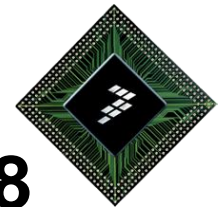


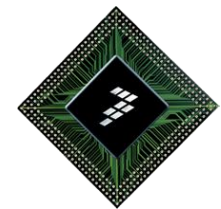
June 2012

# CW for Microcontrollers v10.2 and MQX 3.8



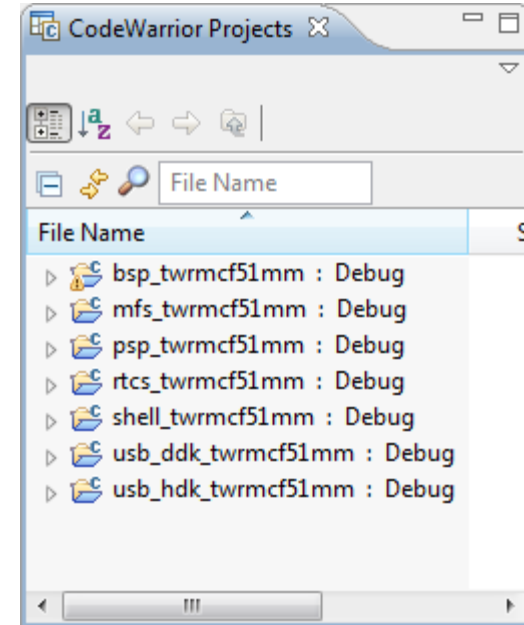
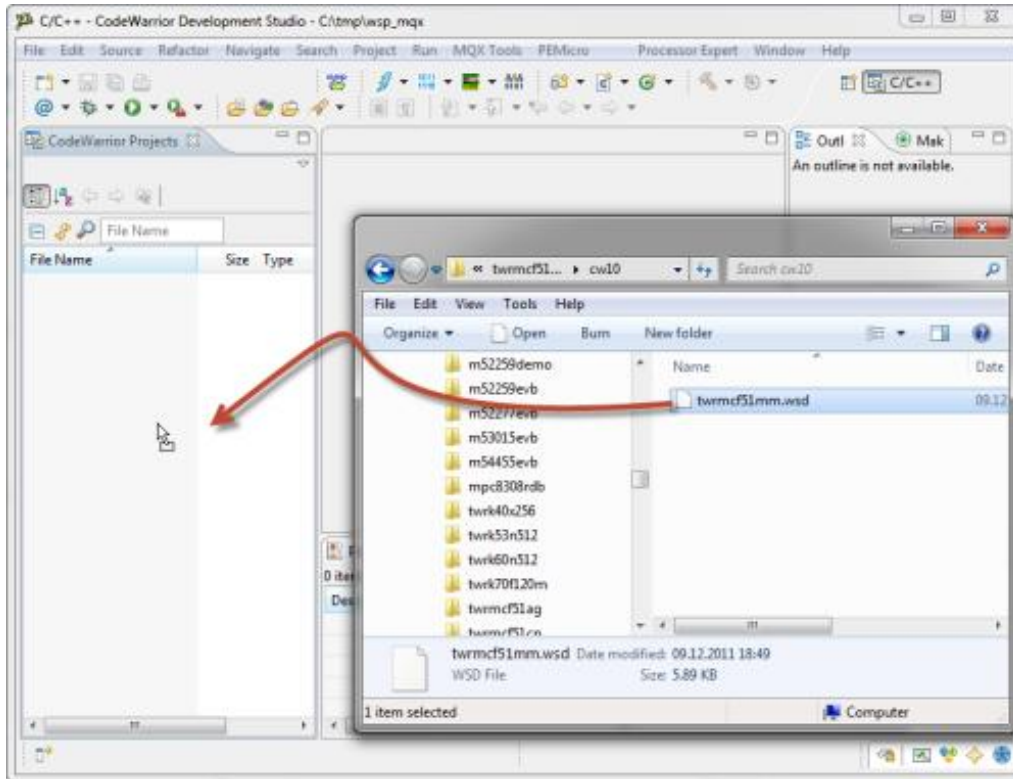
- ▶ Import MQX Libraries
- ▶ Build MQX libraries
- ▶ Import and Debug MXQ Hello World Project
- ▶ New MQX project
- ▶ Debugging with Jlink
- ▶ Eclipse Working Set
- ▶ CW10.2, MQX 3.8 and Processor Expert
- ▶ CW10.2, MQX 3.8 and PE : New LDD driver

# Import MQX Libraries



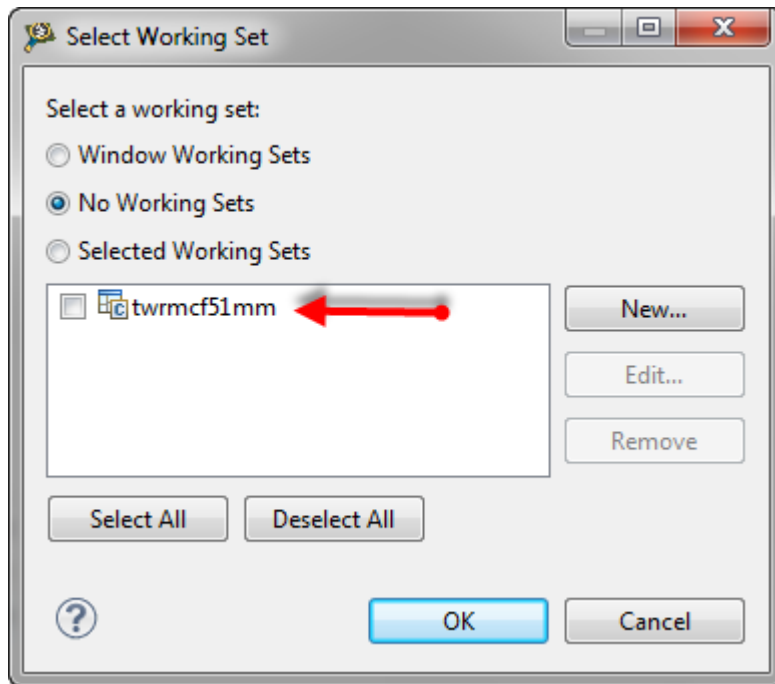
# Importing MQX Library projects

- Navigate to `C:\Freescale\Freescale MQX 3.8\config\<board_name>` and drag `<board>.wsd` to the CodeWarrior
- All BSP libraries will be loaded to your environment automatically

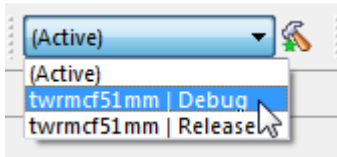


# Importing MQX Library projects

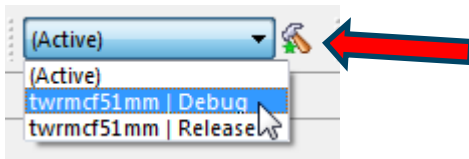
- Beside of the projects, it has imported as well the Working Set:



- Use MQX toolbar to select desired configuration to build



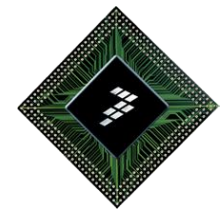
- and push hammer icon to build all MQX libraries for selected working set



**Note:**

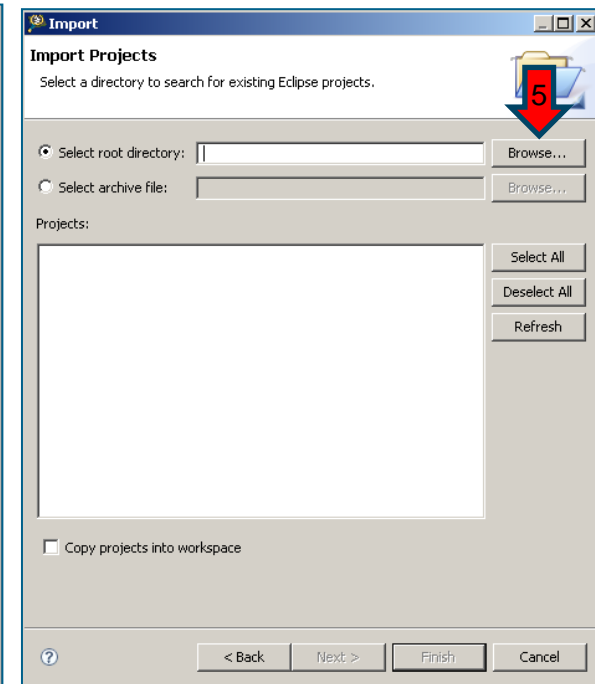
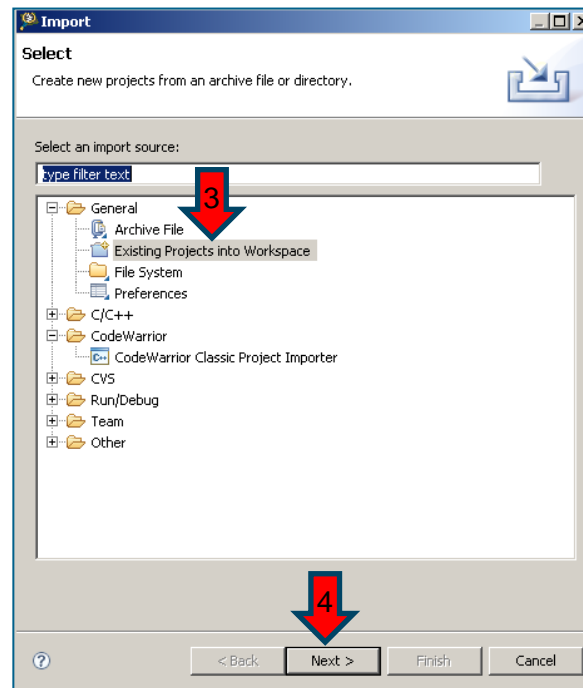
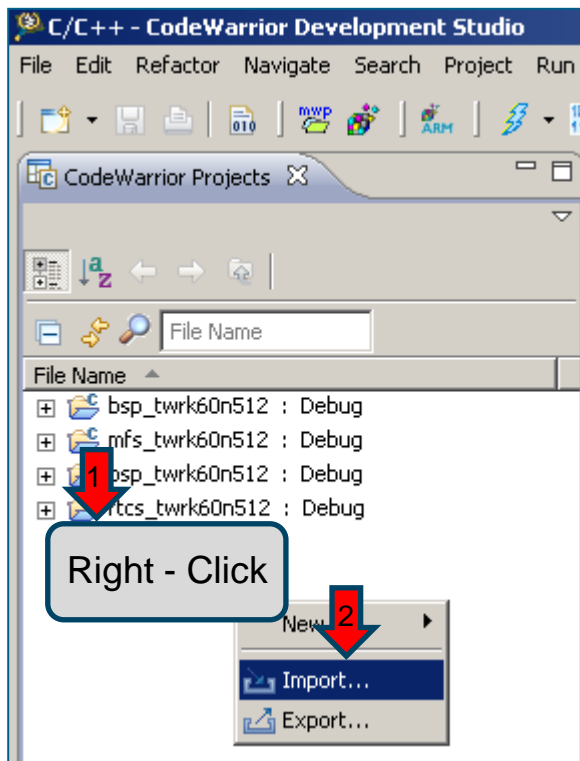
**Debug** configuration of MQX libraries (workingset) have compiler optimization set to lowest level for all imported projects, while **Release** configuration uses the highest possible compiler optimization setting

# Import and Debug MQX Hello World Project



# Import 'Hello World' MQX example

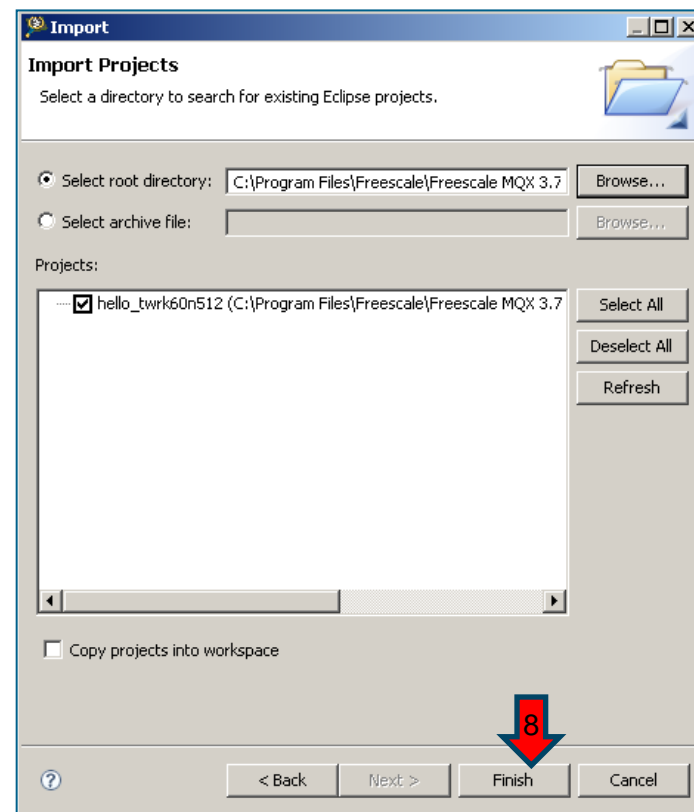
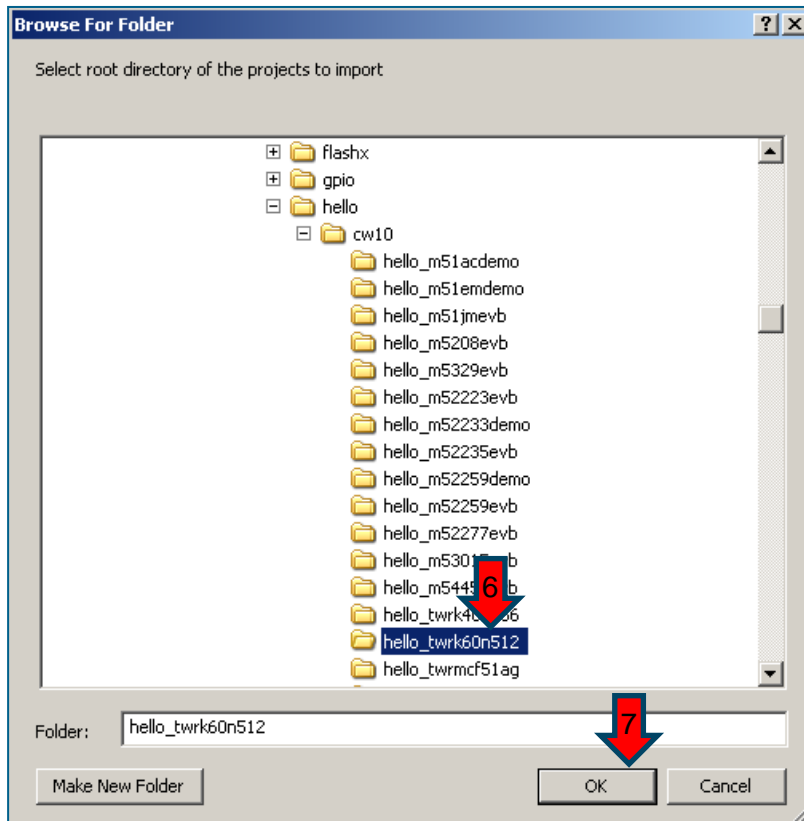
- ▶ Right-Click on Project Explorer and Import
- ▶ Select Existing Projects into Workspace and Browse





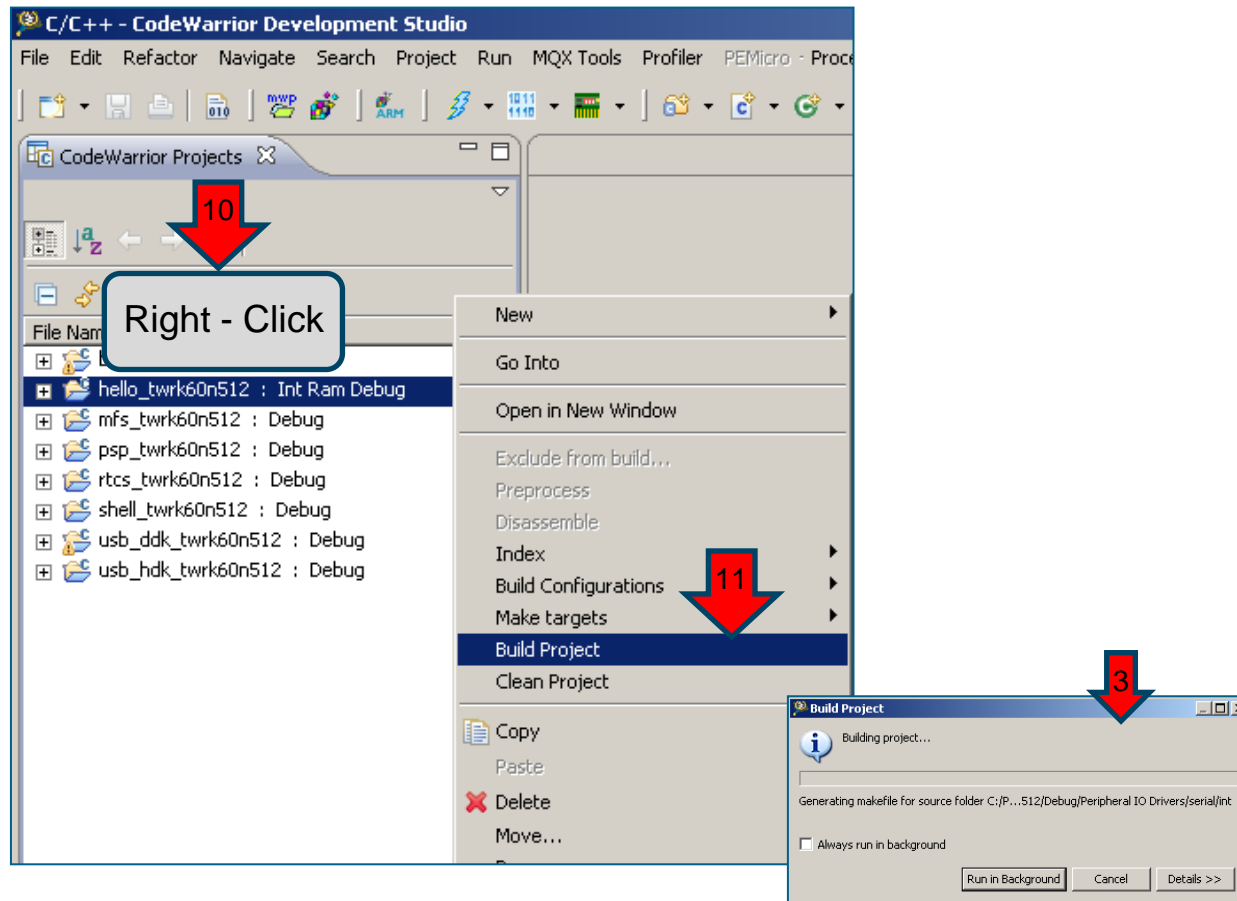
# Import 'Hello World' MQX example

- Select *<install mqx folder>\mqx\examples\hello\CW10\hello\_twrk60n512*



# Build 'Hello World' MQX example

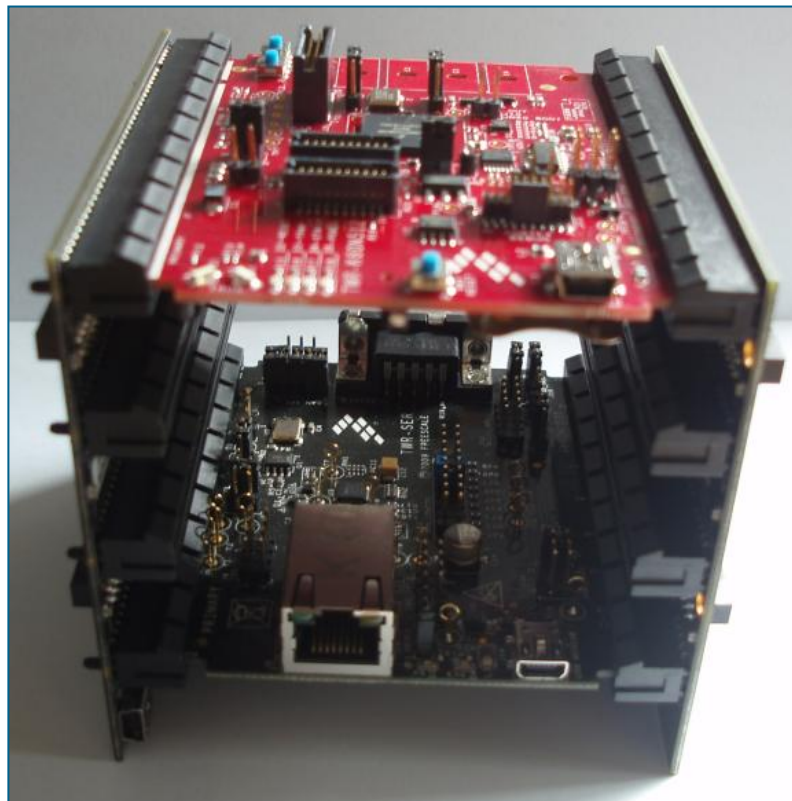
- Right-Click on Project Explorer **hello\_twrk60n512** and Build Project



# Prepare your hardware (example for K60 TWR Kit)

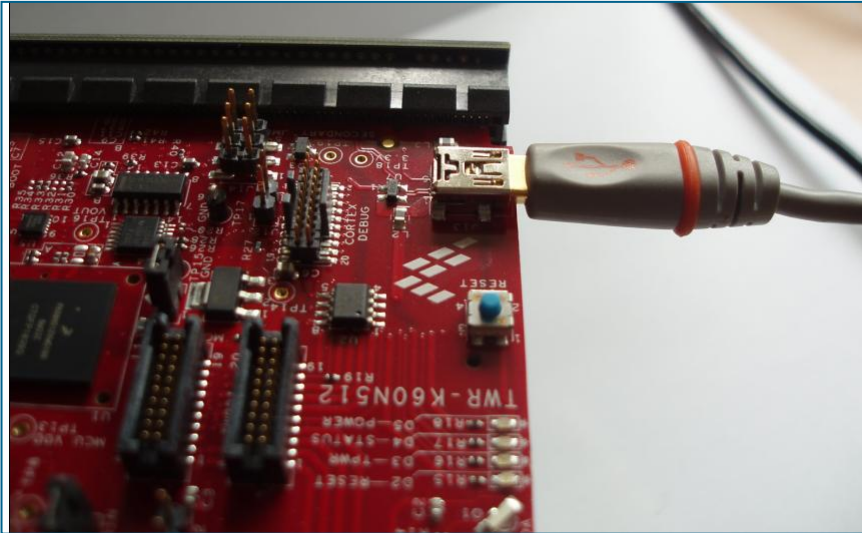
## ► Prepare your Tower System:

- Connect **TWR-SER** and **TWR-K60N512** to **TWR-ELEV** (Primary and Secondary)



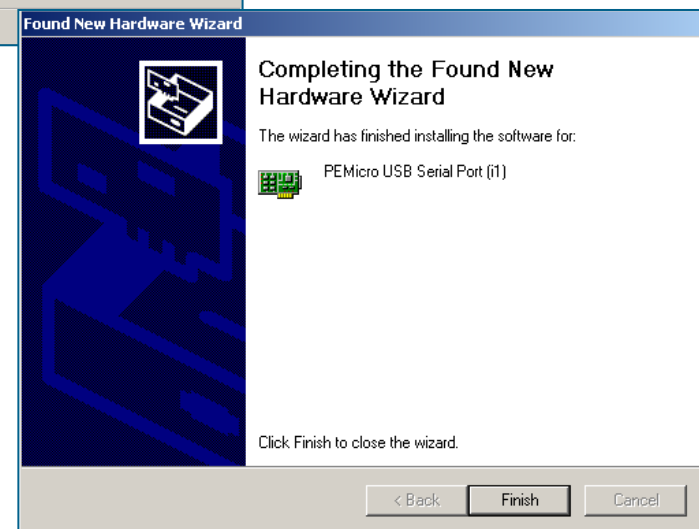
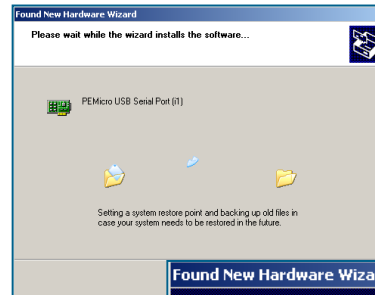
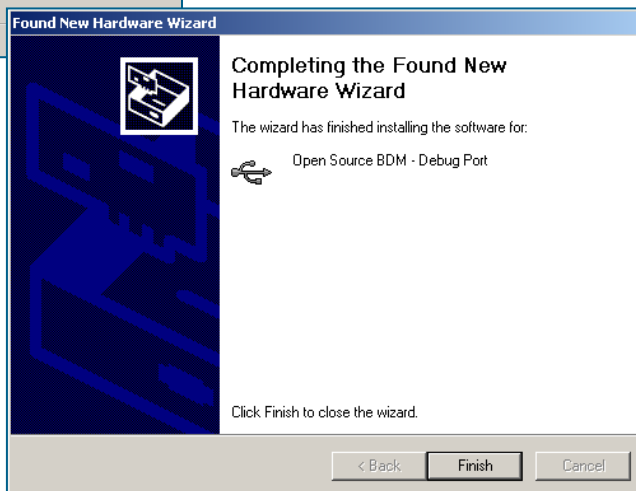
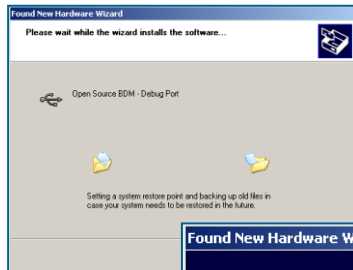
# Prepare your hardware

- Connect USB Cable to **TWR-K60N512** (J13) and laptop



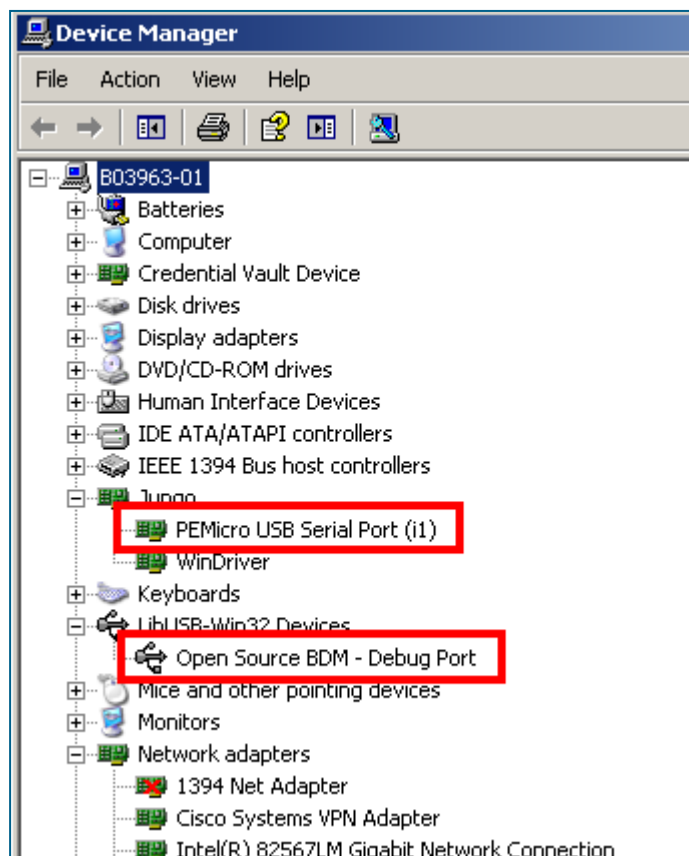
# Install OSJTAG Drivers

- ▶ Windows will detect the new USB device. Follow the installation wizards for BDM and virtual Serial port drivers.



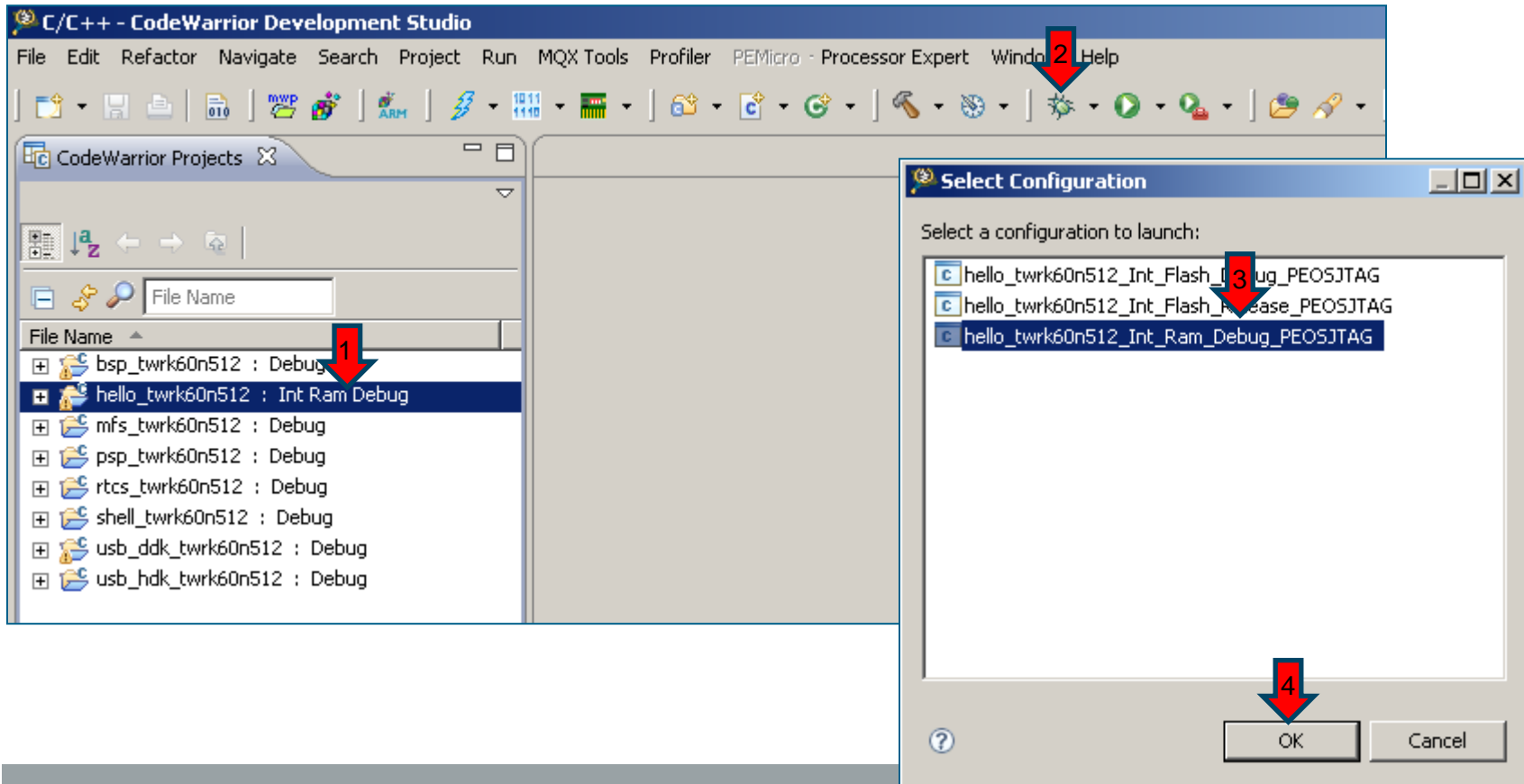
# Install OSJTAG Drivers

- You can see in Device Manager the two new USB devices installed



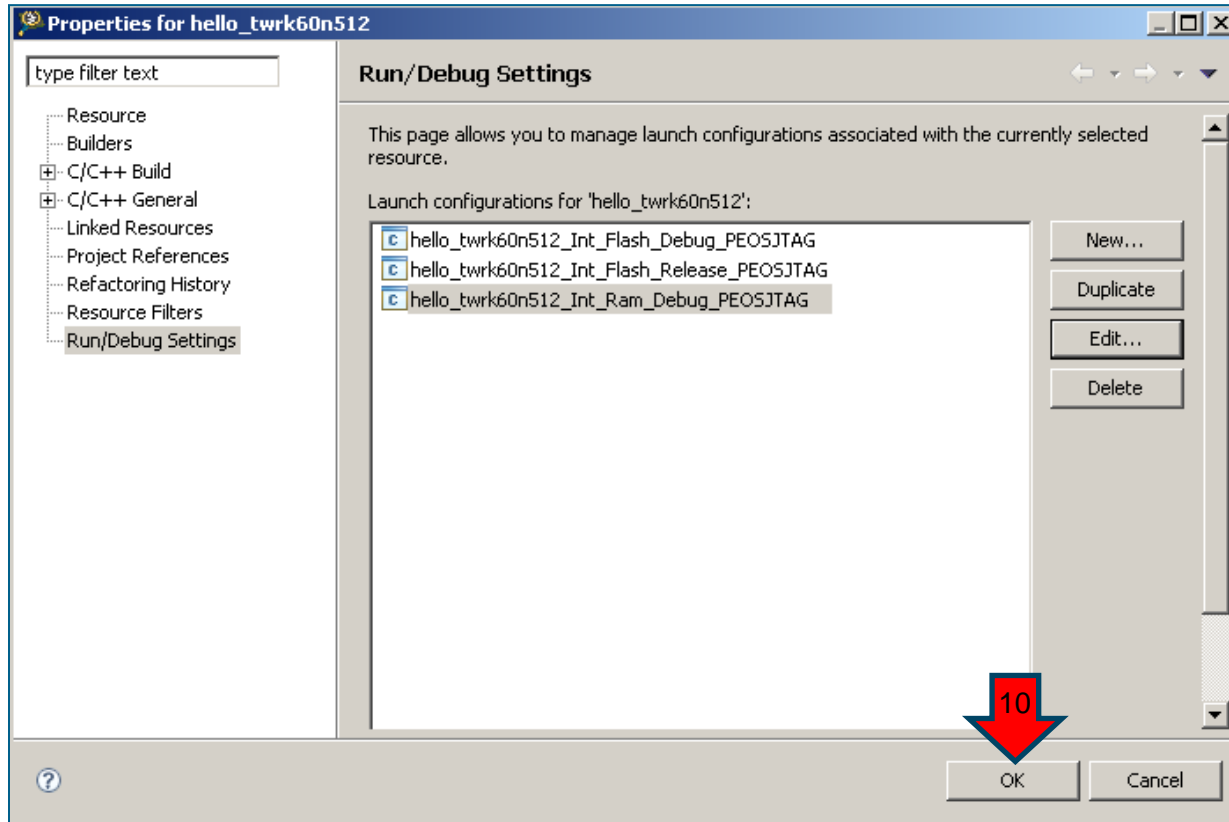
# Debug MQX 'Hello World' example

- ▶ Select **hello\_twrk60n512** project and Click 'Debug icon'
- ▶ Select **hello\_twrk60n512\_Int\_Ram\_Debug\_PEOSJTAG** Connection



# Change Connection Settings

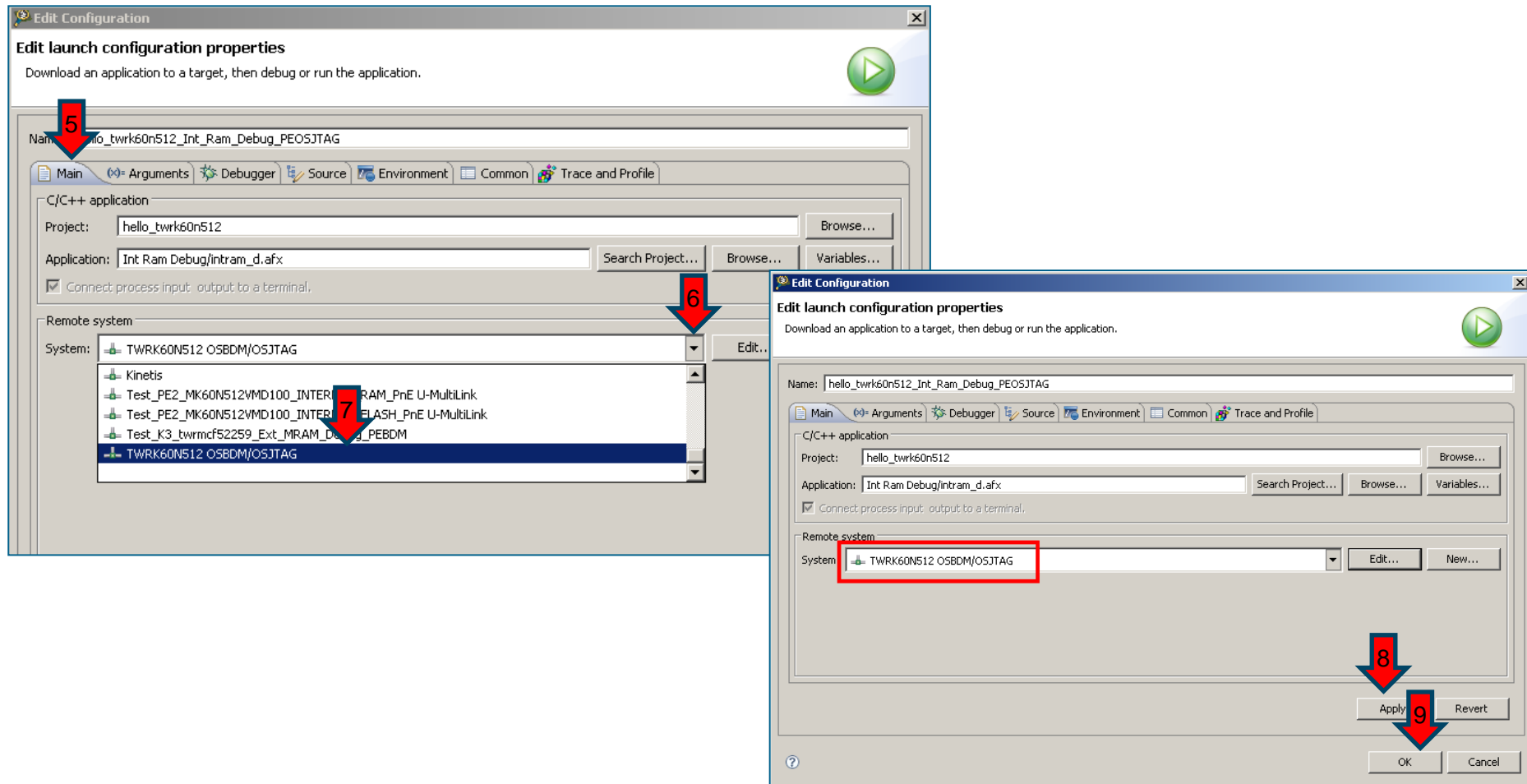
- Click OK





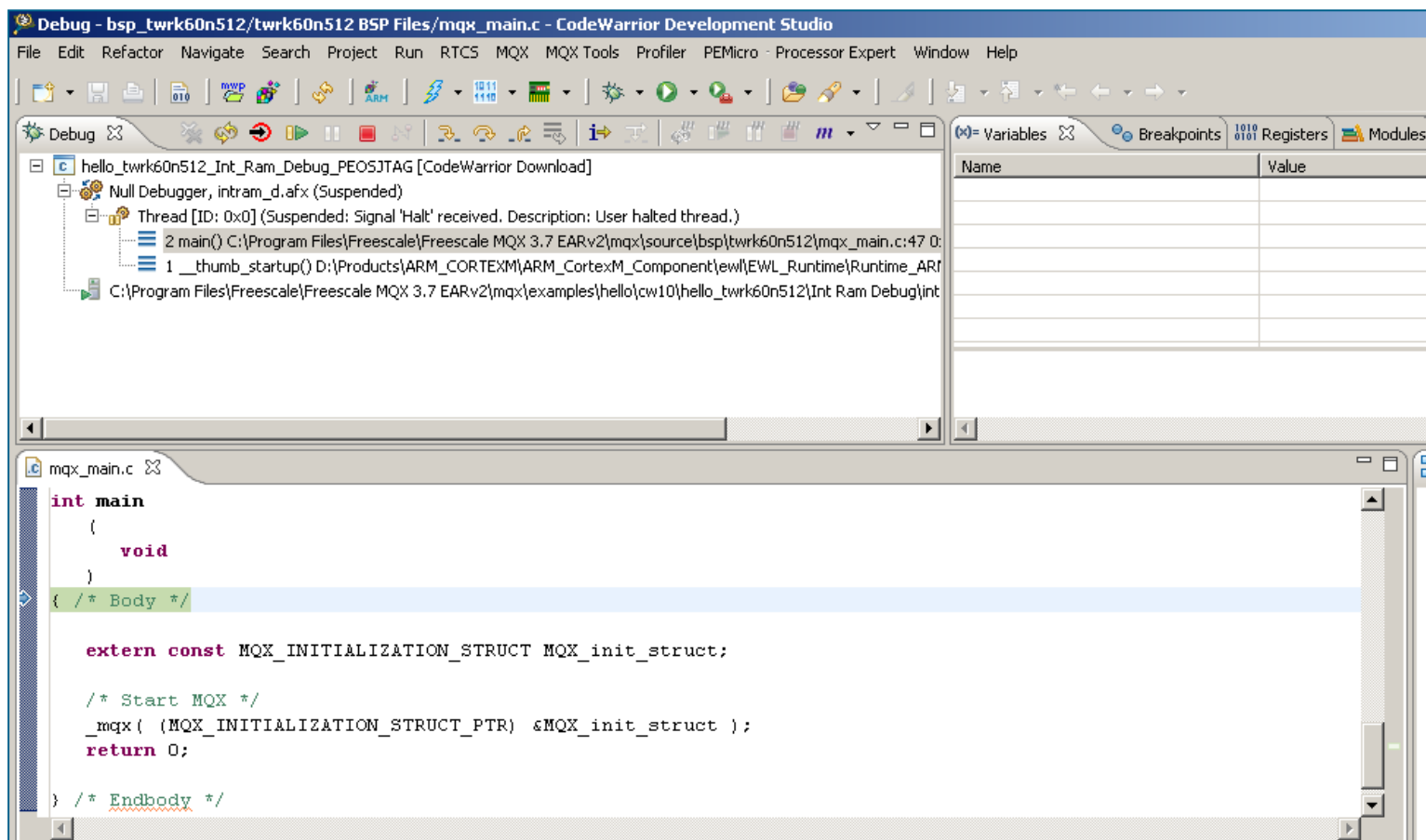
# Change Connection Settings

## ► Select **TWRK60N512 OSBDM/OSJTAG** Remote System



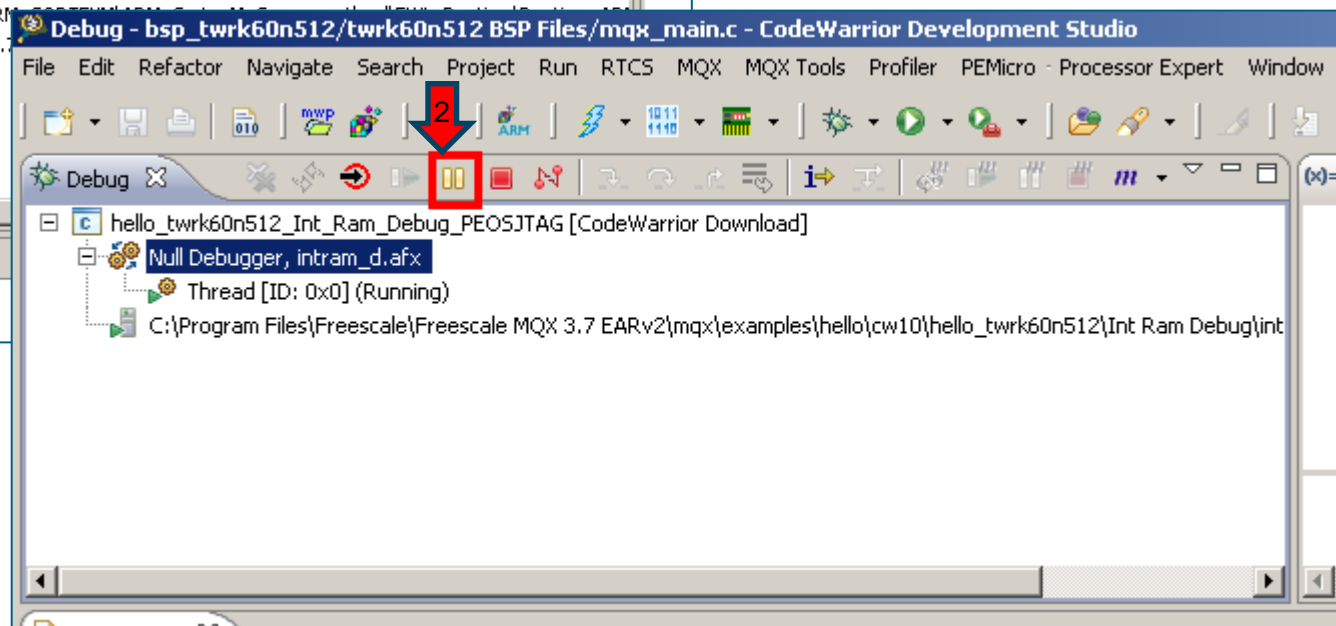
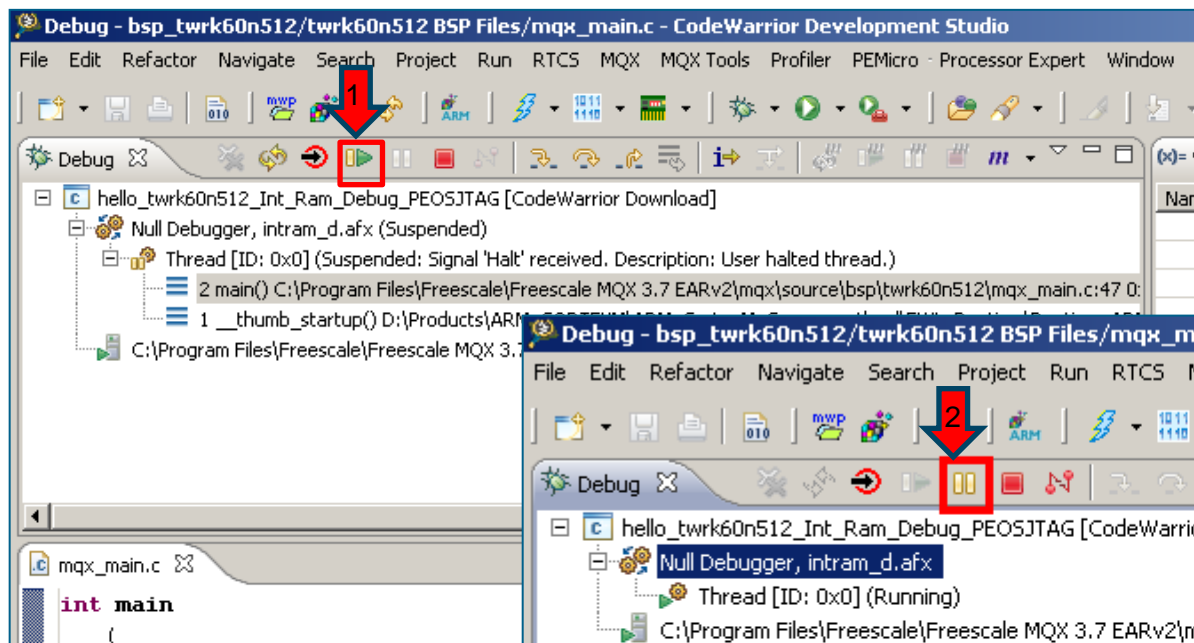
# Update OSJTAG firmware

- You are ready to Run and Debug the project



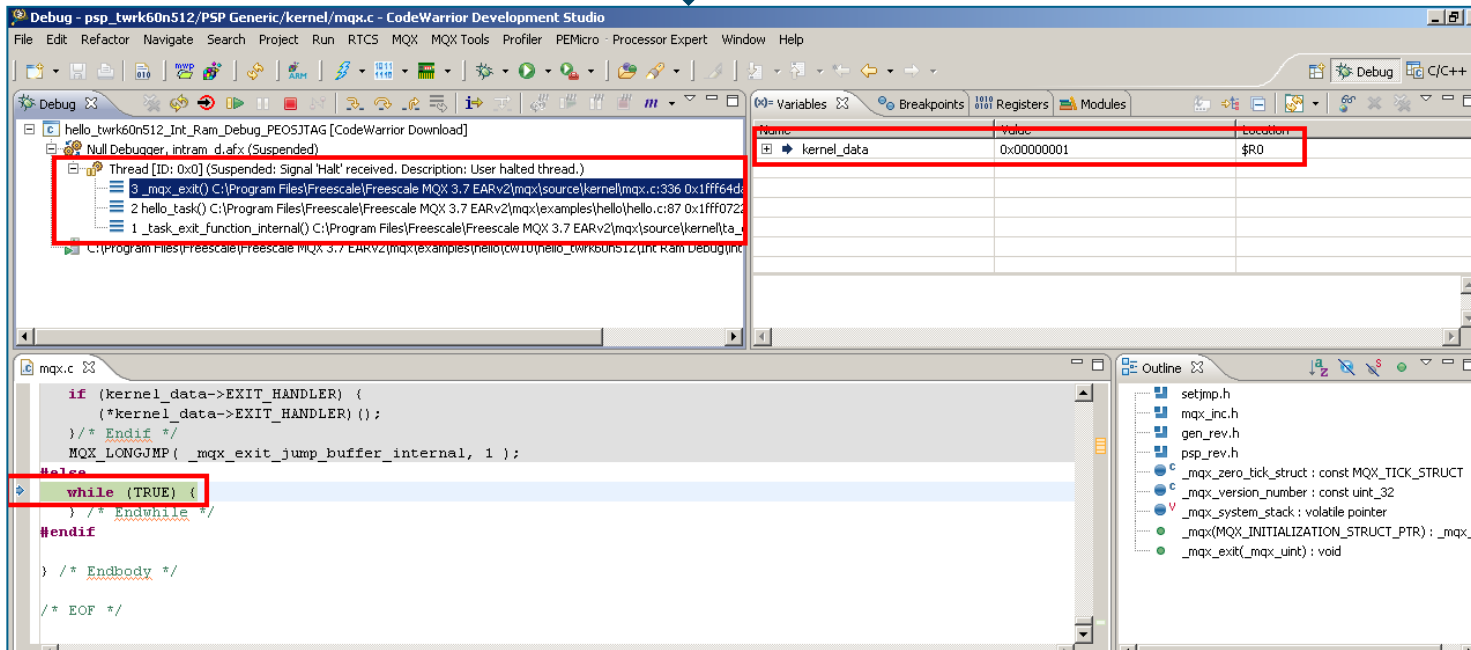
# Run MQX 'Hello World' example

- ▶ Execute the code 'Resume' icon and 'Pause' execution



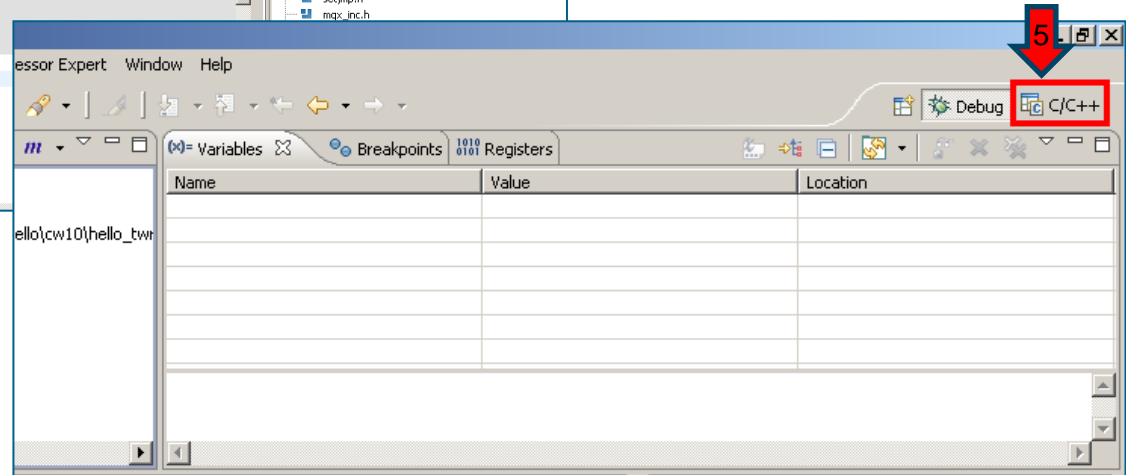
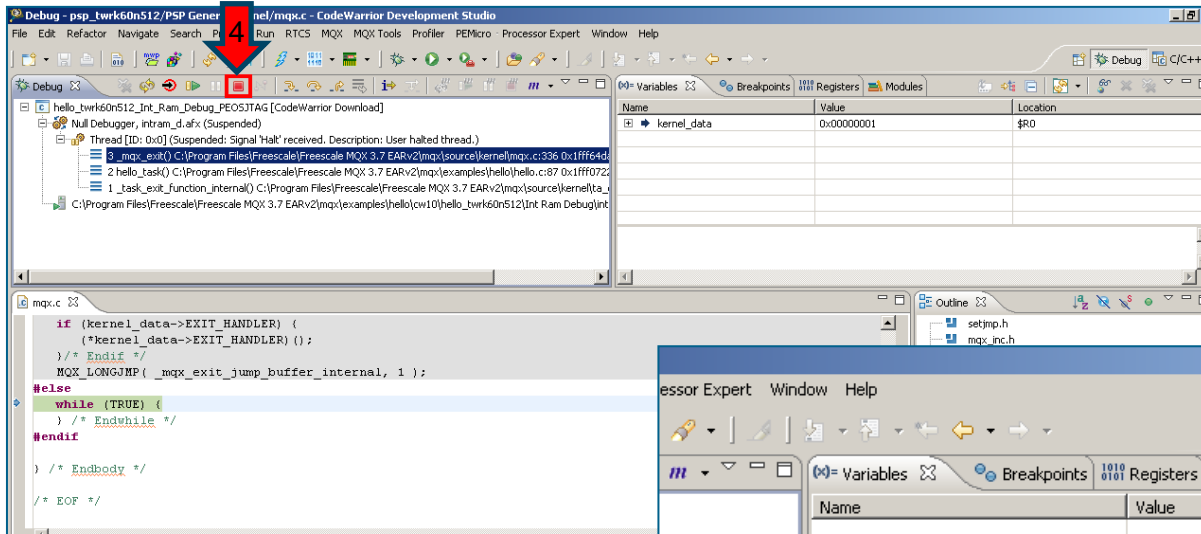
# Run MQX 'Hello World' example

- You can explore the Debugging Eclipse perspective



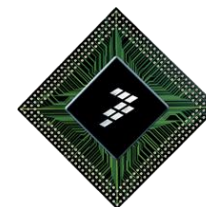
# Run MQX 'Hello World' example

- ▶ Terminate the Debugging session and change Eclipse perspective
- ▶ You have Run and Debug your first MQX CW10 project



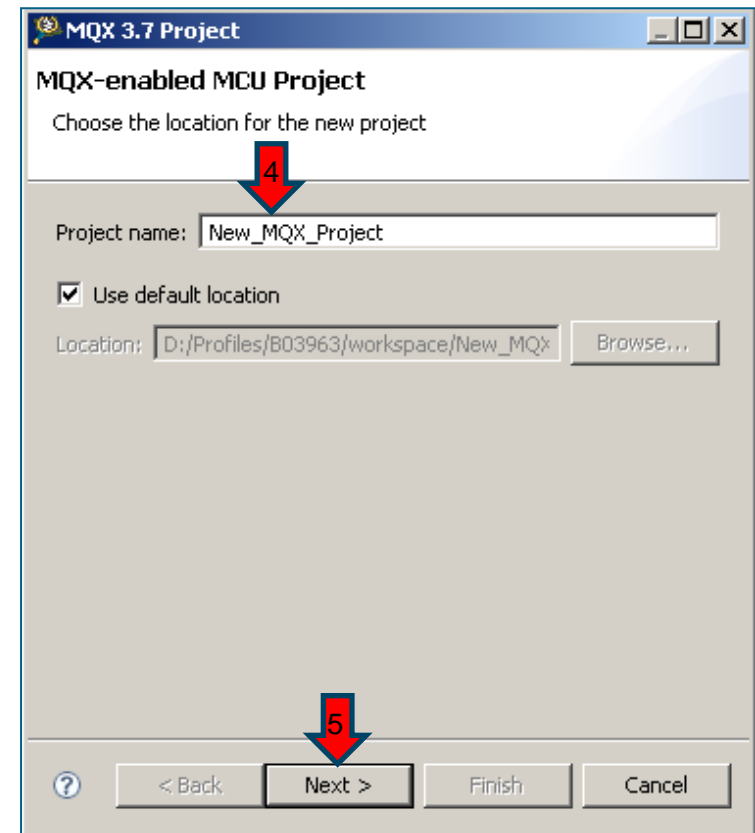
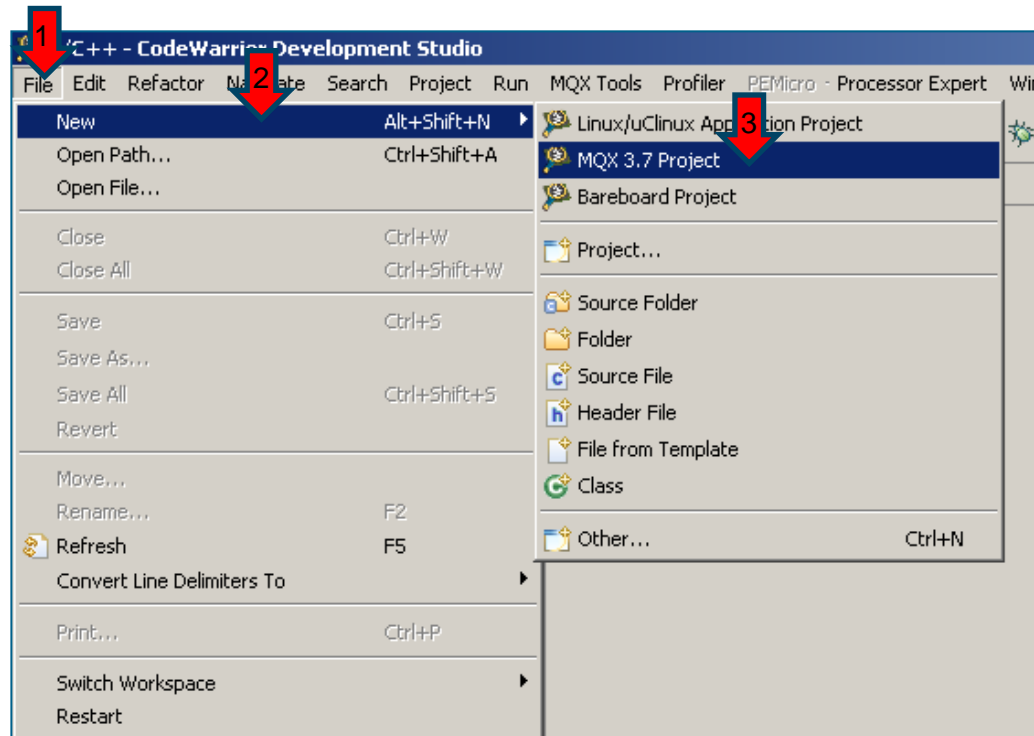
# New MQX project

---

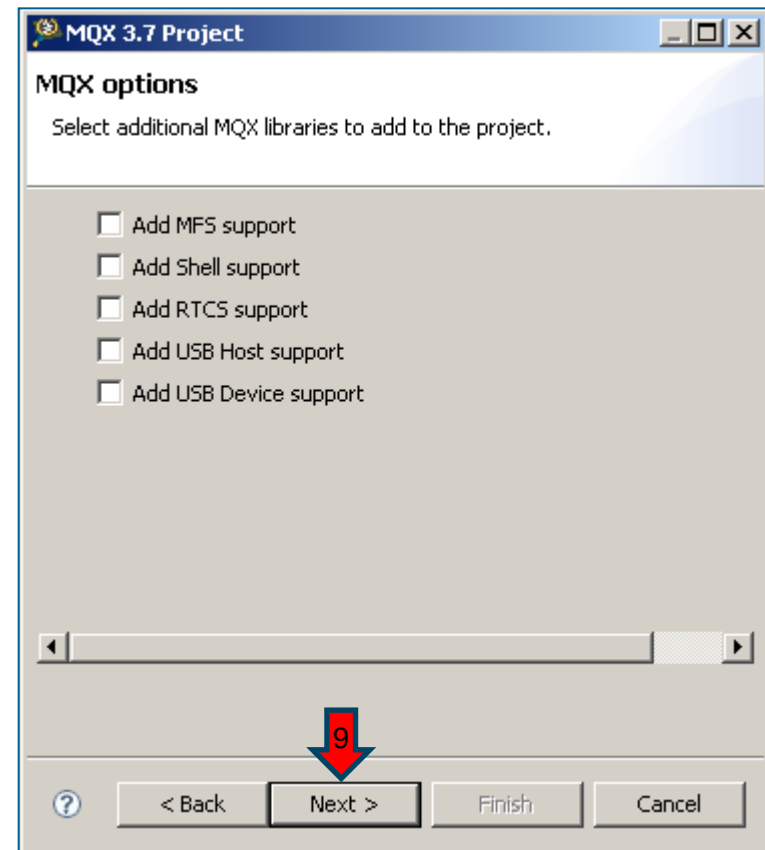
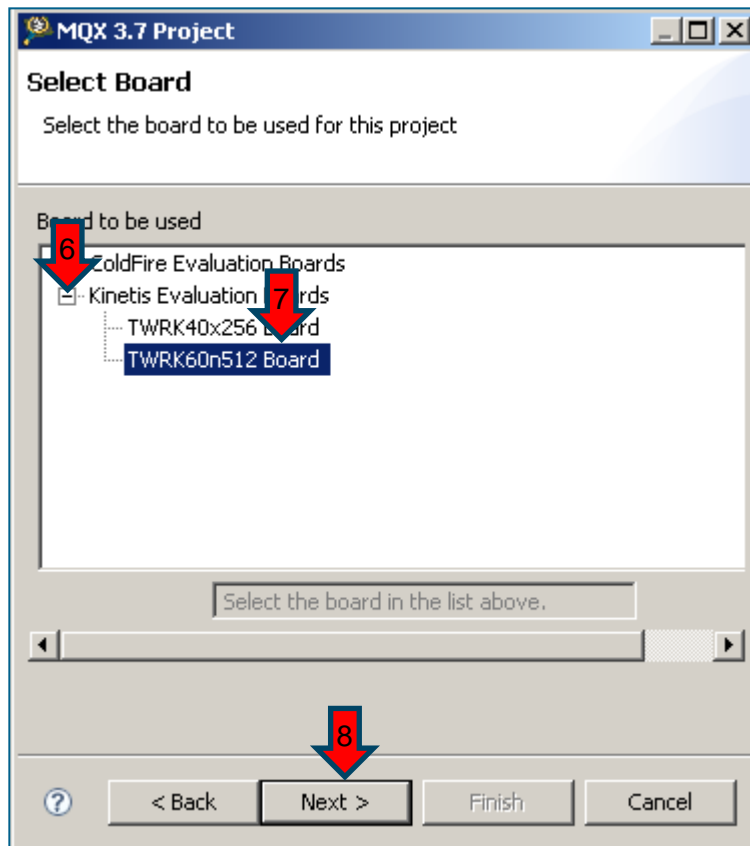


# New MQX Project

- ▶ File -> New -> MQX 3.8 Project
- ▶ Give a name and click Next

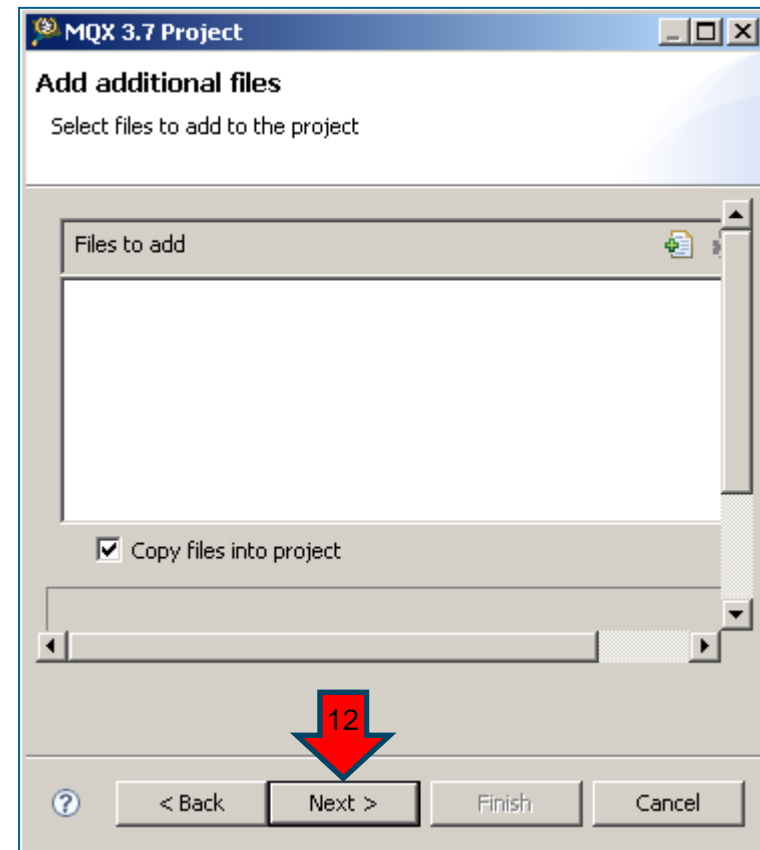
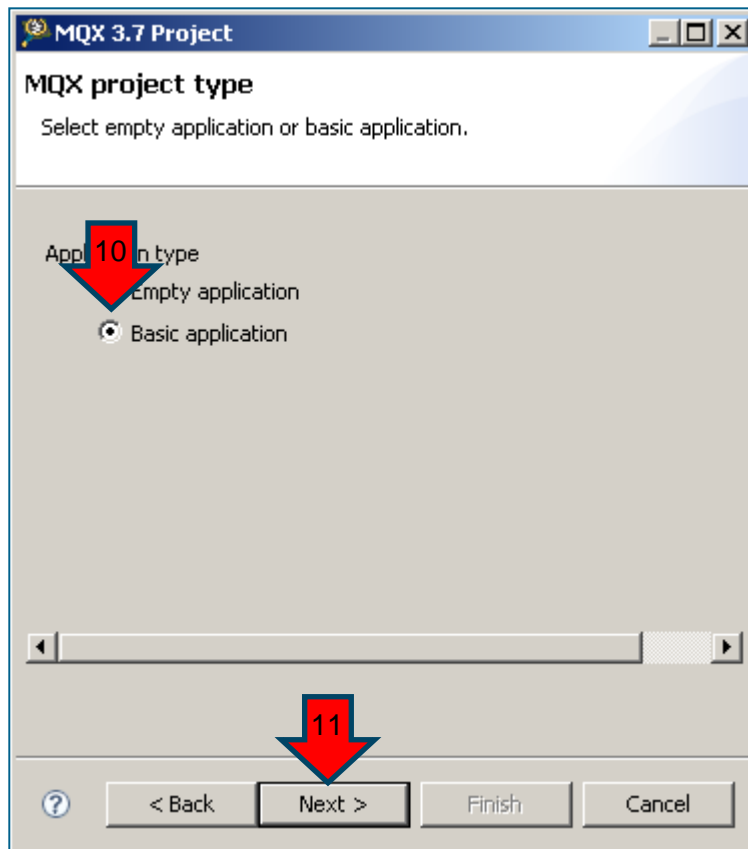


## ► Select **TWRK60n512** Board



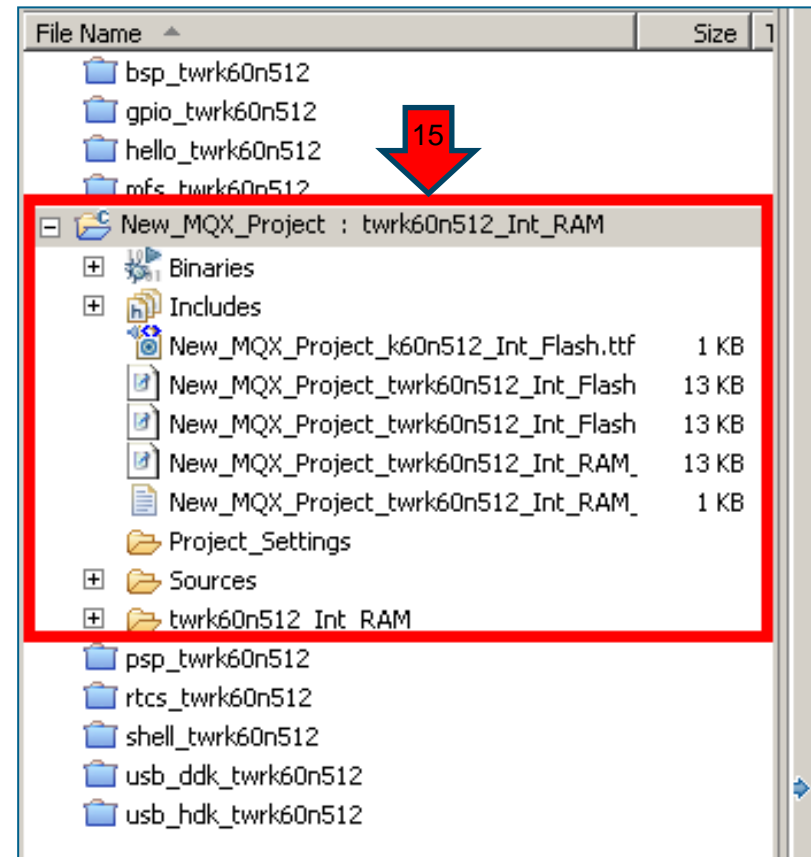
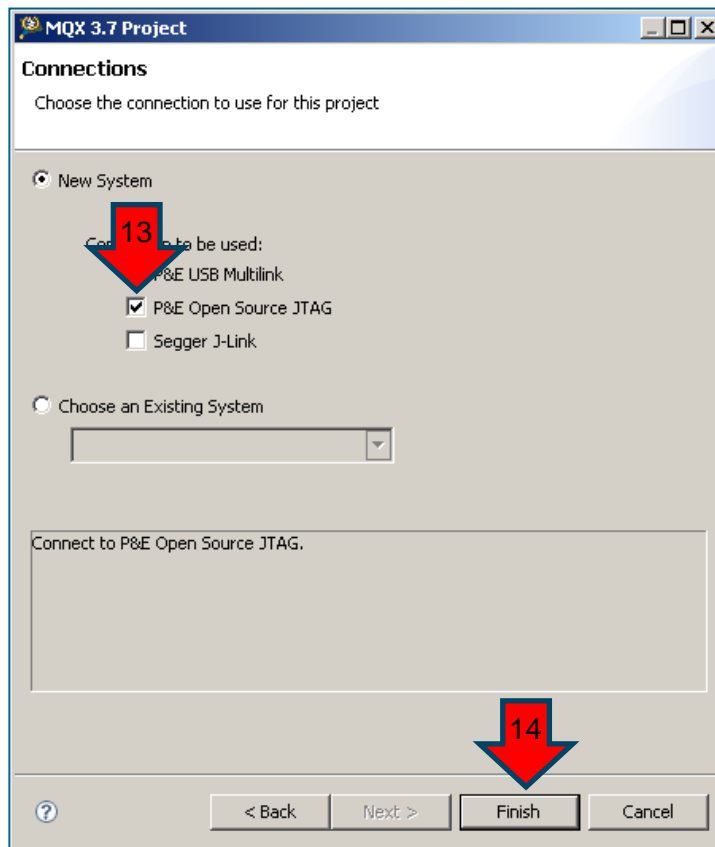


## ► Select Basic application



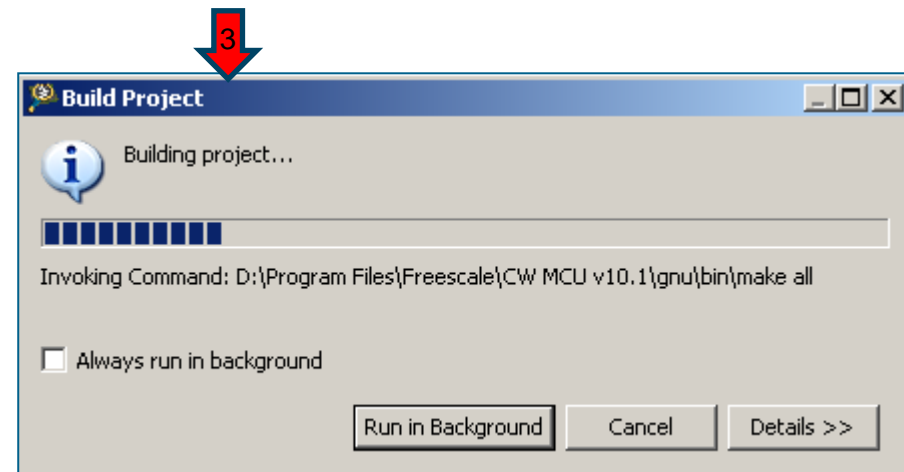
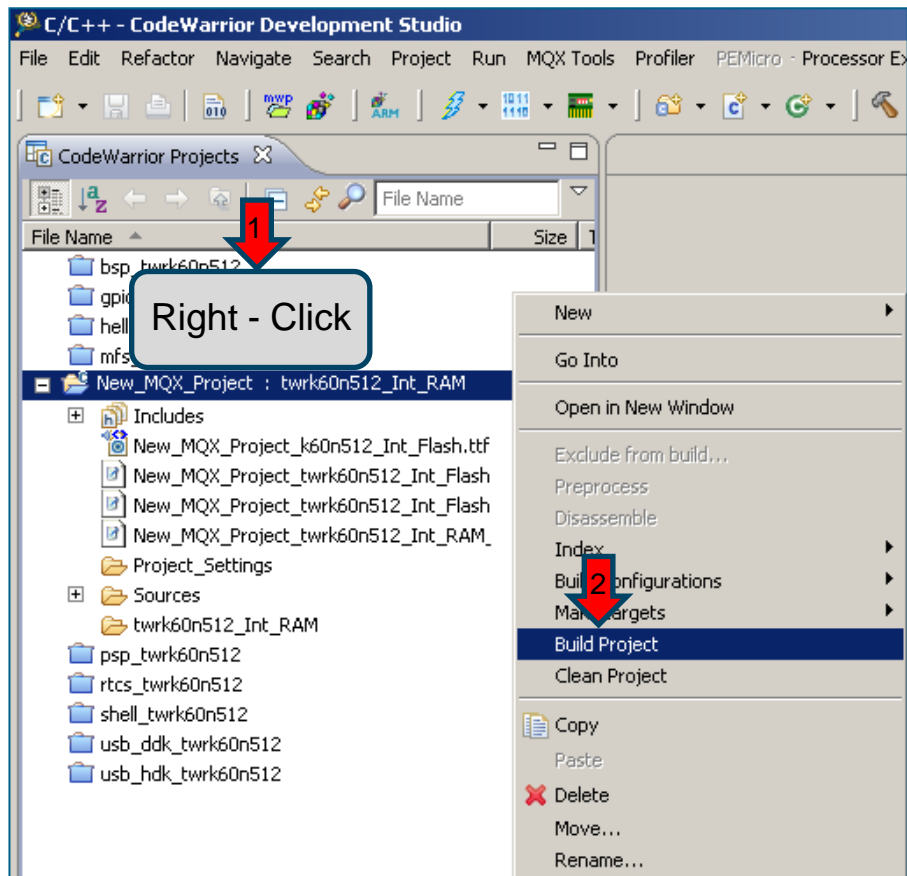
# New MQX Project

- ▶ Select P&E Open Source JTAG
- ▶ Project is created



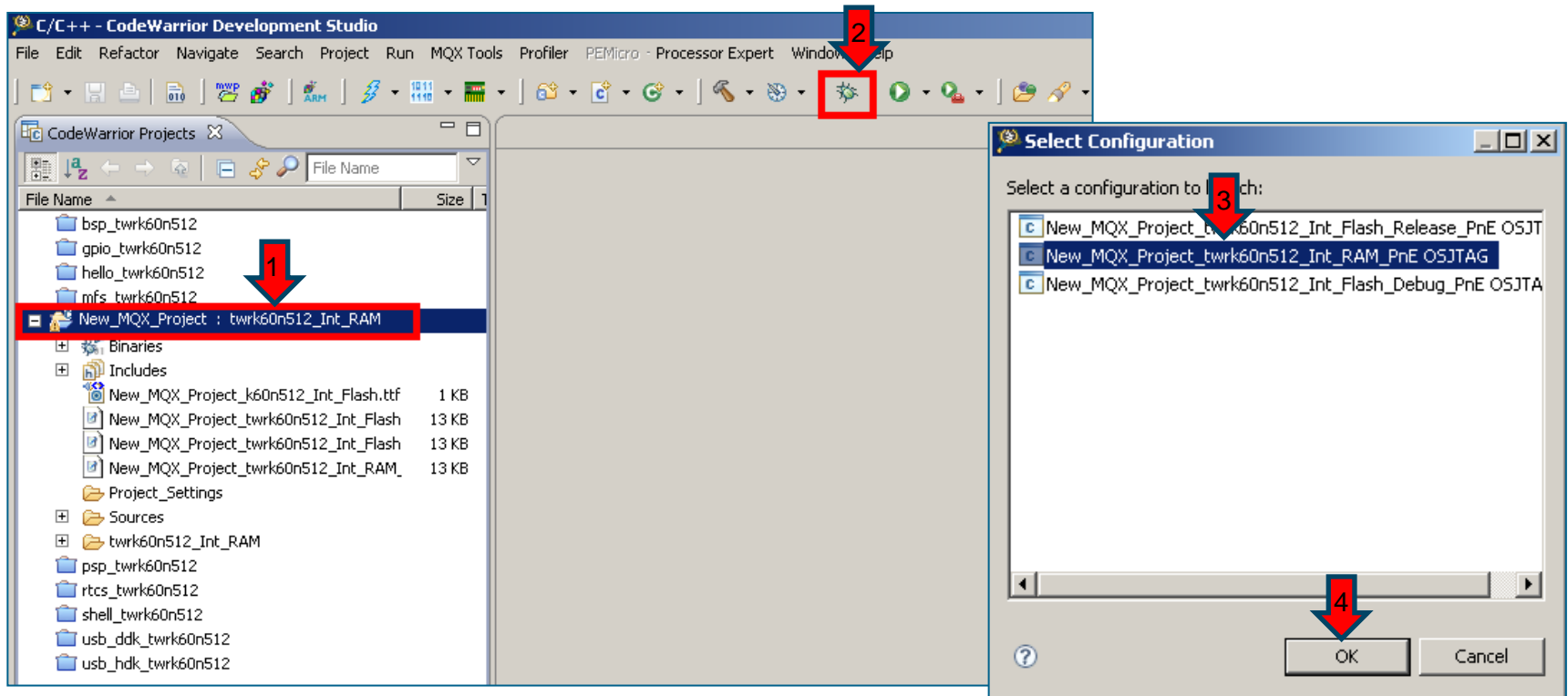
# Build New MQX Project

- Right-Click on Project Explorer **New\_MQX\_Project** and Build Project



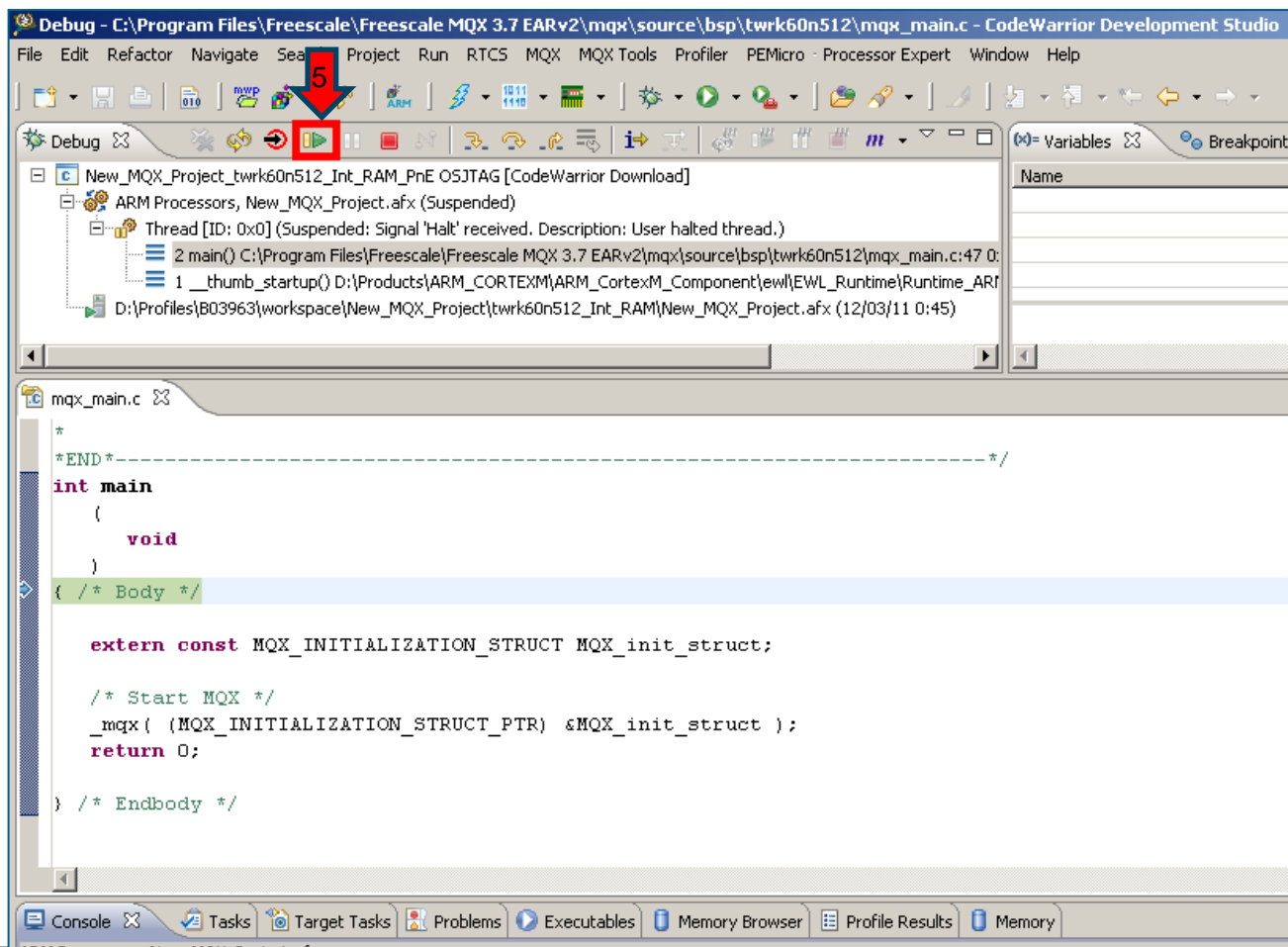
# Debug New MQX Project

- ▶ Select **New\_MQX\_Project : twrk60n512\_Int\_RAM**
- ▶ Select **New\_MQX\_Project\_twrk60n512\_Int\_Ram\_PnE OSJTAG**



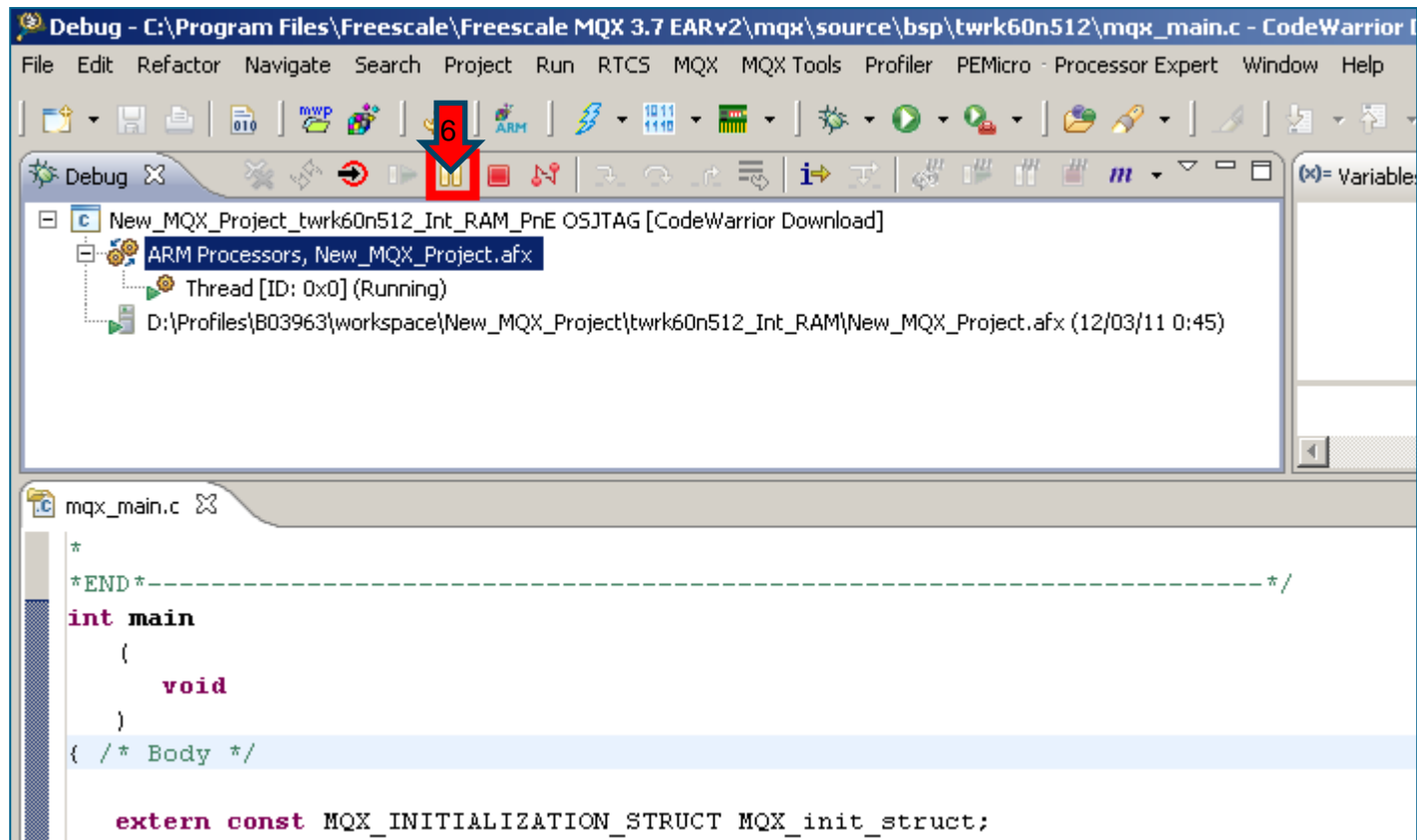
# Run New MQX Project

## ► Execute the code 'Run' icon

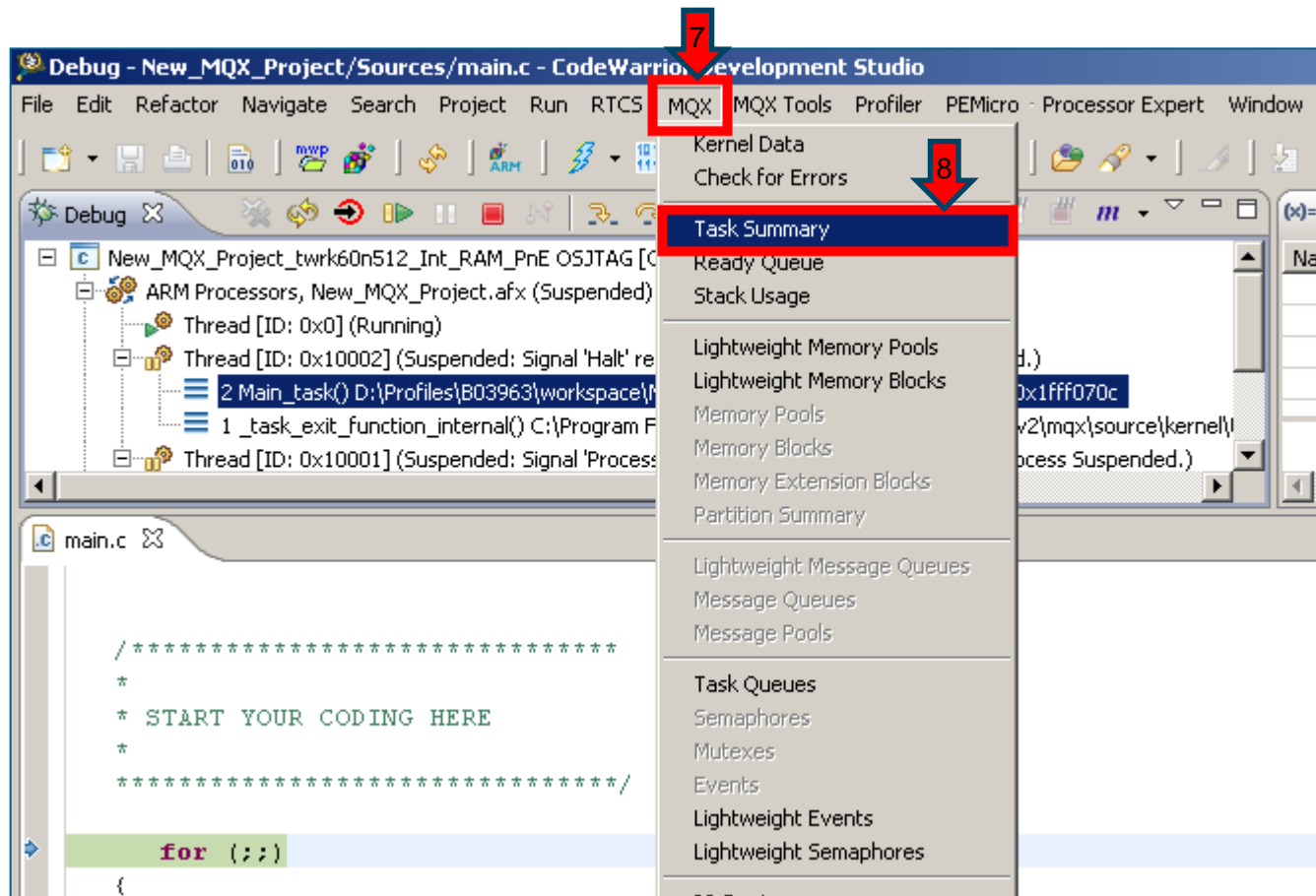


# Run New MQX Project

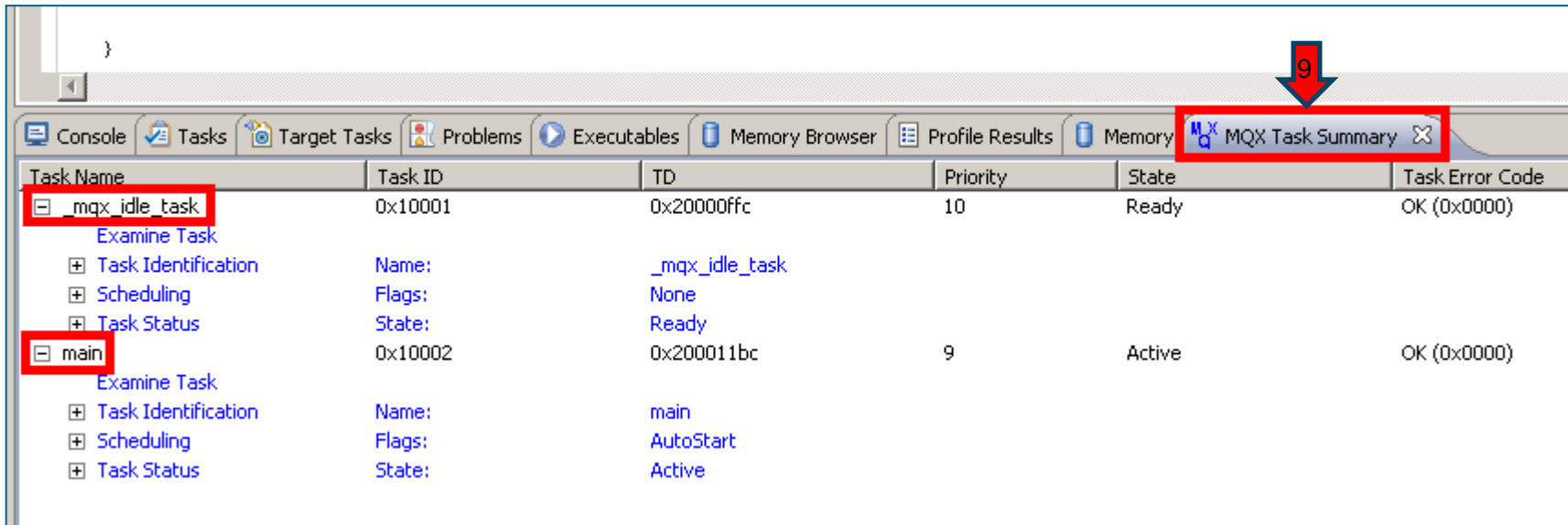
## ► Pause execution



## ► MQX -> Task Summary



## ► Observe Tasks in your Application

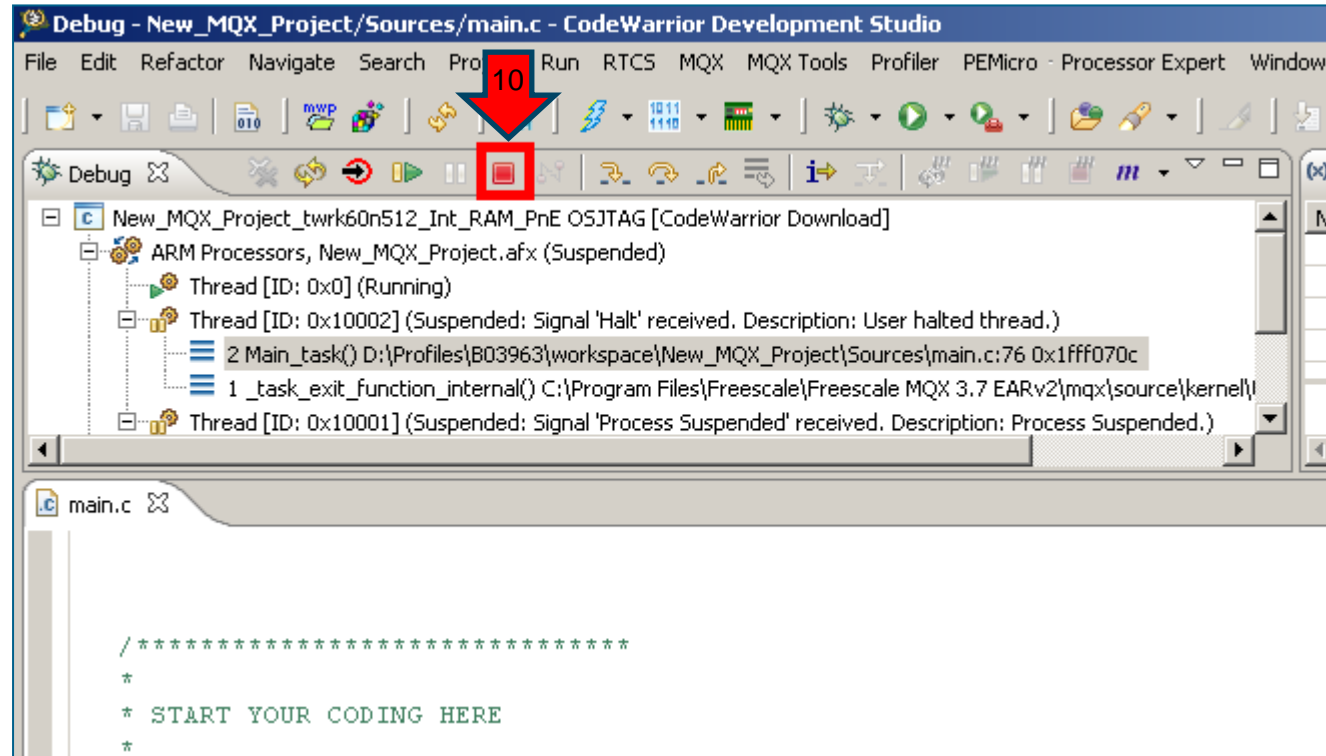


The screenshot displays the 'MQX Task Summary' window, which is highlighted with a red box and a red arrow pointing to it from the top right. The window contains a table of tasks:

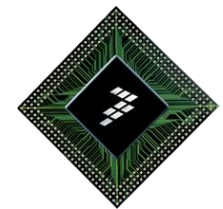
Task Name	Task ID	TD	Priority	State	Task Error Code
<input checked="" type="checkbox"/> _mqx_idle_task	0x10001	0x20000ffc	10	Ready	OK (0x0000)
Examine Task					
<input checked="" type="checkbox"/> Task Identification	Name:	_mqx_idle_task			
<input checked="" type="checkbox"/> Scheduling	Flags:	None			
<input checked="" type="checkbox"/> Task Status	State:	Ready			
<input checked="" type="checkbox"/> main	0x10002	0x200011bc	9	Active	OK (0x0000)
Examine Task					
<input checked="" type="checkbox"/> Task Identification	Name:	main			
<input checked="" type="checkbox"/> Scheduling	Flags:	AutoStart			
<input checked="" type="checkbox"/> Task Status	State:	Active			



## ► Terminate execution

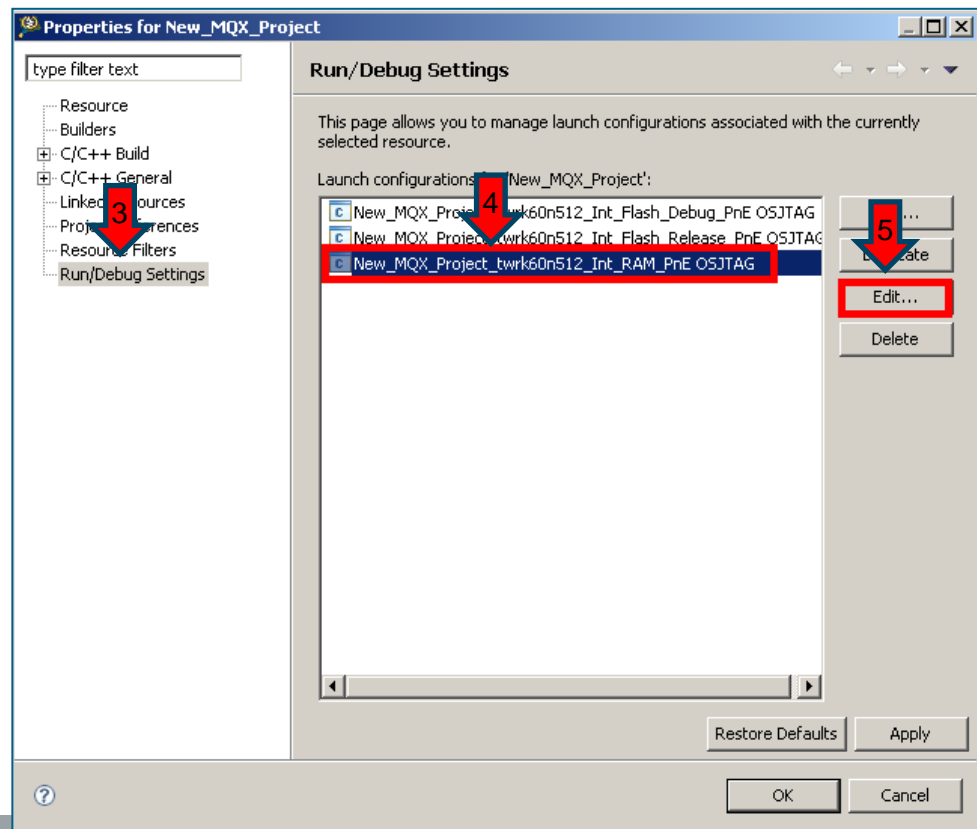
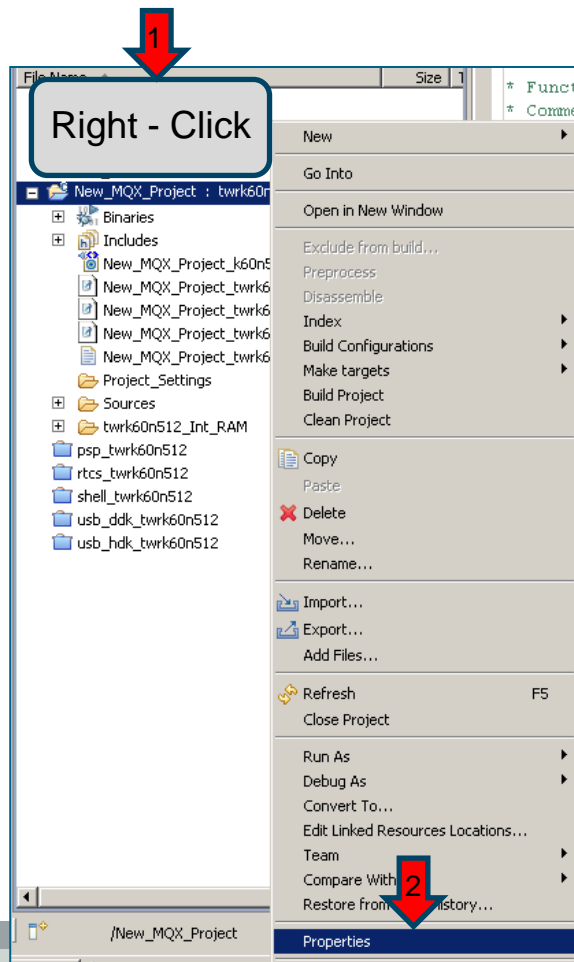


# Debugging with JLink



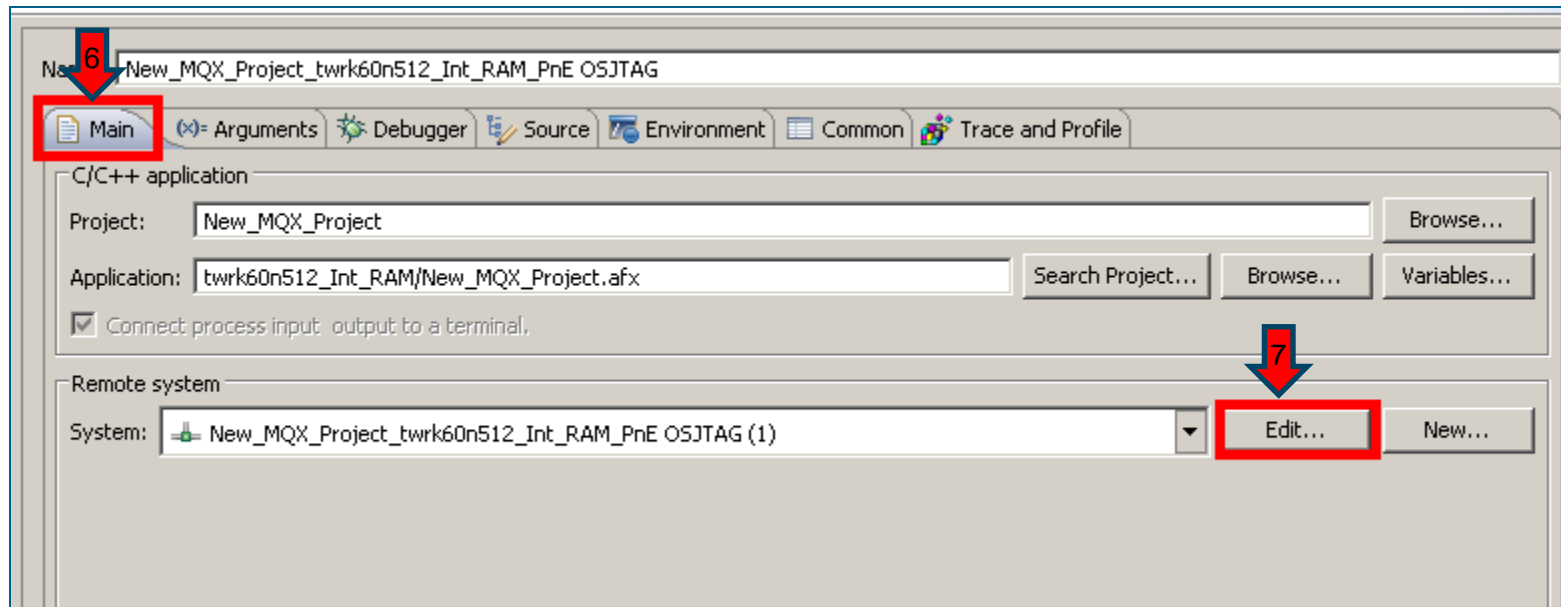
# Debugging with JLINK - Change Connection Type

## ► Edit Connection Settings of the project



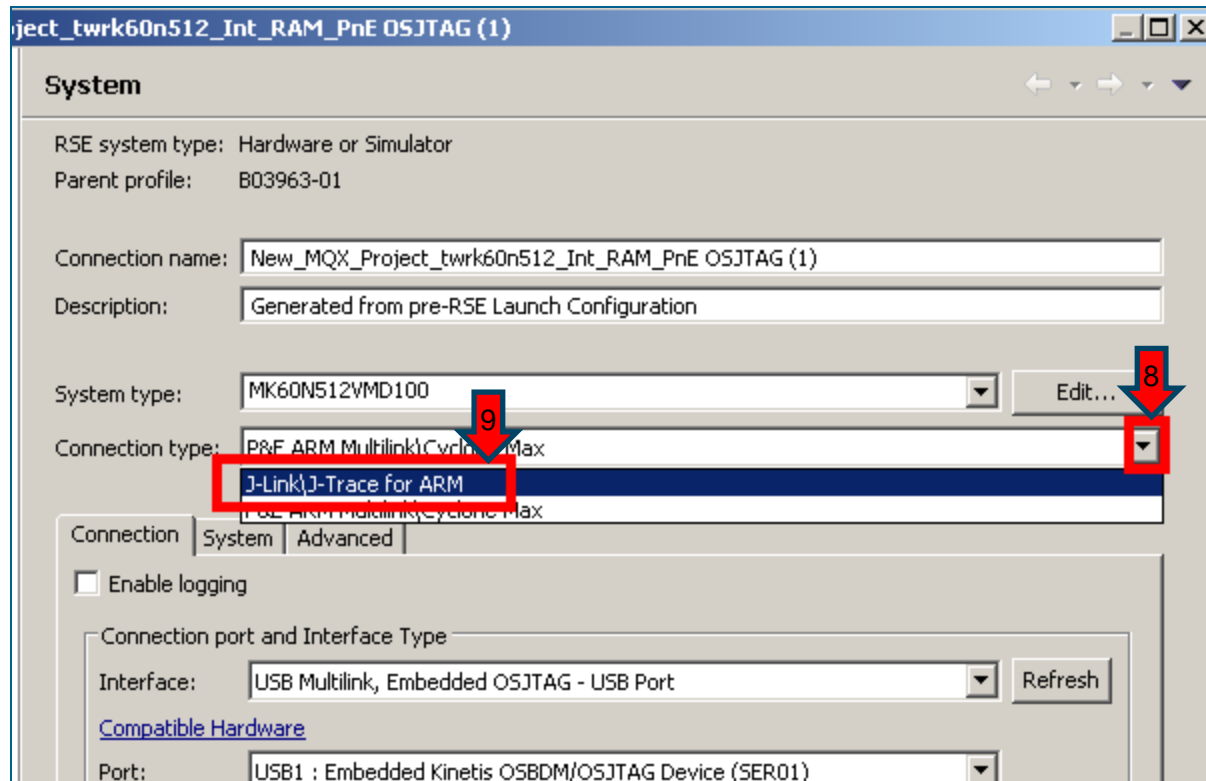
# Change Connection Type

## ► Edit Remote System



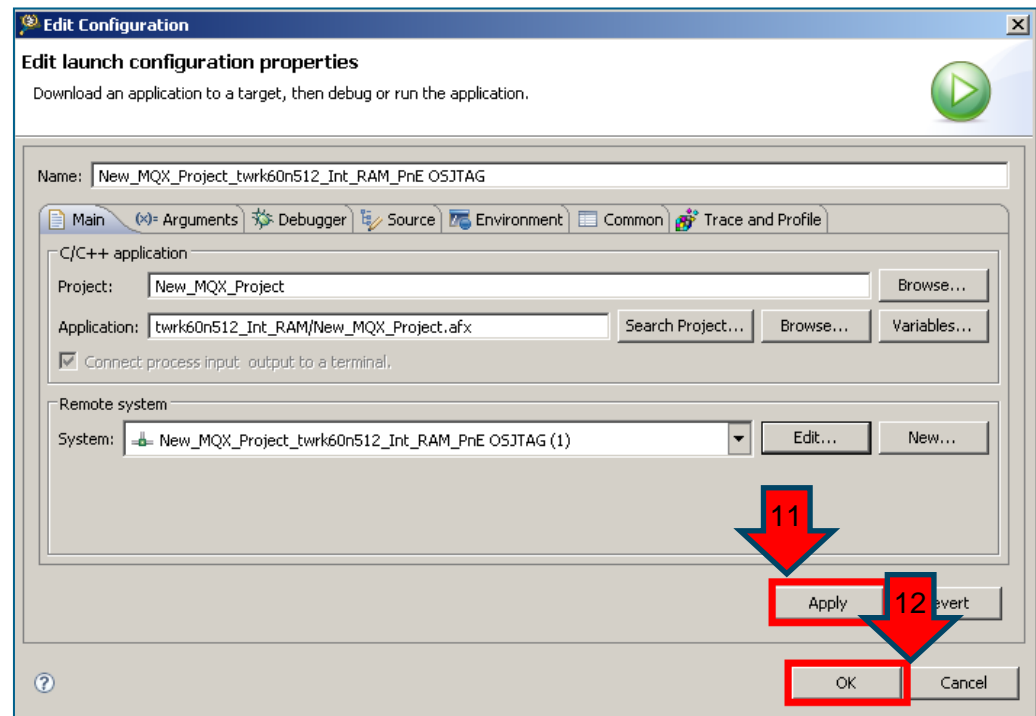
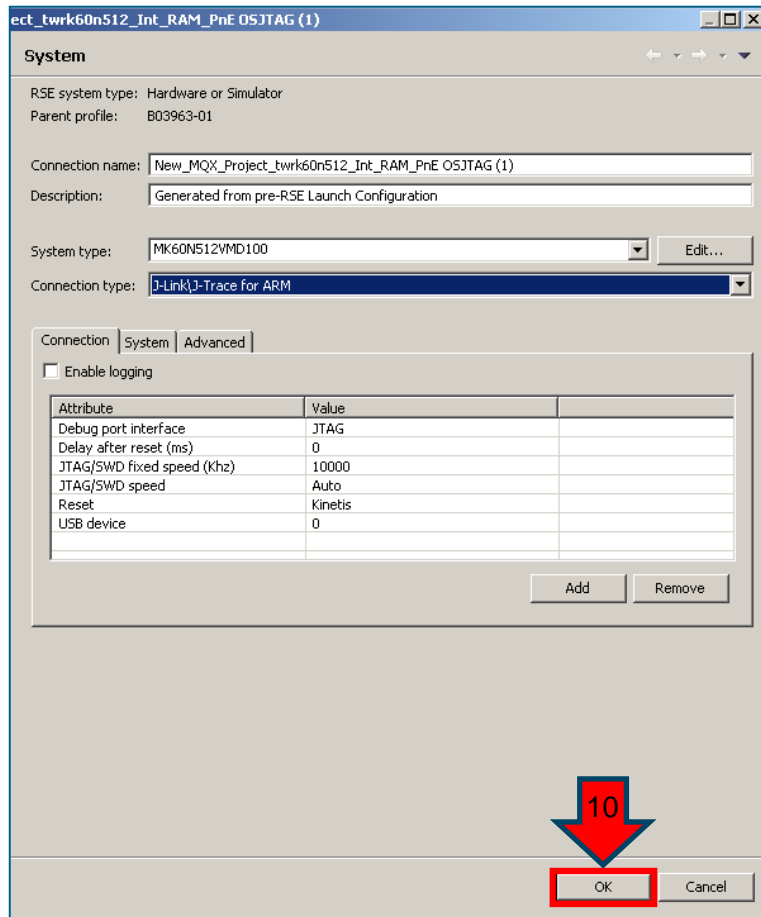
# Change Connection Type

## ► Select J-Link\J-Trace for ARM



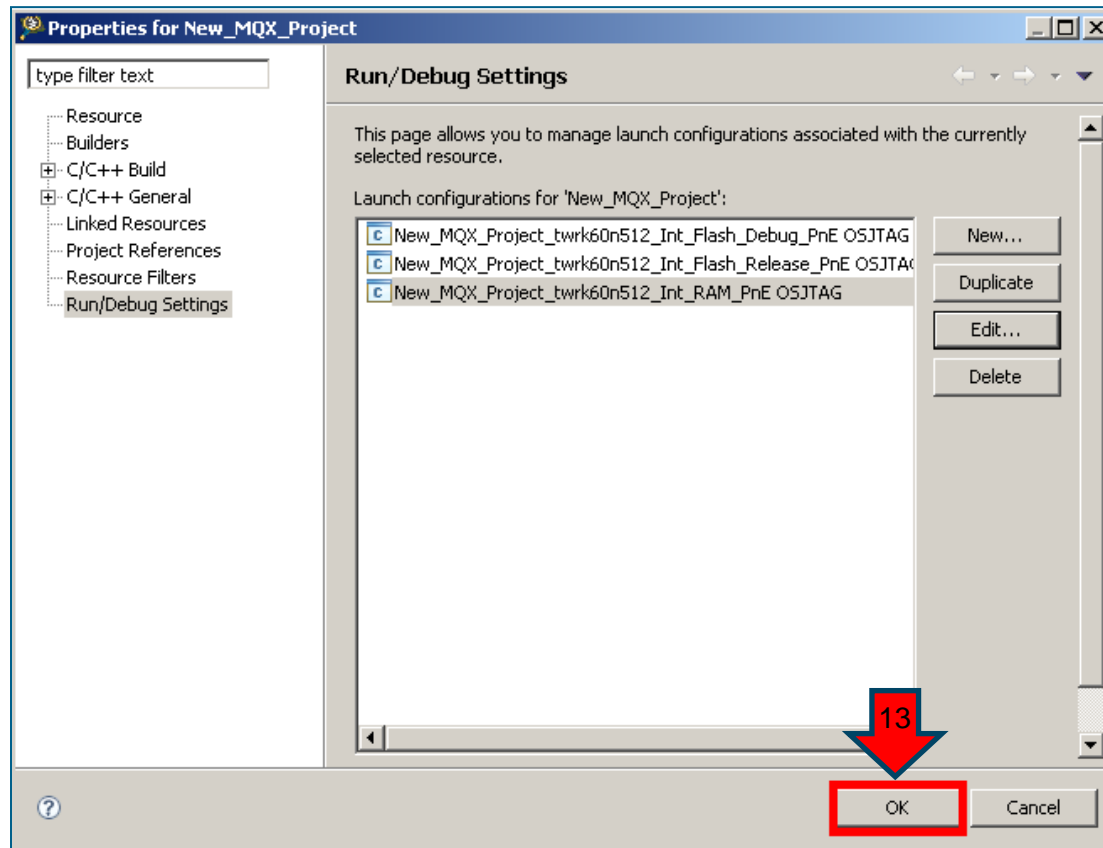
# Change Connection Type

## ► Confirm changes



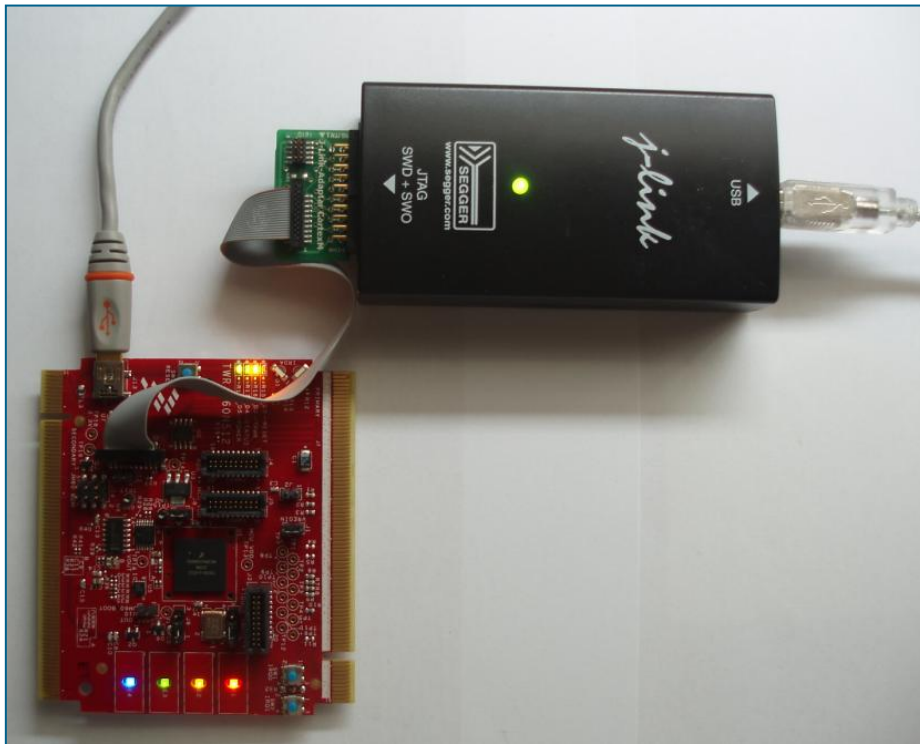
# Change Connection Type

► Click OK



# Debug with JLink

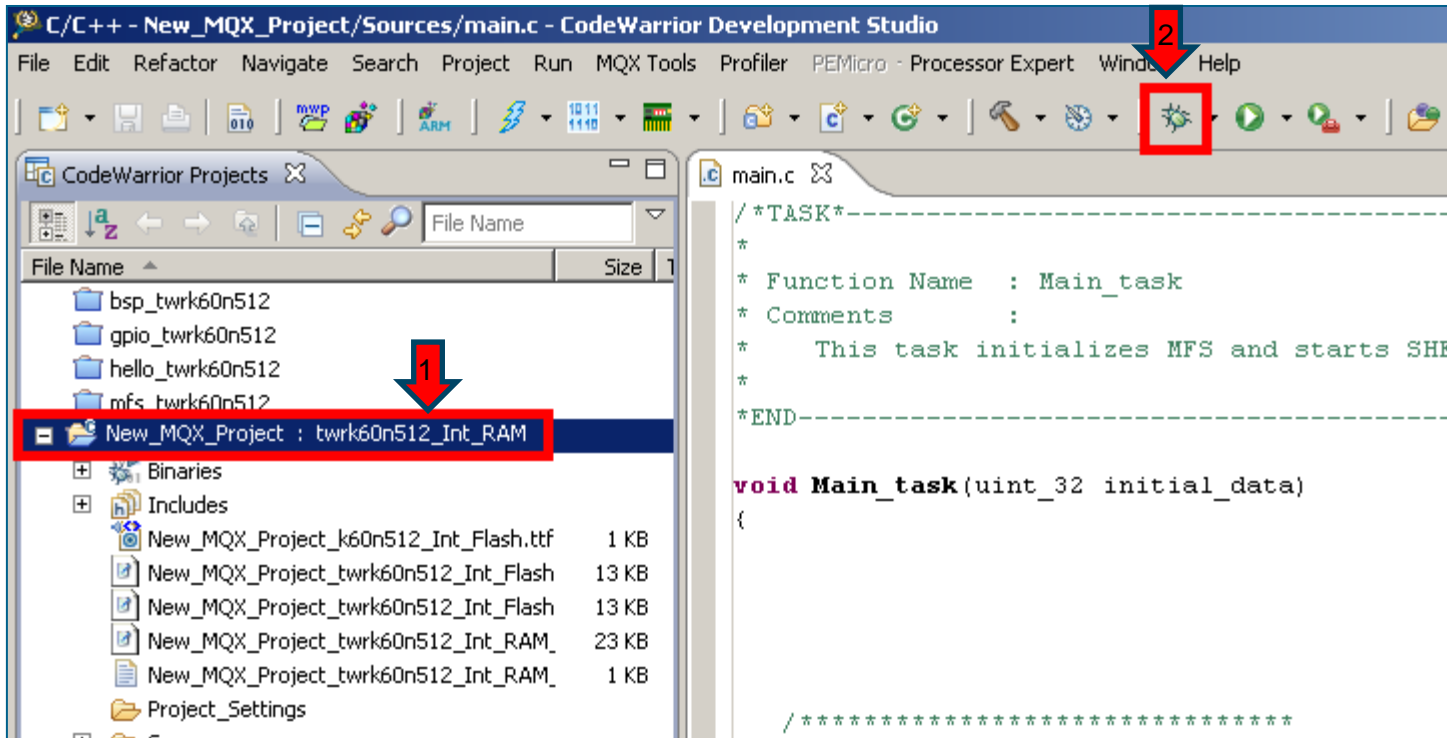
- ▶ Connect J-Link target cable to TWR-K60N512 (J11)
- ▶ Connect USB J-Link cable to laptop
- ▶ Connect USB Cable to TWR-K60N512 (J13) and laptop





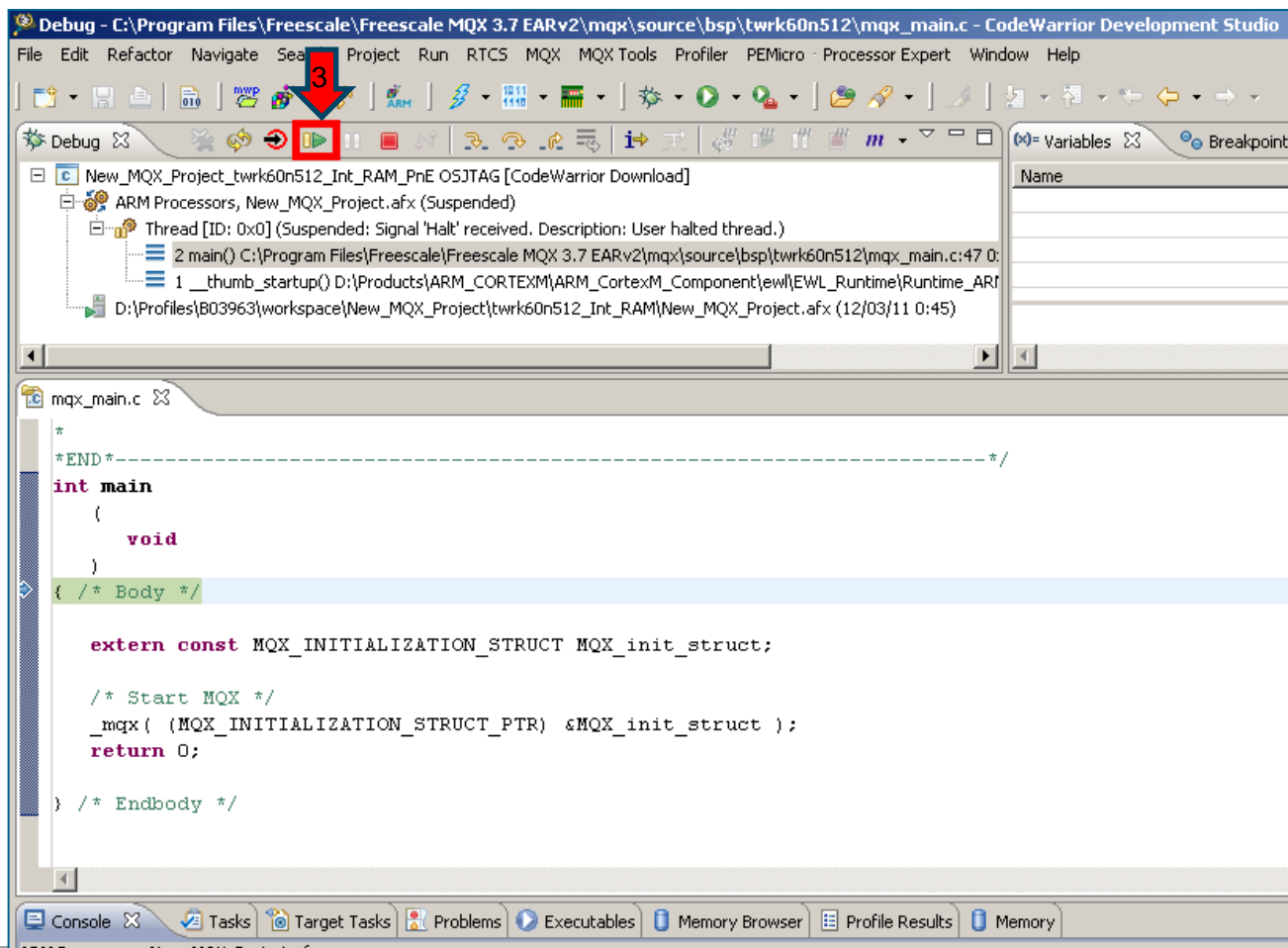
# Change Connection Type

- ▶ Select **New\_MQX\_Project : twrk60n512\_Int\_RAM**
- ▶ Click Debug



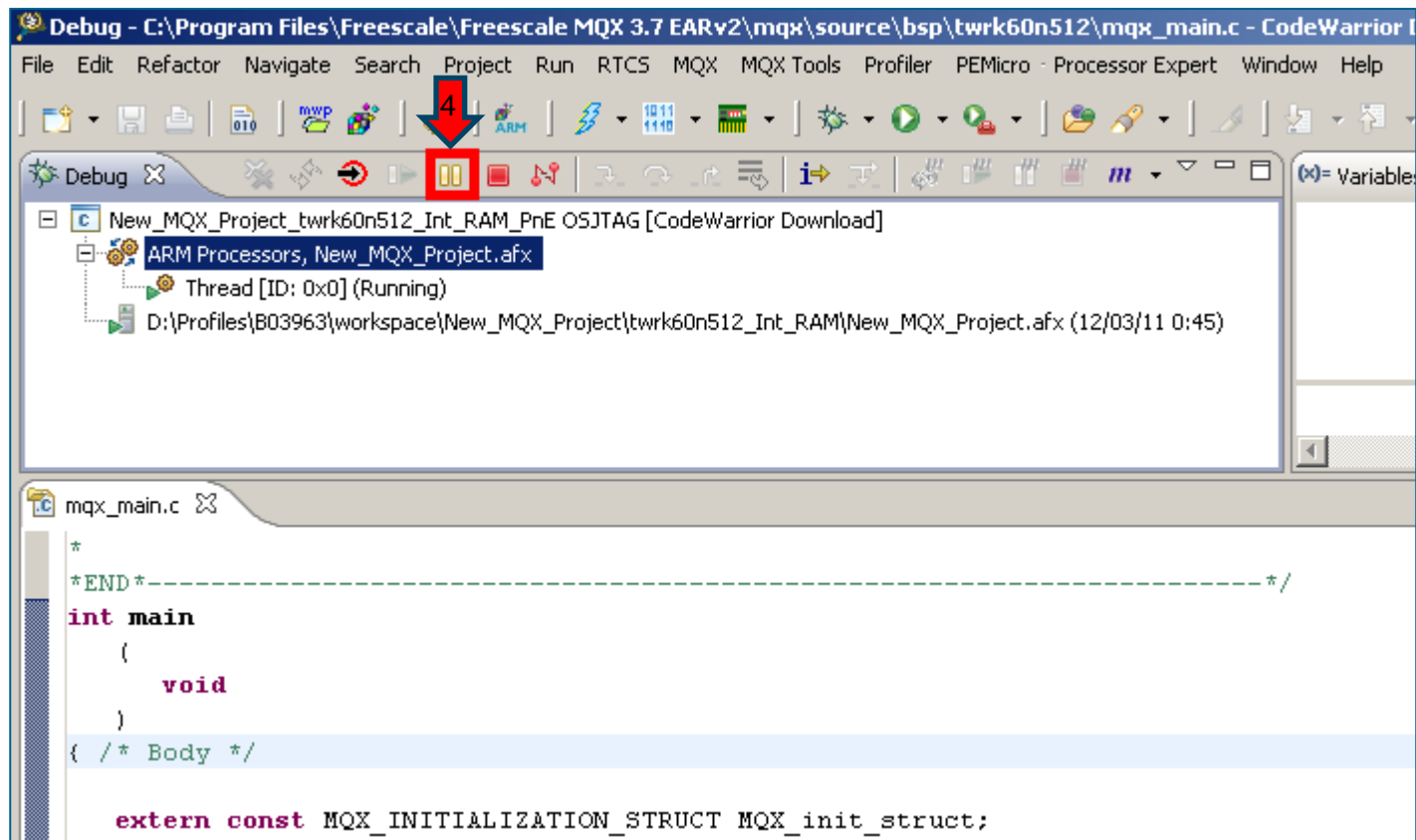
# Run New MQX Project

## ► Execute the code 'Resume' icon

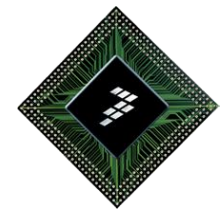


# Run New MQX Project

## ► Pause execution

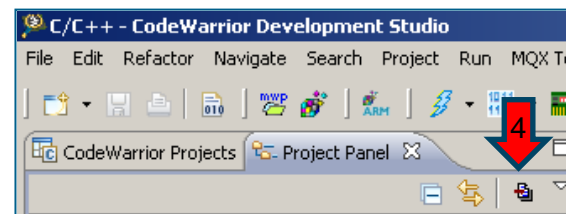
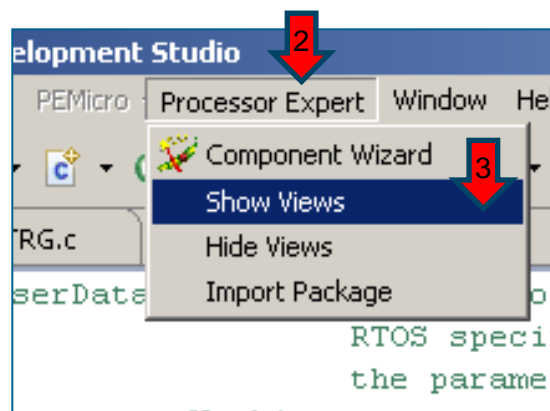
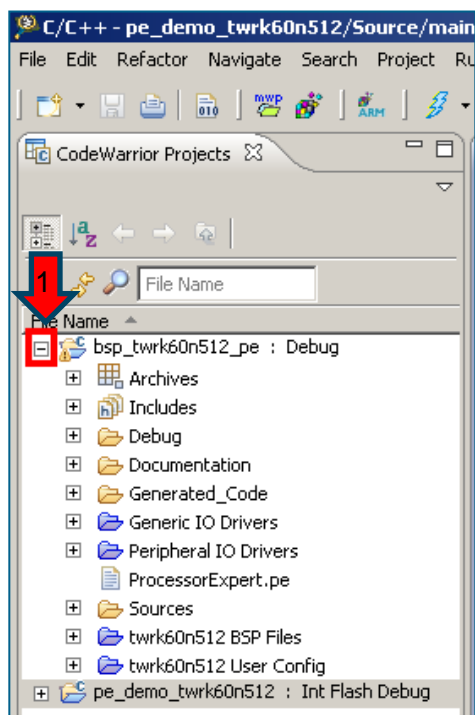


# CW10.2, MQX 3.8 and Processor Expert



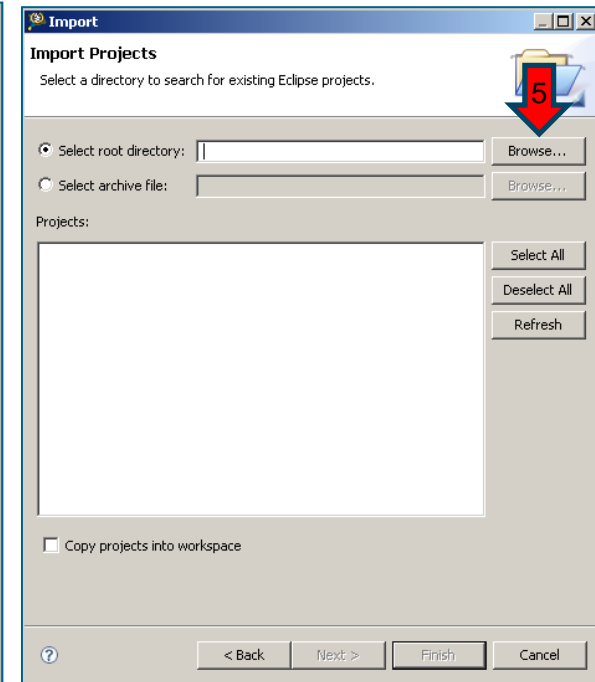
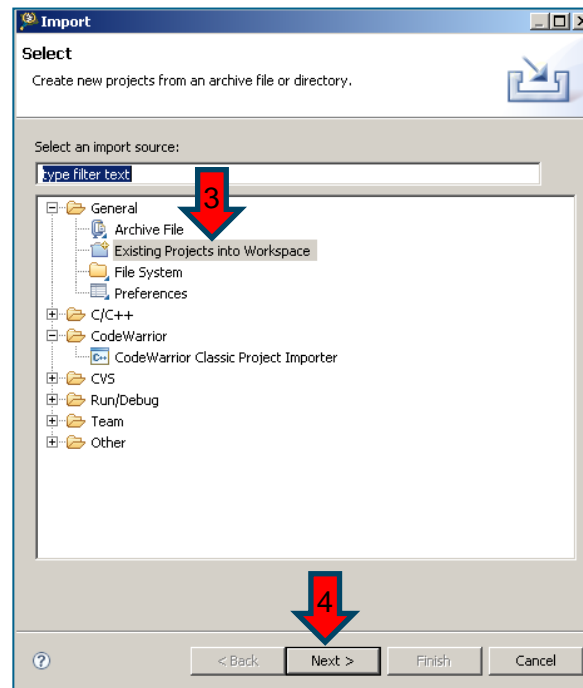
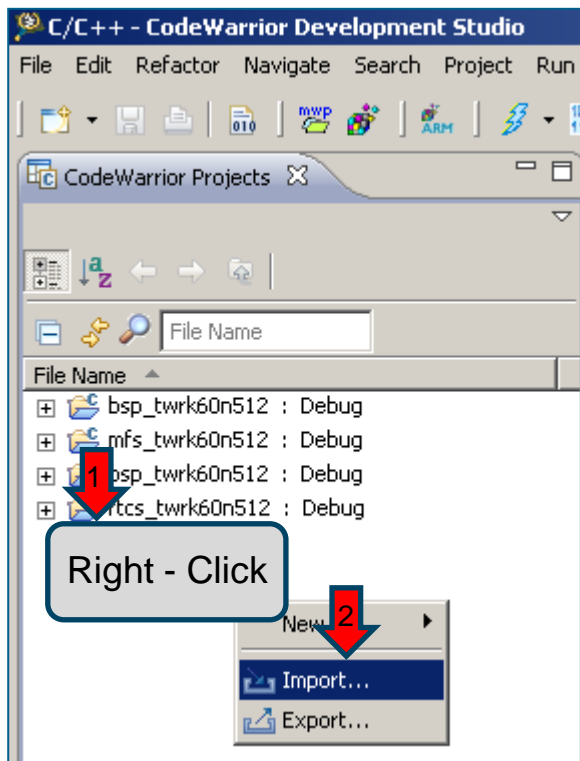
- ▶ All Kinetis BSP projects are Processor Expert Ready and Processor Expert drivers are enabled in MQX RTOS environment

- ▶ Expand **bsp\_twrk60n512** project view
- ▶ Show Processor Expert View
- ▶ Generate code



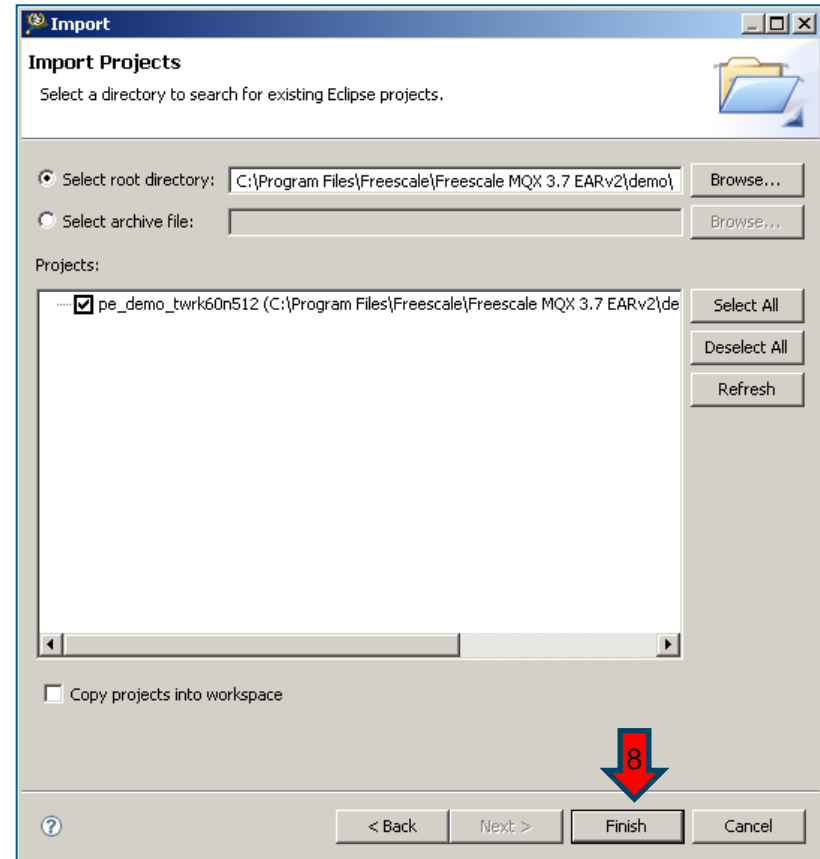
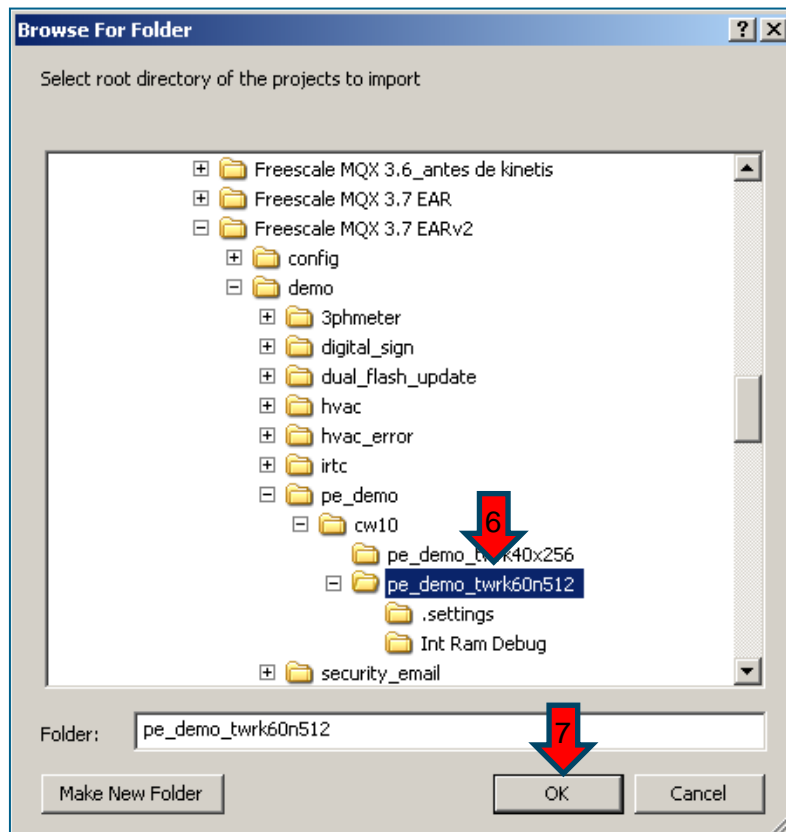
# Import MQX PE Demo

- ▶ Right-Click on Project Explorer and Import
- ▶ Select Existing Projects into Workspace and Browse



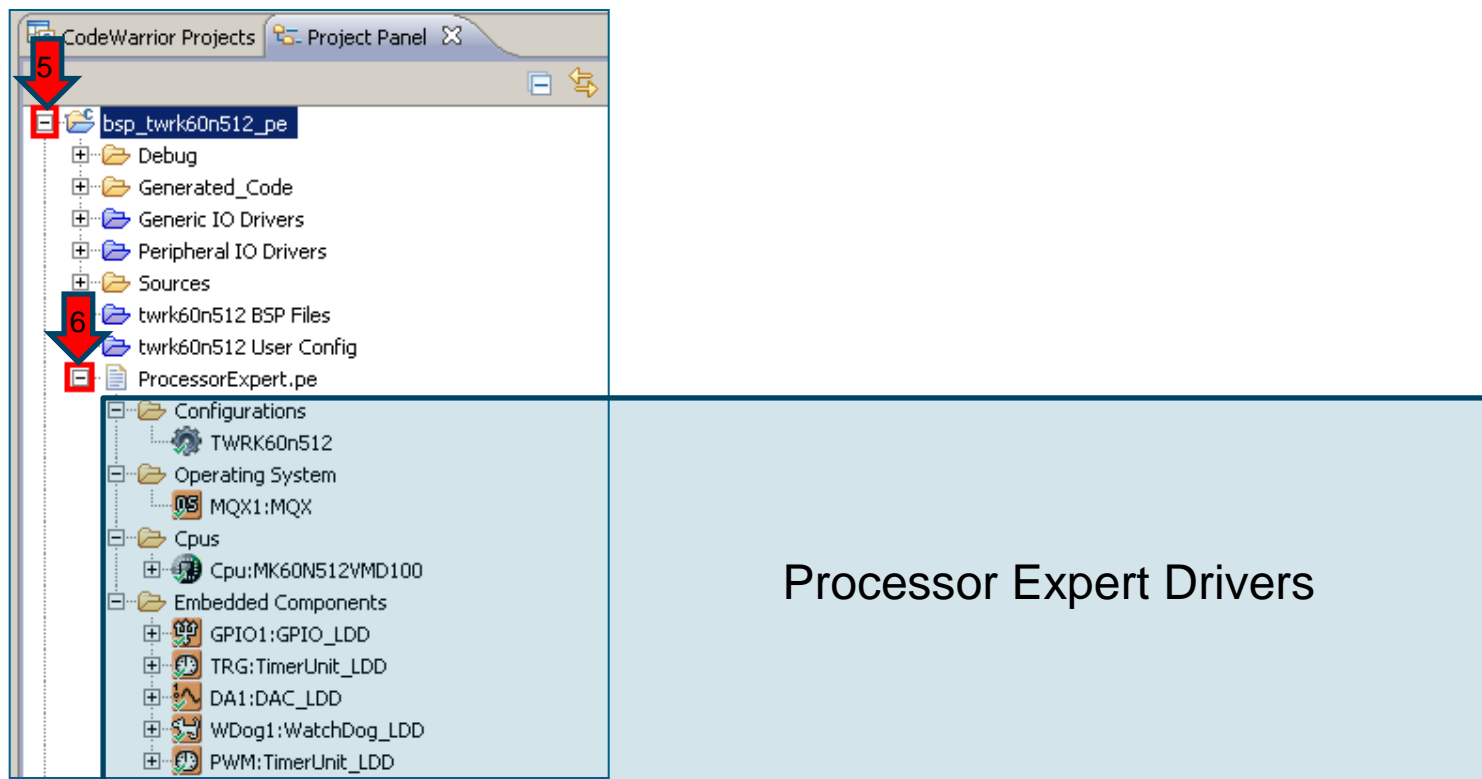
# Import MQX PE Demo

- Select *<install mqx folder>\mqx\pe\_demo\pe\_demo\_twrk60n512*





- Expand **bsp\_twrk60n512** project view



- Click on PE components to watch the properties

**7** CodeWarrior Projects | Project Panel

**8** bsp\_twrk60n512\_pe

- Debug
- Generated\_Code
- Generic IO Drivers
- Peripheral IO Drivers
- Sources
- twrk60n512 BSP Files
- twrk60n512 User Config
- ProcessorExpert
- Configuration
  - TWRK60N512
  - Operating System
  - MQX1:MQX
  - Cpus
    - Cpu:MK60N512VMD100
  - Embedded Components
    - GPIO1:GPIO\_LDD
    - TRG:TimerUnit\_LDD
    - DA1:DAC\_LDD
    - WDog1:WatchDog\_LDD
    - PWM:TimerUnit\_LDD

**9** \*Configuration Inspector

Name	Value
<b>Optimizations</b>	
Ignore range checking	no
Ignore enable test	no
Utilize after reset values	no
Complete initialization in Peripheral Init.Component	no

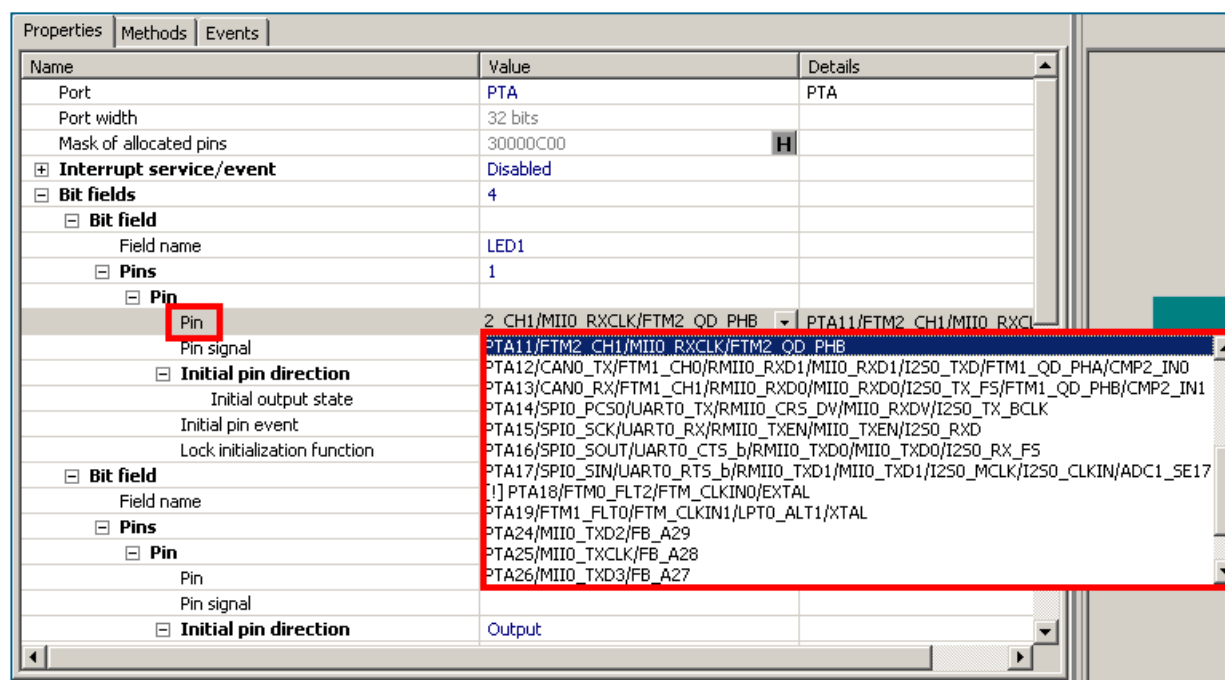
**Properties** | Methods | Events

Name	Value
Component name	MQX1
<b>Settings</b>	
<b>RTOS adaptor</b>	
<b>Memory allocation</b>	
User function for memory allocation	no
User function for memory deallocation	no
User function name	
<b>Critical section</b>	
User function for entering critical section	no
User function for exiting critical section	no

**Properties** | Methods | Events

Name	Value	Details
Port	PTA	PTA
Port width	32 bits	
Mask of allocated pins	30000C00	H
<b>Interrupt service/event</b>	Disabled	
<b>Bit fields</b>	4	
<b>Bit field</b>		
Field name	LED1	
<b>Pins</b>	1	
<b>Pin</b>		
Pin	PTA11/FTM2_CH1/MII0_RXCLK/FTM2...	PTA11/FTM2_CH1/MII0_RXCLK
Pin signal	LED_D17	
<b>Initial pin direction</b>	Output	
Initial output state	1	
Initial pin event	Disabled	
Lock initialization function	no	
<b>Bit field</b>		
Field name	LED2	
<b>Pins</b>	1	
<b>Pin</b>		
Pin	PTA28/MII0_TXER/FB_A25	PTA28/MII0_TXER/FB_A25
Pin signal		
<b>Initial pin direction</b>	Output	

- ▶ Processor Expert gives you a easy way to add device drivers to MQX BSP.
- ▶ In BSP example two Timers GPIO are included
- ▶ Properties of component can be changed easily, for example GPIO pin



- ▶ **GPIO1** component in BSP is driving LED's in Tower board
- ▶ **TRG** Timer will generate a 64KHz interrupt

\*Component Inspector

Basic Advanced

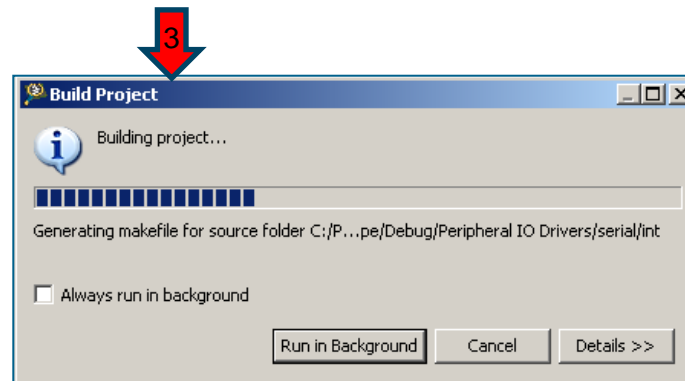
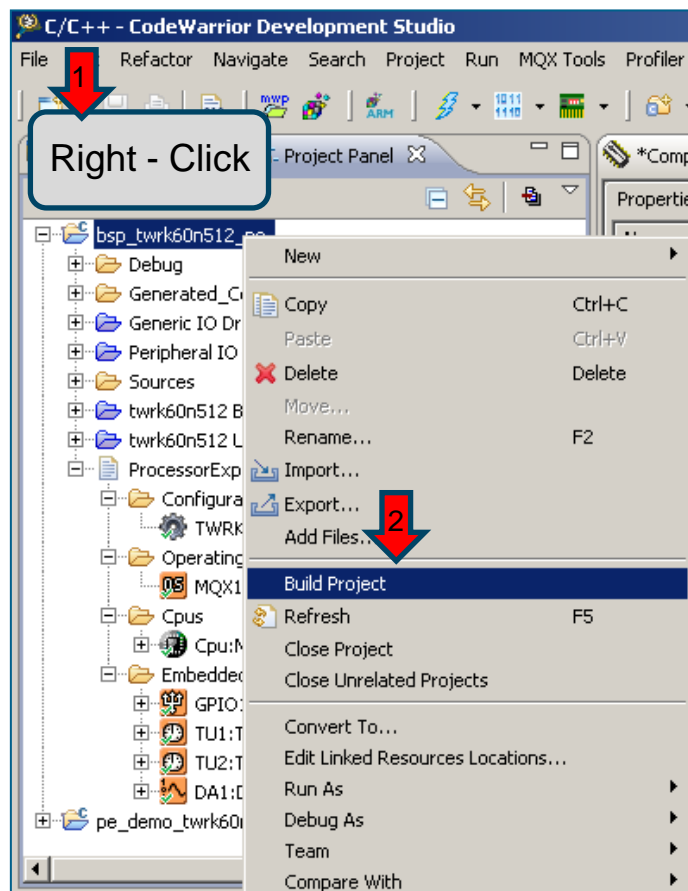
Properties Methods Events

Name	Value	Details
Module name	FTM2	FTM2
Counter	FTM2_CNT	FTM2_CNT
Counter direction	Up	
Counter width	16 bits	
Value type	uint16_t	uint16_t
<input type="checkbox"/> Input clock source	Internal	
Counter frequency	48 MHz	48 MHz
<input type="checkbox"/> Counter restart	On-match	
Period device	FTM2_MOD	FTM2_MOD
Period	64000 Hz	64000 Hz
<input type="checkbox"/> Interrupt	Enabled	
Interrupt priority	medium priority	8
Channel list	0	
<input type="checkbox"/> Initialization		

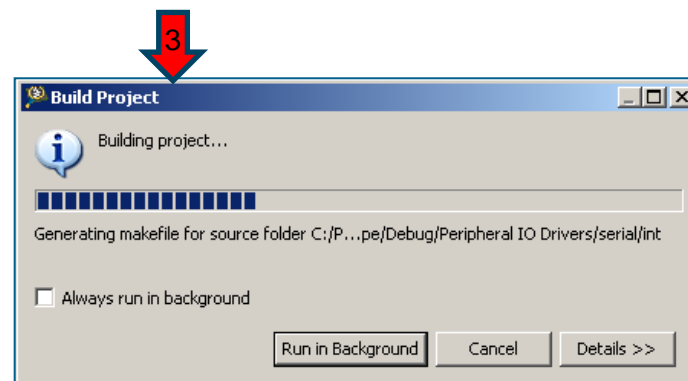
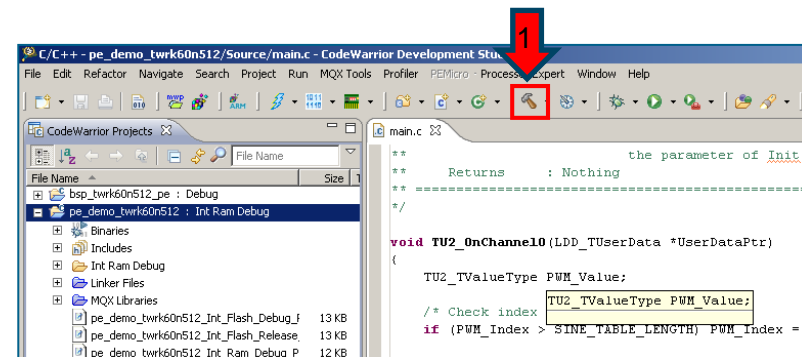
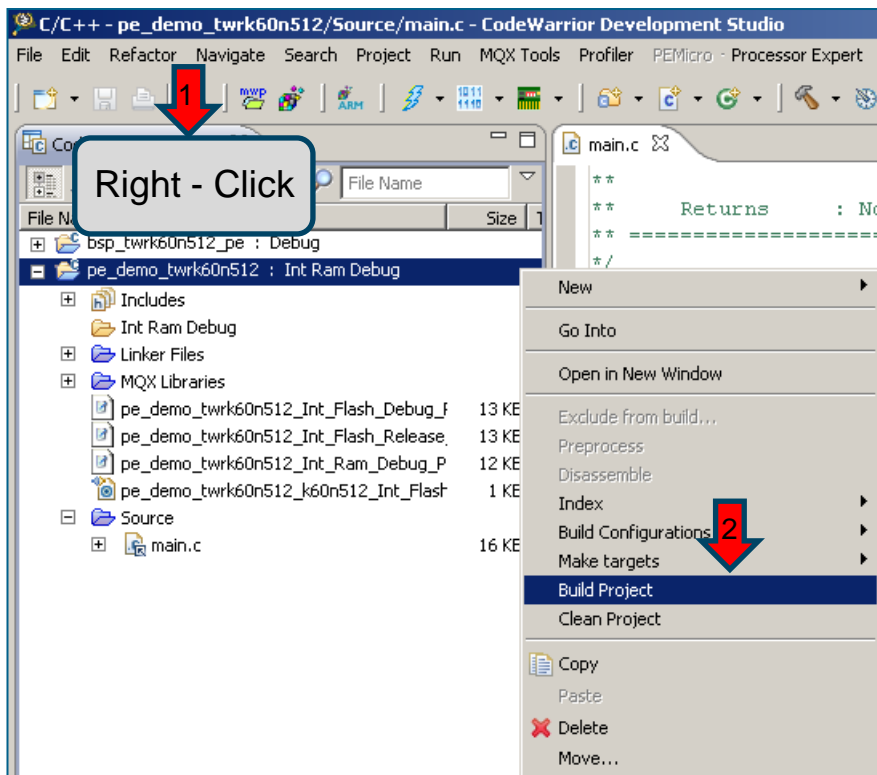
- **PWM** configures Channel 0 in Flex Timer 0 a PWM of 4096 timer-ticks

Properties   Methods   Events		
Name	Value	Details
Module name	FTM0	FTM0
Counter	FTM0_CNT	FTM0_CNT
Counter direction	Up	
Counter width	16 bits	
Value type	uint16_t	uint16_t
<input type="checkbox"/> Input clock source	Internal	
Counter frequency	48 MHz	48 MHz
<input type="checkbox"/> Counter restart	On-match	
Period device	FTM0_MOD	FTM0_MOD
Period	4096 timer-ticks	4096 timer-ticks
<input type="checkbox"/> Interrupt	Enabled	
Interrupt priority	medium priority	8
<input type="checkbox"/> Channel list	1	
<input type="checkbox"/> Channel 0		
<input type="checkbox"/> Mode	Compare	
Compare	FTM0_COV	FTM0_COV
Offset	1 timer-ticks	1 timer-ticks
<input type="checkbox"/> Output on compare	Set	
Output on overrun	Clear	
Initial state	Low	
Output pin	PTC1/SPI0_PCS3/UART1_RTS_b/FTM...	PTC1/SPI0_PCS3/UART1_RTS_b/FTM...
<input type="checkbox"/> Interrupt	Disabled	
<input type="checkbox"/> Initialization		
Enabled in init. code	no	
<input type="checkbox"/> Event mask		

- Right-Click on Project Explorer **bsp\_twrk60n512** and Build Project

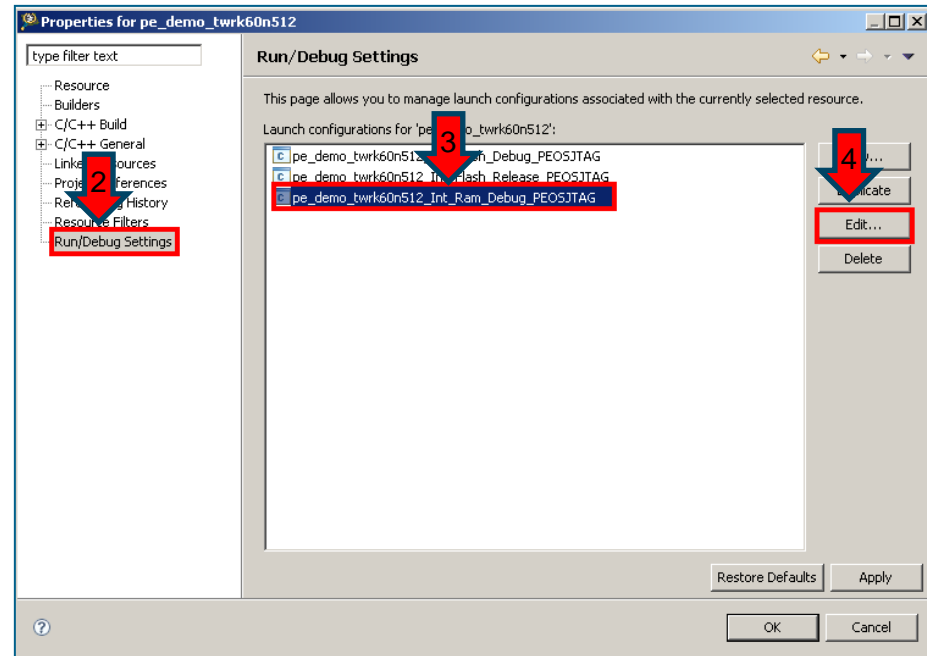
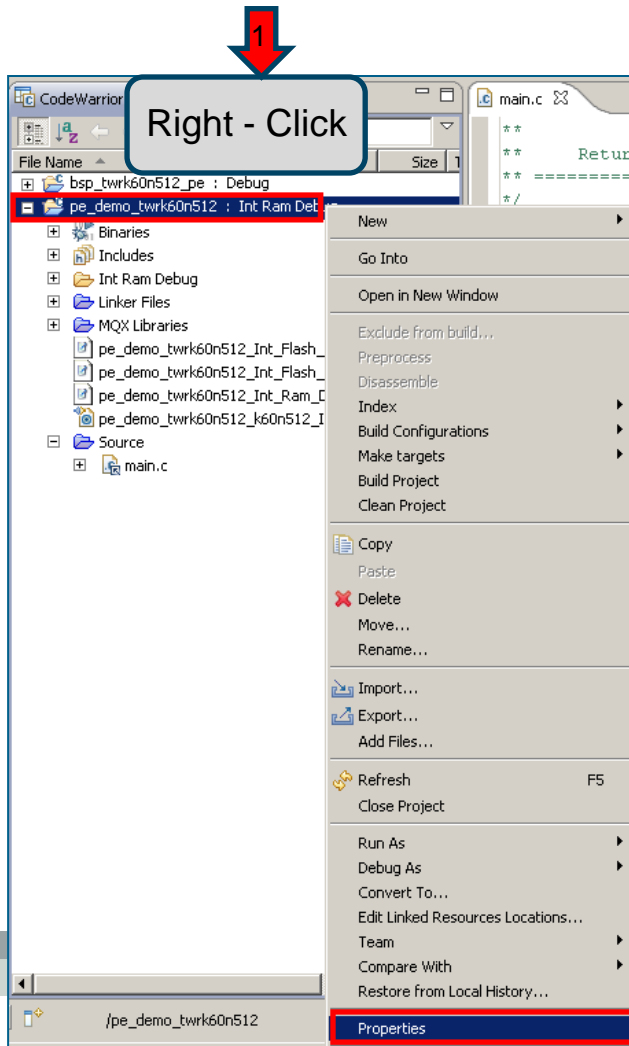


- ▶ Right-Click on Project Explorer **pe\_demo\_twrk60n512** and Build Project or click on icon 



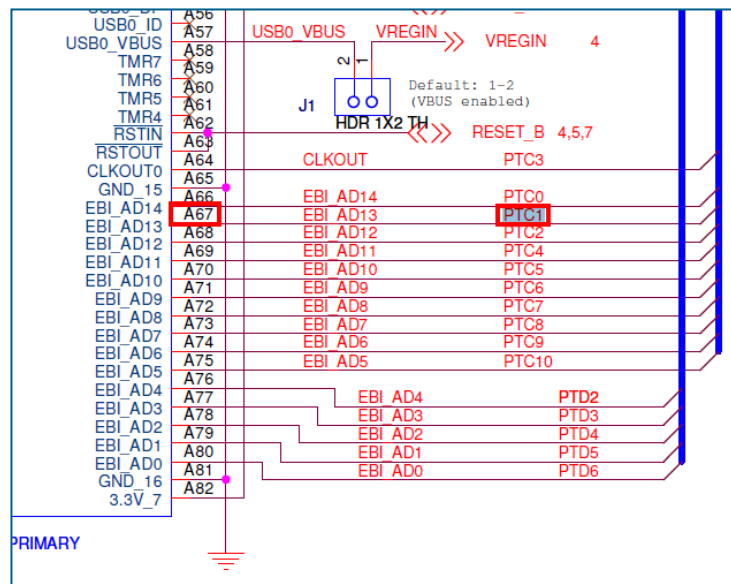
# Debug Connection MQX PE Demo

- ▶ Right-Click on Project Explorer **pe\_demo\_twrk60n512** and Build Project
- ▶ Edit **pe\_demo\_twrk60n512\_Int\_Ram\_Debug\_PEOSJTAG**

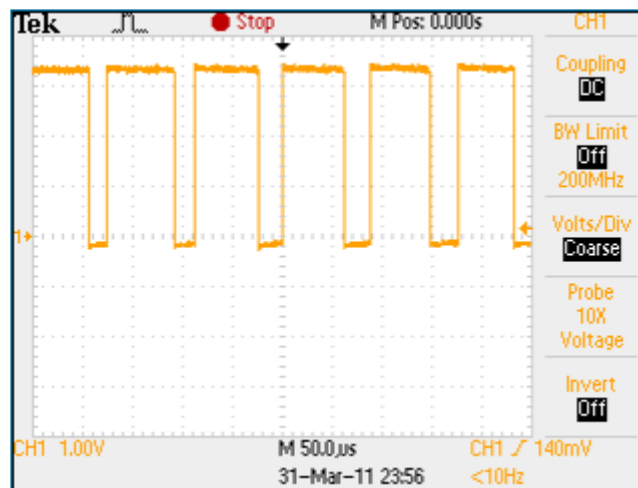




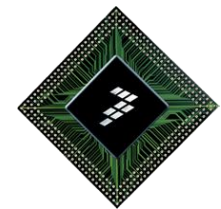
## ► Check PWM output on A67



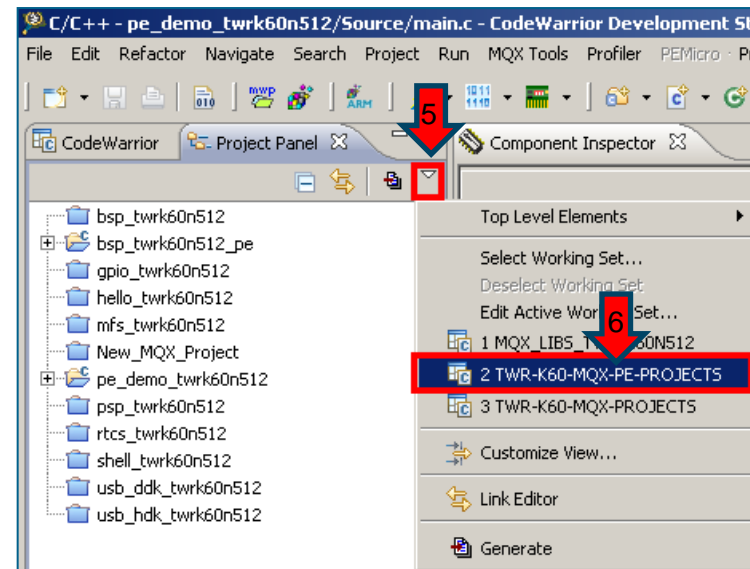
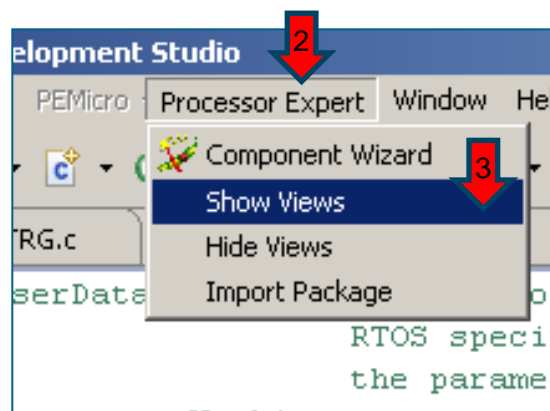
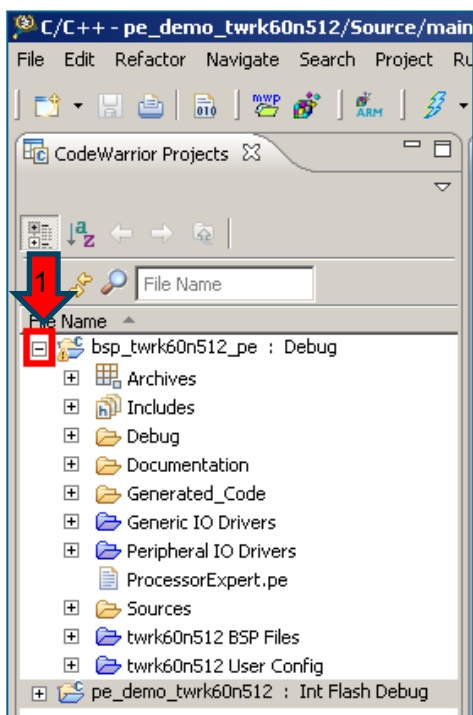
Properties	Methods	Events
Name	Value	Details
Module name	FTM0	FTM0
Counter	FTM0_CNT	FTM0_CNT
Counter direction	Up	
Counter width	16 bits	
Value type	uint16_t	uint16_t
<b>Input clock source</b>		
Counter frequency	Internal	48 MHz
<b>Counter restart</b>		
Period device	FTM0_MOD	FTM0_MOD
Period	4096 timer-ticks	4096 timer-ticks
<b>Interrupt</b>		
Interrupt priority	Enabled	medium priority
<b>Channel list</b>		
<b>Channel 0</b>		
<b>Mode</b>		
Compare	FTM0_C0V	FTM0_C0V
Offset	1 timer-ticks	1 timer-ticks
<b>Output on compare</b>		
Output on overrun	Set	
Initial state	Clear	
Output pin	PTC1/SPI0_PCS3/UART1_RTS_b/FTM...	PTC1/SPI0_PCS3/UART1_RTS_b/FTM...
<b>Interrupt</b>	Disabled	
<b>Initialization</b>		
Enabled in init. code	no	



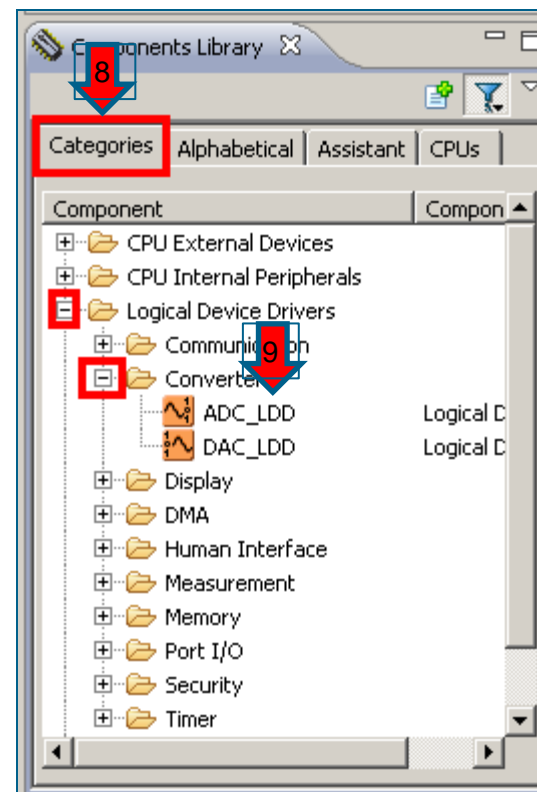
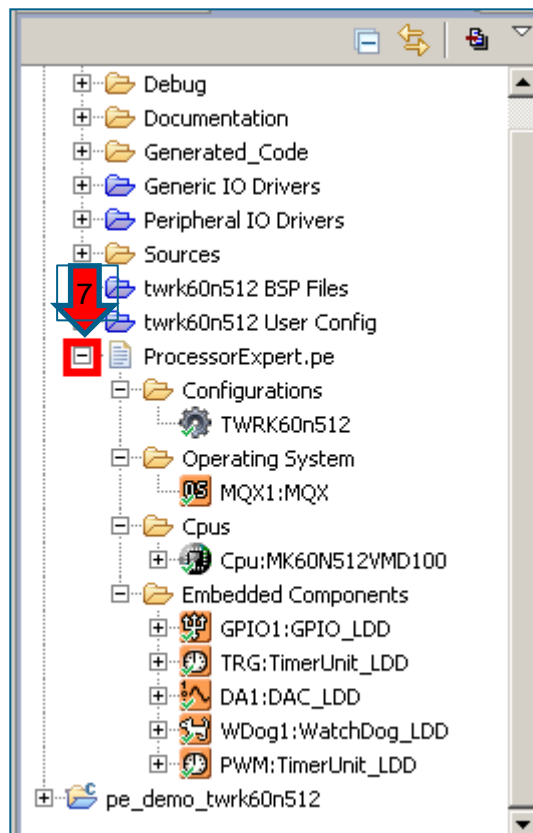
# CW10.2, MQX 3.8 and PE : New LDD driver



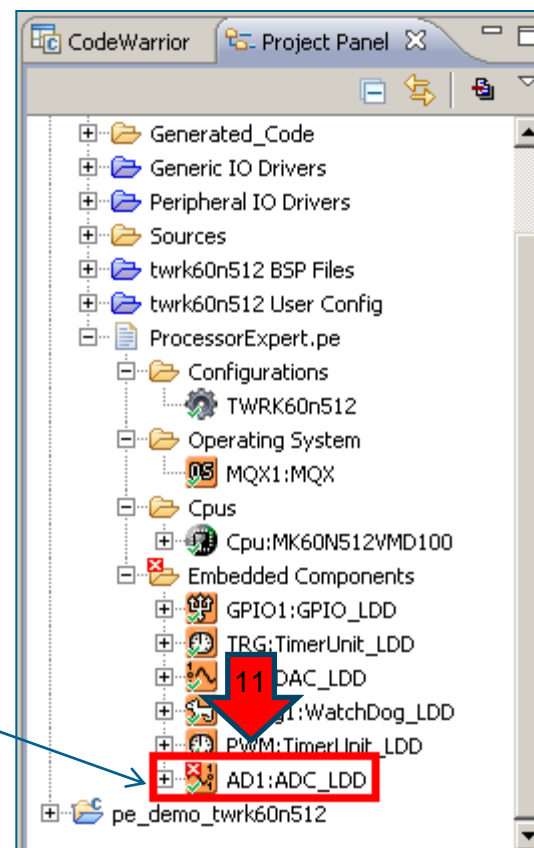
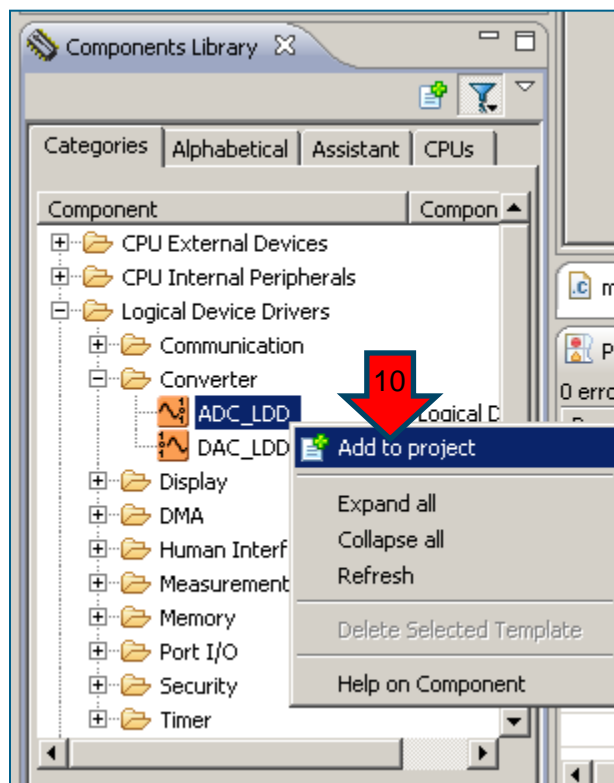
- ▶ Expand **bsp\_twrk60n512** project view
- ▶ Show Processor Expert View
- ▶ Select PE Projects Working Set



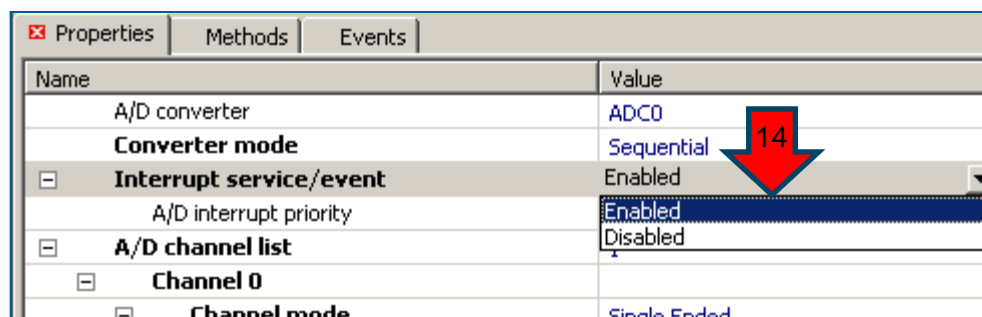
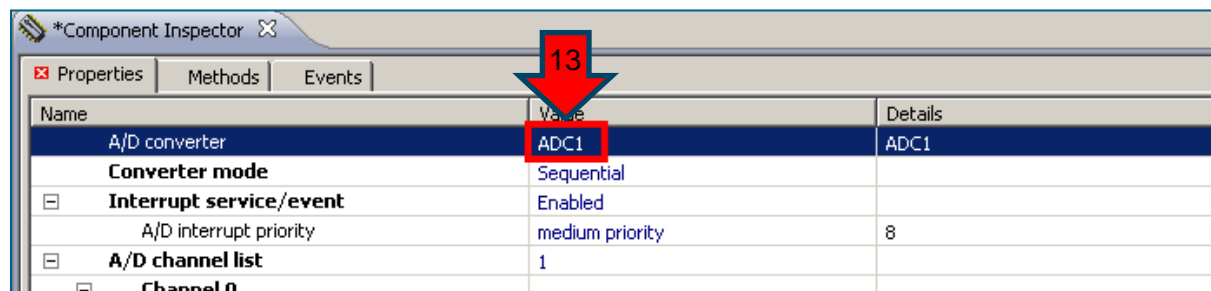
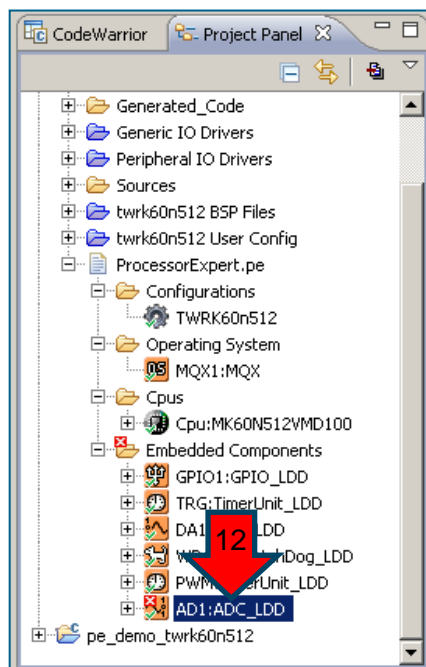
- ▶ Expand Processor Expert Project View
- ▶ Search ADC\_LDD in Components Library window



- ▶ Right click on the component
- ▶ Select Add to project



- ▶ Double click on ADC\_LDD
- ▶ Select ADC1
- ▶ Enable Interrupt service



- ▶ Select **ADC1\_DM1** Channel
- ▶ Enable Static sample groups
- ▶ Open Conversion Time Window

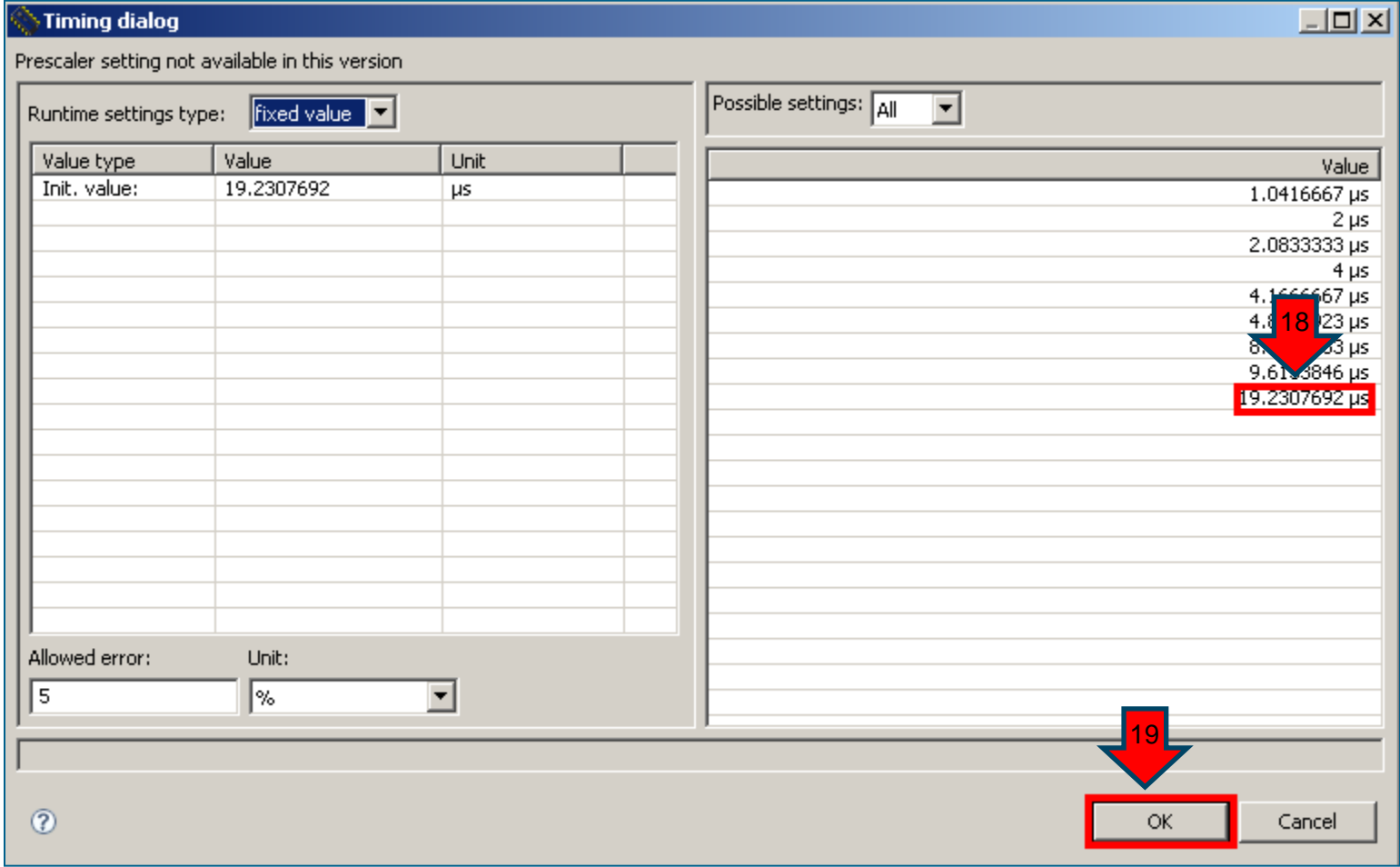
[-] A/D channel list	1	
[-] Channel 0		
[-] Channel mode	Single Ended	
[-] Input		
A/D channel (pin)	ADC1_DM1	ADC1_DM1

[-] Static sample groups	Enabled	
[-] Sample group list	1	
[-] Group 0		
[-] Sample list	1	
[-] Sample 0	Enabled	Sample group settings.
Channel index	0	D

A/D channel (pin)	ADC1_DM1	ADC1_DM1
[+] Static sample groups	Disabled	
A/D resolution	Autoselect	16 bits
! Conversion time		...
ADC clock		Unassigned timing

\_\_\_\_\_

- 





## ► ADC LLD Driver is configured

Properties	Methods	Events
Name	Value	Details
A/D converter	ADC1	ADC1
<b>Converter mode</b>	Sequential	
<input checked="" type="checkbox"/> <b>Interrupt service/event</b>	Enabled	
A/D interrupt priority	medium priority	8
<input checked="" type="checkbox"/> <b>A/D channel list</b>	1	
<input checked="" type="checkbox"/> <b>Channel 0</b>		
<input checked="" type="checkbox"/> <b>Channel mode</b>	Single Ended	
<input checked="" type="checkbox"/> <b>Input</b>		
A/D channel (pin)	ADC1_DM1	ADC1_DM1
<input checked="" type="checkbox"/> <b>Static sample groups</b>	Enabled	
<input checked="" type="checkbox"/> <b>Sample group list</b>	1	
<input checked="" type="checkbox"/> <b>Group 0</b>		
<input checked="" type="checkbox"/> <b>Sample list</b>	1	
<input checked="" type="checkbox"/> <b>Sample 0</b>	Enabled	
Channel index	0	D
A/D resolution	Autoselect	16 bits
Conversion time	4µs	4.167 µs
ADC clock	5.999 MHz (166.667 ns)	Clock conf. 0: 5.999 MHz (166.667 ns)
Single conversion time - Single-ended	10.104 us	Clock conf. 0: 10.104 us
Single conversion time - Differential	11.604 us	Clock conf. 0: 11.604 us
Additional conversion time - Single-ended	4.166 us	Clock conf. 0: 4.166 us
Additional conversion time - Differential	5.666 us	Clock conf. 0: 5.666 us
Result type	unsigned 16 bits, right justified	
<input checked="" type="checkbox"/> <b>Initialization</b>		This property allows to select one of result Description for the current value (unsigned
Enabled in init. code	yes	
<input checked="" type="checkbox"/> <b>Event mask</b>		
OnMeasurementComplete	Enabled	

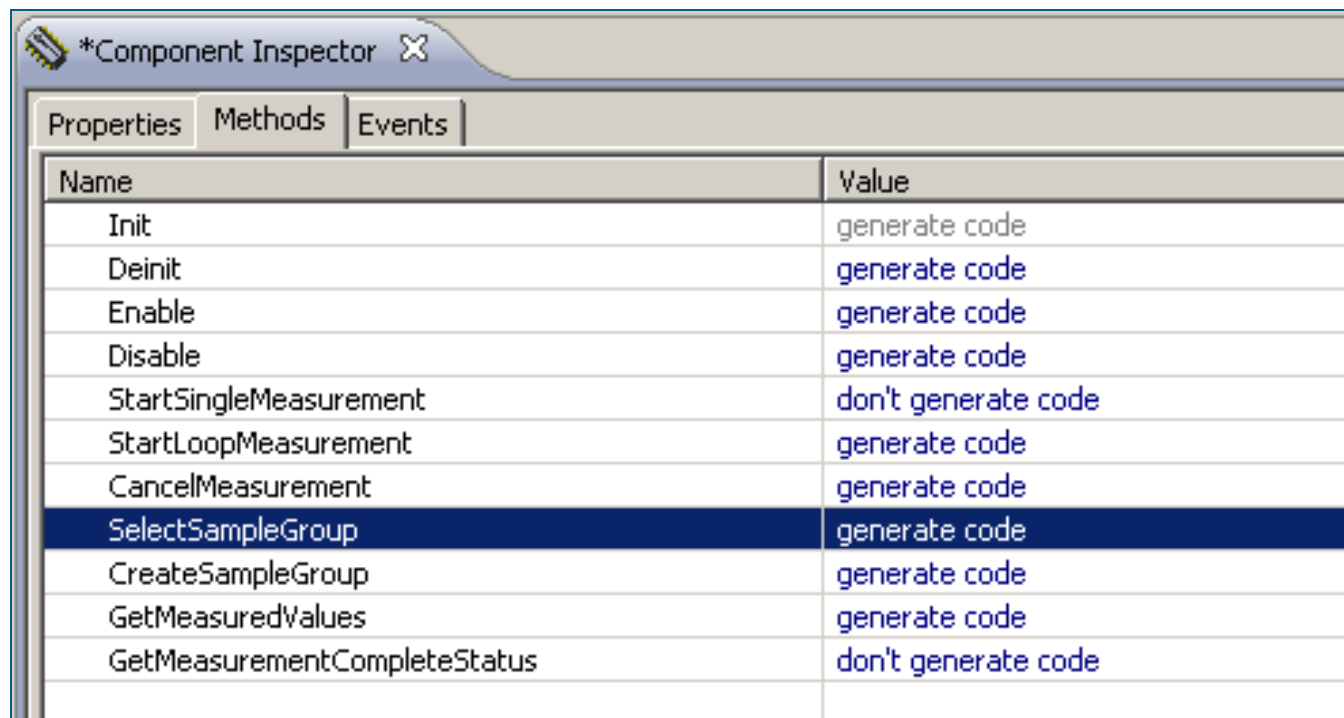
- ▶ Click Methods Tab
- ▶ Click to generate code for methods

\*Component Inspector

Properties	Methods	Events
Name	Value	
Init	generate code	
Deinit	generate code	
Enable	don't generate code	
Disable	don't generate code	
StartSingleMeasurement	generate code	
StartLoopMeasurement	don't generate code	
CancelMeasurement	don't generate code	
SelectSampleGroup	don't generate code	
CreateSampleGroup	generate code	
GetMeasuredValues	generate code	
GetMeasurementCompleteStatus	don't generate code	

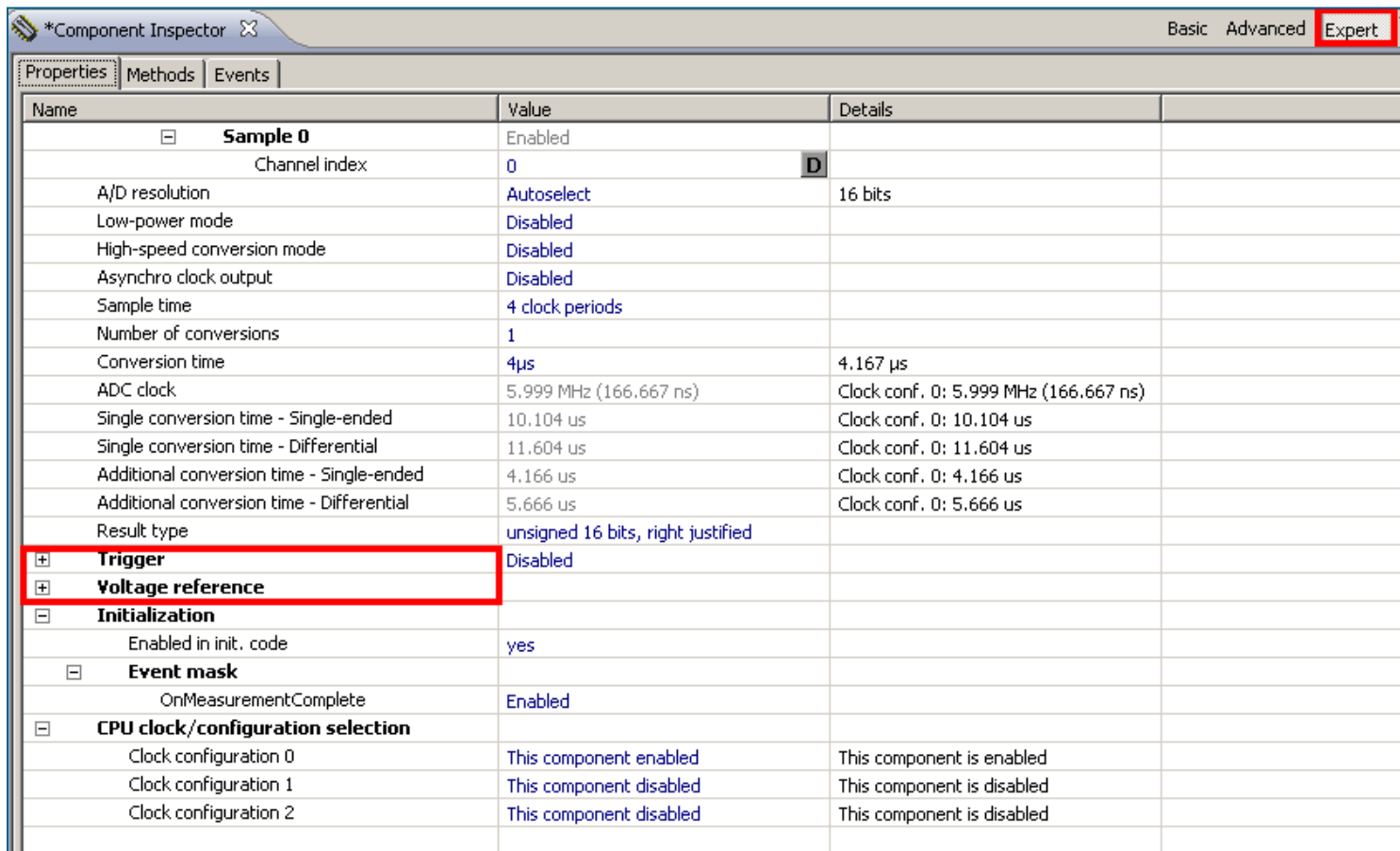
Properties	Methods	Events
Name	Value	
Init	generate code	
Deinit	generate code	
Enable	generate code	
Disable	generate code	
StartSingleMeasurement	generate code	
StartLoopMeasurement	don't generate code	
CancelMeasurement	generate code	
SelectSampleGroup	don't generate code	
CreateSampleGroup	generate code	
GetMeasuredValues	generate code	
GetMeasurementCompleteStatus	don't generate code	

- Set 'generate code' for the next Methods:



Name	Value
Init	generate code
Deinit	generate code
Enable	generate code
Disable	generate code
StartSingleMeasurement	don't generate code
StartLoopMeasurement	generate code
CancelMeasurement	generate code
SelectSampleGroup	generate code
CreateSampleGroup	generate code
GetMeasuredValues	generate code
GetMeasurementCompleteStatus	don't generate code

- You can configure more parameters of the components selecting Expert View

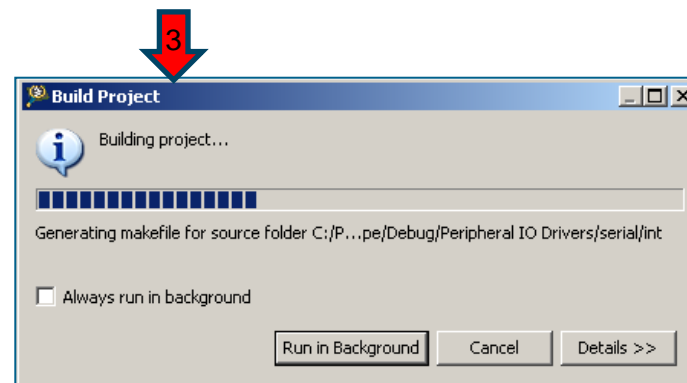
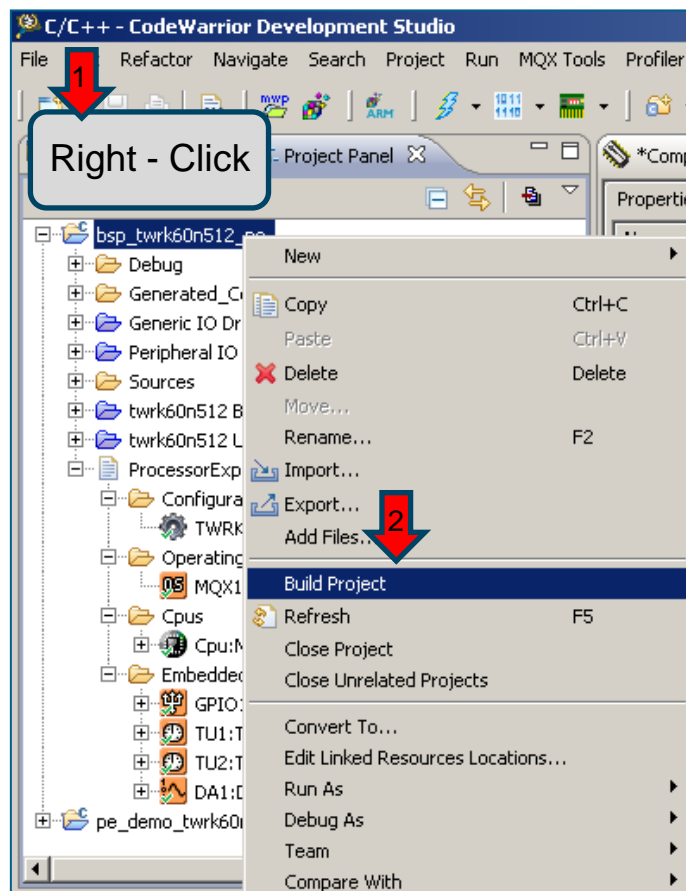


The screenshot shows the \*Component Inspector window with the Expert View tab selected. The 'Trigger' and 'Voltage reference' sections are highlighted with a red box. The 'Initialization' section is also visible, showing 'Enabled in init. code' as 'yes'. The 'Event mask' section shows 'OnMeasurementComplete' as 'Enabled'. The 'CPU clock/configuration selection' section shows three configurations: 'Clock configuration 0' (enabled), 'Clock configuration 1' (disabled), and 'Clock configuration 2' (disabled).

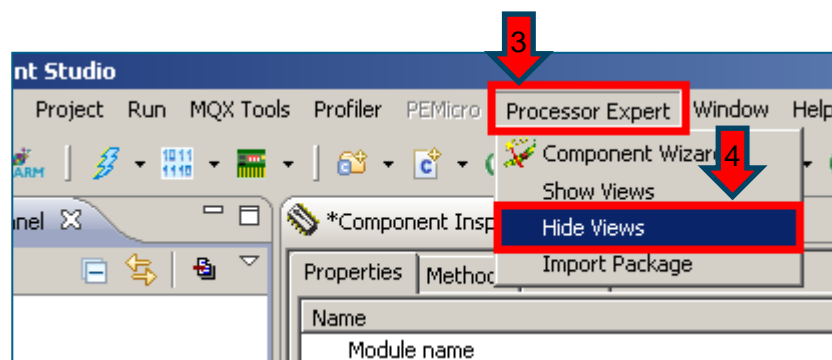
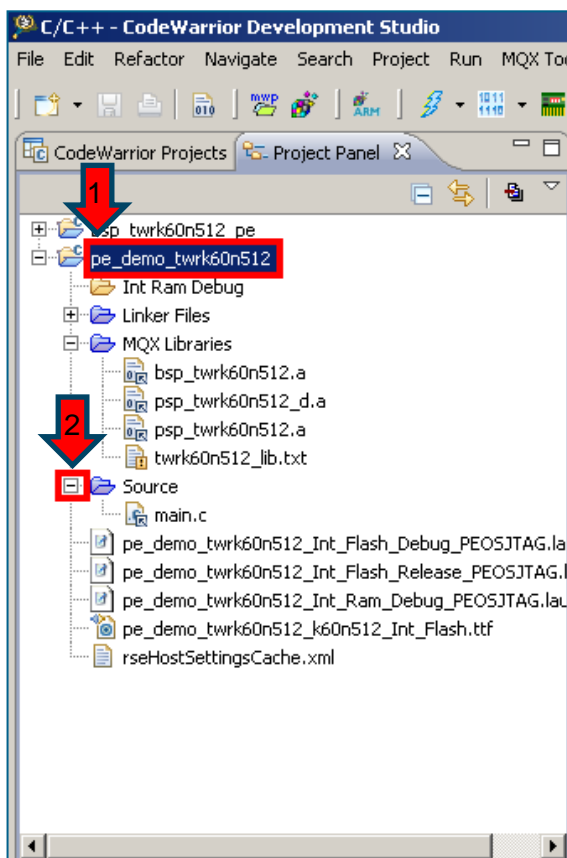
Name	Value	Details
<b>Sample 0</b>	Enabled	
Channel index	0	
A/D resolution	Autoselect	16 bits
Low-power mode	Disabled	
High-speed conversion mode	Disabled	
Asynchro clock output	Disabled	
Sample time	4 clock periods	
Number of conversions	1	
Conversion time	4µs	4.167 µs
ADC clock	5.999 MHz (166.667 ns)	Clock conf. 0: 5.999 MHz (166.667 ns)
Single conversion time - Single-ended	10.104 us	Clock conf. 0: 10.104 us
Single conversion time - Differential	11.604 us	Clock conf. 0: 11.604 us
Additional conversion time - Single-ended	4.166 us	Clock conf. 0: 4.166 us
Additional conversion time - Differential	5.666 us	Clock conf. 0: 5.666 us
Result type	unsigned 16 bits, right justified	
<b>Trigger</b>	Disabled	
<b>Voltage reference</b>		
<b>Initialization</b>		
Enabled in init. code	yes	
<b>Event mask</b>		
OnMeasurementComplete	Enabled	
<b>CPU clock/configuration selection</b>		
Clock configuration 0	This component enabled	This component is enabled
Clock configuration 1	This component disabled	This component is disabled
Clock configuration 2	This component disabled	This component is disabled

# Build BSP project

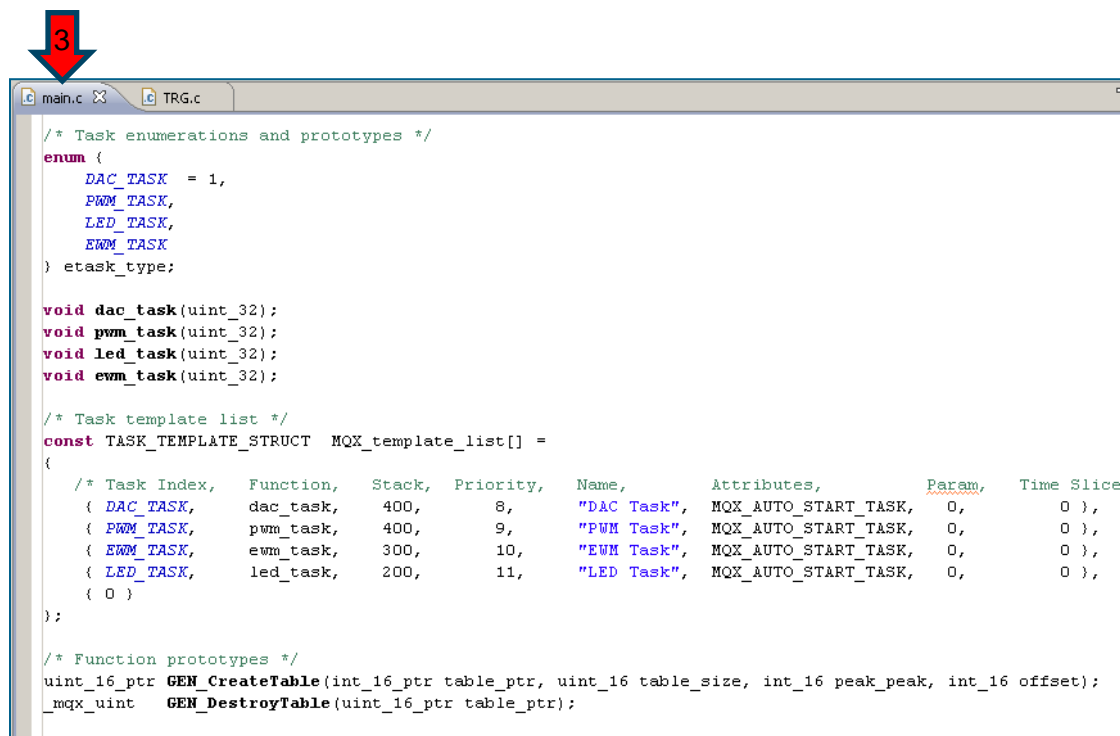
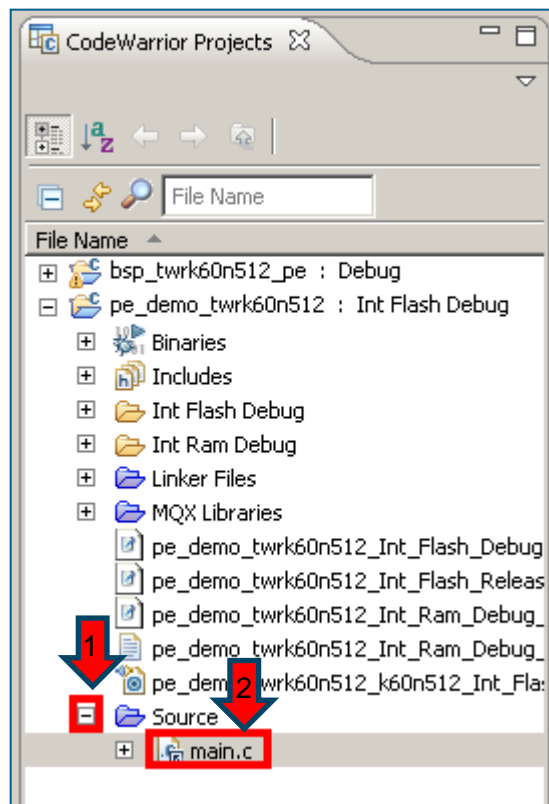
- Right-Click on Project Explorer **bsp\_twrk60n512** and Build Project



- ▶ Expand **pe\_demo\_twrk60n512** project view
- ▶ Hide Processor Expert View



- Double click in **main.c** to view code



## ► Add new task : ADC

```

/* Task enumerations and prototypes */
enum {
    DAC_TASK = 1,
    PWM_TASK,
    LED_TASK,
    EWM_TASK,
    ADC_TASK
} etask_type;

void dac_task(uint_32);
void pwm_task(uint_32);
void led_task(uint_32);
void ewm_task(uint_32);
void adc_task(uint_32);

/* Task template list */
const TASK_TEMPLATE_STRUCT MQX_template_list[] =
{
    /* Task Index,    Function,    Stack,    Priority,    Name,        Attributes,        Param,    Time Slice
    { DAC_TASK,      dac_task,    400,      8,          "DAC Task",   MQX_AUTO_START_TASK, 0,        0 },
    { PWM_TASK,      pwm_task,    400,      9,          "PWM Task",   MQX_AUTO_START_TASK, 0,        0 },
    { EWM_TASK,      ewm_task,    300,      10,         "EWM Task",   MQX_AUTO_START_TASK, 0,        0 },
    { LED_TASK,      led_task,    200,      11,         "LED Task",   MQX_AUTO_START_TASK, 0,        0 },
    { ADC_TASK,      adc_task,    200,      12,         "ADC Task",   MQX_AUTO_START_TASK, 0,        0 },
    { 0 }
};

```



## ► Add Task function and code

```
#define SAMPLE_GROUP_SIZE 1U
volatile AD1_TResultData MeasuredValues[SAMPLE_GROUP_SIZE];
LDD_TUserData *MyADCPtr;
LDD_TError Error;

void adc_task
(
    uint_32 initial_data
)
{
    MyADCPtr = AD1_Init((LDD_TUserData *)NULL);           /* Initialize the device */
    Error = AD1_SelectSampleGroup(MyADCPtr, 0U);           /* Select sample group 0 */
    Error = AD1_StartLoopMeasurement(MyADCPtr);            /* Start continuous measurement */
    Error = AD1_Enable(MyADCPtr);
    while(1)
    {
        /* Suspend task for 100ms */

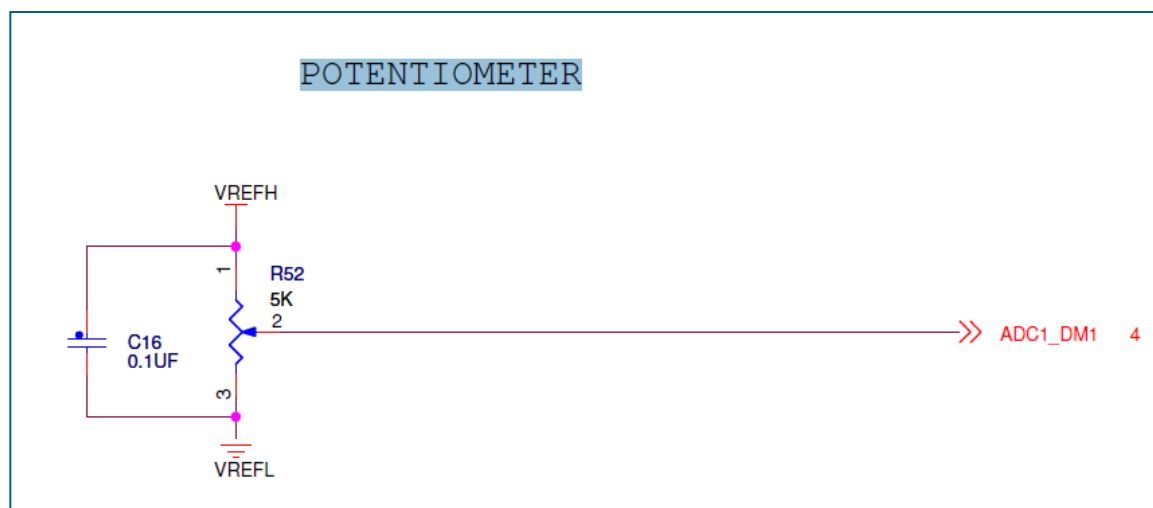
        if(MeasuredValues[0]>2000) GPIO1_ToggleFieldBits(LED_DeviceData, LED4, 1);
        _time_delay(200);
    }
}
```

## ► Add **ADC1** Event function code

8

```
void AD1_OnMeasurementComplete(LDD_TUserData *UserDataPtr)
{
    Error = AD1_GetMeasuredValues(MyADCPtr, (LDD_TData *)&MeasuredValues); /* Read measured values */
}
/* EOF */
```

## ► ADC1 channel is connected to TWR-K60N512 Potentiometer

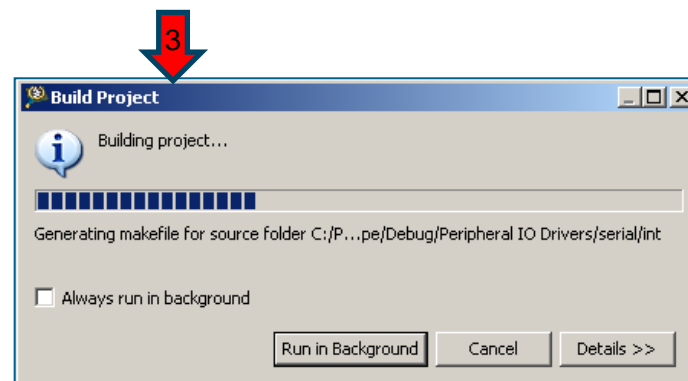
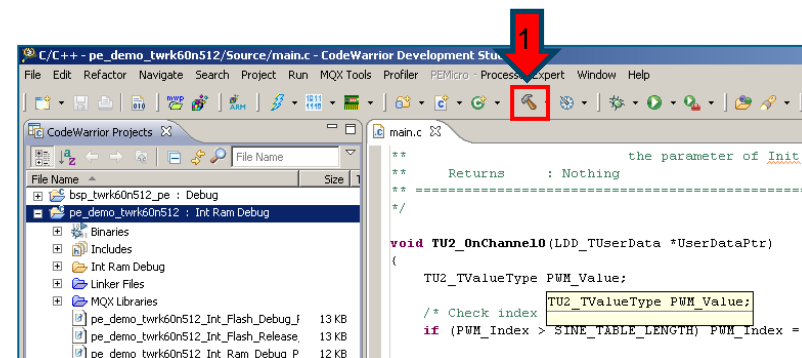
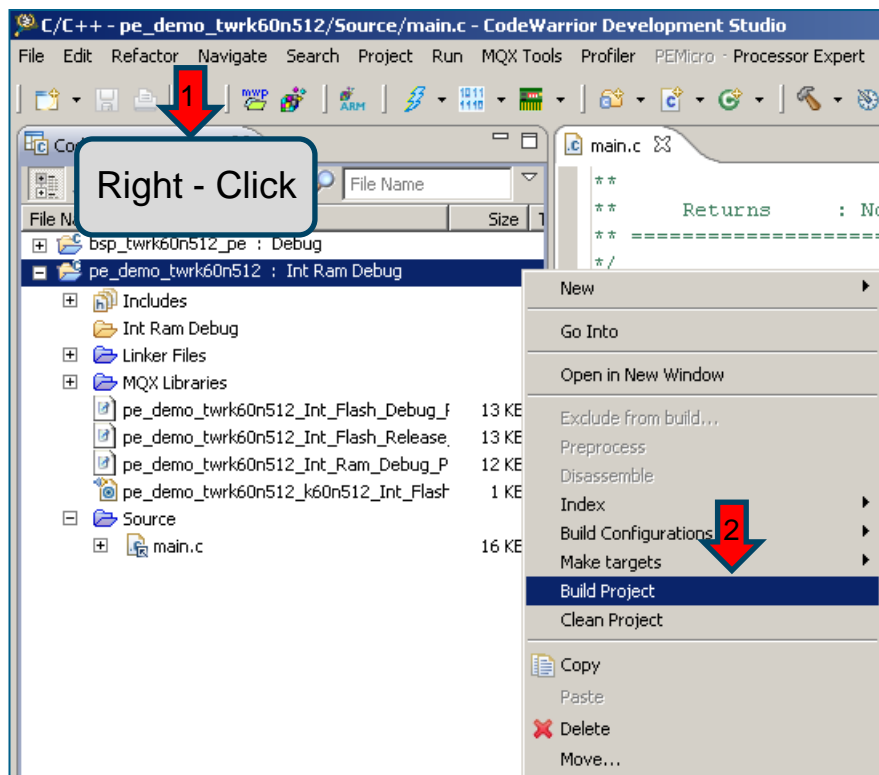


- ▶ When ADC value is greater than 20000, LED4 (Blue) toggles

```
while(1)
{
    /* Suspend task for 100ms */
    if (MeasuredValues[0]>2000) GPIO1_ToggleFieldBits(LED_DeviceData, LED4, 1);
    _time_delay(200);
}
```

- ▶ Moving potentiometer R52 can start/stop LED4 toggle

- ▶ Right-Click on Project Explorer **pe\_demo\_twrk60n512** and Build Project or click on icon 



- ▶ Test the new functionality in the application and the new LDD driver

