33_4, P21_2
                          O O PIN52 P02_8, P13_3
                         O O PIN49 P21_0
            PIN48 PIN48
            PIN46 P11_3
                          O O PIN47 P11_6
            PIN44 P33_3
                          O O PIN45 P11_2
                          O O PIN43 P11_11
            PIN40 P33_0
                          O O PIN41 P11_9
            PIN38 P33_1
                               PIN39 P00 7
                              PIN37 P00_6
            PIN34 P00_12
                          O O PIN35 P00_5
            PIN32 P00 11
                          O O PIN33 P00 4
            PIN30 P00_10
                          O O PIN31 P00_3
            PIN28 P00_9
            PIN26 P00 8
                         O O PIN27 P00 1
                         O O PIN25 P00 0
                          O O PIN23 P14_1
            VDD / 5V ?
```

Q&A

Thank you for your attention

OOOOO Architecture and
Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU

ACE Lab (hn02301@gmail.com)

