

# **PWM\_Generation with DTM Example**

**Smart Embedded System Lab  
Kookmin University**

**이 자료 및 교육 과정 전반에 대한 저작권은  
국민대학교 산학협력단에 있습니다.**

**이 자료를 활용한 외부 교육을 금합니다.  
이 자료의 재배포 및 무단 전재, 복제 등을 금합니다.**

## PWM 실습

1. Objective
2. Tresos 설정
3. Tasking 설정
4. PLS 디버깅

# Exercise PWM - Objective

## ➤ Objective

- General PWM 신호 생성 실습
- Tresos 설정
  - Driver 설정 및 실습
    - ✓ Pwm Driver
    - ✓ Mcu Driver
    - ✓ Port Driver
- PWM\_Generation 실습
  - 레지스터 이용한 DTM 적용

GTM	Port	Output Freq.	파형	비고
TOM0_0	P33.10	10kHz		Reference PWM
TOM0_1	P33.9	10kHz		Center Aligned
TOM0_2	P33.11	10kHz		Shifted(50%) PWM

Dead time 미적용  
Dead time 적용

# Exercise PWM – Objective

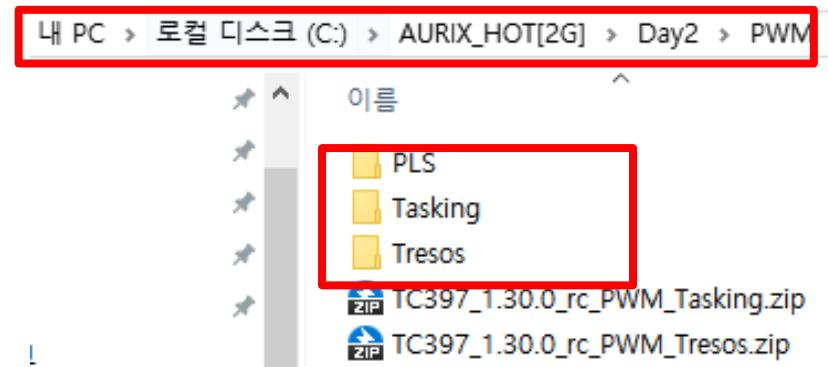
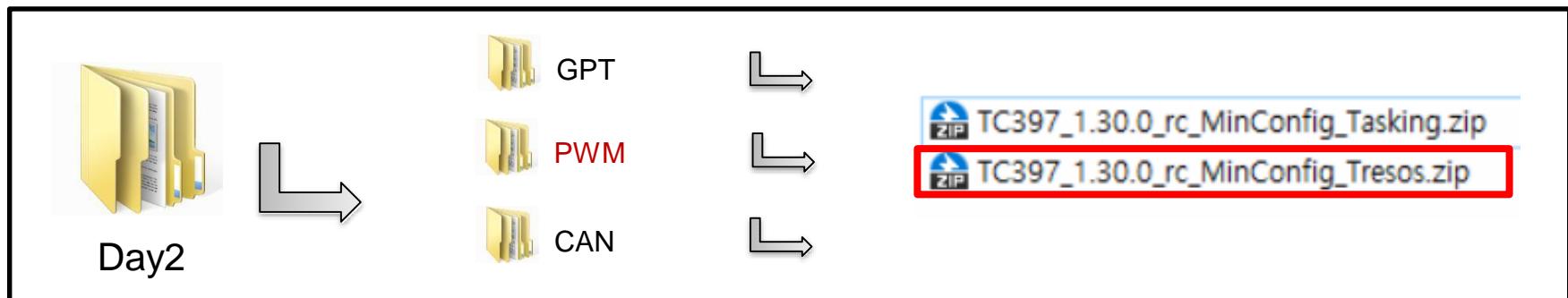
## ➤ 워크스페이스 폴더 아키텍처

### ▪ C:\AURIX\_HOT\

- Tresos 폴더 생성 : ..\Day2\PWM\Tresos
- Tasking 폴더 생성 : ..\Day2\PWM\Tasking
- PLS 폴더 생성 : ..\Day2\PWM\PLS

<WorkSpace 경로>

<Import파일>



## PWM 실습

1. Objective
2. Tresos 설정
3. Tasking 설정
4. PLS 디버깅

# EB Tresos Configurations

## ➤ Tresos 실행



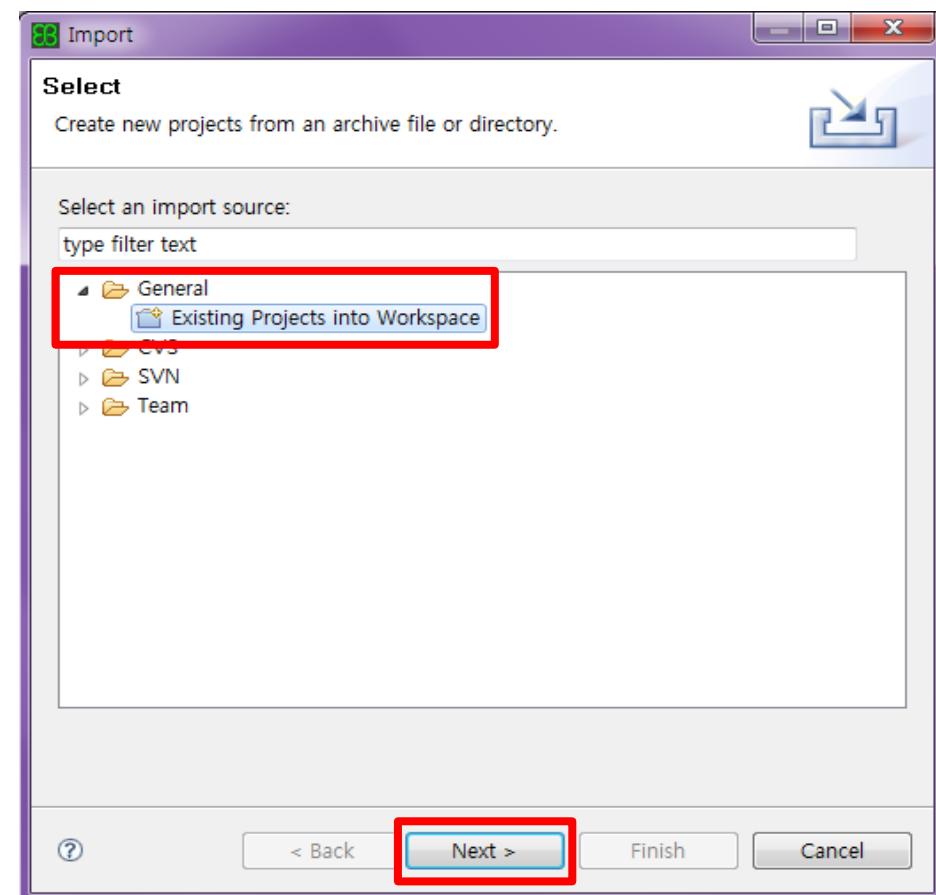
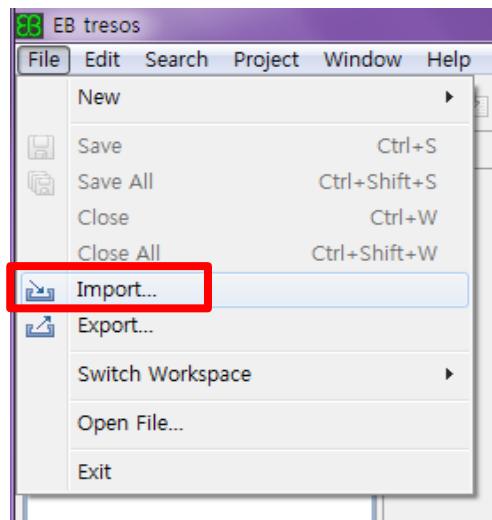
클릭



# Exercise PWM – EB Tresos Configurations

## ➤ 프로젝트 생성

- File -> Import 클릭
- General -> Existing Projects into Workspace 선택 -> Next 클릭

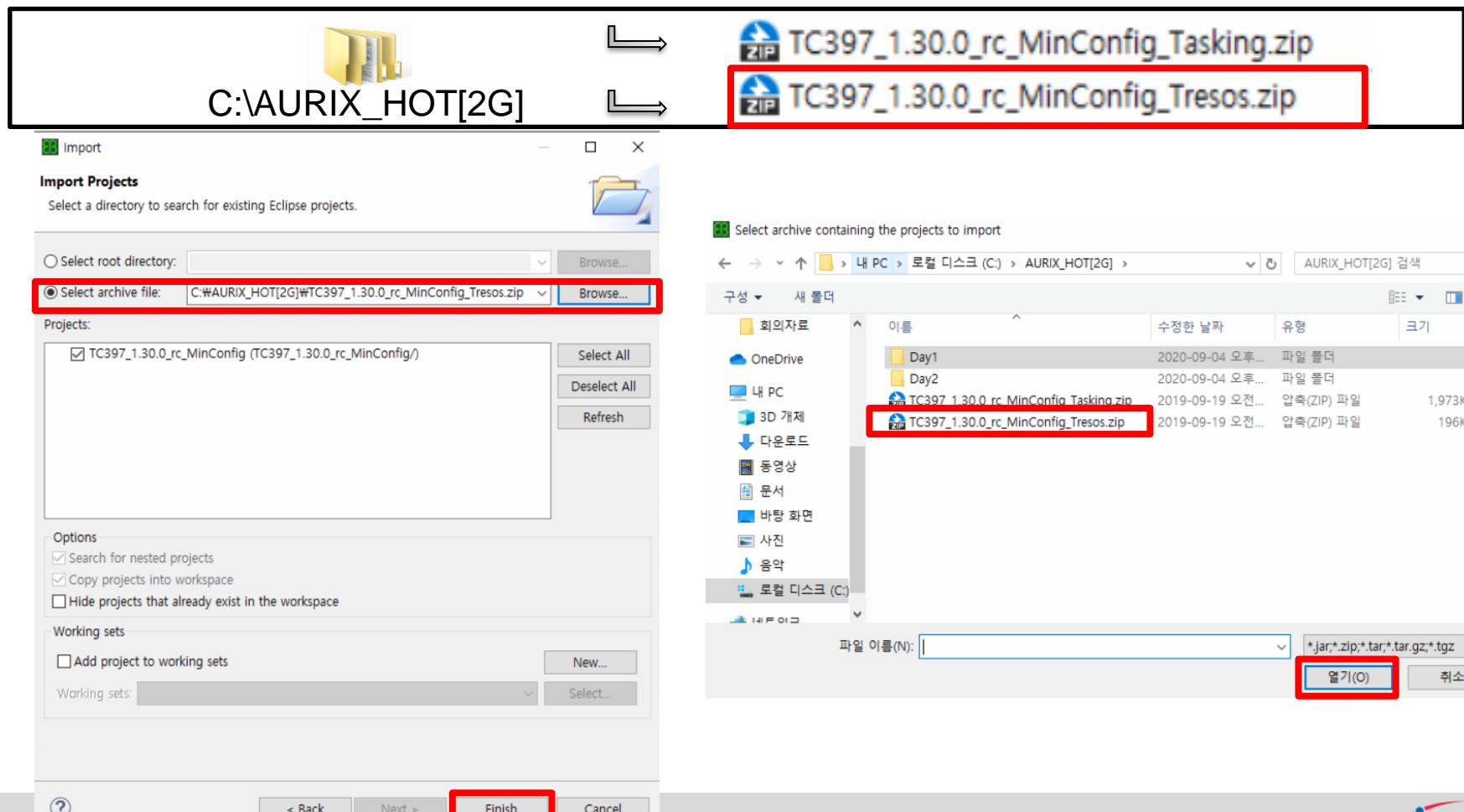


# Exercise PWM – EB Tresos Configurations

## ➤ Tresos\_Min\_Config 프로젝트 Import

- Select archive file 선택 -> Browse 클릭

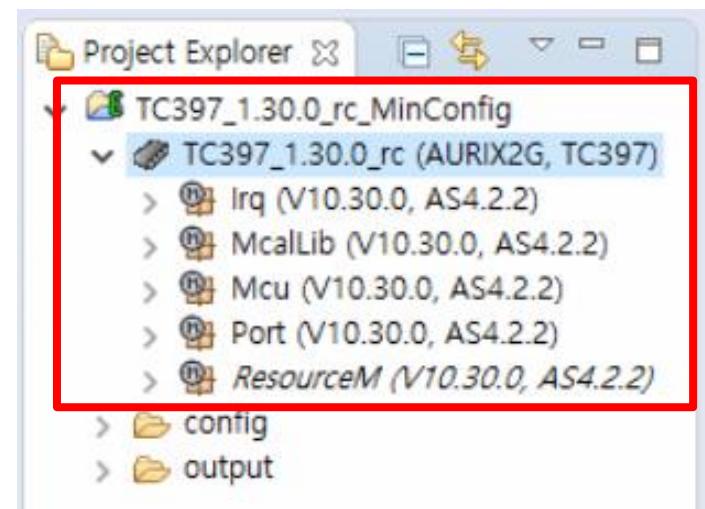
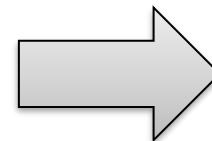
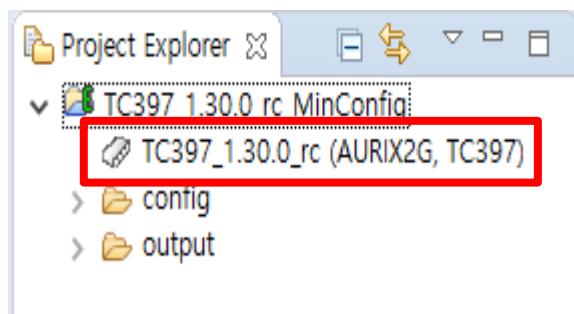
- 배포한 TC397\_1.30.0\_rc\_MinConfig\_Tresos.zip 선택 -> 열기



# Exercise PWM – EB Tresos Configurations

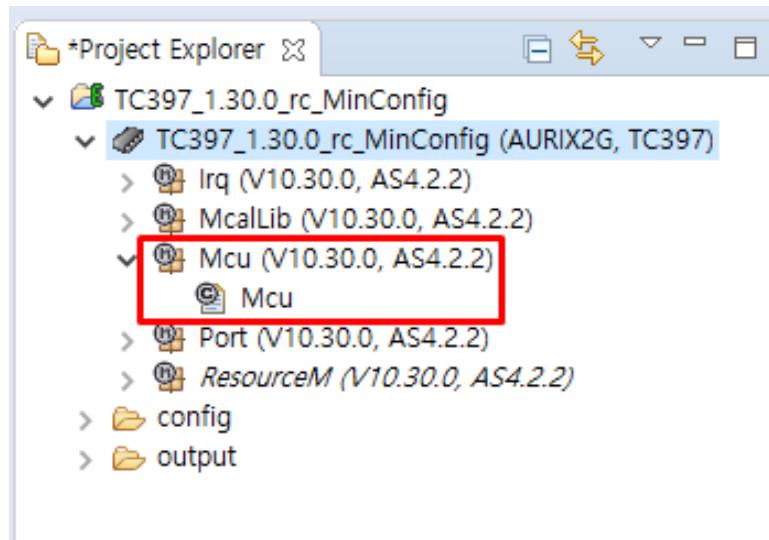
## ➤ Tresos\_Min\_Config 프로젝트 Import

- 각 프로젝트의  더블 클릭
  - 모듈이 오른쪽 그림과 같이 생성 됨



# Exercise PWM – EB Tresos Configurations (cont.)

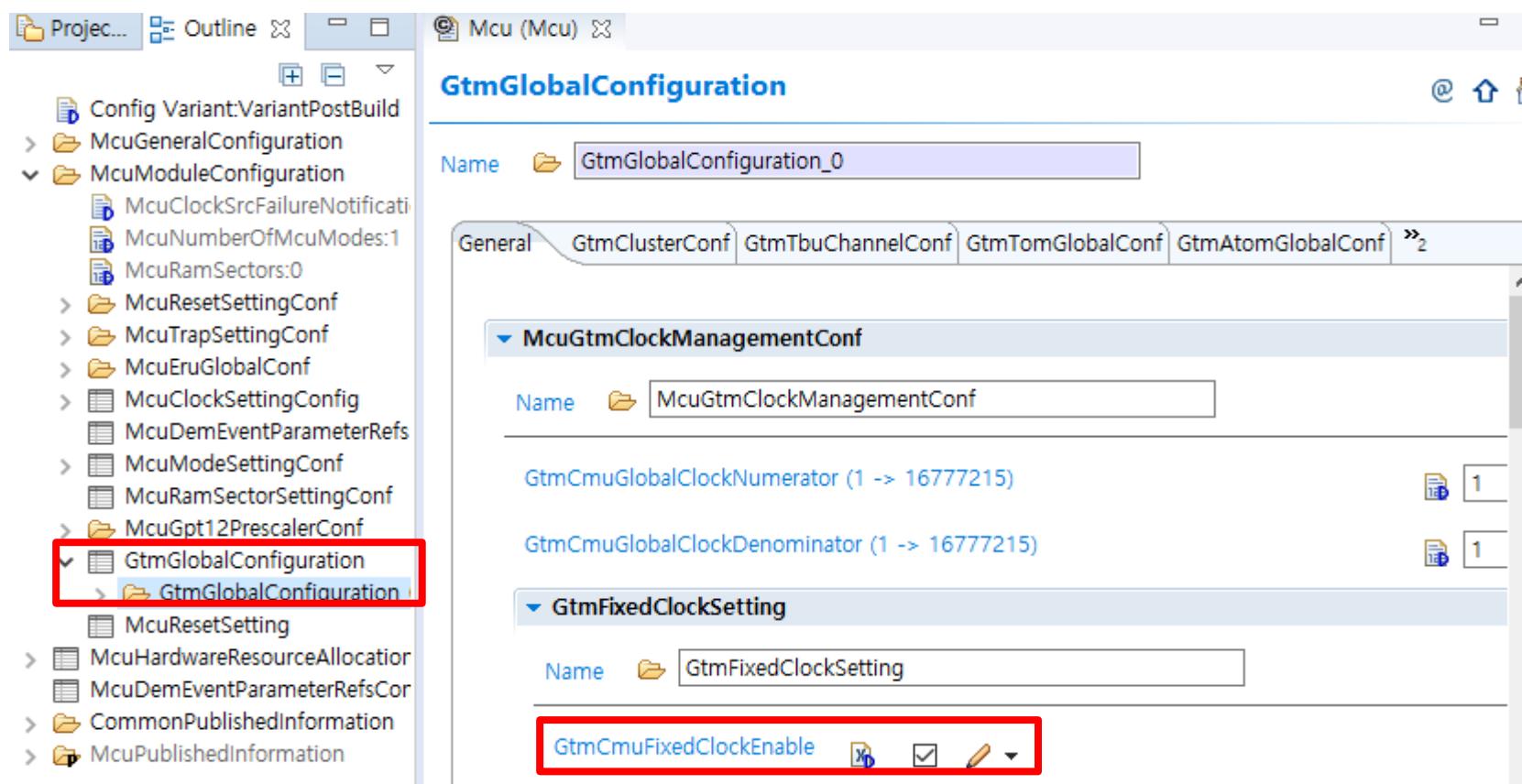
- Mcu 모듈 설정
  - Mcu 더블 클릭



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Mcu 모듈 설정

- **McuModuleConfiguration** → **GtmGlobalConfiguration** → **GtmGlobalConfiguration\_0**
  - GtmCmuFixedClockEnable : 체크



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Mcu 모듈 설정

- McuClockSettingConfig를 살펴보면 GTM Clock은 2.0E8 ( 200MHz)로 설정

- GtmClusterConf

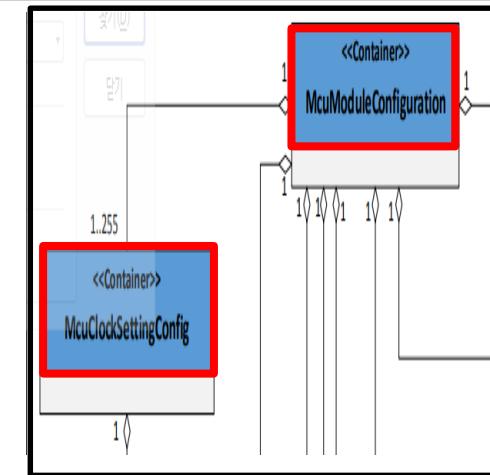
- ✓ 기본 클럭을 나눠줄 수 있는 기능

- ✓ EB기본 설정은 200MHz를 2를 나눈 100MHz를 사용하도록 되어 있으므로 200MHz를 사용하려면 2를 나누지 않는 SEL1로 설정해야함

The screenshot shows two configuration panels side-by-side. On the left is the 'McuClockSettingConfig' panel, which has a 'Name' field set to 'McuClockSettingConfig\_0'. This field is highlighted with a red rectangle. Below it are sections for 'McuClockReferencePoint' (set to 'McuErayFrequency (dynamic range)' with value '8.0E7') and 'McuGTMFrequency (dynamic range)' (set to '2.0E8'). On the right is the 'GtmClusterConf' panel, also with a 'Name' field set to 'GtmClusterConf\_0'. Under the 'General' tab, there is a dropdown menu for 'GtmCmuClusterInputClockDividerEnable\*' with several options. One option, 'CLS\_CLK\_CFG\_ENABLED\_WITHOUT\_DIV\_SEL1', is highlighted with a red rectangle. Below this is a 'GtmClusterConfClockSetting' section with a 'Name' field set to 'GtmClusterConfClockSetting'. A dropdown menu here also shows 'CLS\_CLK\_CFG\_ENABLED\_WITHOUT\_DIV\_SEL1' as the selected option.

- Fixed Clock의 경우 다음과 같이 구성 되어 있다.

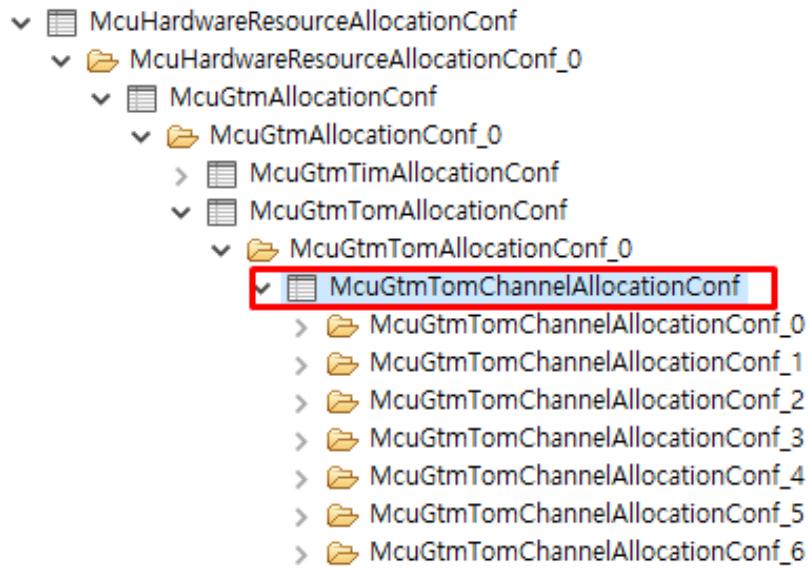
- Fixed Clock 0 : 100MHz → 200MHz
  - Fixed Clock 1 : 6.25MHz → 12.5MHz
  - Fixed Clock 2 : 390KHz → 780KHz
  - Fixed Clock 3 : 24KHz → 48KHz
  - Fixed Clock 4 : 1.5KHz → 3KHz



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Mcu 모듈 설정

- McuHardwareResourceAllocationConf  
  ▪ McuGtmAllocationConf  
    ▪ McuGtmAllocationConf\_0  
    ▪ McuGtmTomAllocationConf  
      ▪ McuGtmTomAllocationConf\_0
  - McuGtmTomAllocationConf\_0,1,2을 GTM\_TOM\_CHANNEL\_USED\_BY\_PWM으로 선택

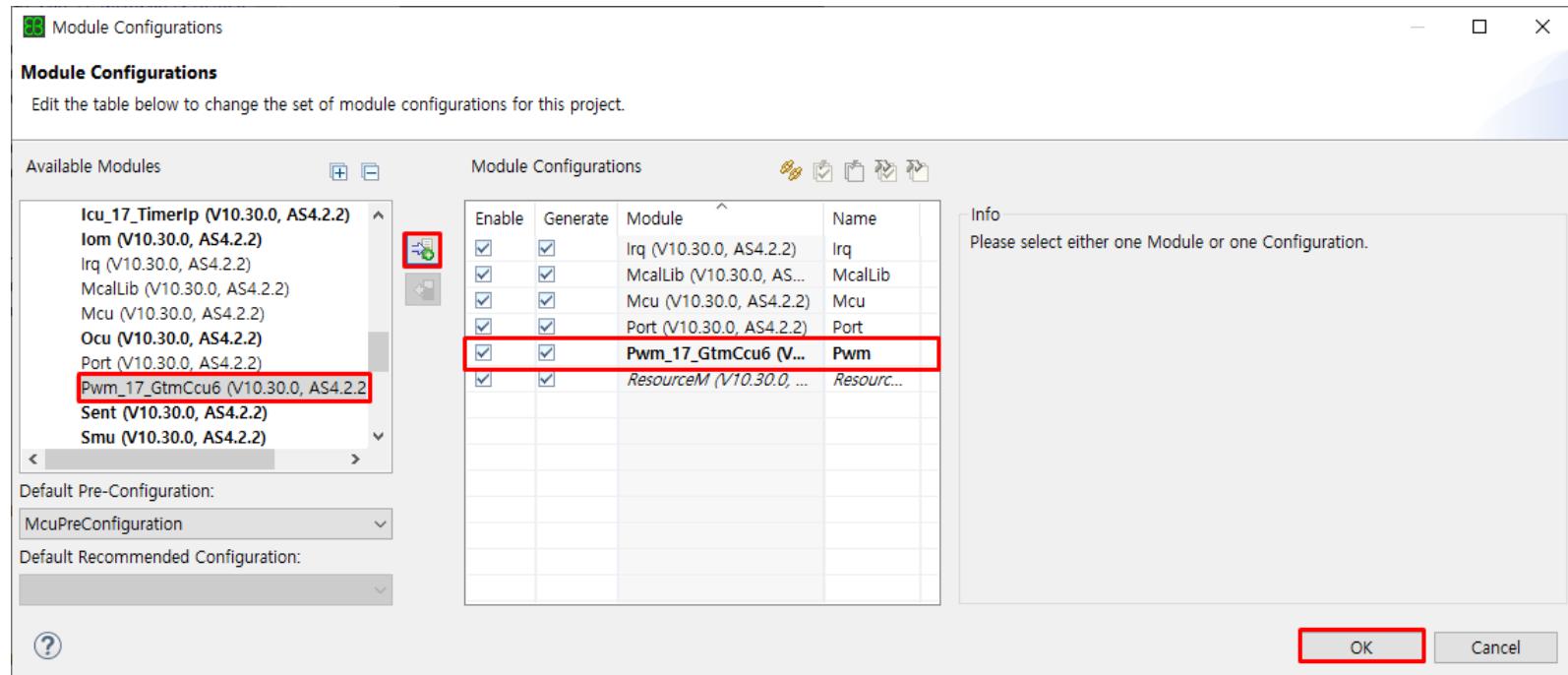


McuGtmTomAllocationConf				
Name				
McuGtmTomChannelAllocationConf				
McuGtmTomChannelAllocationConf				
Index	Name	McuGtmTomChannelAllocationConf	X	McuTom...
0	McuGtmT...	GTM_TOM_CHANNEL_USED_BY_PWM	☒	
1	McuGtmT...	GTM_TOM_CHANNEL_USED_BY_PWM	☒	
2	McuGtmT...	GTM_TOM_CHANNEL_USED_BY_PWM	☒	
3	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
4	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
5	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
6	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
7	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
8	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
9	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
10	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
11	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
12	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
13	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
14	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	
15	McuGtmT...	GTM_TOM_CHANNEL_NOT_USED	☒	

# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 추가

- **TC397\_1.30.0\_rc\_Minconfig (AURIX2G, TC397)에서 오른쪽 마우스 클릭**
  - Module Configuration 클릭
- **Pwm\_17\_GtmCcu6 (V10.30.0, AS4.2.2)**

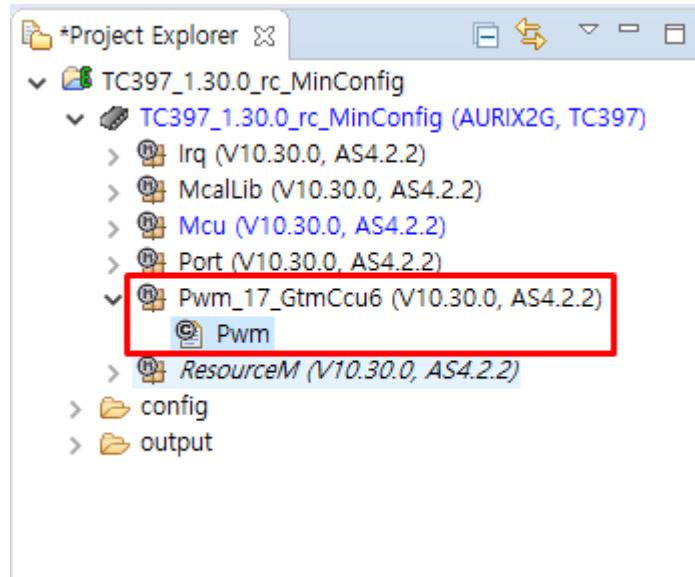


# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

- Pwm 더블 클릭
- Pwm 모듈은 다음과 같은 기능을 제공

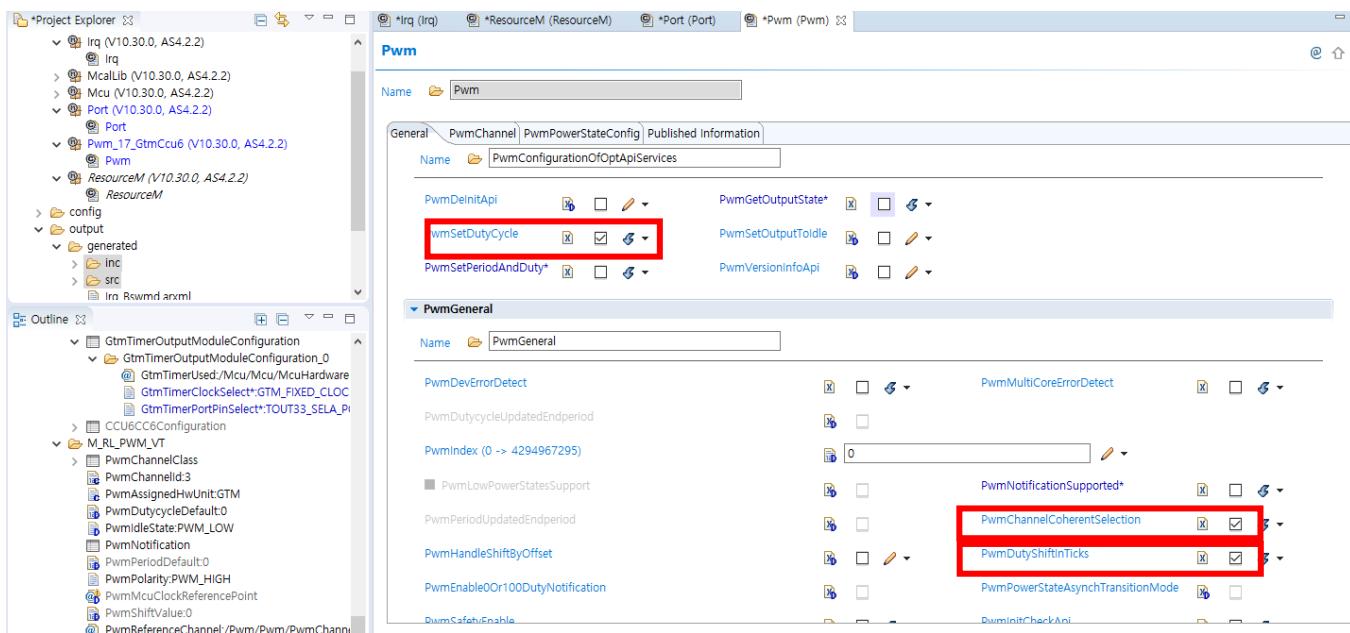
- Can be configured with multiple channels and has features to change the duty cycle, period during runtime.
- Support notification to the upper layer upon configured edge.



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

- **PwmSetDutyCycle -> 체크**
  - Pwm\_17\_Gtm\_SetDutyCycle API를 사용할 수 있도록 설정
- **PwmChannelCoherentSelection -> 체크**
  - Period/duty cycle을 즉시 update 할 수 있도록 설정
- **PwmDutyShiftInTicks -> 체크 ( PWM Duty cycle type 설정 )**
  - Enable일 경우, Duty Cycle and Shift value in timer ticks (0 ~ Same value as Period value )
  - Disable일 경우, Duty Cycle and Shift value in percentage (0x0000 ~ 0x8000)

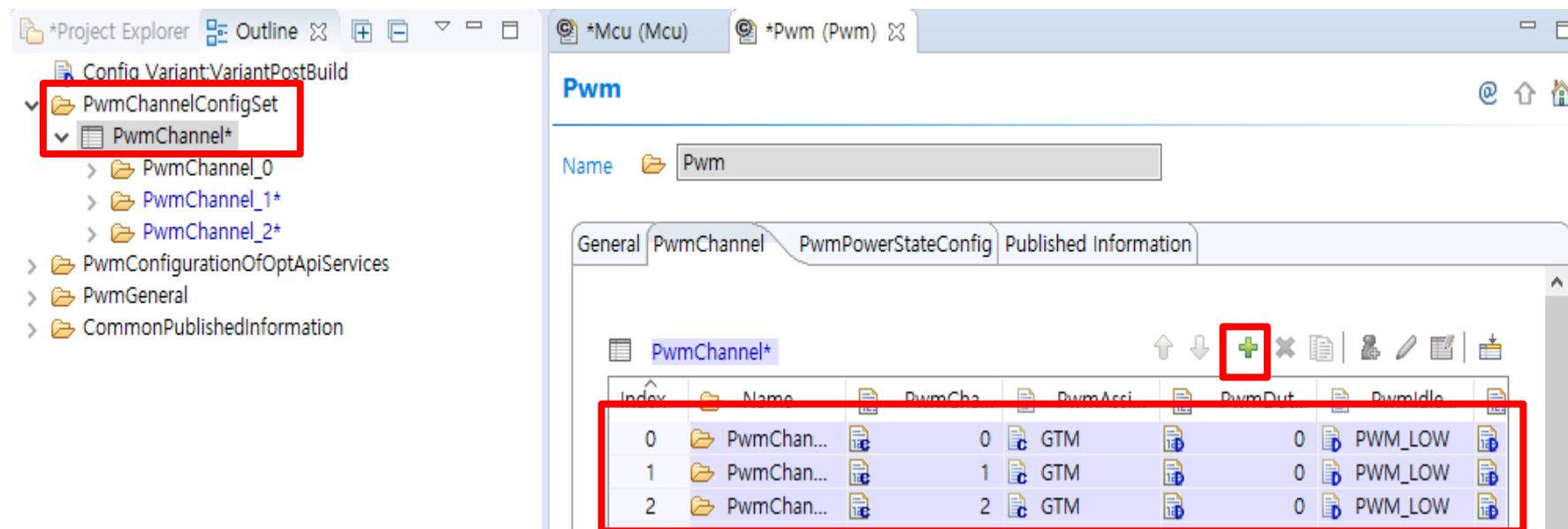


# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

### ▪ PwmChannelConfigSet

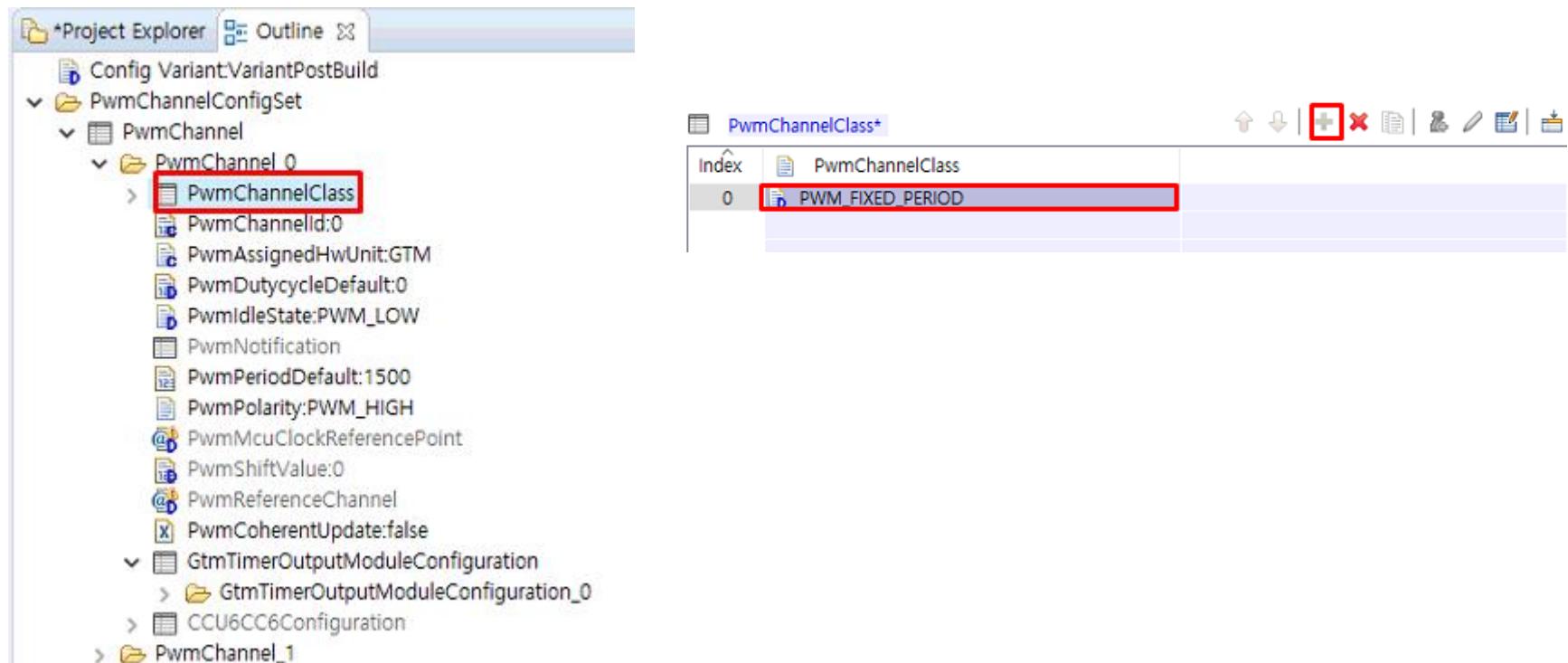
- 버튼(Add new element with default values) 클릭하여 3개의 채널 추가



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

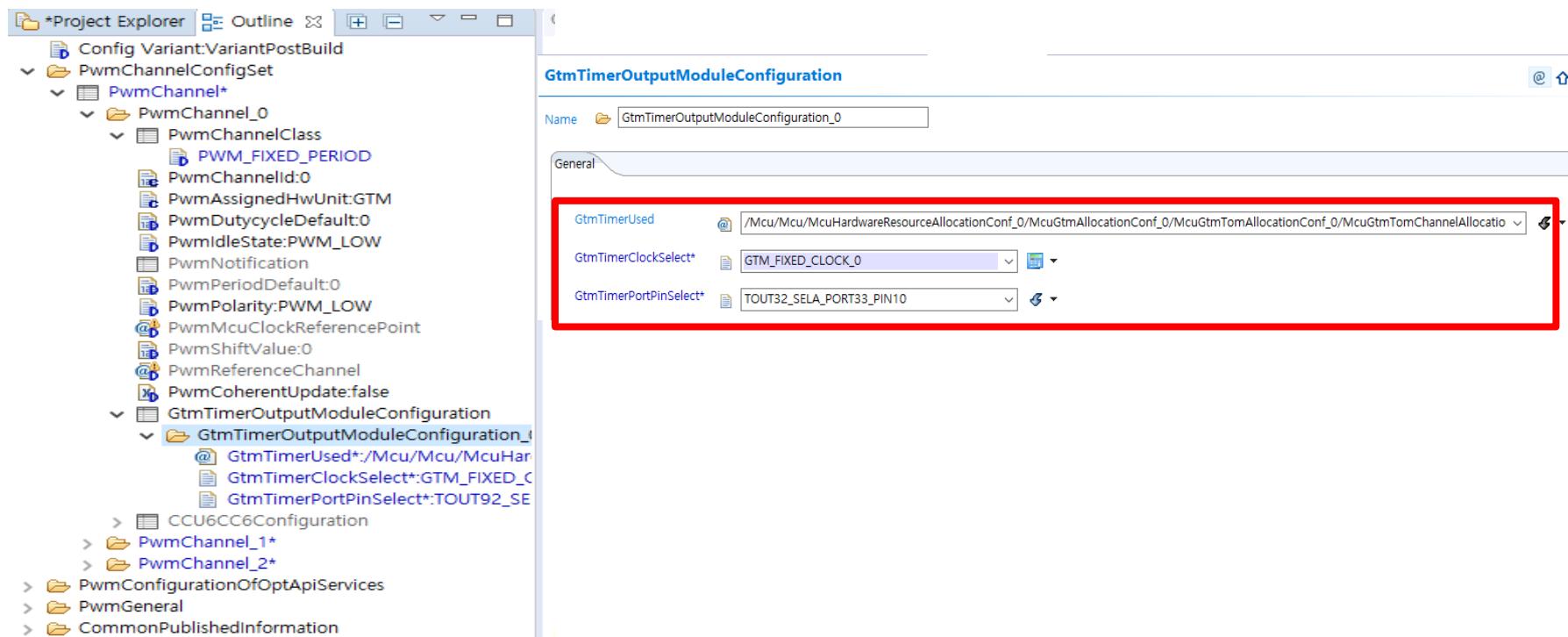
- **PwmChannelConfigSet** → **PwmChannel** → **PwmChannel\_0** → **PwmChannelClass**
  - + 클릭 후 **PWM\_FIXED\_PERIOD** 선택



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

- PwmChannelConfigSetWPwmChannelWPwmChannel\_0WGtmTimerOutputModuleConfigurationeConfigurationWGtmTimerOutputModuleConfiguration\_0
- GtmTimerUsed : McuGtmTomAllocationConf\_0WTomChannelAllocationConf\_0
- GtmTimerClockSelect : Gtm\_FIXED\_CLOCK\_0
- GtmTimerPortPinSelect : TOUT32\_SELA\_PORT33\_PIN10 선택



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

### ▪ PwmChannelConfigSetWPwmChannelWPwmChannel\_0

- PwmPeriodDefault : 20000 입력 -> ( Period값 입력 )
- PwmPolarity : PWM\_HIGH 선택 -> ( Duty 시간 동안의 High 의미 )

The screenshot shows the EB Tresos configuration interface. On the left, there is a tree view of the project structure:

- Config Variant: VariantPostBuild
- PwmChannelConfigSet
  - PwmChannel\*
    - PwmChannel\_0 (highlighted with a red box)
    - PwmChannel\_1
    - PwmChannel\_2\*
  - PwmConfigurationOfOptApiServices
  - PwmGeneral
  - CommonPublishedInformation

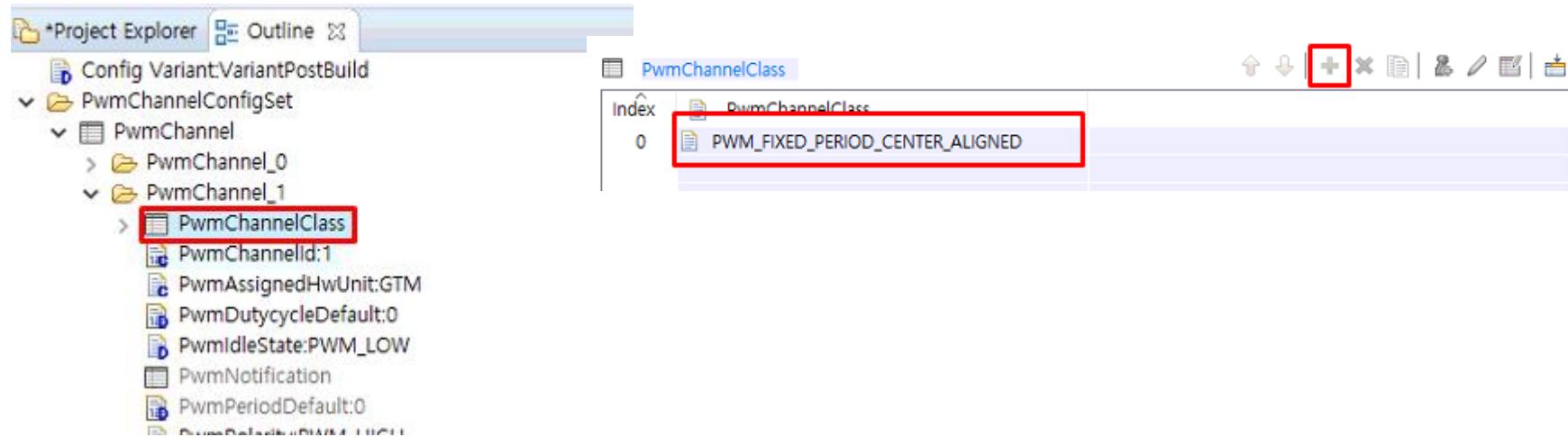
On the right, the configuration details for PwmChannel\_0 are displayed in a tabular form:

Setting	Value
PwmChannelId	0
PwmAssignedHwUnit	GTM
PwmDutyCycleDefault	0
PwmIdleState	PWM_LOW
PwmPeriodDefault	20000 (highlighted with a red box)
PwmPolarity	PWM_HIGH (highlighted with a red box)
PwmMcuClockReferencePoint	
PwmShiftValue	0
PwmReferenceChannel	
PwmCoherentUpdate	

# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

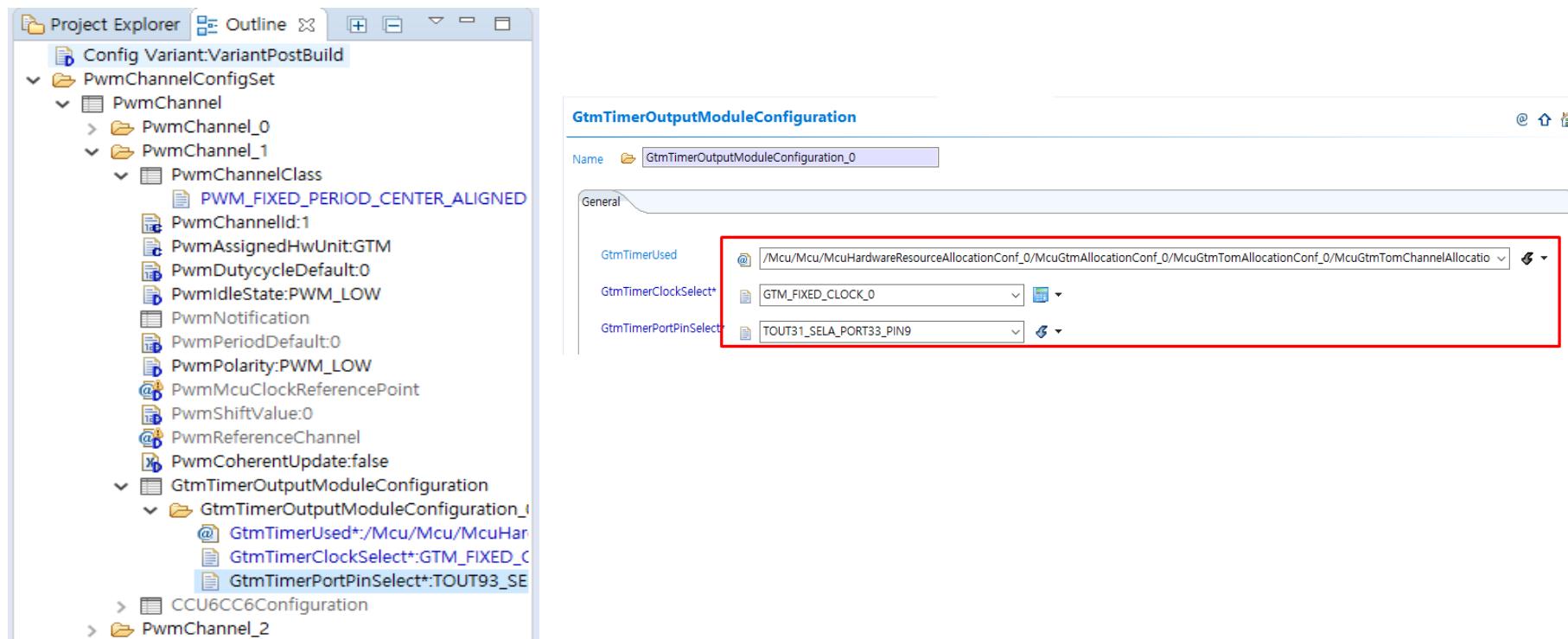
- **PwmChannelConfigSet** → **PwmChannel** → **PwmChannel\_1** → **PwmChannelClass**
  - + 클릭 후 **PWM\_FIXED\_CENTER\_ALIGNED** 선택



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

- **PwmChannelConfigSet** → **PwmChannel** → **PwmChannel\_1** → **GtmTimerOutputModuleConfiguration\_0**
  - GtmTimerUsed : McuGtmTomAllocationConf\_0 → TomChannelAllocationConf\_1
  - GtmTimerClockSelect : Gtm\_FIXED\_CLOCK\_0
  - GtmTimerPortPinSelect : TOUT31\_SELA\_PORT33\_PIN9 선택

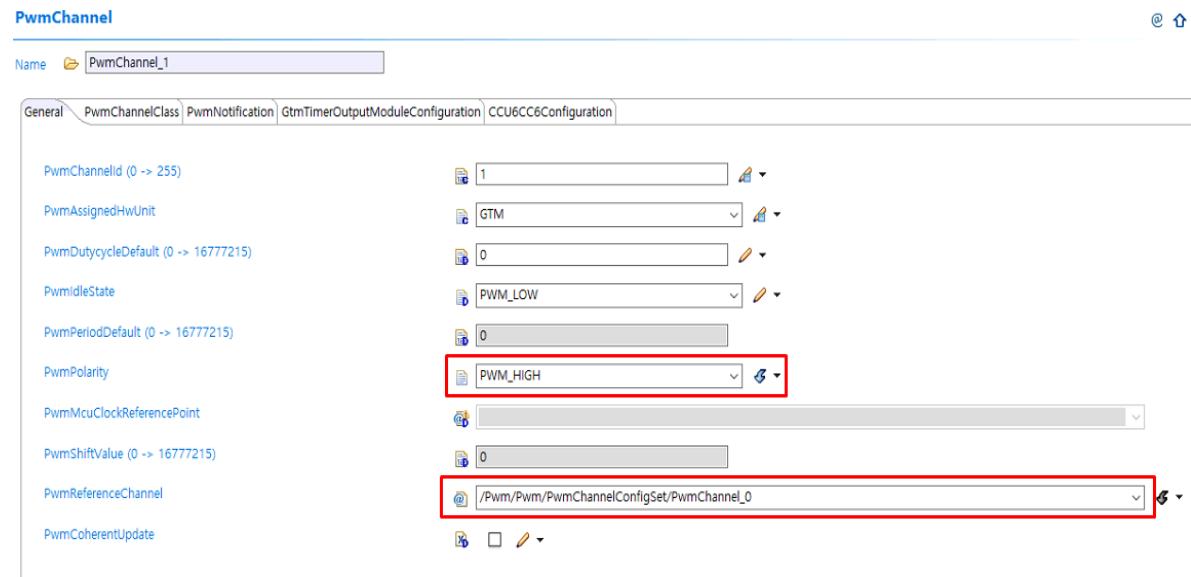
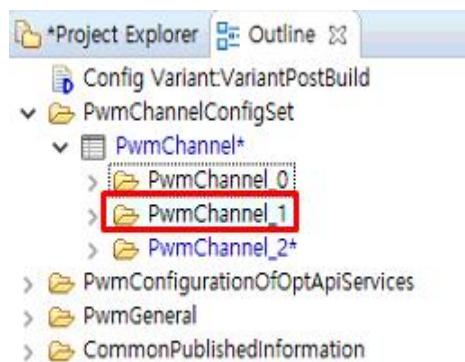


# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

- **PwmChannelConfigSet** → **PwmChannelConfigSet\_0** → **PwmChannel** → **PwmChannel\_1**

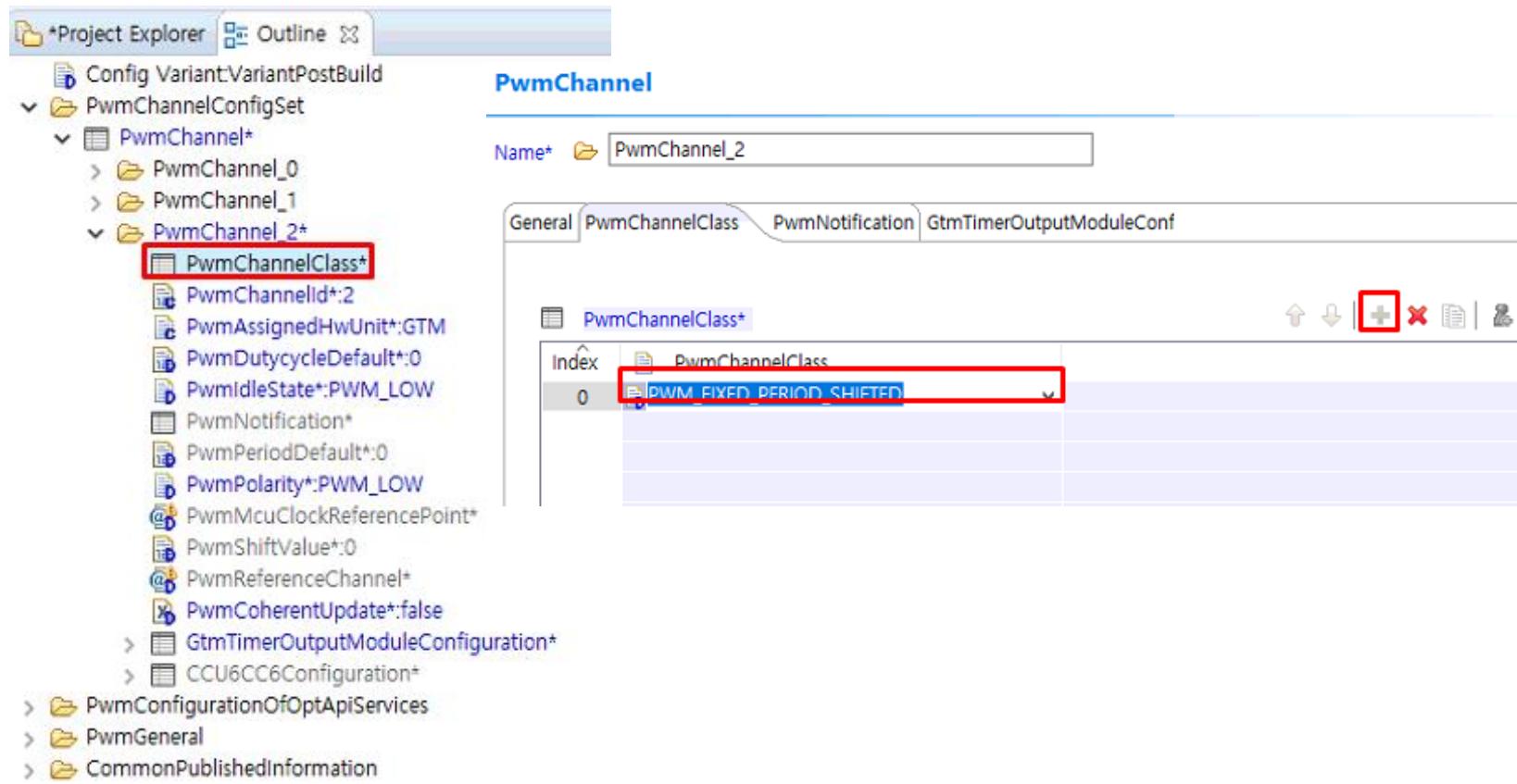
- PwmPolarity : PWM\_HIGH 선택 -> ( Duty 시간 동안의 High 의미 )
- PwmReferenceChannel : PwmChannel\_0 (기준 채널 설정)



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

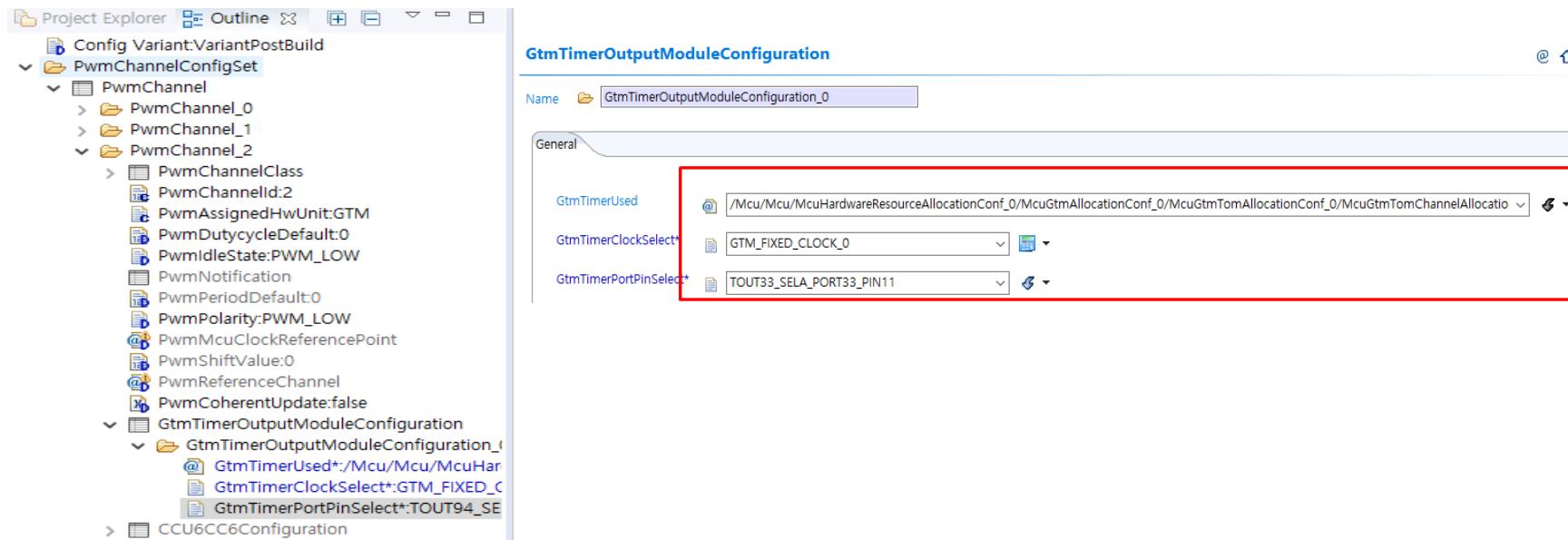
- PwmChannelConfigSet → PwmChannel → PwmChannel\_2 → PwmChannelClass
- + 클릭 후 PWM\_FIXED\_PERIOD\_SHIFTED 선택



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

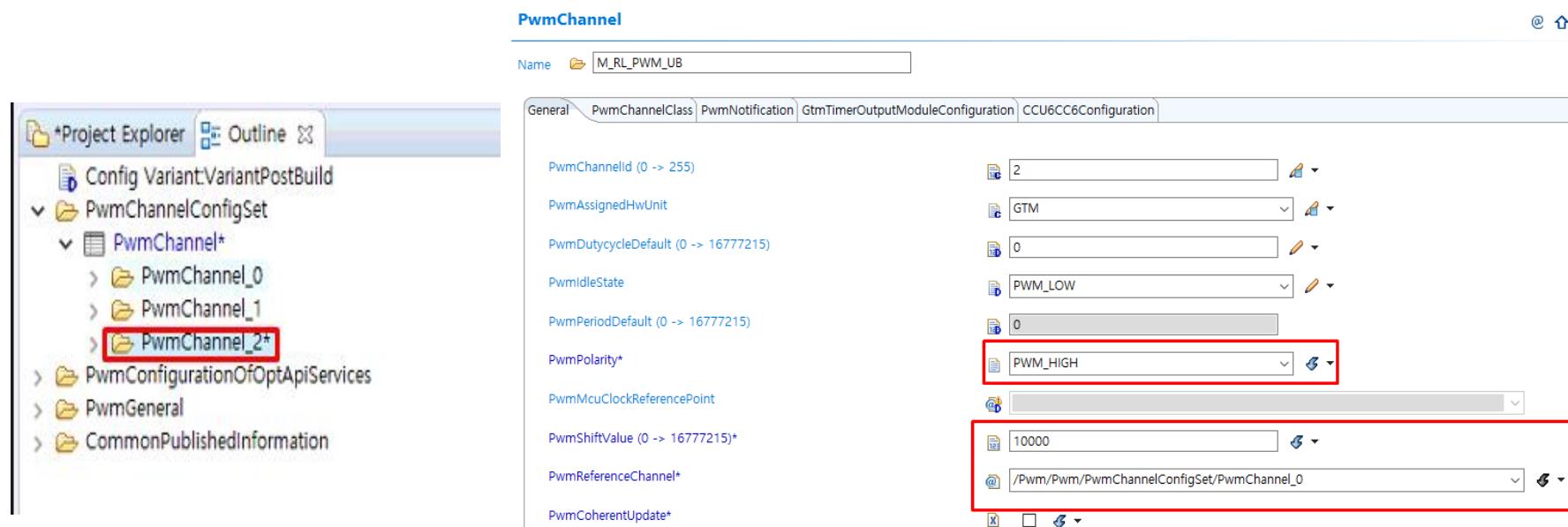
- **PwmChannelConfigSet** → **PwmChannel** → **PwmChannel\_2** → **GtmTimerOutputModuleConfiguration\_0**
  - GtmTimerUsed : McuGtmTomAllocationConf\_0 → TomChannelAllocationConf\_2
  - GtmTimerClockSelect : Gtm\_FIXED\_CLOCK\_0
  - GtmTimerPortPinSelect : TOUT33\_SELE\_PORT33\_PIN11 선택



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Pwm 모듈 설정

- **PwmChannelConfigSet** → **PwmChannelConfigSet\_0** → **PwmChannel** → **PwmChannel\_2**
  - PwmPolarity : PWM\_HIGH 선택 -> ( Duty 시간 동안의 High 의미 )
  - PwmShiftValue : 10000 (이동값 입력)
  - PwmReferenceChannel : PwmChannel\_0 (기준 채널 설정)

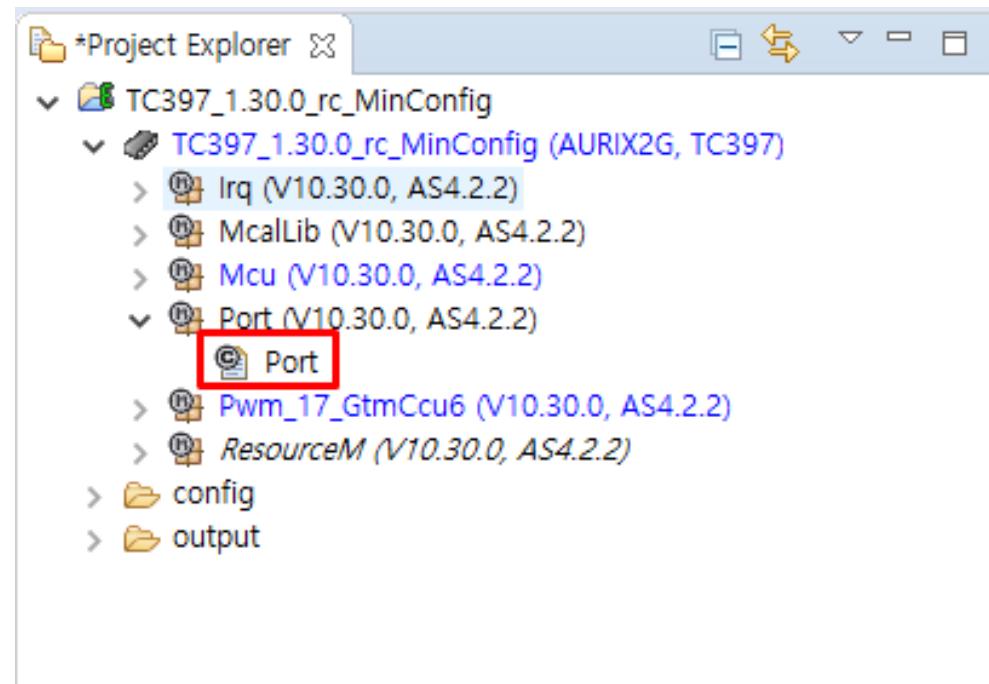


# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Port 모듈 설정

### ▪ Port 더블 클릭

- PORT\_33\_PIN\_10
  - ✓ PwmChannel0
- PORT\_33\_PIN\_9
  - ✓ PwmChannel1
- PORT\_33\_PIN\_11
  - ✓ PwmChannel2

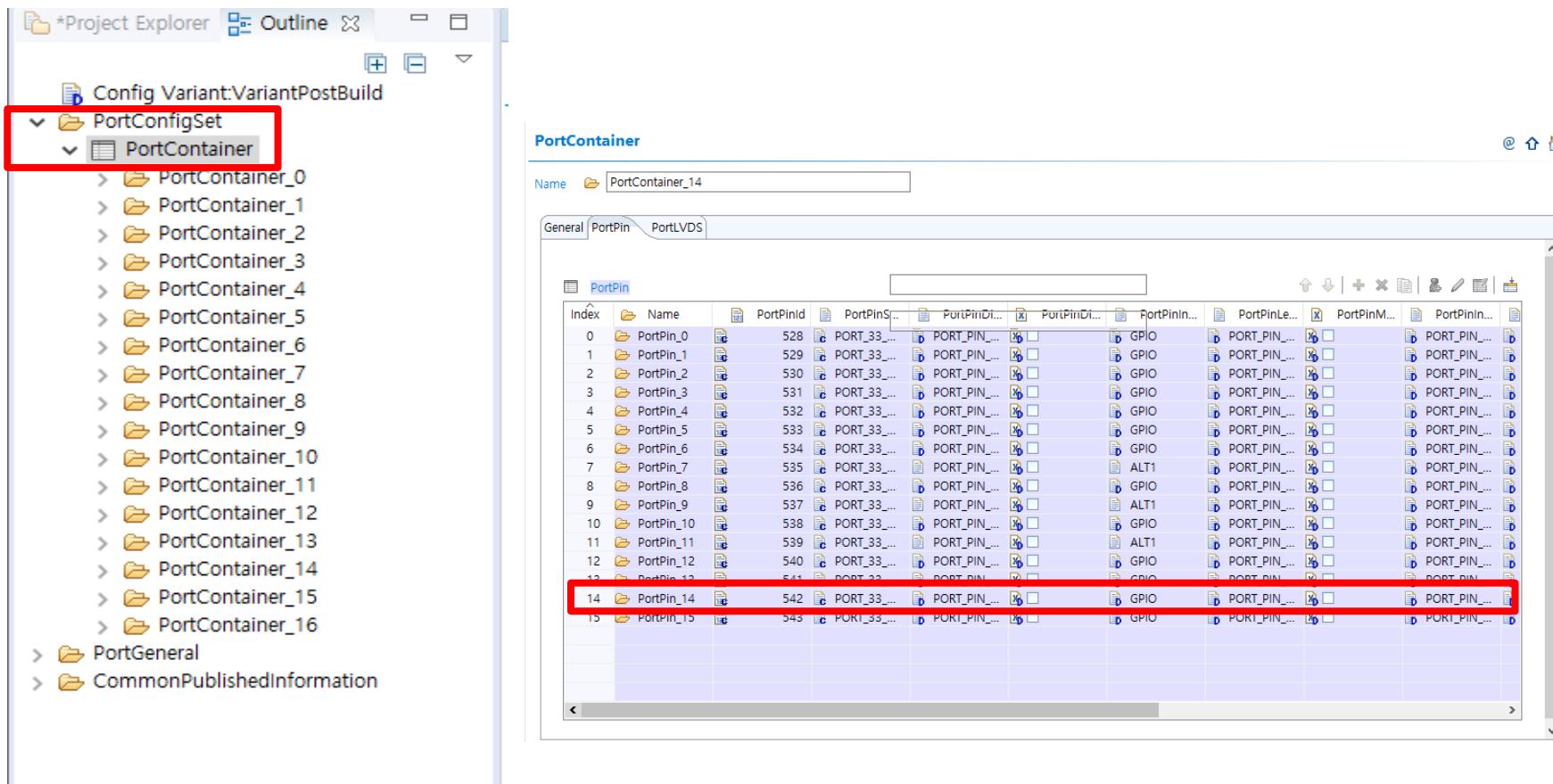


# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Port 모듈 설정

### ▪ PortConfigSet W PortContainer

- PortContainer\_14로 진입( PortNumber : 33 )



# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Port 모듈 설정

### ▪ PortConfigSet ➔ PortContainer ➔ PortContainer\_14 ➔ PortPin

- PortPinDirection : PortPin\_9,10,11 -> PORT\_PIN\_OUT 선택
- PortPinInitialMode : PortPin\_9,10,11 -> ALT1 선택

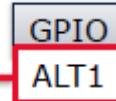
Index	Name	PortPinId	PortPinSymbolicName	PortPinDirection	PortPinDigitalValue	PortPinInitialMode	PortPinLevel
0	PortPin_0	528	PORT_33_PIN_0	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
1	PortPin_1	529	PORT_33_PIN_1	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
2	PortPin_2	530	PORT_33_PIN_2	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
3	PortPin_3	531	PORT_33_PIN_3	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
4	PortPin_4	532	PORT_33_PIN_4	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
5	PortPin_5	533	PORT_33_PIN_5	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
6	PortPin_6	534	PORT_33_PIN_6	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
7	PortPin_7	535	PORT_33_PIN_7	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
8	PortPin_8	536	PORT_33_PIN_8	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
9	PortPin_9	537	PORT_33_PIN_9	PORT_PIN_OUT	0	ALT1	PORT_PIN_OUT
10	PortPin_10	538	PORT_33_PIN_10	PORT_PIN_OUT	0	ALT1	PORT_PIN_OUT
11	PortPin_11	539	PORT_33_PIN_11	PORT_PIN_OUT	0	ALT1	PORT_PIN_OUT
12	PortPin_12	540	PORT_33_PIN_12	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
13	PortPin_13	541	PORT_33_PIN_13	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
14	PortPin_14	542	PORT_33_PIN_14	PORT_PIN_IN	0	GPIO	PORT_PIN_IN
15	PortPin_15	543	PORT_33_PIN_15	PORT_PIN_IN	0	GPIO	PORT_PIN_IN

# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ Port 모듈 설정

- PortPinInitialMode : PORT\_PIN\_MODE\_ALT1 설정 이유

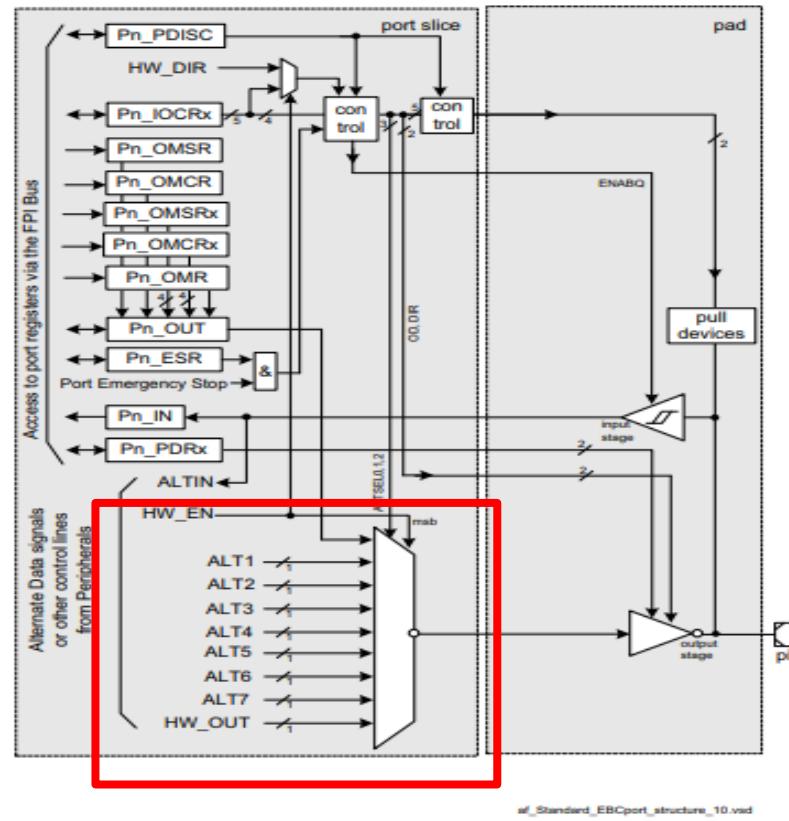
Table 481 Example Port Table

Pin	Symbol	Ctrl.	Buffer Type	Function	
10	Pxx.y	00	FAST / PU1 / VEXT	General-purpose input	
	TIMm_n			GTM_TIN	
	Pxx.y			General-purpose output	
	TOMa_b	01		GTM_TOUT	
	TOMc_d	GTM_TOUT			
	IOM_REFv_w	IOM reference input		ALT2	
	ASCLINz_RTS	02		ASCLIN0 output	ALT3
	...	...		...	ALT4
	ETH0_MDIO	0		Ethernet output	ALT5
					ALT6
					ALT7

# Exercise PWM – EB Tresos Configurations (cont.)

## ➤ General Structure of a Port Pin

- MUX 구조로 포트 설정



# EcuM - EB Tresos Configurations (cont.)

## ➤ EcuM 모듈 설정

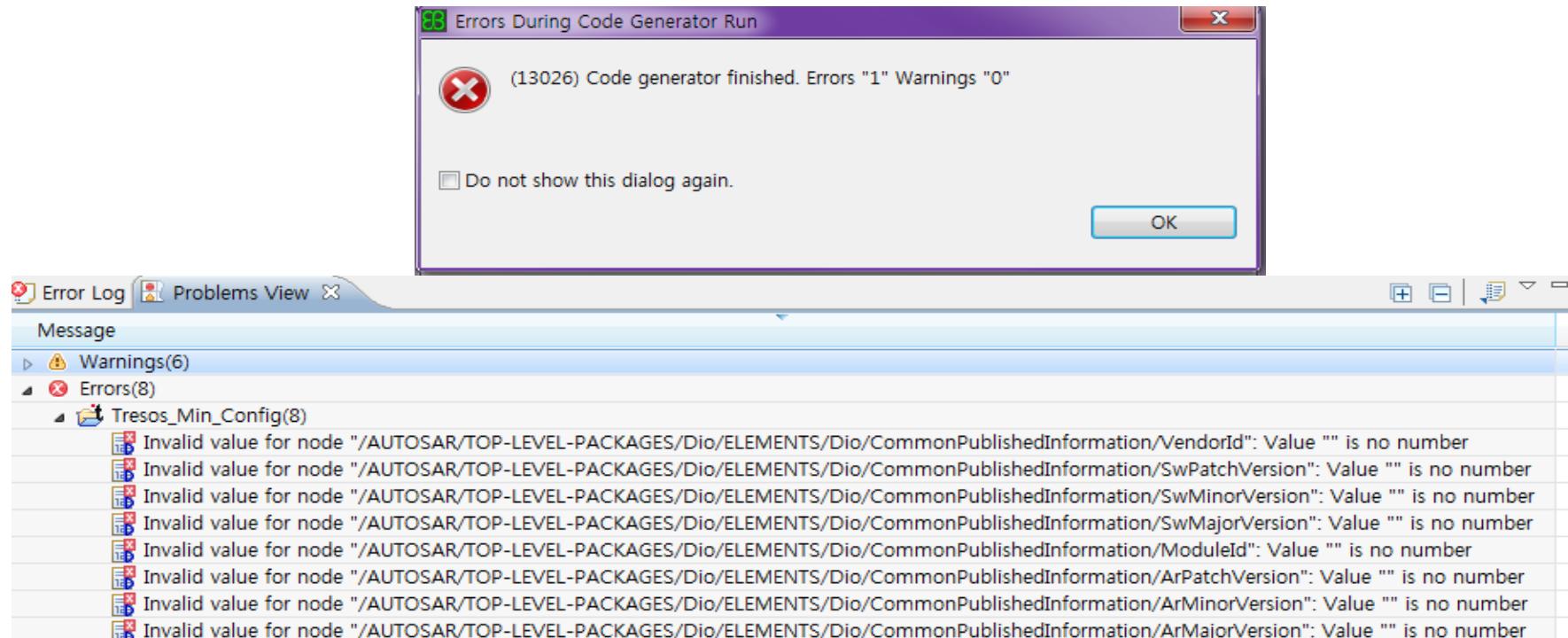
- 1G에는 EcuM 드라이버를 통해 Driver 초기화를 사용
  - 2G의 TC39x Tresos 및 MC-ISAR에서는 EcuM 드라이버를 설정할 수 없음
    - ✓ User가 Tasking Tool에서 초기화 코드를 작성해주어야 함.

# Build Generation Error - EB Tresos Configurations (cont.)

## ➤ Save and Code generation

- 저장 후, generate code  를 클릭
  - Error "0" Warings "0" 일 경우, 코드 생성이 완료

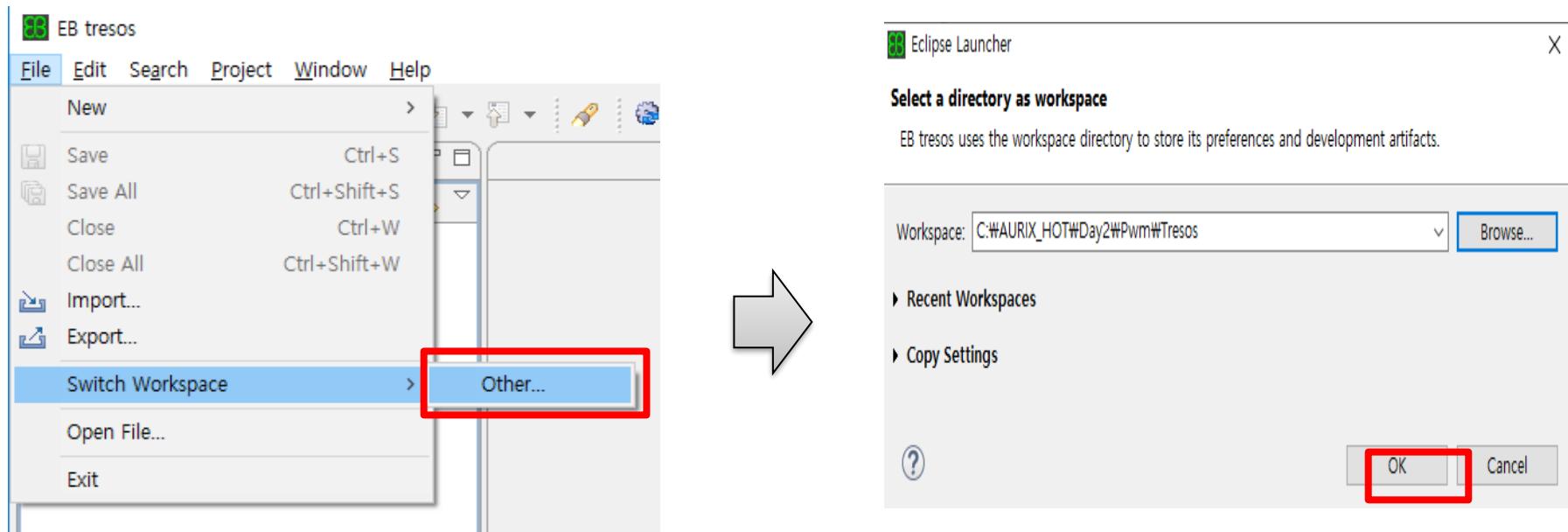
- 만약, 다음과 같은 Error가 발생 하면 재실행



# Build Generation Error - EB Tresos Configurations (cont.)

## ➤ Save and Code generation

- File > Switch Workspace > Other를 클릭.
- Workspace Launcher에서 OK 클릭.



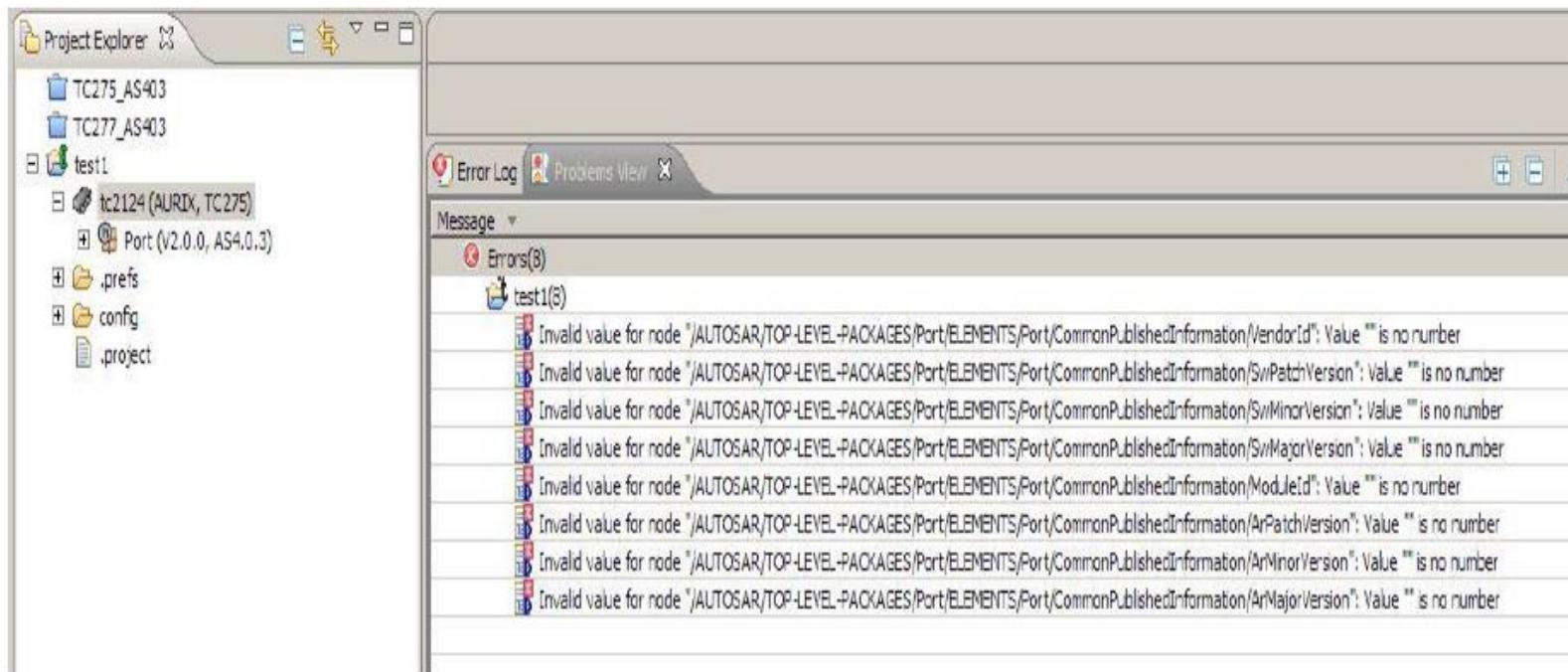
- EB tresos program이 재실행되면서 Error가 발생하지 않음.
- 또는 EB tresos를 종료 후 재실행해도 Error가 발생하지 않음.

# Build Generation Error - EB Tresos Configurations (cont.)

## ➤ Save and Code generation

### ▪ Tresos issue

- This document explains the known issue found in the tresos tool v13.0.0
  - ✓ When creating a new project in tresos workspace the following problem is reported due to the container CommonPublishedInformation



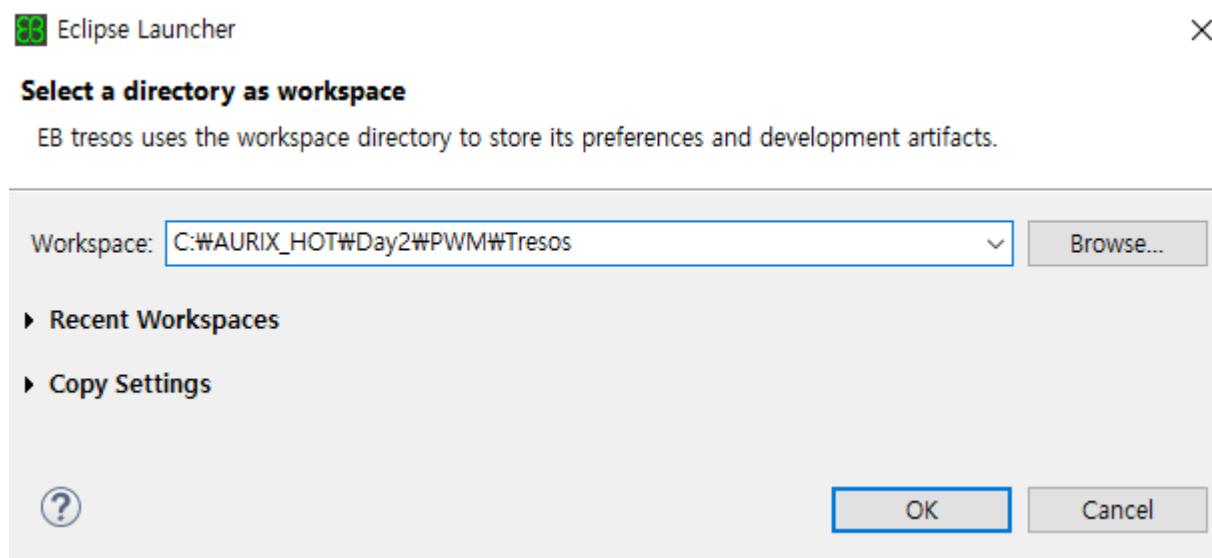
참조 : ReadMeTresosIssue.pdf

# Build Generation Error - EB Tresos Configurations (cont.)

## ➤ Save and Code generation

- This issue can be fixed by reopening the same project.

- Click File -> Switch Workspace -> Other을 클릭한 후, 바로 OK를 다시 누른다. 그러면 Tresos가 재실행 되면서 Error가 발생 하지 않는다.



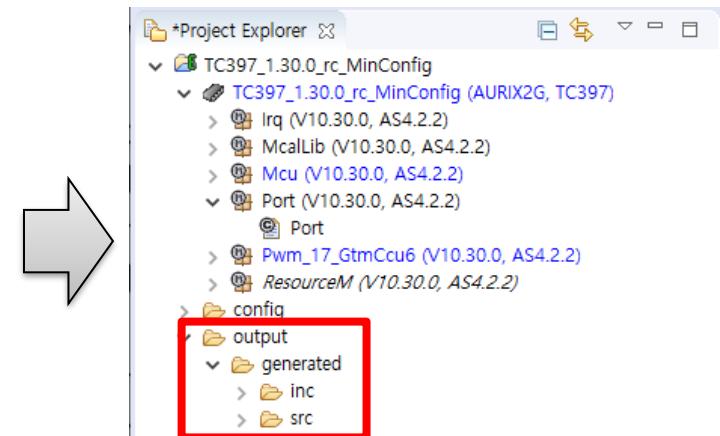
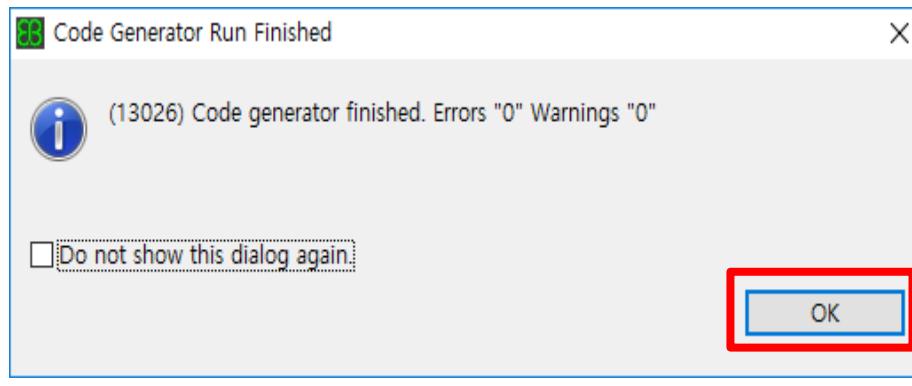
- 또는, Tresos를 종료 한 후 다시 실행 시켜도 Error를 없앨 수 있다.

# Build Generation Error - EB Tresos Configurations (cont.)

## ➤ Save and Code generation

### ▪ Regenerate code 를 클릭

- Error "0" Warings "0" 일 경우, 코드 생성이 완료  
✓ Output 폴더 생성 됨



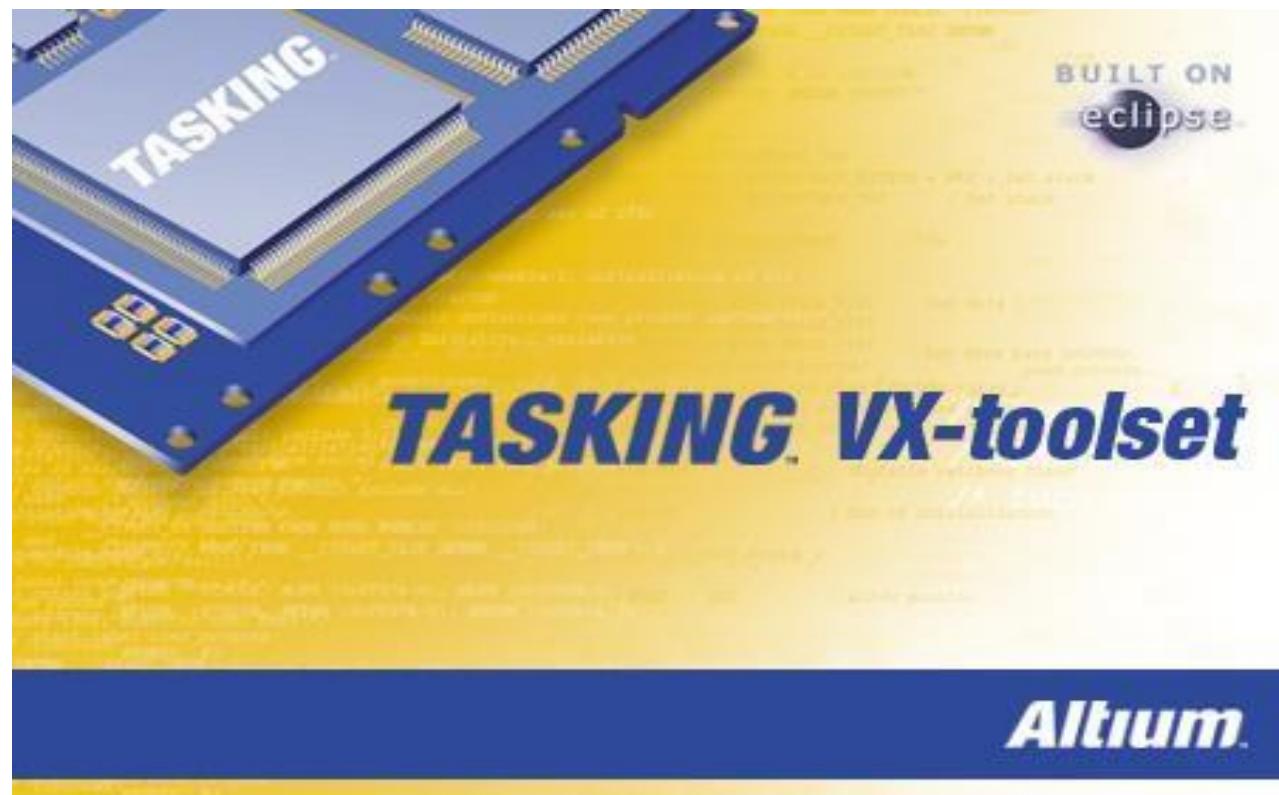
## PWM 실습

1. Objective
2. Tresos 설정
3. Tasking 설정
4. PLS 디버깅

# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Tasking 실행

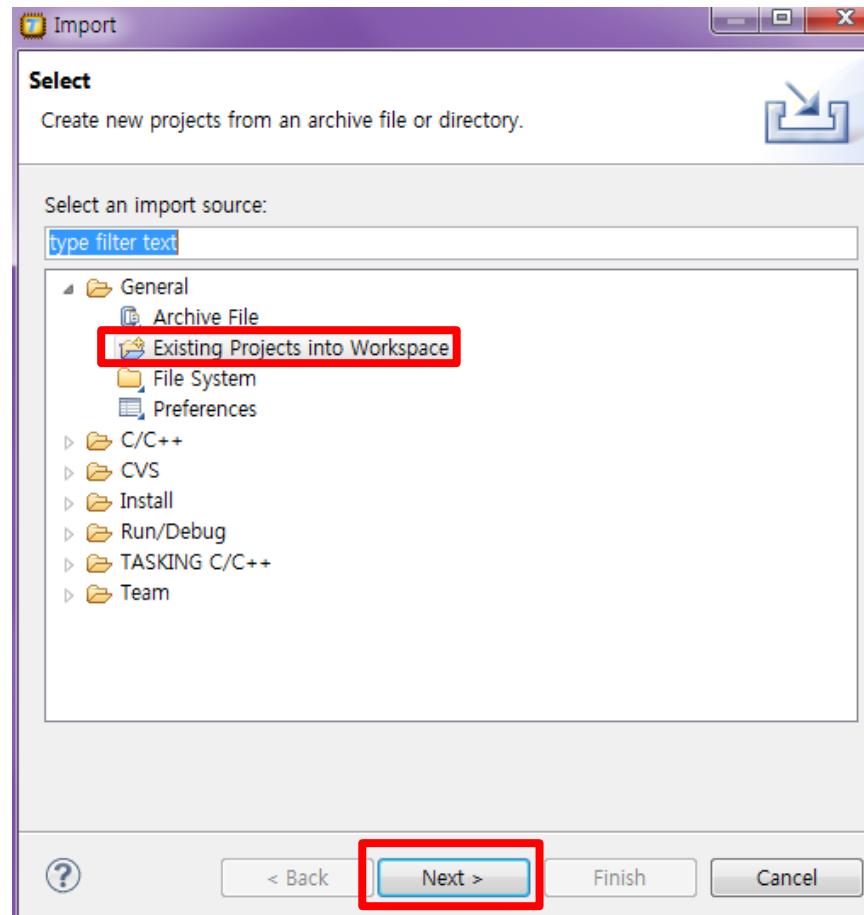
-  클릭



# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Project Import

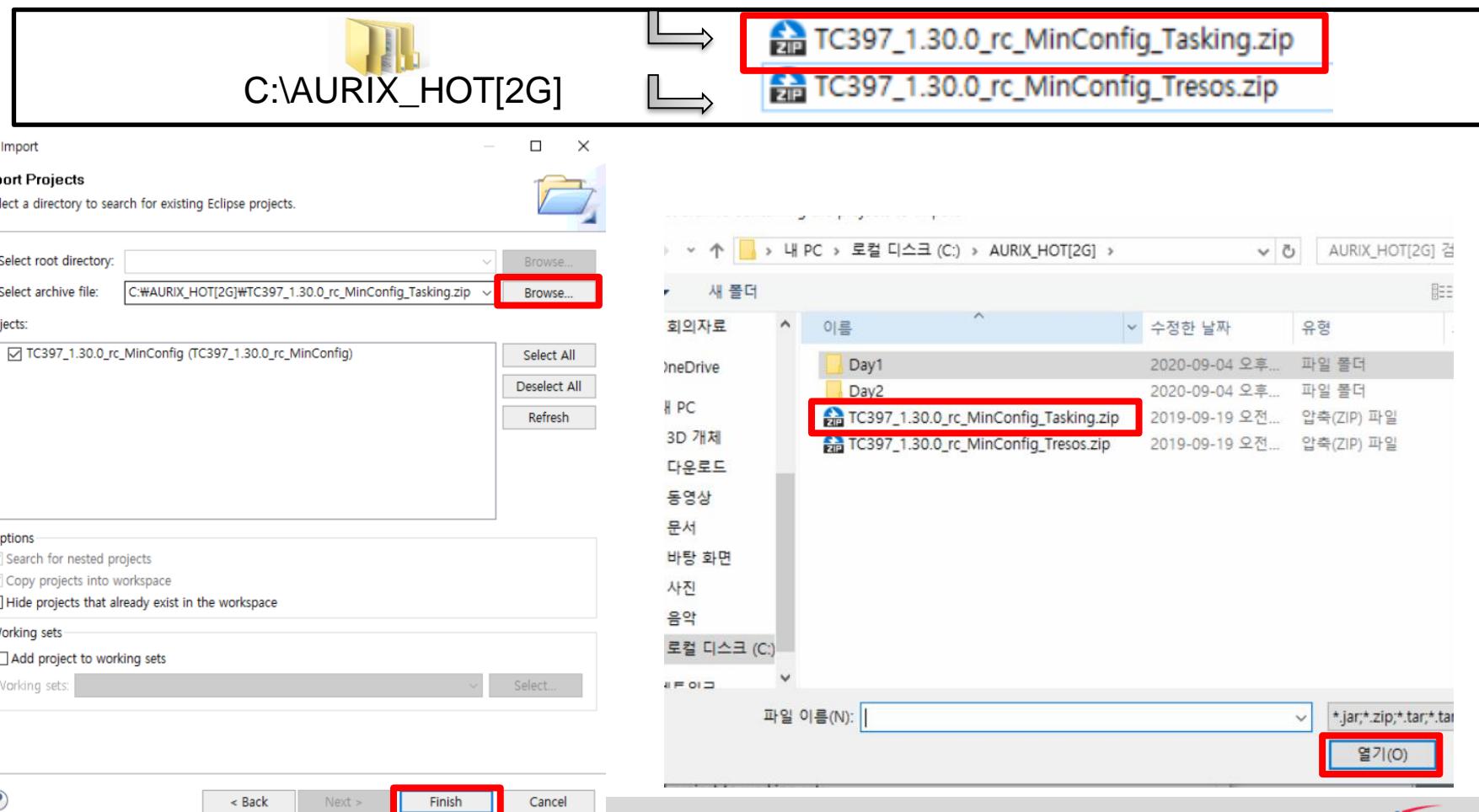
- General -> Existing Projects into Workspace 선택 -> Next 클릭



# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Project Import

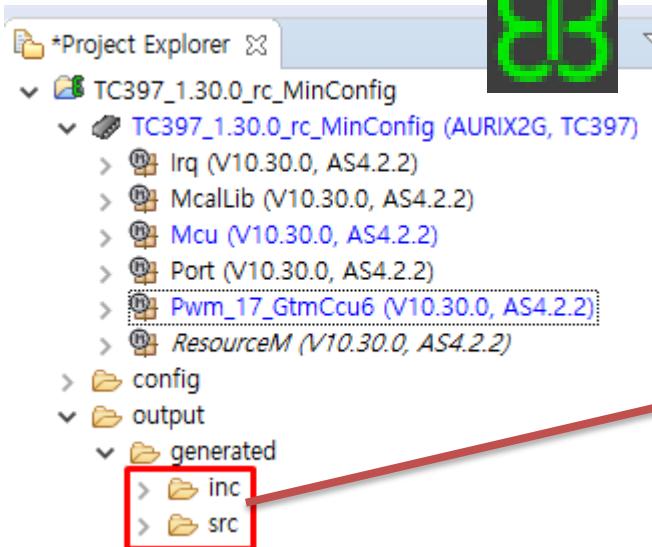
- Select archive file 선택 -> Browse 클릭
  - 배포한 TC397\_1.30.0\_rc\_MinConfig\_Tasking.zip 선택 -> 열기



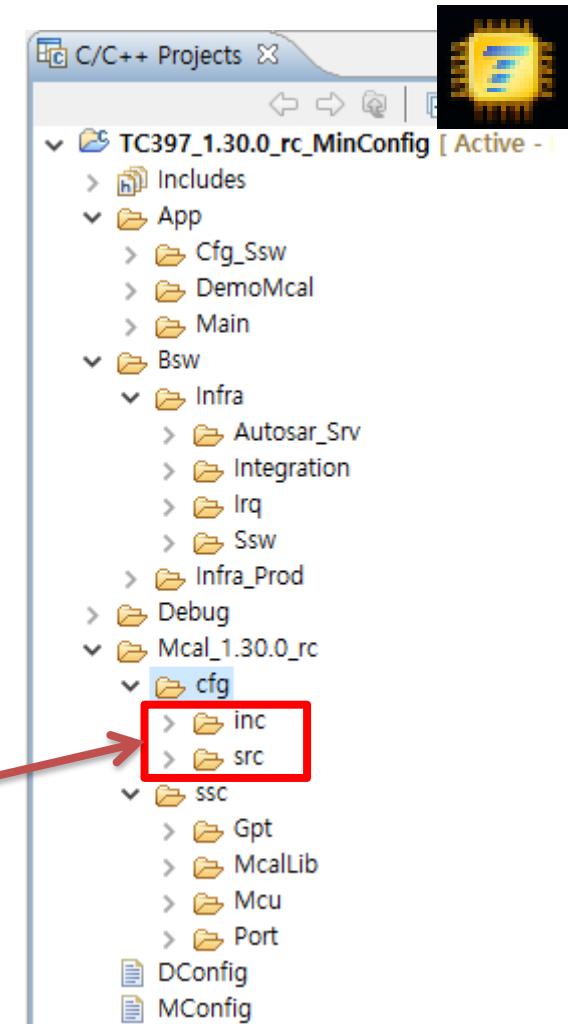
# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Tresos에서 생성한 Code를 Tasking 경로에 복사

- Tresos에서
  - inc / src 복사
- Tasking에서
  - SourceCode/Mcal\_1.30.0\_rc/cfg 경로에 붙여 넣기
- Tresos에서 생성 된, C파일과 h 파일은  
Tresos에서 설정한 여러 가지 값들을 코드  
로 생성 하여, Tasking에서 참조 할 수 있도록  
구성 되어 있다.

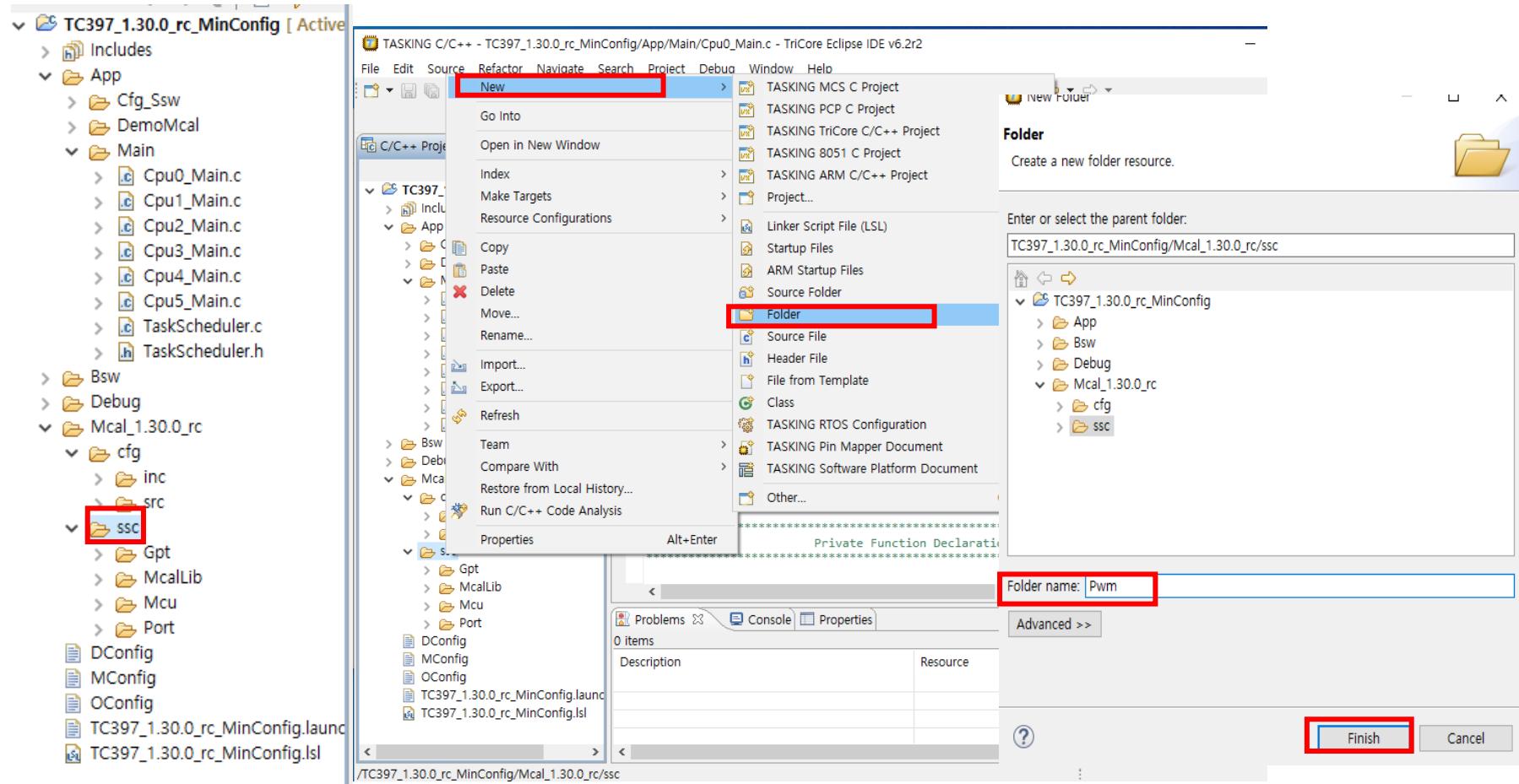


복사/붙여넣기



# Exercise PWM – Tasking VX-toolset (cont.)

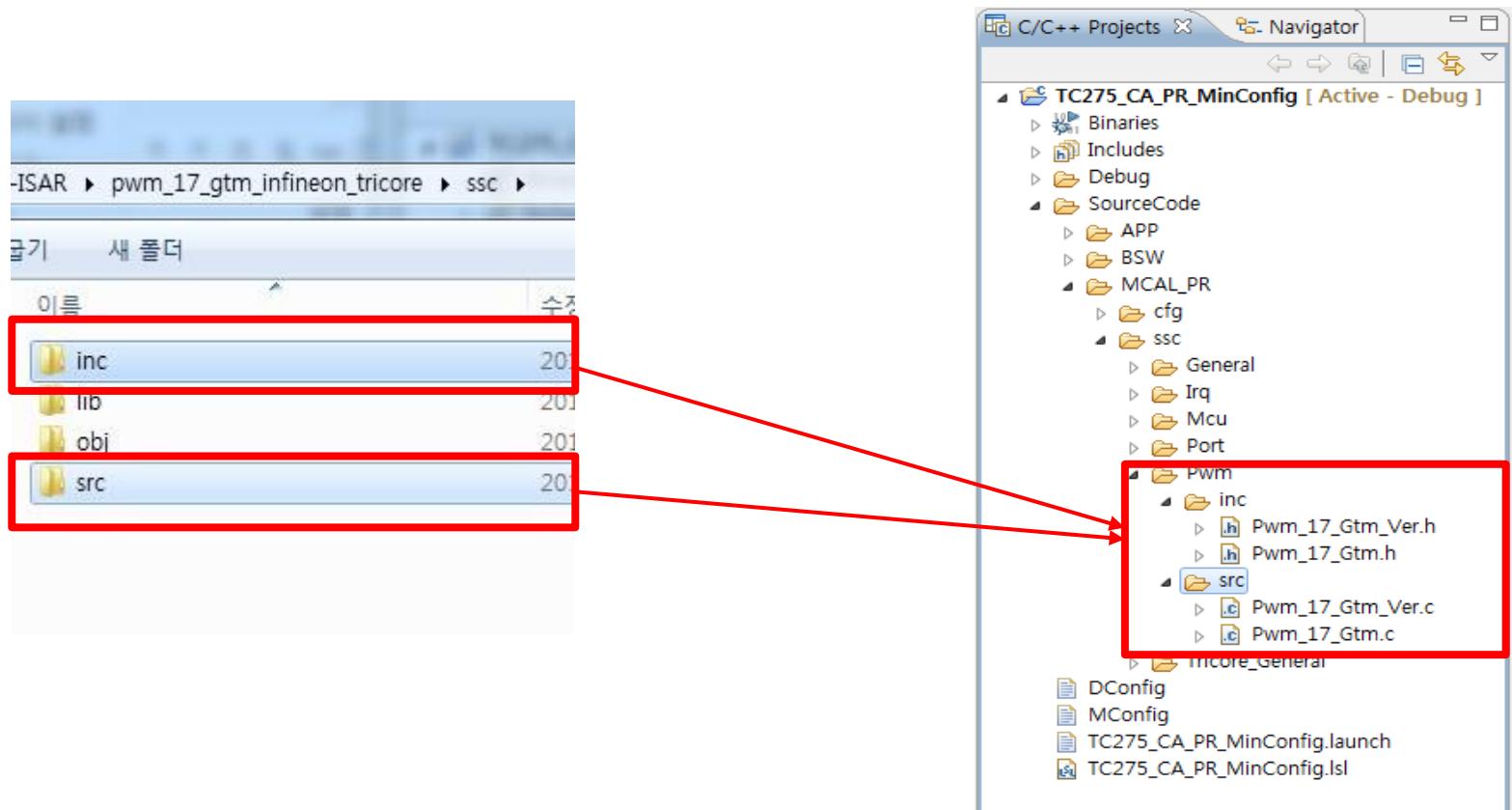
- Pwm Api를 사용하기 위해 MCAL에서 제공 하는 C파일과 Header파일 추가
  - SourceCode/MCAL\_1.30.0\_rc/ssc에 Pwm 폴더 추가



# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Pwm Api를 사용하기 위해 MCAL에서 제공 하는 C파일과 Header파일 추가

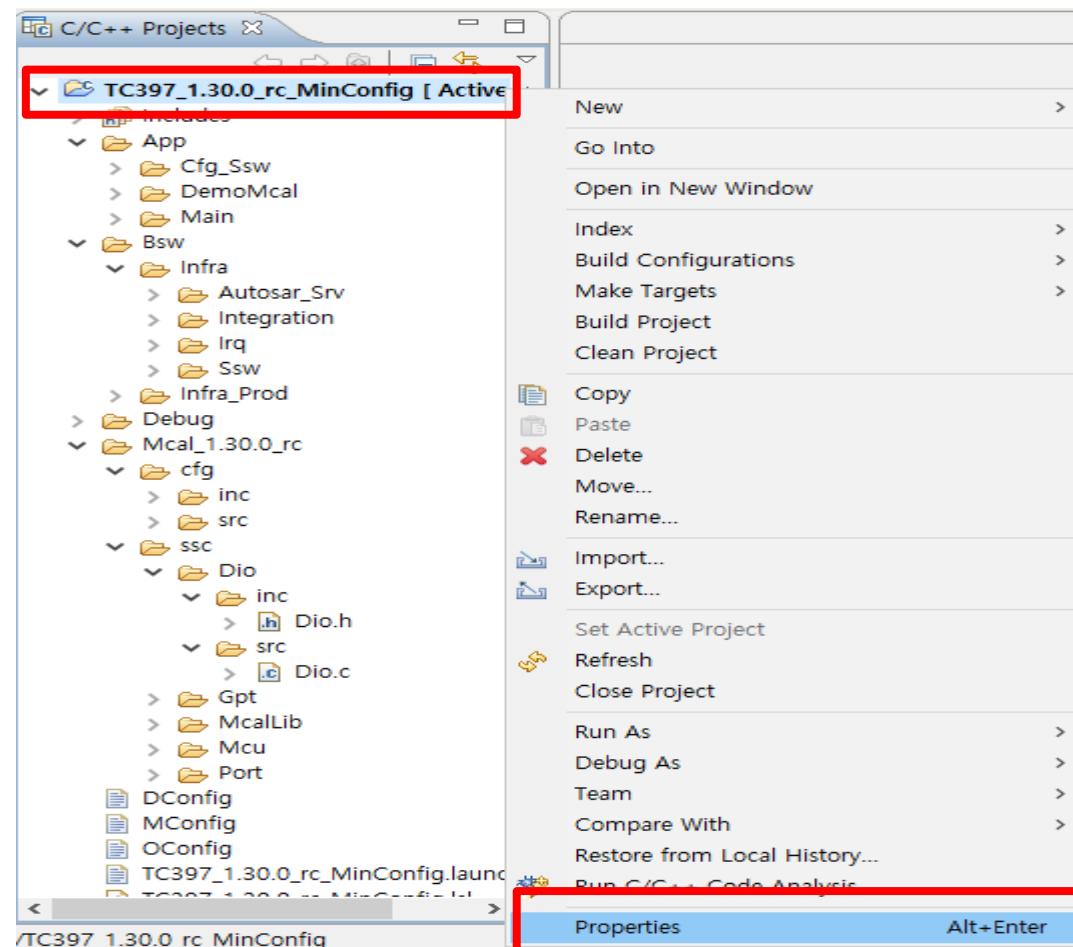
- C:\WEB\Wtresos2\WMC-ISAR\_AS42x\_AURIX2G\_TC38xA\_TC39xB\_TC35xA\_TC37xA\_1.30.0-rc\WMcIsar\WSrc\WMcal\WTricore\WPwm\_17\_GtmCcu6\Wssc
- inc와 src 복사 -> SourceCode/MCAL/ssc에 Pwm 폴더에 붙여넣기



# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Pwm Path 설정

- TC397\_1.30.0\_rc\_Minconfig 에서 마우스 우클릭
  - Properties 클릭

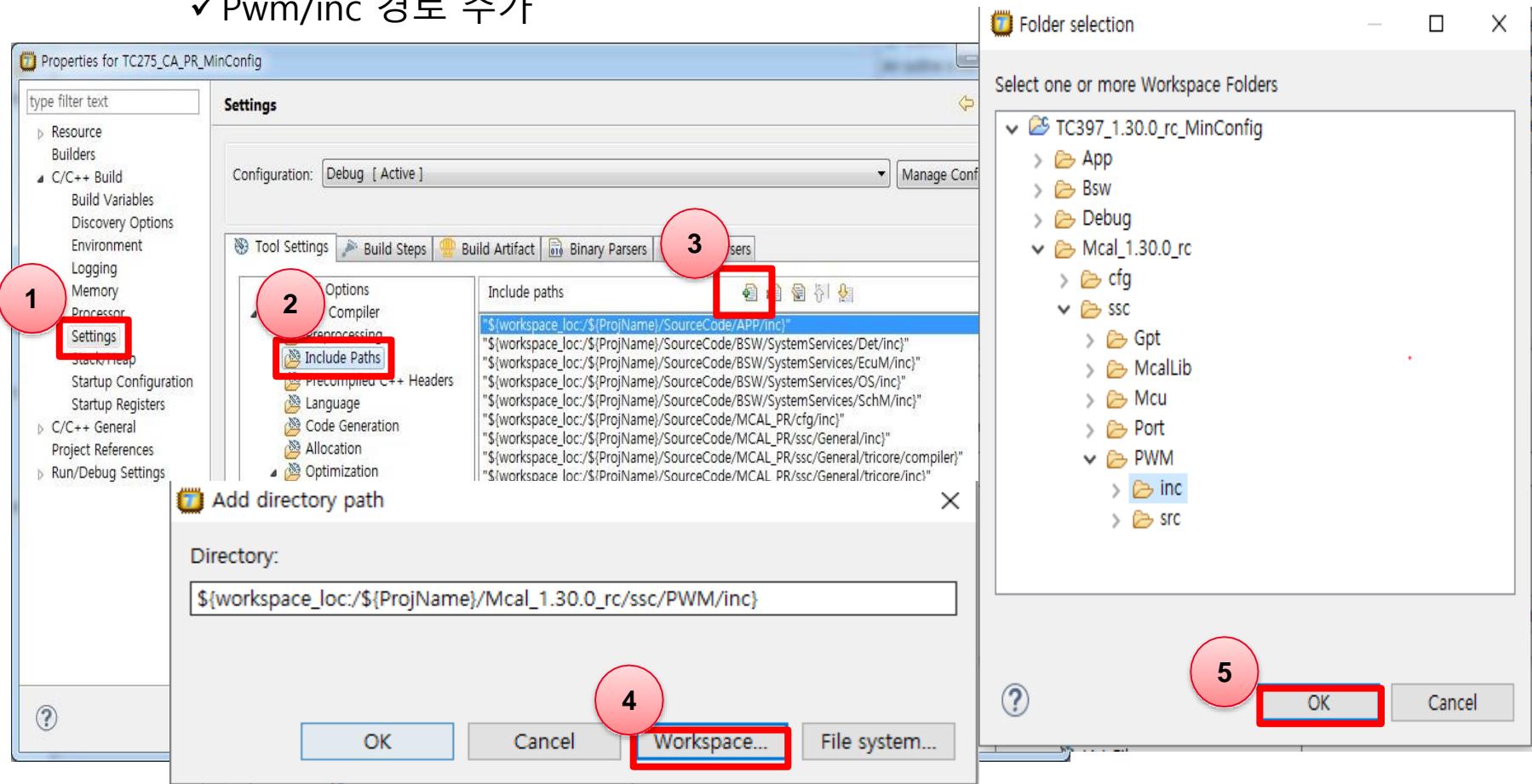


# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Pwm Path 설정

### ▪ C/C++ Build -> Settings

- C/C++ Compiler -> Include Paths  
✓ Pwm/inc 경로 추가

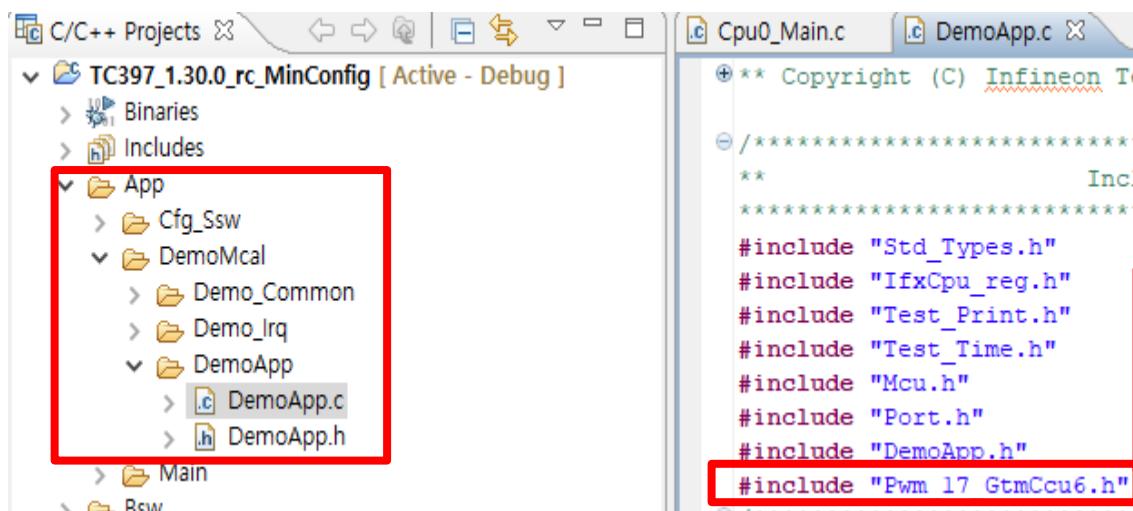


# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ DemoApp.c 초기화 코드 작성

### ▪ DemoApp 작성

- App → DemoMcal → DemoApp
  - ✓ void DemoApp\_Pwm\_17\_GtmCcu6\_Init(void);
  - ✓ DemoApp\_Pwm\_17\_GtmCcu6\_Init();
  - ✓ void DemoApp\_Pwm\_17\_GtmCcu6\_Init(void){
  - ✓ const Pwm\_17\_GtmCcu6\_ConfigType \*ConfigPtr = NULL\_PTR;
  - ✓ ConfigPtr = &Pwm\_17\_GtmCcu6\_Config;
  - ✓ Pwm\_17\_GtmCcu6\_Init(ConfigPtr);



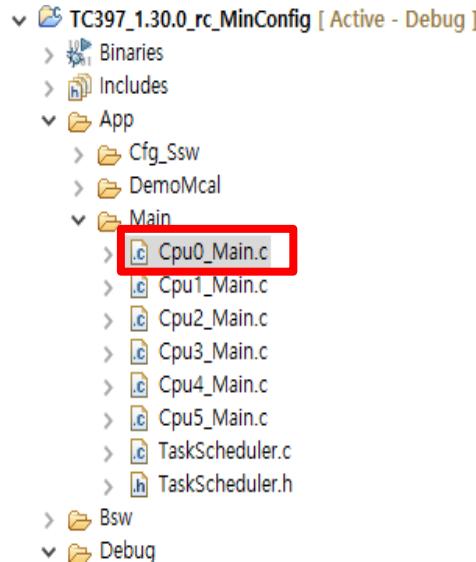
```
void DemoApp_Init(void);
void DemoApp_Mcu_Init(void);
void DemoApp_Port_Init(void);
void DemoApp_Pwm_17_GtmCcu6_Init(void);

void DemoApp_Init(void)
{
    DemoApp_Mcu_Init();
    DemoApp_Port_Init();
    DemoApp_Pwm_17_GtmCcu6_Init();
}

void DemoApp_Pwm_17_GtmCcu6_Init(void)
{
    const Pwm_17_GtmCcu6_ConfigType *ConfigPtr = NULL_PTR;
    ConfigPtr = &Pwm_17_GtmCcu6_Config;
    Pwm_17_GtmCcu6_Init(ConfigPtr);
}
```

# Exercise PWM – Tasking VX-toolset (cont.)

- Cpu0\_Main.c 소스 코드 작성
  - #include "Pwm\_17\_GtmCcu6.h" 추가



```
** Copyright (C) Infineon Technologies (2016)
** Includes
****

#include "Ifx_Ssw_Infra.h"
#include "IFX_Os.h"

#include "TaskScheduler.h"
#include "IfxSrc_reg.h"
#include "DemoApp.h"
#include "Gpt.h"
#include "Irq.h"

#include "Pwm_17_GtmCcu6.h"

**** Imported Compiler Switch Check ****

**** Private Macro Definitions ****
```

# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Dead Time 구현

### ▪ Pwm 파형 생성

- 채널 주기 설정
- ✓ SetDutyCycle Api 사용

```
uint16 duty = 10000;
```

```
Pwm_17_GtmCcu6_SetDutyCycle(Pwm_17_GtmCcu6Conf_PwmChannel_PwmChannel_0,duty);  
Pwm_17_GtmCcu6_SetDutyCycle(Pwm_17_GtmCcu6Conf_PwmChannel_PwmChannel_1,duty);  
Pwm_17_GtmCcu6_SetDutyCycle(Pwm_17_GtmCcu6Conf_PwmChannel_PwmChannel_2,duty);
```



### ▪ Dead Time Module enable

- DTM 모듈 clock 설정
- Tom 채널 경로 enable
- Shadow Register Enable

```
GTM_CDTM0_DTM0_CTRL.U = 0x40;
```

```
GTM_CDTM0_DTM0_CH_CTRL2.B.DTO_1 = 1;  
GTM_CDTM0_DTM0_CH_CTRL2.B.DTO_2 = 1;
```

```
GTM_CDTM0_DTM0_CH_CTRL2_SR.B.DTO_1_SR = 1;  
GTM_CDTM0_DTM0_CH_CTRL2_SR.B.DTO_2_SR = 1;
```



### ▪ Dead Time Module 적용

- Dead time 범위 : 0~2^10-1
- Dead time을 적용할 edge 설정
  - ✓ Rising edge or falling edge

```
uint16 delayTime = 1000;
```

```
GTM_CDTM0_DTM0_CH1_DTV.B.RELRISE = delayTime;  
GTM_CDTM0_DTM0_CH2_DTV.B.RELRISE = delayTime;
```

# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Cpu0\_Main.c 소스 코드 작성

### ▪ Cpu0\_Main.c 작성

- 각 채널 별 duty 설정

- ✓ uint16 duty = 10000;
- ✓ Pwm\_17\_GtmCcu6\_SetDutyCycle(Pwm\_17\_GtmCcu6Conf\_PwmChannel\_PwmChannel\_0,duty);
- ✓ Pwm\_17\_GtmCcu6\_SetDutyCycle(Pwm\_17\_GtmCcu6Conf\_PwmChannel\_PwmChannel\_1,duty);
- ✓ Pwm\_17\_GtmCcu6\_SetDutyCycle(Pwm\_17\_GtmCcu6Conf\_PwmChannel\_PwmChannel\_2,duty);

```
uint16 duty = 10000;

void core0_main (void)
{
    unsigned short cpuWdtPassword;
    unsigned short safetyWdtPassword;

    ENABLE();
    /*
     * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdog in the demo if it is required and also service the watchdog periodically
     */
    cpuWdtPassword = Ifx_Ssw_getCpuWatchdogPassword(&MODULE_SCU.WDTCPU[0]);
    safetyWdtPassword = Ifx_Ssw_getSafetyWatchdogPassword();
    Ifx_Ssw_disableCpuWatchdog(&MODULE_SCU.WDTCPU[0], cpuWdtPassword);
    Ifx_Ssw_disableSafetyWatchdog(safetyWdtPassword);

    DemoApp_Init();          // Initialize Mcu and Port module

    Pwm_17_GtmCcu6_SetDutyCycle(Pwm_17_GtmCcu6Conf_PwmChannel_PwmChannel_0, duty);
    Pwm_17_GtmCcu6_SetDutyCycle(Pwm_17_GtmCcu6Conf_PwmChannel_PwmChannel_1, duty);
    Pwm_17_GtmCcu6_SetDutyCycle(Pwm_17_GtmCcu6Conf_PwmChannel_PwmChannel_2, duty);
```

# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Cpu0\_Main.c 소스 코드 작성

### ▪ Cpu0\_Main.c 작성

- 레지스터를 이용한 Dead Time 적용

```
uint16 delayTime = 1000;
```

```
GTM_CDTM0_DTM0_CTRL.U = 0x40;
```

DTM 모듈과 TOM이 연결되기 때문에  
TOM 모듈에서 사용하는 fixed clock으로 설정

```
GTM_CDTM0_DTM0_CH_CTRL2.B.DT0_1 = 1;  
GTM_CDTM0_DTM0_CH_CTRL2.B.DT0_2 = 1;
```

Dead Time을 적용할 채널(Tom Channel)  
경로 Enable

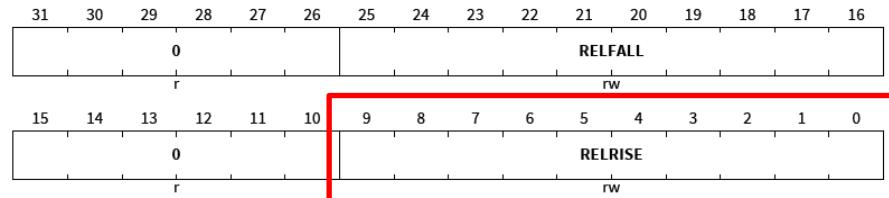
```
GTM_CDTM0_DTM0_CH_CTRL2_SR.B.DT0_1_SR = 1;  
GTM_CDTM0_DTM0_CH_CTRL2_SR.B.DT0_2_SR = 1;
```

Dead Time 값을 저장할 Shadow  
Register Enable

```
GTM_CDTM0_DTM0_CH1_DTV.B.RELRISE = delayTime;  
GTM_CDTM0_DTM0_CH2_DTV.B.RELRISE = delayTime;
```

Rising edge일 때 delay시킬 tick 수  
(범위: 0 ~  $2^{10}-1$ )

```
while(1)  
{  
    __nop();  
}
```

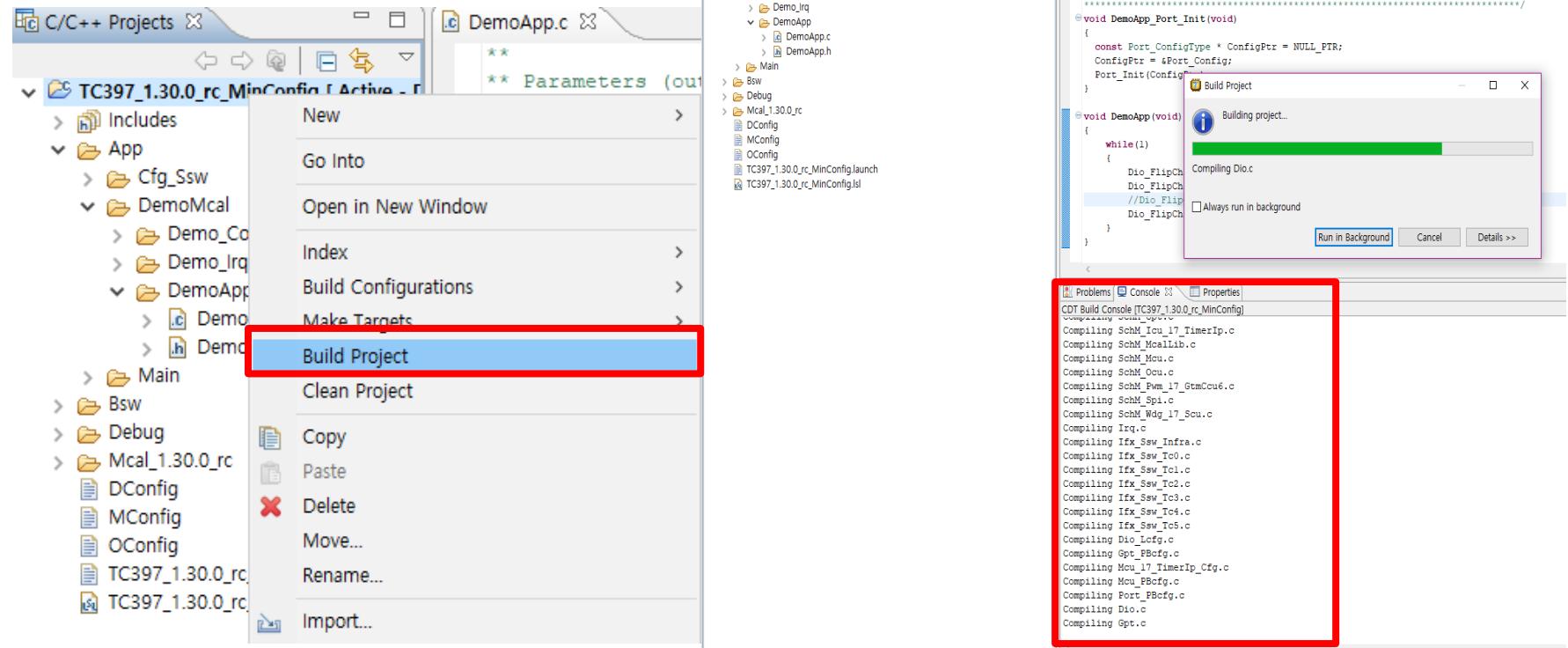


# Exercise PWM – Tasking VX-toolset (cont.)

## ➤ Project Build

### ▪ TC397\_1.30.0\_rc\_MinConfig에서 마우스 우클릭

- Build Project 또는  클릭
- 빌드 완료 시, ..\Debug 폴더에 \*.elf 파일 생성됨

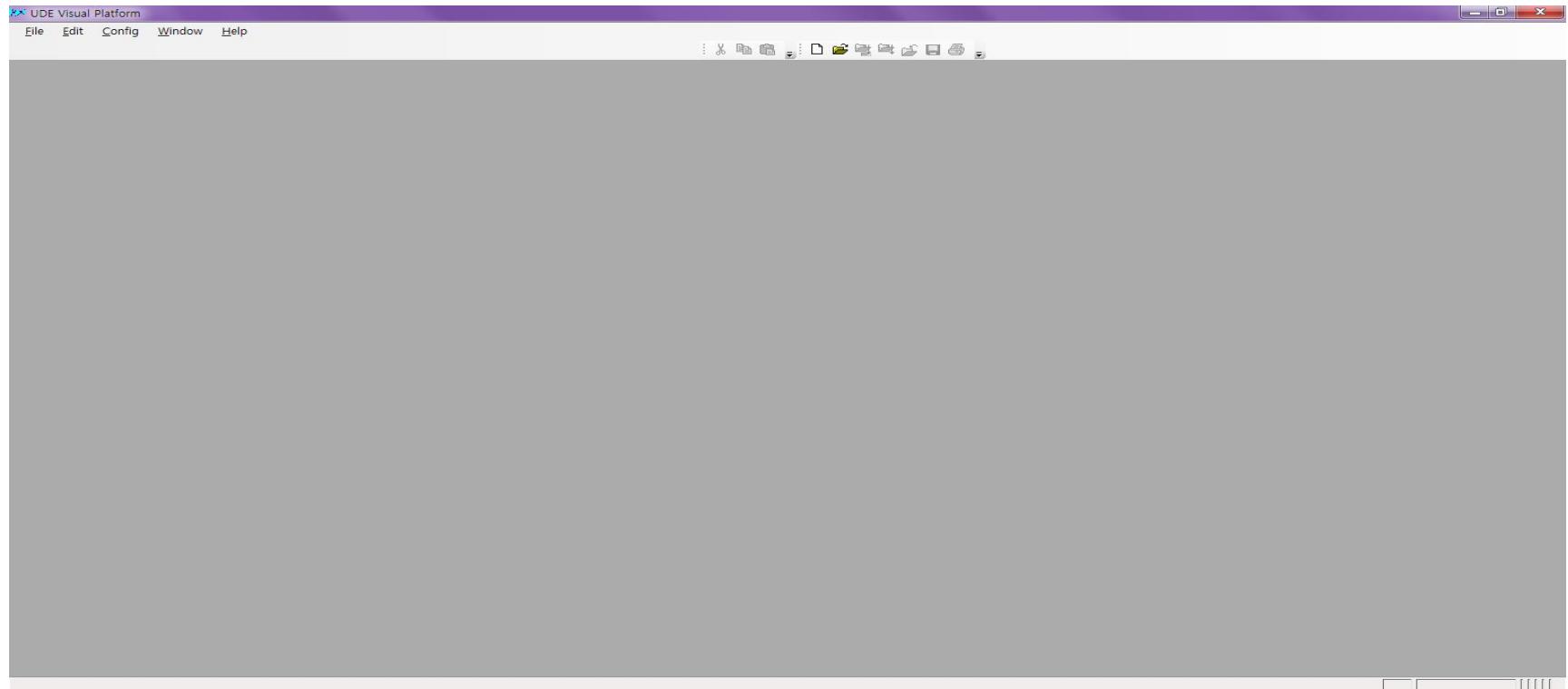


## PWM 실습

1. Objective
2. Tresos 설정
3. Tasking 설정
4. PLS 디버깅

## ➤ Tresos 실행

-  클릭

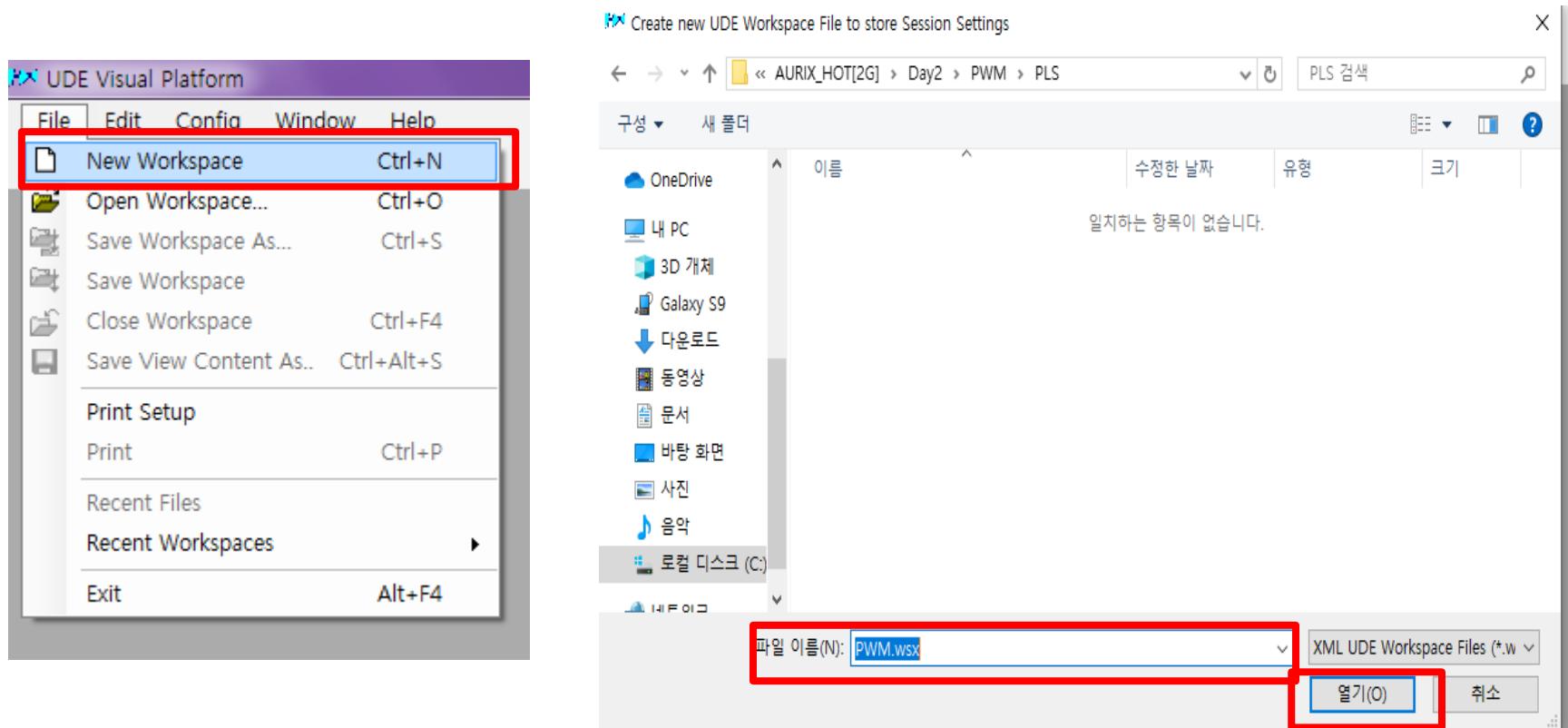


# Exercise PWM – PLS

## ➤ Workspace 생성

### ▪ File -> New Workspace 클릭

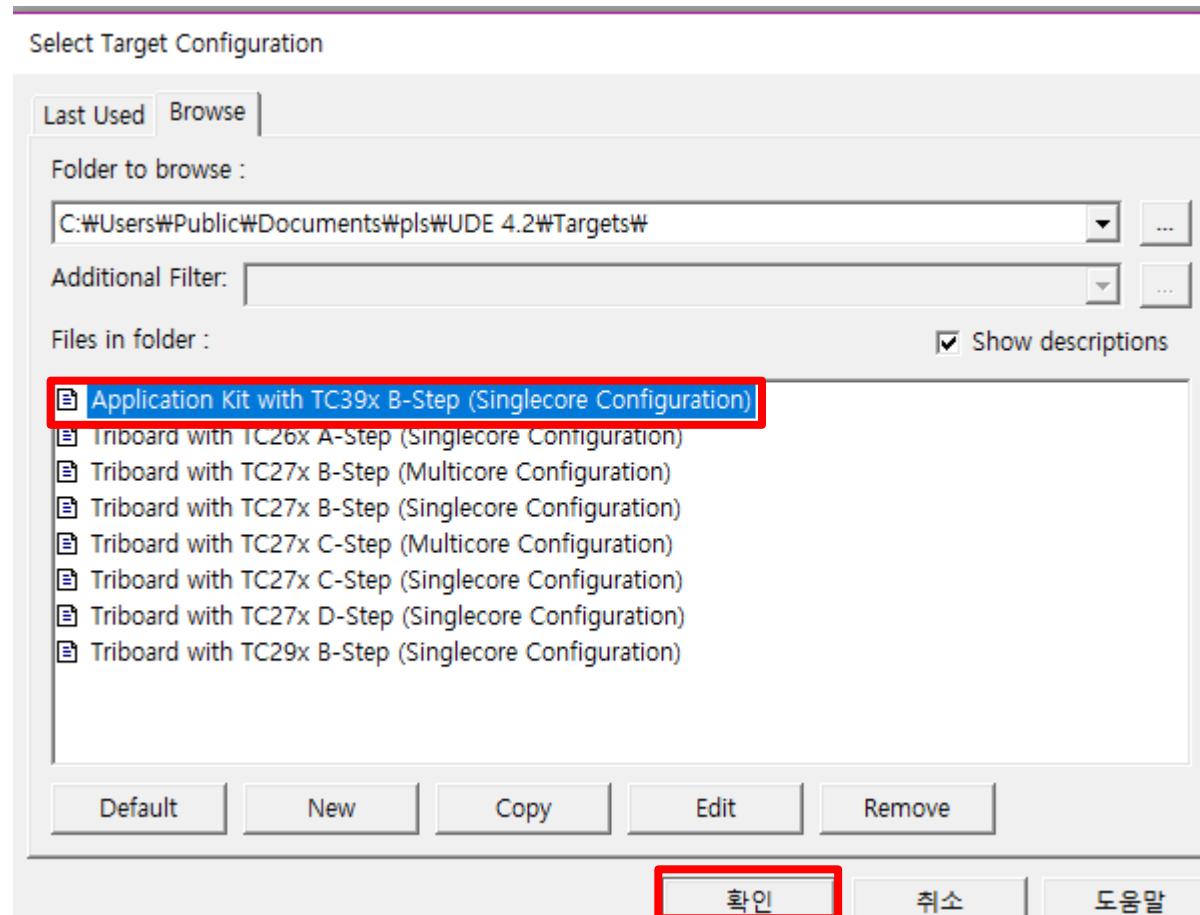
- C:\AURIX\_HOT[2G]\Day2\PWM\PLS 경로에서  
파일 이름에 PWM.wsx 입력 -> 열기



# Exercise PWM – PLS

## ➤ Target board 설정

- Application Kit with TC39x B-Step (Singlecore Configuration) 선택 -> 확인



# Exercise PWM – PLS

## ➤ Target board 연결 확인

- 다음과 같은 메시지가 뜨면, 연결 완료

The screenshot shows the UDE STK 5.0 interface. The assembly code window displays the instruction at address 0xA0000000, which is MOVH.A a15, 0x8000. The message log window at the bottom shows several successful connection messages, with entries 8, 9, and 12 highlighted with a red box.

**Target Manager**

- TargetManager
- Target0
  - Controller0
    - Core0 (selected)
    - Core1
    - Core2
    - Core3
    - Core4
    - Core5
    - GTM
    - ED
  - Memory

**code <0xA0000000-0xA00003FF>**

Address	Instruction	Operands
0xA0000000	MOVH.A	a15, 0x8000
0xA0000004	D9 FF 34 B0	LEA a15, [a15]0x2F4
0xA0000008	DC 0F	JI a15
0xA000000A	00 90	RET
0xA000000C	80 23	MOV.D d3.a2
0xA000000E	8F 43 1F 00	SH d0,d3,-0xC
0xA0000012	37 03 70 33	EXTR.U d3,d3,0x6,0x10
0xA0000016	26 20	AND d0,d2
0xA0000018	A6 30	OR d0,d3
0xA000001A	00 70	FRET
0xA000001C	61 00 04 00	FCALL 0xA0000024 ↓
0xA0000020	74 2F	ST.W [a2].d15
0xA0000022	00 70	FRET
0xA0000024	54 F0	LD.W d0,[a15]
0xA0000026	37 00 70 08	EXTR.U d0,d0,0x10,0x10
0xA000002A	8F 00 01 00	SH d0,d0,0x10
0xA000002E	A6 0F	OR d15,d0
0xA0000030	00 70	FRET
0xA0000032	00 00	NOP
0xA0000034	4E 41	JGTZ d4, 0xA0000036 ↓
0xA0000036	4E 00	JGTZ d0, 0xA0000036 ←
0xA0000038	00 A0	DEBUG
0xA000003A	3C FF	J 0xA0000038 ↑
...	...	...

**Messages**

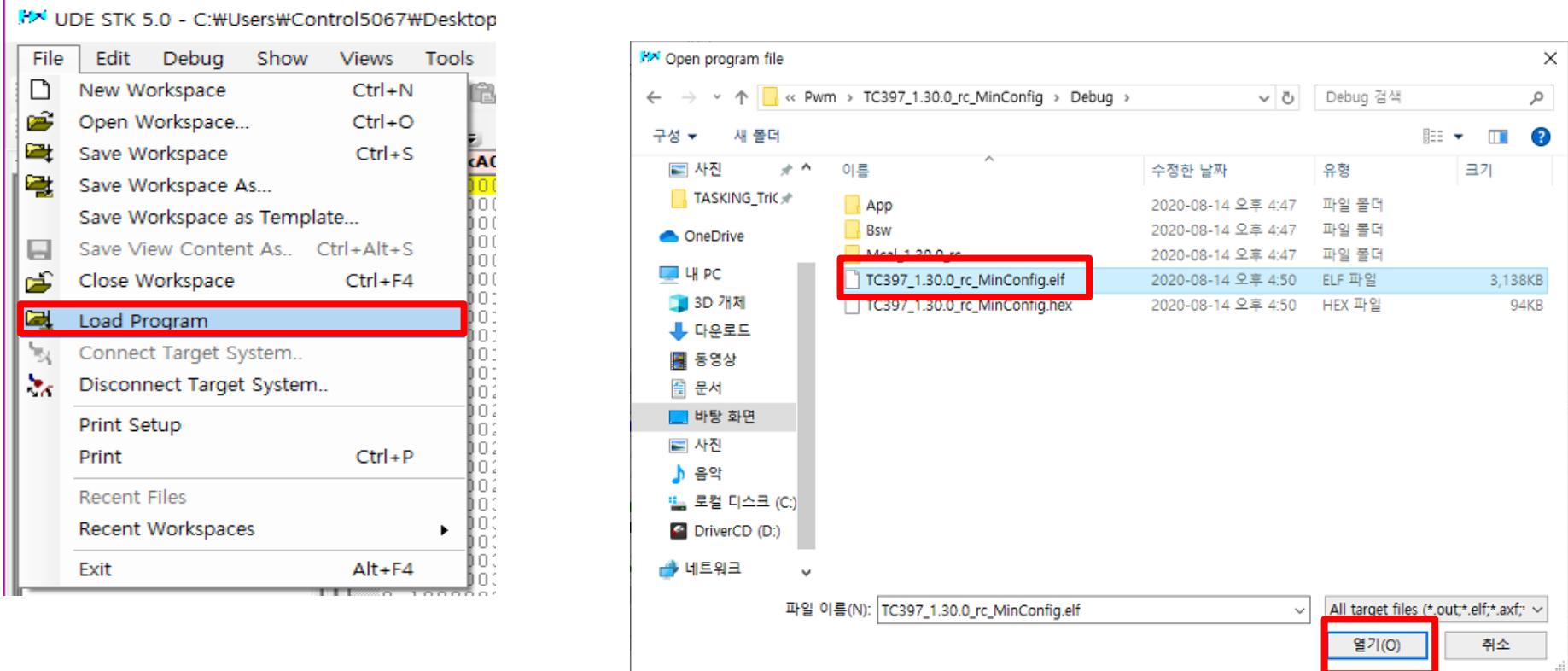
I...	Type	Time	Source	Message
5	Info	16:58:43...	Workspace	Target configuration file C:\Users\Public\Documents\pl
6	Info	16:58:51...	Core0::PFLASH	No ABM configuration found --> 'Safe ABM header handle'
7	Info	16:58:51...	Core0::PFLASH	No HSM configuration found --> 'Safe HSM vector handle'
8	Success	16:58:51...	Core0::PFLASH	FLASH programming for device '16 MByte OnChip Program'
9	Success	16:58:51...	Core0::DF_EEPROM	FLASH programming for device '1 MByte OnChip Data FLASH'
10	Info	16:58:51...	Core0::DF_EEPROM	DF_EEPROM: Normal sensing mode detected
11	Info	16:58:51...	Core0::DF1	DF1: Normal sensing mode detected
12	Success	16:58:51...	Core0::UDED... (Core0)	Connection to TC39xB target established: TriCore (Core0)

# Exercise PWM – PLS

## ➤ Load Program

### ▪ File -> Load Program

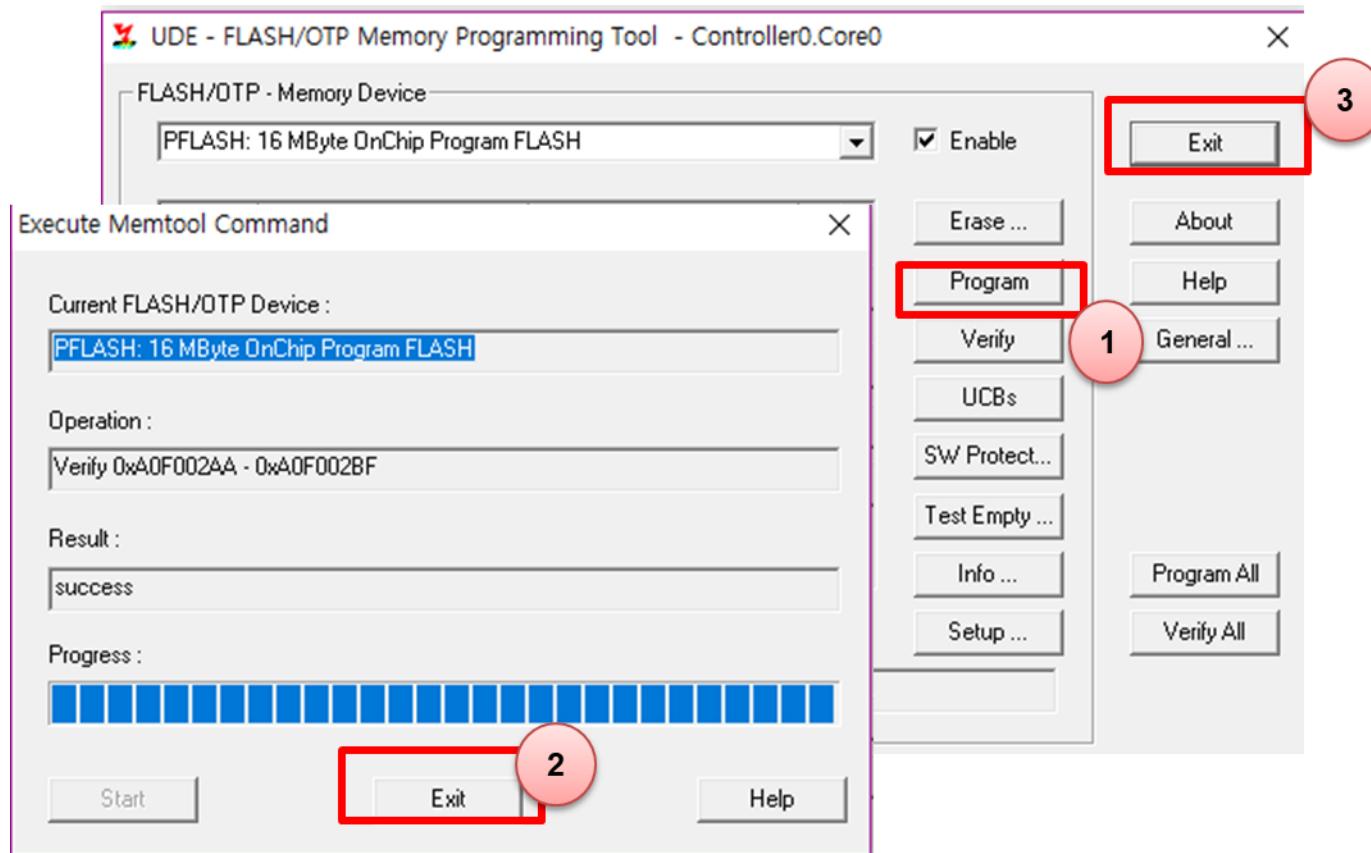
- day2 PWM Tasking TC397\_1.30.0\_rc\_MinConfig Debug 경로의 TC397\_1.30.0\_rc\_MinConfig.elf 파일 선택 -> 열기



# Exercise PWM – PLS

## ➤ Load Program

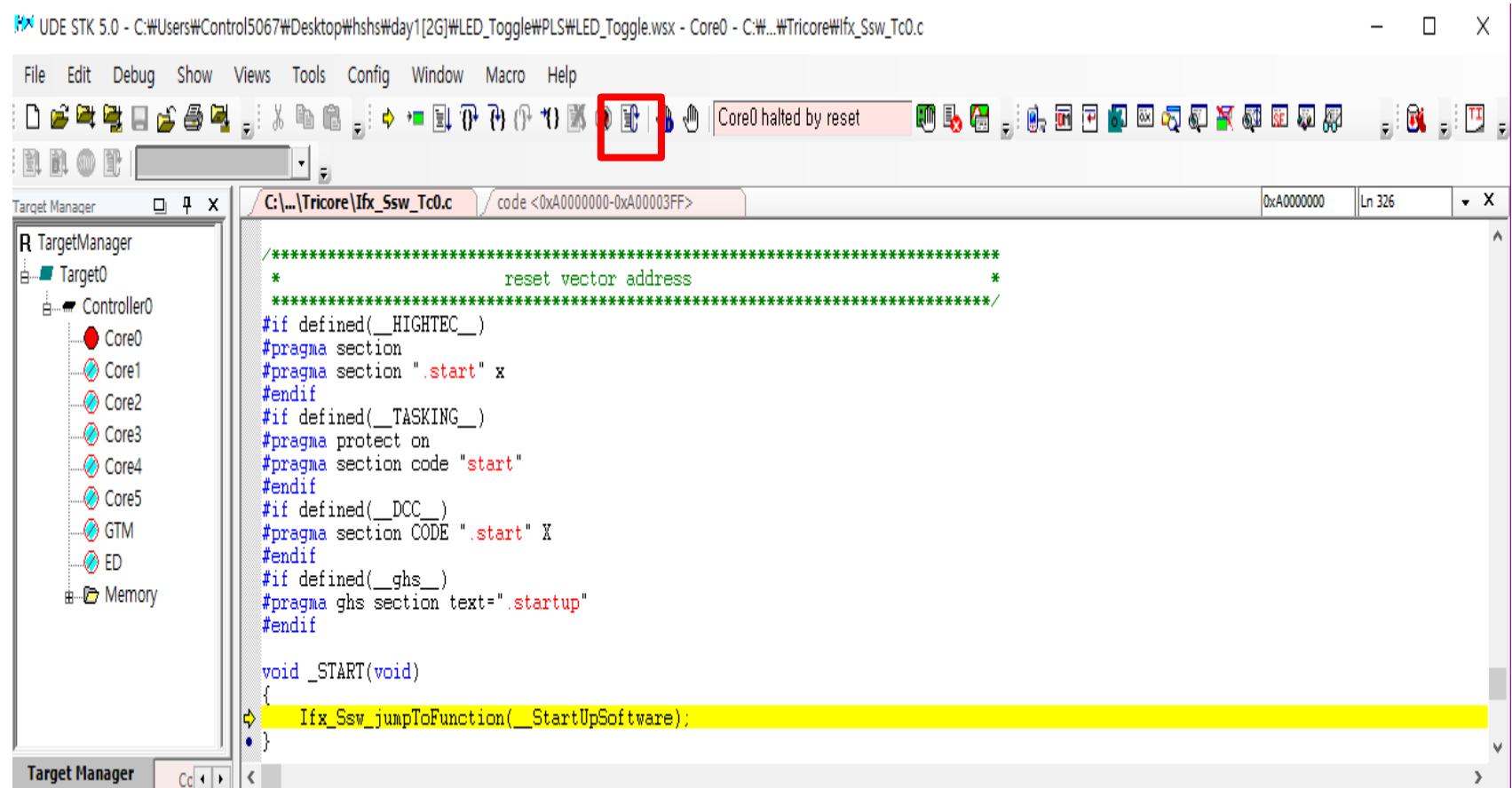
- Program 클릭
- Execute Memtool Command 창에서 진행이 완료 되면, Exit 클릭
- FLASH/OTP Memory Programming Tool 창에서 Exit 클릭



# Exercise PWM – PLS

## ➤ Start Program

- 또는 F5(Start Program) 클릭



# Exercise PWM – PLS

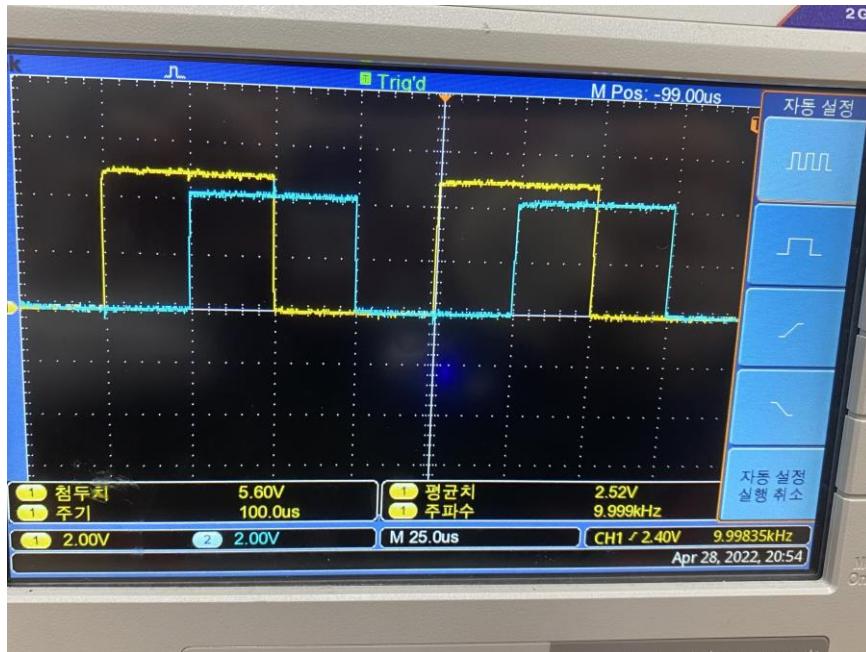
## ➤ Pwm 신호 확인

- P33.10, P33.9을 통해 Reference pwm과 Center aligned 적용된 pwm 신호 확인

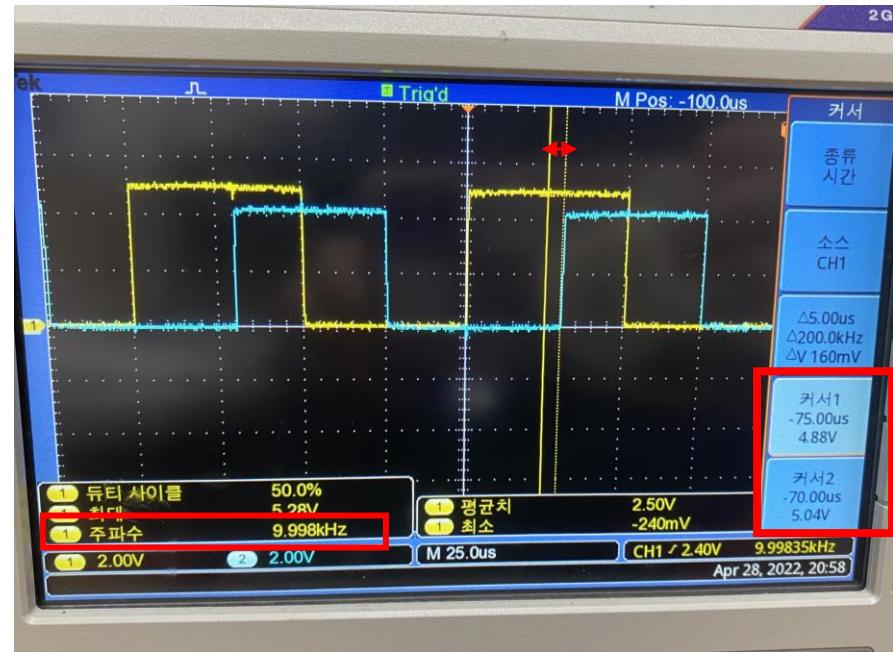
- 10kHz주기의 PWM 파형 생성
- Rising Edge에서의 5us의 DTM 적용

$$\checkmark \frac{1}{200\text{Mhz}(GTM\ CLOCK)} \times 1000(\text{Ticks}) = 5\mu\text{s}$$

Reference pwm  
Center aligned pwm



DTM 적용 전 PWM 파형



DTM 적용 후 PWM 파형

# Exercise PWM – PLS

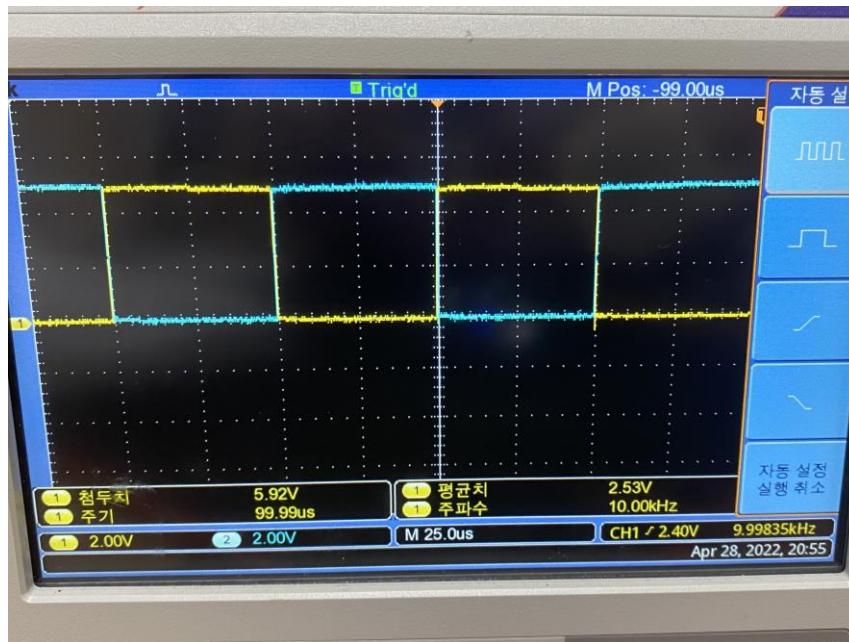
## ➤ Pwm 신호 확인

- P33.10, P33.11을 통해 Reference pwm과 Shifted pwm 신호 확인

- 10kHz의 PWM 파형 생성
- Rising Edge에서의 5us의 DTM 적용

$$\checkmark \frac{1}{200\text{Mhz}(GTM\ CLOCK)} \times 1000(\text{Ticks}) = 0.5\mu\text{s}$$

Reference pwm  
Shifted pwm



DTM 적용 전 PWM 파형



DTM 적용 후 PWM 파형