



Installing and using CW 10.6 for TPMS applications

Revision 3

May 2015

Table of Contents

1. Installing Code Warrior for MCUs v10.6	3
a. General information.....	3
b. Downloading the IDE	4
c. Installing CW 10.6	4
2. Installing the necessary updates for TPMS applications.....	6
3. Programming the Starter project.....	13
a. Downloading the project	13
b. Importing the project in CW 10.6	13
c. Programming the device.....	15
i. Hardware set up.....	15
ii. Debug Configuration	17
d. Running the program	23
4. Troubleshooting.....	26
a. During the installation of the update.....	26
b. When opening a TPMS project with CW 10.6.....	28
c. When building a TPMS project with CW 10.6.....	28
d. When entering the debugger.....	29
e. When running the program after having reprogrammed the device.....	33

1. Installing Code Warrior for MCUs v10.6

a. General information

Code Warrior v10.6 for MCUs is a new version based on the Eclipse development platform. It is compatible with the following host platforms:

- Microsoft® Windows XP 32-bit and 64-bit (Professional Edition)
- Microsoft Windows 7 32-bit and 64-bit (Home Premium Edition and Professional Edition)
- Microsoft Windows 8 32-bit and 64-bit (Home Premium Edition and Professional Edition)

More information concerning the system requirements, target interfaces, new features... can be found on the website. For that, go to www.freescale.com > *Software and Tools* > *Code Warrior Development Tools*. On this page choose the Eclipse IDE for RS08/S08 8-Bit MCUs (MCUs used in the TPMS applications).

Freescale ► CodeWarrior Embedded Software Development Tools

CodeWarrior Embedded Software Development Tools

CodeWarrior embedded software development studio is a complete integrated development environment (IDE) that provides a highly visual and automated framework to accelerate development of the most complex embedded applications.

Choosing the right CodeWarrior Embedded Software Development Suite

CodeWarrior Embedded Software Development Suites

CodeWarrior Development Suites for Networked Applications (recommended for all StarCore DSPs, PowerQUICC processors and devices within the QorIQ Processing Platforms portfolio)

- Development Studio - Developer Suite
- Development Studio - Specialist Suite
- Development Studio - Architect Suite
- Development Studio - LS Tower Suite

CodeWarrior Development Suites

- Development Studio - Professional Suite
- Development Studio - Standard Suite
- Development Studio - Basic Suite
- Development Studio - Special Suite (free trial)

CodeWarrior Embedded Software Development Tools

CodeWarrior Development Studios

CodeWarrior Development Studio for MCUs

- 68K MCUs (Classic IDE)
- ColdFire MCUs (Eclipse IDE)
- ColdFire V1 MCUs (Classic IDE)
- HCS12(X) 16-Bit MCUs (Classic IDE)
- Kinetis MCUs (Eclipse IDE)
- mobileGT MCUs (Classic IDE)
- MPC5xx MCUs (Classic IDE)
- Qorivva 56xx MCUs (Eclipse IDE)
- Qorivva 5xxx MCUs (Classic IDE)
- RS08/S08 8-Bit MCUs (Eclipse IDE)**
- RS08/S08 8-Bit MCUs (Classic IDE)
- S12Z 16-Bit MCUs (Eclipse IDE)



Download now: CodeWarrior Embedded Software Development Studio for Microcontrollers v10.6

Freescale's CodeWarrior Development Studio for Microcontrollers v10.6 is a full new product release with build tools, debugger, and Processor Expert improvements; 64-bit support for S12Z devices; New Processor Expert Component Inspector available; and P&E Cyclone Universal [FX] support.

Download it now ►

Download



Click to download a free trial of CodeWarrior embedded software development suites, evaluation versions, updates, patches, and board support packages.

Download CodeWarrior now ►

Design Resources

Getting Started

- CodeWarrior Development Tool Suites Factsheet (pdf)
- CodeWarrior Development Suite Quick Start Guide (pdf)
- Getting Started with Freescale Development Tools (pdf)
- CodeWarrior Development Suites Overview Factsheet (pdf)

Freescale Community: CodeWarrior Embedded Software Development Tools

Licensing and Registration

Design Support

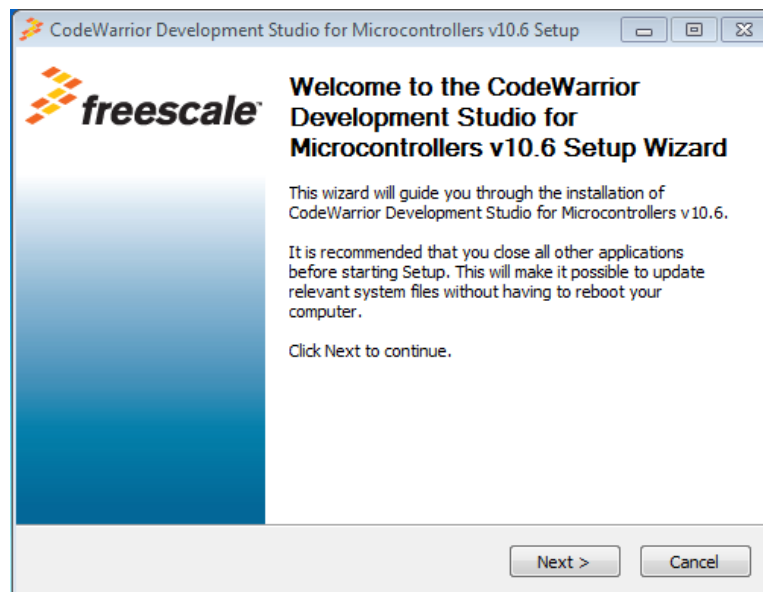
b. Downloading the IDE

The evaluation version is free and available on this page.

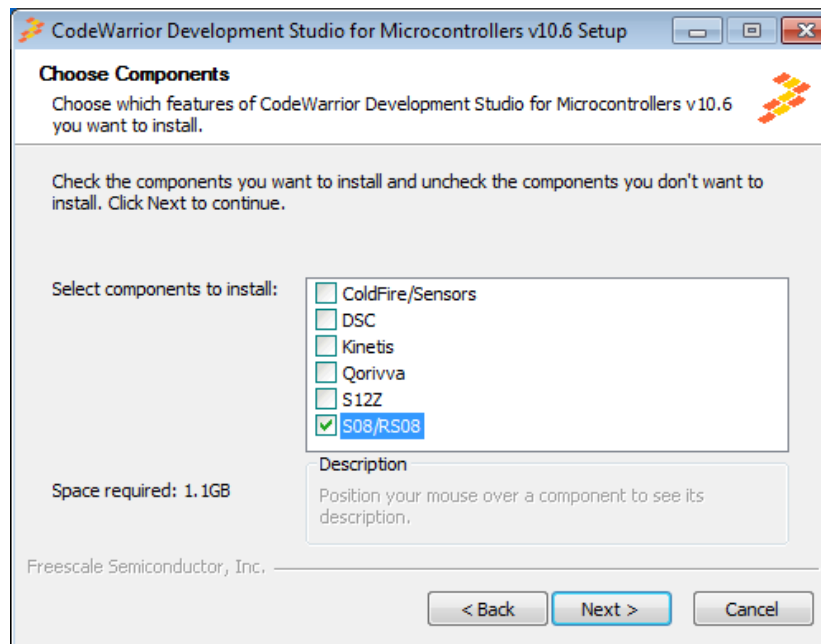
The screenshot shows the Freescale website's product page for CodeWarrior IDE. The navigation bar includes links for Products, Applications, Software & Tools, Training & Communities, Sample & Buy, and About. The main content area features the title 'CodeWarrior for MCUs (Eclipse IDE) - ColdFire, 56800/E DSC, Kinetis, Freescale 56xx, RS08/S08, S12Z v10.6' and a star rating. Below the title are tabs for Overview, Documentation, Downloads, Buy / Specifications, Hardware & Tools, and Training & Support. The 'Downloads' tab is active, displaying a description of the IDE and a 'Download Eval' button circled in red. To the right, a preview of the IDE's 'CW MCU v10.6 C/C++ Perspective' is shown.

c. Installing CW 10.6

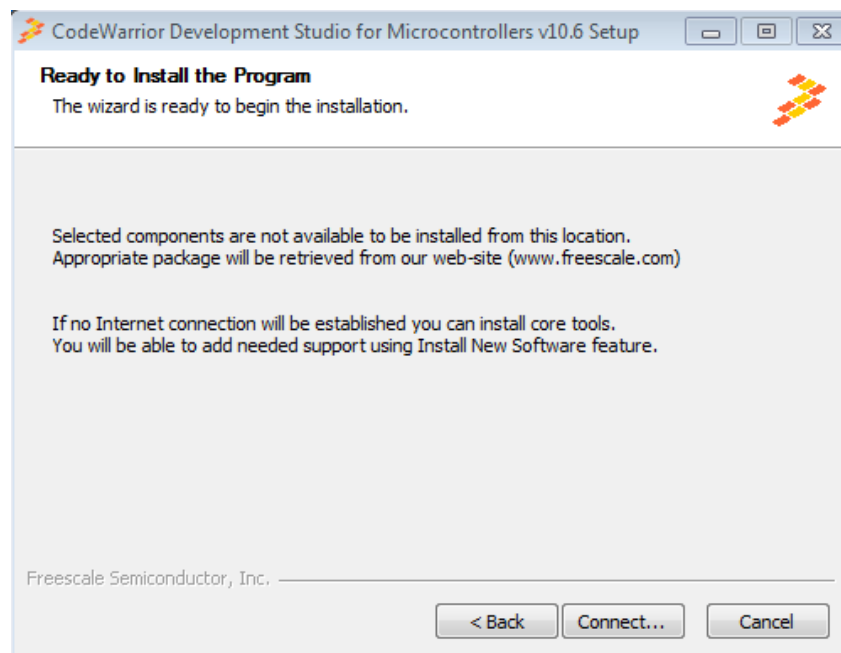
Once downloaded the application can be executed.



After having accepted the terms of the license agreement the following window appears. For TPMS applications, the only necessary components to be installed are **S08/RS08** components. However more components can be installed for further applications.



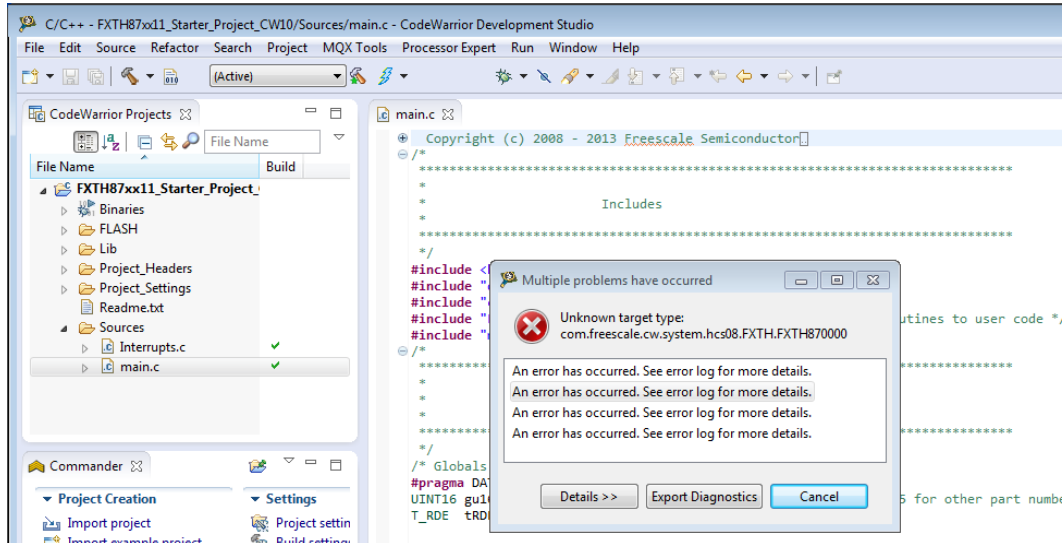
Then click *Connect...*



Once the installation is complete CW 10.6 for MCUs can be used. However, in order to be able to use the IDE for TPMS applications, it is necessary to install some updates (c.f. to next section).

2. Installing the necessary updates for TPMS applications

If a TPMS project is opened with CW 10.6 and if the updates for TPMS applications have not been installed, the following error window will appear.

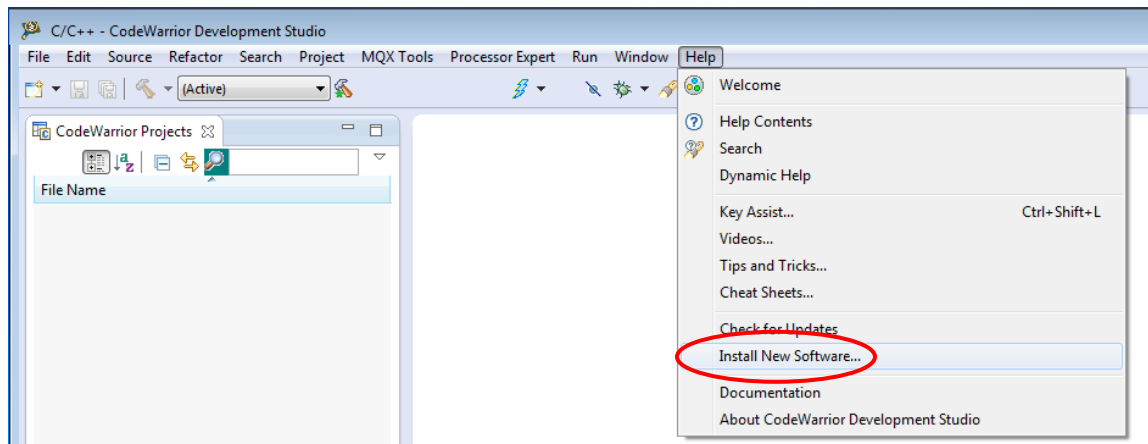


To prevent this error from occurring, a general update needs to be installed. This update can be downloaded from the website. Go to the following link (need to sign in or register):
<https://freescale.flexnetoperations.com/control/frse/download?agree=Accept&element=5901381>

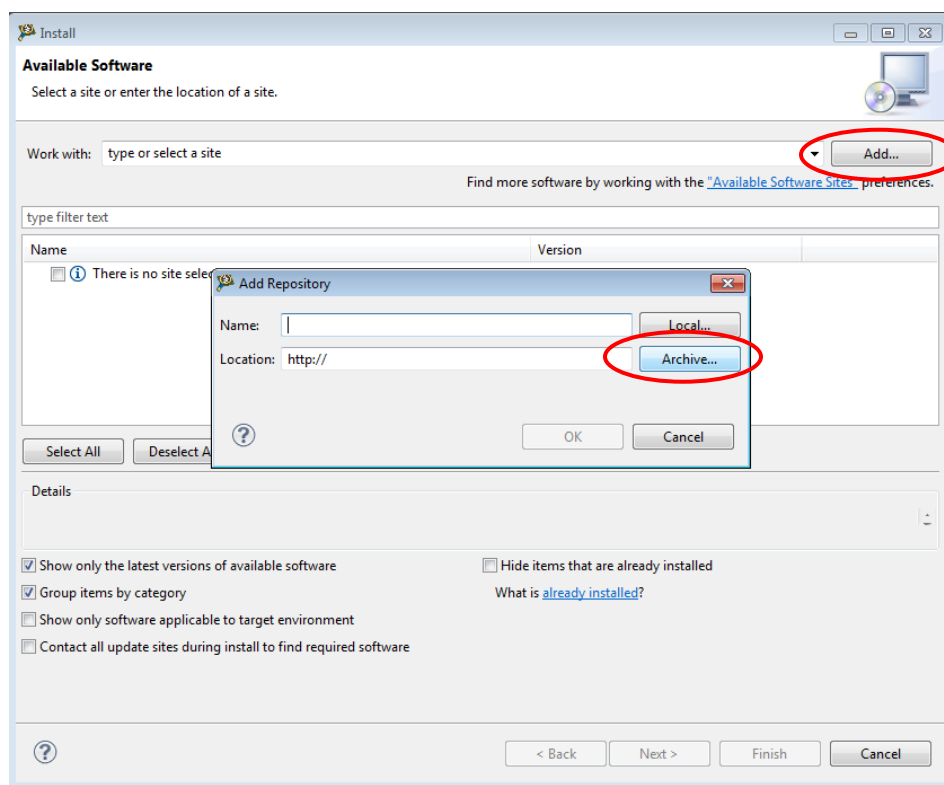
Download the **CW MCU v10.6 General Update 3.0.0** zip file. Do **not** unzip it.

Show All Files		10 Files	
	File Description	File Size	File Name
+	Service Pack	4.9 MB	CodeWarrior for MCU v10.6 S12ZVM Service Pack 1.0.0.zip
+	Installer	432.6 MB	CodeWarrior for Microcontrollers (Windows Hosted, Eclipse based) v10.6 Online.exe
+	Installer	1.4 GB	CodeWarrior for Microcontrollers (Windows Hosted, Eclipse based) v10.6 Offline.exe
+	General Update	923.3 MB	CW MCU v10.6 General Update 3.0.0.zip
+	General Update	790.9 MB	CW MCU v10.6 General Update 2.0.0.zip
+	General Update	693.6 MB	CW MCU v10.6 General Update 1.0.0.zip
+	Service Pack	394.9 KB	CW MCU v10.6 MC1323x Service Pack v1.0.0.zip
+	Service Pack	7.5 MB	CW MCU v10.6 S12ZVC Service Pack v1.0.1.zip
+	Service Pack	5.1 MB	CW MCU v10.6 S12ZVHL Service Pack v1.0.2.zip
+	Service Pack	4.4 MB	CW MCU v10.6 Wireless Charging MWCT1xxx Service Pack v1.0.1.zip

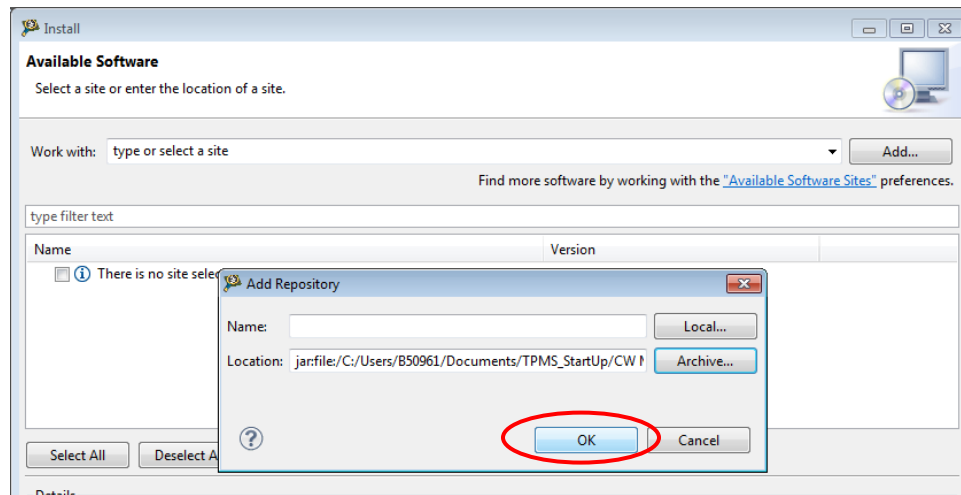
To install this file start CW v10.6, open the *Help* window and click on *Install New Software*.



Then click on *Add* and then on *Archive*.

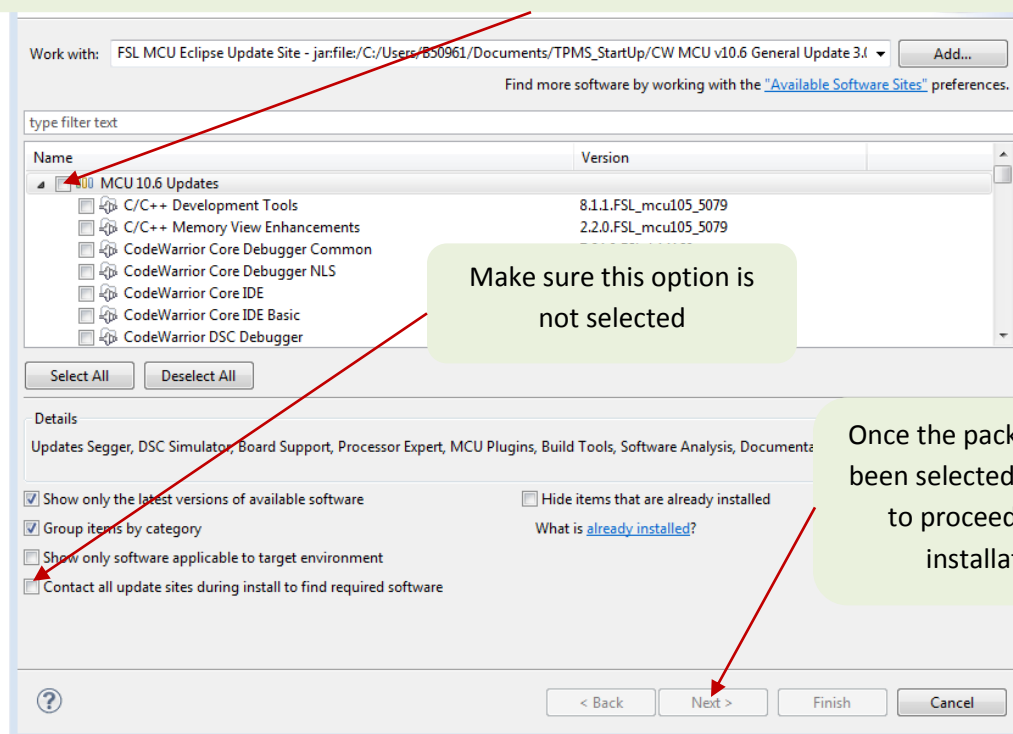


Select your zip file and click **OK**.



At this point the user can choose which packages of the general update have to be installed (either specific packages or the whole update if all the components related to the update have been installed). **If updates that correspond to components that have not been installed are selected then errors will occur.** In what follows the list of the necessary packages for TPMS applications is given.

Tick the box to install the complete update – only if **all the components** have been installed. **Otherwise just tick the necessary packages** (the ones related to the installed components). See below for the list of these packages for TPMS.



List of the necessary packages to be installed for TPMS applications:

If all the components related to the general update have not been installed then the whole update cannot be installed. The specific packages need to be selected. To install the minimum number of packages for TPMS applications (only the packages related to TPMS) follow the procedure below. Otherwise install everything (if all the components have been installed) and once the installation is complete restart CW 10.6. After that TPMS projects can be used with CW 10.6.

Procedure to install the packages specific to TPMS applications:

The necessary Service Pack for TPMS is the package **MCU v10.6 S08 Service Pack for HCS08 FXTH870000** included in the general update 3.0. However, this package alone cannot be installed. In order to install this package other additional packages are necessary as well (packages that contain information to install the S08 Service Pack). The list of all these packages is given below.

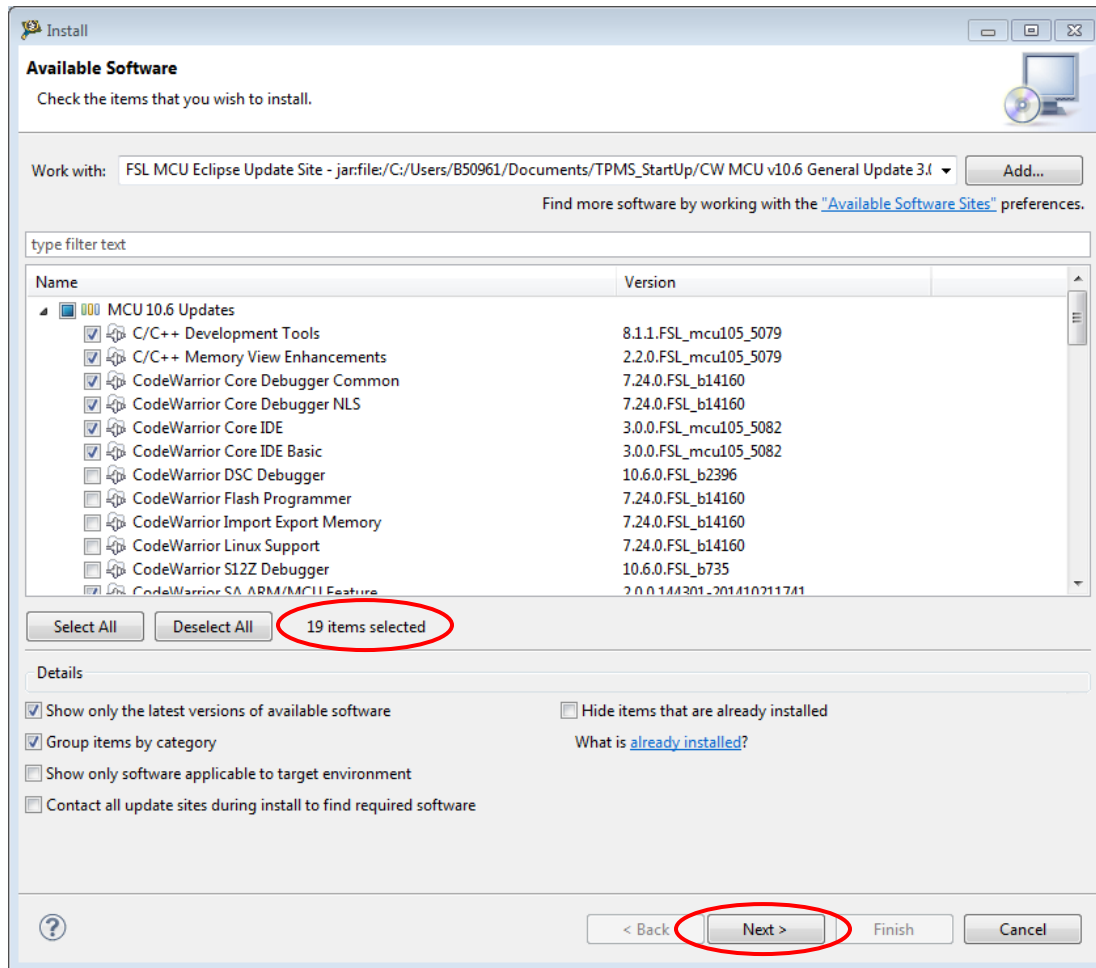
Name	Version	Id
MCU v10.6 S08 Service Pack for HCS08 FXTH870000	1.0.5	com.freescale.mcu10_6.HCS08_FXTH8700...
S08/RS08 Wizard, Debugger	1.0.5	com.freescale.hcs.architecture.feature.gro...
MCU v10.6 Core update	3.0.0	com.freescale.mcu10_6.update.feature.gr...
C/C++ Development Tools	8.1.1.FSL_mcu105_5...	org.eclipse.cdt.feature.group
C/C++ Memory View Enhancements	2.2.0.FSL_mcu105_5...	org.eclipse.cdt.debug.ui.memory.feature....
CodeWarrior Core Debugger Common	7.24.0.FSL_b14160	com.freescale.coretech.debugger.commo...
CodeWarrior Core Debugger NLS	7.24.0.FSL_b14160	com.freescale.coretech.debugger.nls.feat...
CodeWarrior Core IDE	3.0.0.FSL_mcu105_5...	com.freescale.core.ide.feature.group
CodeWarrior Core IDE Basic	3.0.0.FSL_mcu105_5...	com.freescale.core.ide.basic.feature.group
CodeWarrior SA ARM/MCU Feature	2.0.0.144301-20141...	com.freescale.sa.arm.feature.group
CodeWarrior SA ARM/MCU NLS Feature	2.0.0.144301-20141...	com.freescale.sa.arm.nls.feature.group
CodeWarrior SA Core Feature	2.0.0.144301-20141...	com.freescale.sa.feature.group
CodeWarrior SA Core NLS Feature	2.0.0.144301-20141...	com.freescale.sa.nls.feature.group
Common Documentation	5.0.0	com.freescale.doc.feature.group
Component Development Environment	1.6.3.CW_b1441-0004	com.freescale.processorexpert.cde.featur...
DDR Configuration	1.0.0.b1429-0610	com.freescale.processorexpert.feature.ddr...
MCU Wizard and Tools Plugins	10.6.0.FSL_b06012	com.freescale.mcu.feature.group
PEMicro LCD Feature	1.4.18	com.pemicro.lcd.feature.feature.group
Processor Expert for MCU	10.6.3.RT6_b1446-0...	com.freescale.processorexpert.feature.mc...

18
necessary
additional
packages to
install

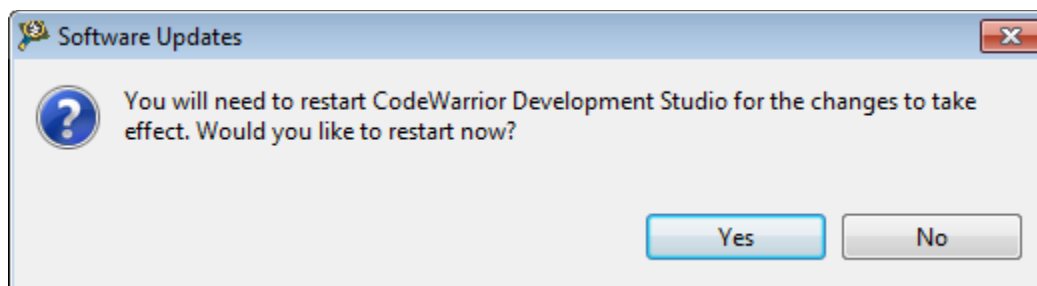
Package we want to install

So for TPMS there are **at least 19** packages to install: the **MCU v10.6 S08 Service Pack for HCS08 FXTH870000** plus the additional 18 packages listed above. These 19 packages must be selected (ticked) in the *Install* window.

Select the right ones, then click *Next*, accept the terms of the license agreement and start the installation of the update.



When the following window appears, click Yes.

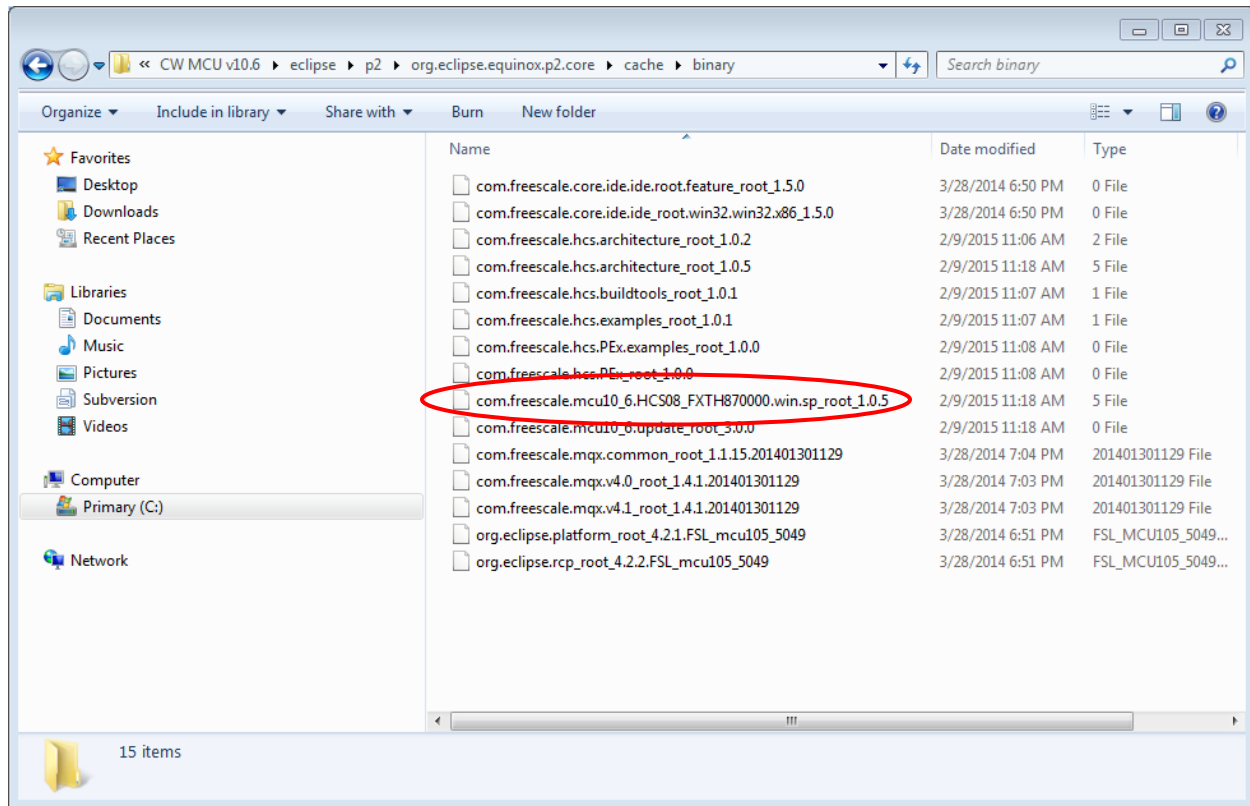


After that the installation is complete. TPMS projects can now be used with CW v10.6.

Note 1:

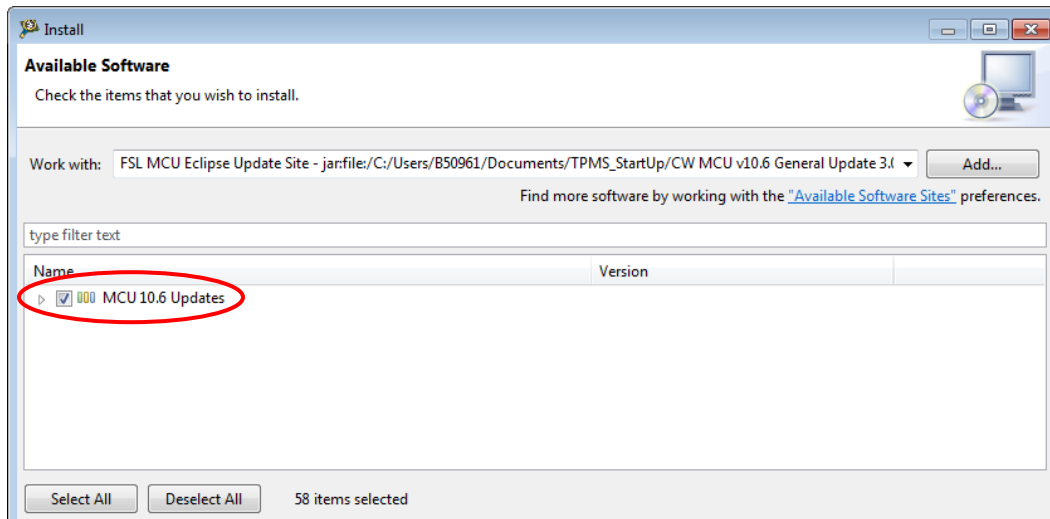
Once the installation of the update is complete the newly installed TPMS service pack can be seen in the following folder:

(CW Installation Path) > CW MCU v10.6 > eclipse > p2 > org.eclipse.equinox.p2.core > cache > binary

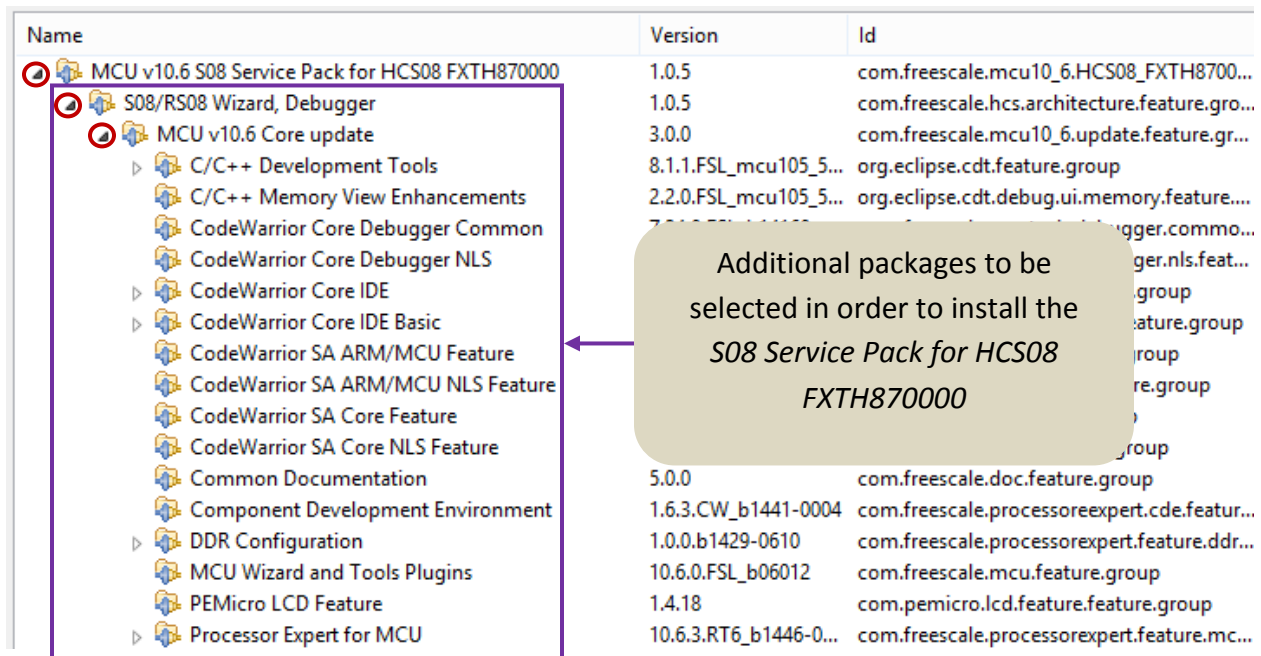
**Note 2 – General information about update installation:**

A general update contains several packages. In order to install a specific service pack (a specific package among all the other packages and not the whole update) it is often necessary to install additional packages because there are dependencies between them. There is a way to know the minimum list of the necessary complementary packages to be installed for a given package.

To see this list, first select all the packages of the update. Then click next.



On the next window the list of the packages appears. When clicking on the little arrow next to the name of the package the list of the dependencies is displayed. In this example it means that in order to install the S08 Service Pack it is necessary to install also all the other packages listed below.



Then go back to the previous window and select only the necessary packages. Then proceed to the installation.

3. Programming the Starter project

a. Downloading the project

A TPMS starter project can be found in the web site in: *Products > Sensors > Tire Pressure Monitoring Sensors > FXTH87*. This is the TPMS main page. To download the project, go to *Software & Tools*.

Freescall > Sensors > Pressure Sensors > Tire Pressure Monitoring Sensors

FXTH87: Freescale FXTH87 Tire Pressure Monitor Sensor Family ☆

Overview Documentation **Software & Tools** Buy / Parametrics Training & Support

[Data Sheet](#) [Application Notes](#) [Buy](#) [Sample](#) [Export to Excel](#) [Collapse All Sections](#)

Software Development Tools (1) Close All Sort by Modified Date

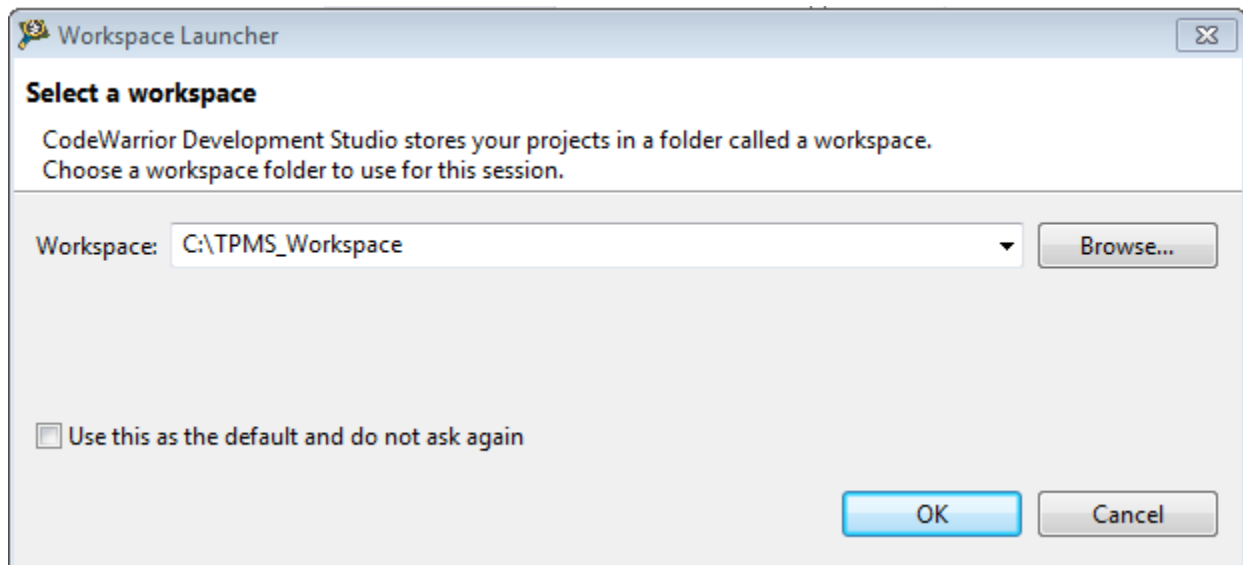
▼ Lab and Test Software (1)

ID and Description	Vendor	Availability	Favorite
FXTH87XX11_STARTERPJ_CW10: FXTH87xx11 Starter Project CW10. Size (K): 125 Format: zip Rev #: 0 Modified: 5/12/2014	FREESCALE	Download	☆

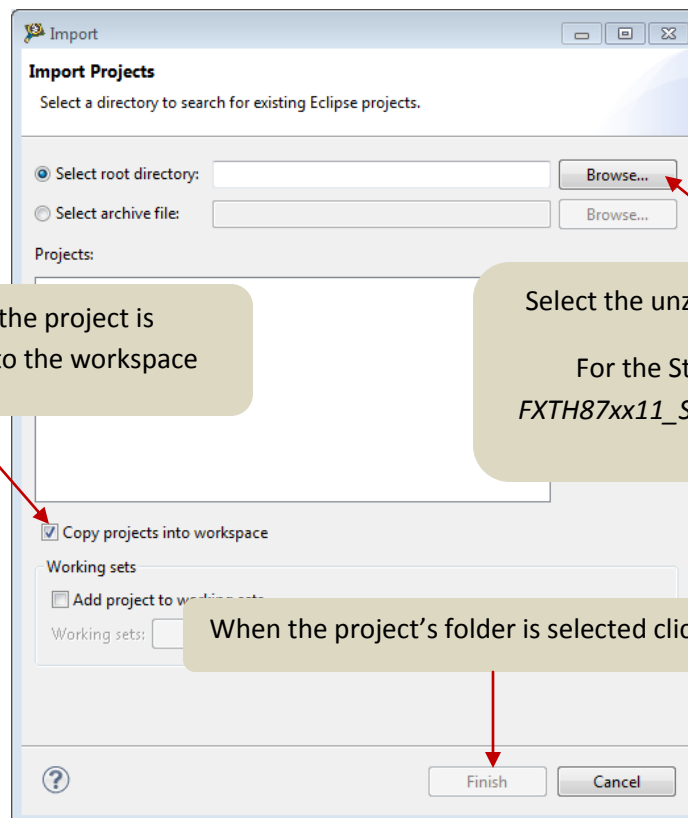
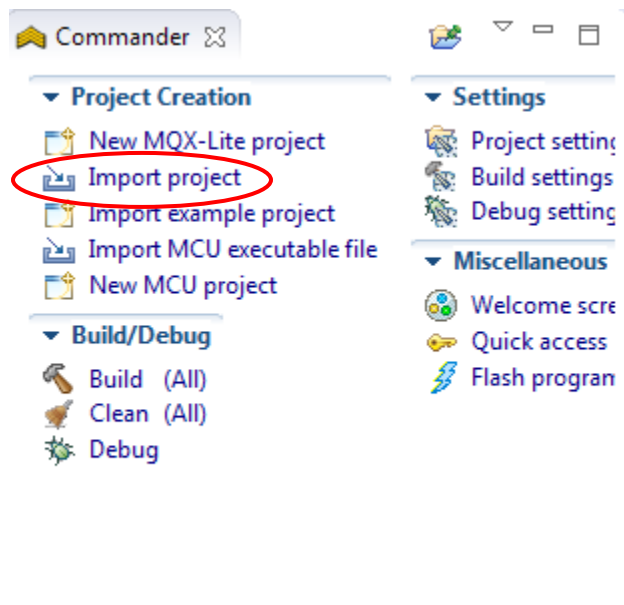
Download the project and unzip it.

b. Importing the project in CW 10.6

Start Code Warrior v10.6. Select the workspace i.e. the folder in which your project will be stored.



When the IDE is started click on *Import Project*.



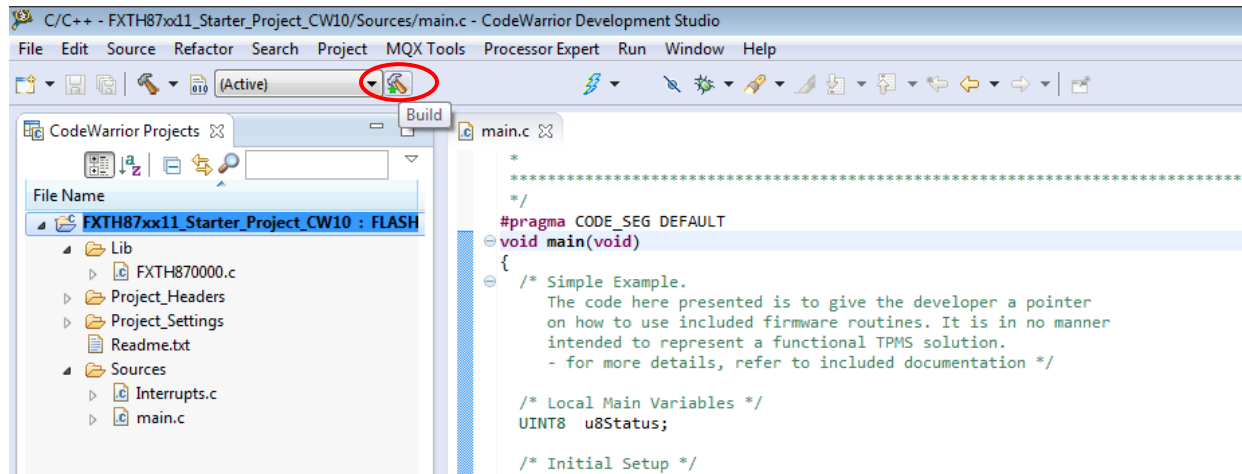
Tick the box so that the project is automatically copied into the workspace

Select the unzipped folder of the project.

For the Starter project select the *FXTH87xx11_Starter_Project_CW10* folder

When the project's folder is selected click *Finish*

Once the project is opened, build it. It might be necessary to clean it before building it (right click on the project and select *Clean Project*).

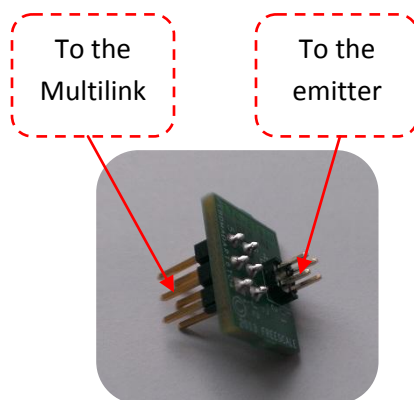


c. Programming the device

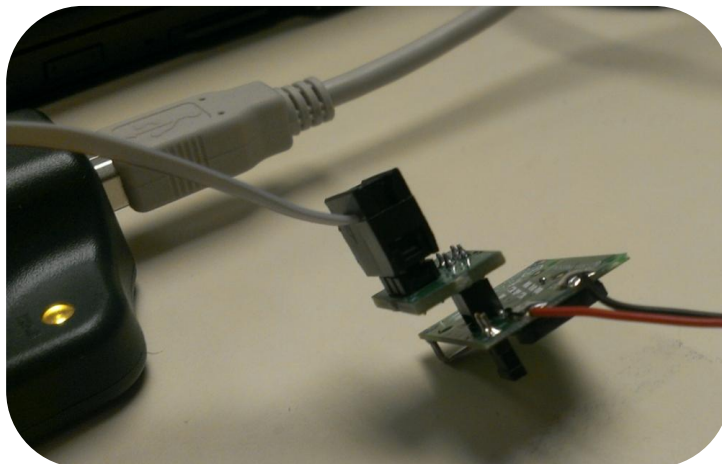
i. Hardware set up

To program the TPMS emitter a P&E's USB Multilink Universal is required. For more information about this product visit [P&E Multilink](#) web page.

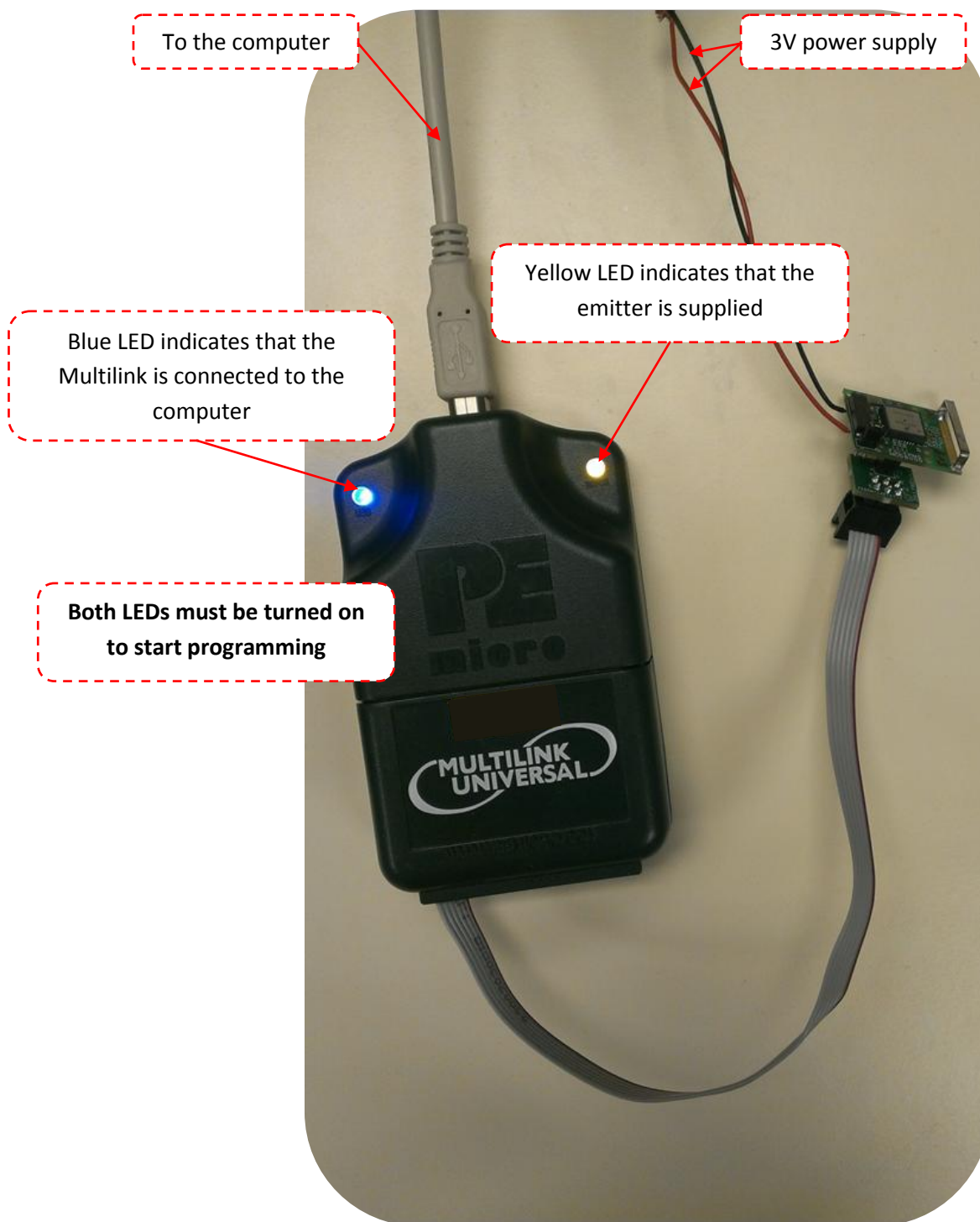
The Multilink connects to the TPMS emitter board via the connector board as shown below:



Connector board

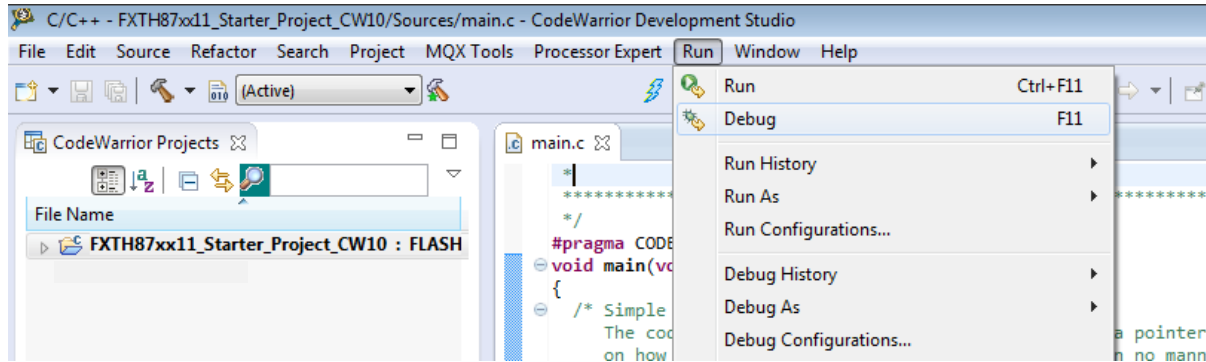


Connection between the Multilink and the TPMS emitter



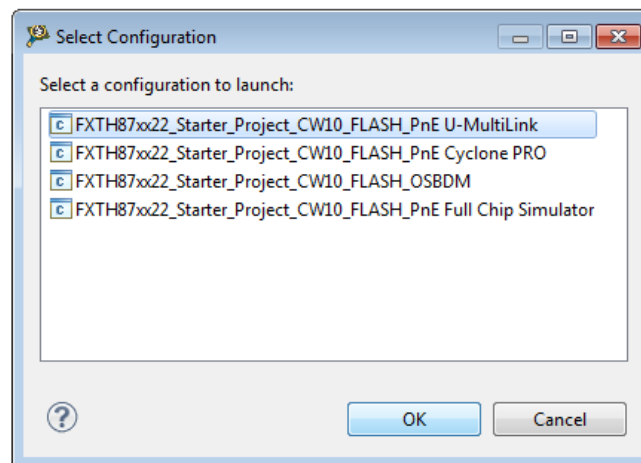
ii. Debug Configuration

After having built the project go to *Run* and select *Debug*.



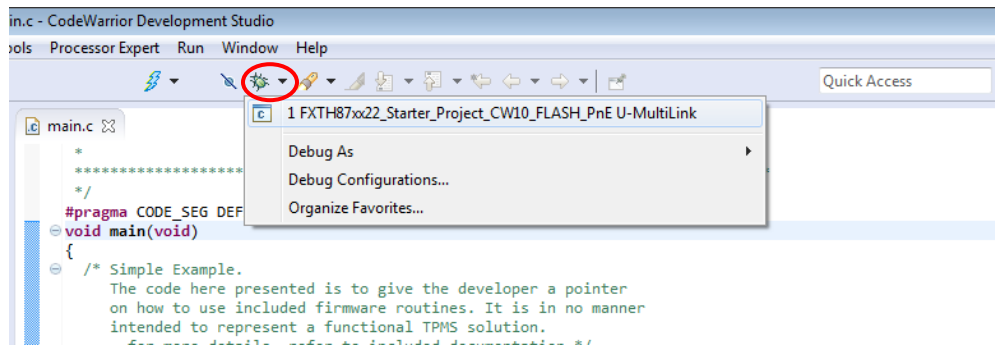
The following window should appear. Choose the right configuration and click *OK*. If the P&E micro Multilink Universal is used, select the *FLASH_PnE U-Multilink* configuration.

If no configurations are available, see below *Adding or modifying a debug configuration*. If configurations are available but problems occur when entering the debugger, see the Troubleshooting part of this document (some settings may need to be (re)selected) or do the configurations again (see below).



Note:

Once a configuration has been selected and used, it is directly available by clicking on the arrow next to the debug button.

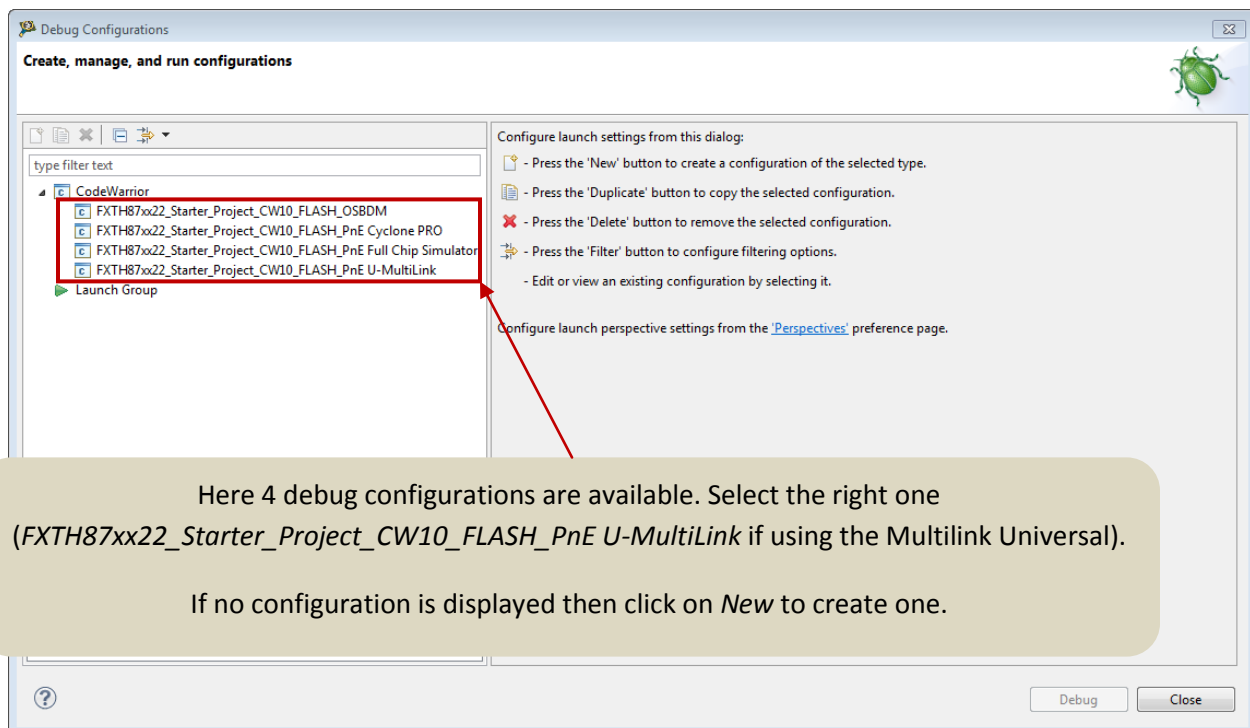


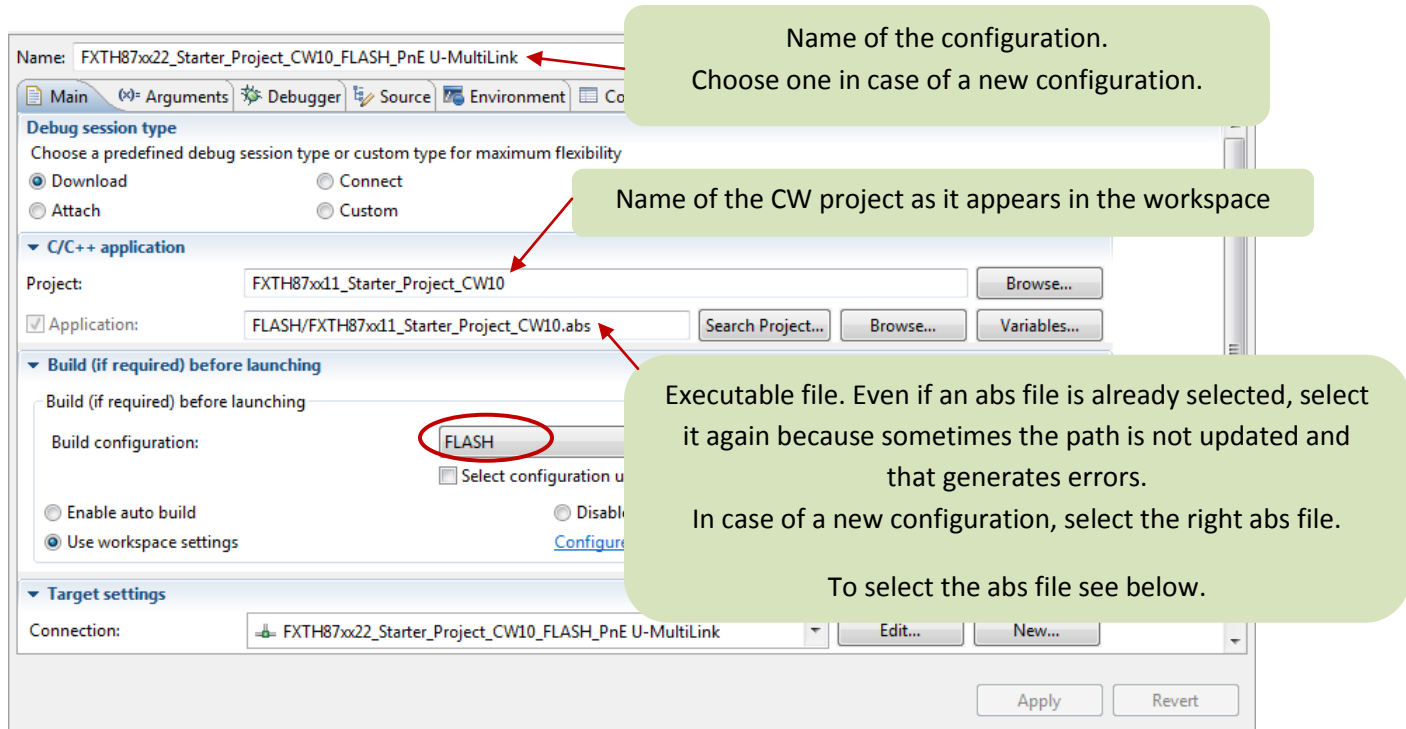
Adding or modifying a debug configuration

Below is explained how to set or modify a debug configuration.

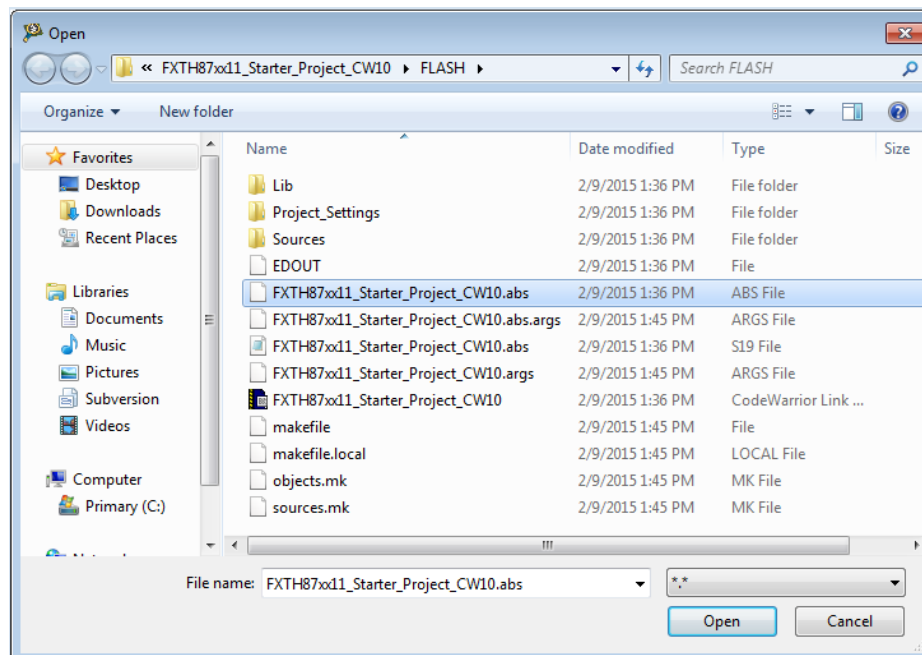
In the *Debug Configurations* window, the following parameters have to be chosen: the project executable file (.abs), the connection interface and the target type.

First open the *Debug Configurations* window by clicking on *Run > Debug Configurations*. This window can also be accessed through the project properties (right click on the project then click on *Properties > Run/Debug Settings*).

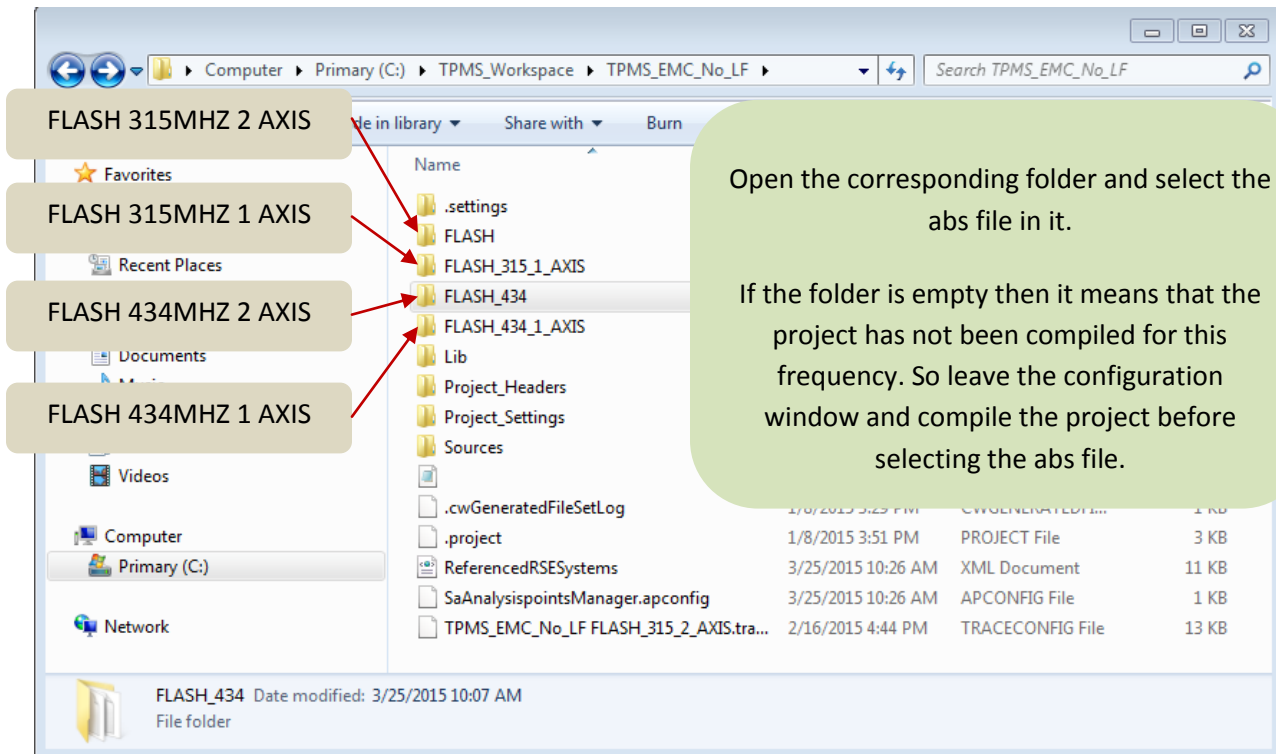




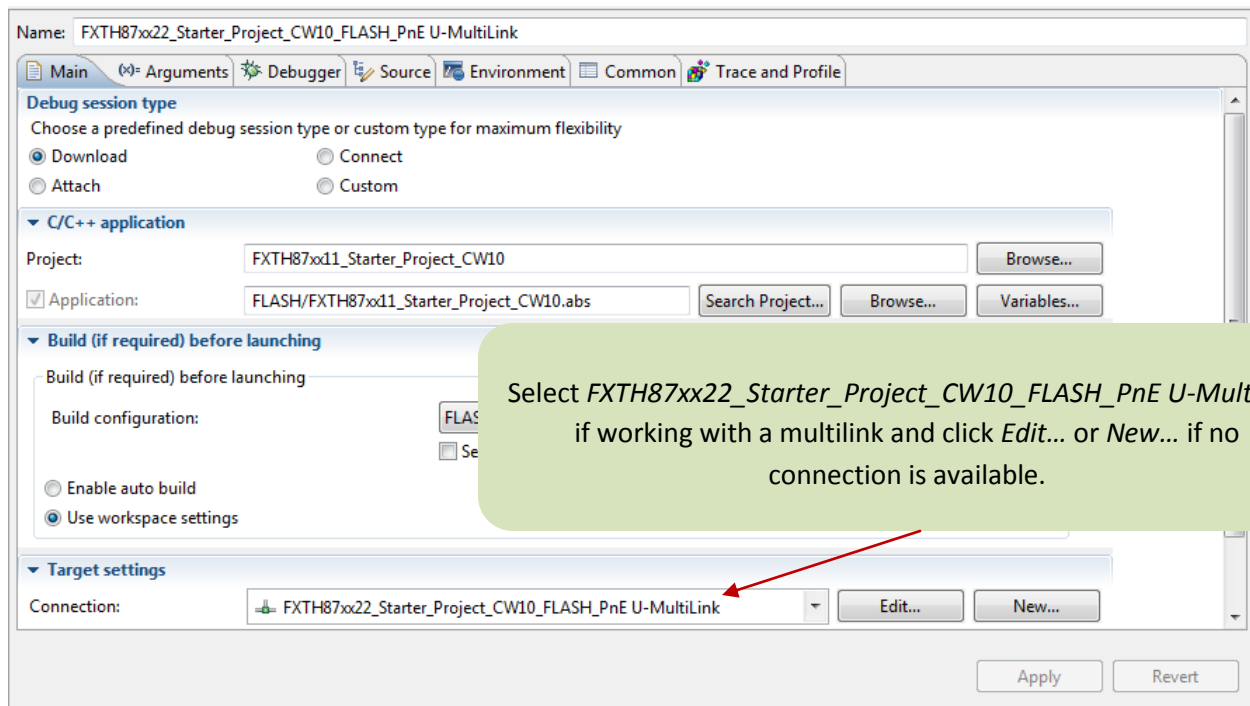
To select the right abs file click on *Browse...* Then go to the FLASH folder (it is in the project's folder) and choose the abs file (be careful: do not select the s19 file that has the same name).



Note: in some projects several abs files are available, depending on frequency chosen during the compilation (315MHz 2 axis, 434MHz 2 axis...). In this case, select the abs file corresponding to the chosen frequency.

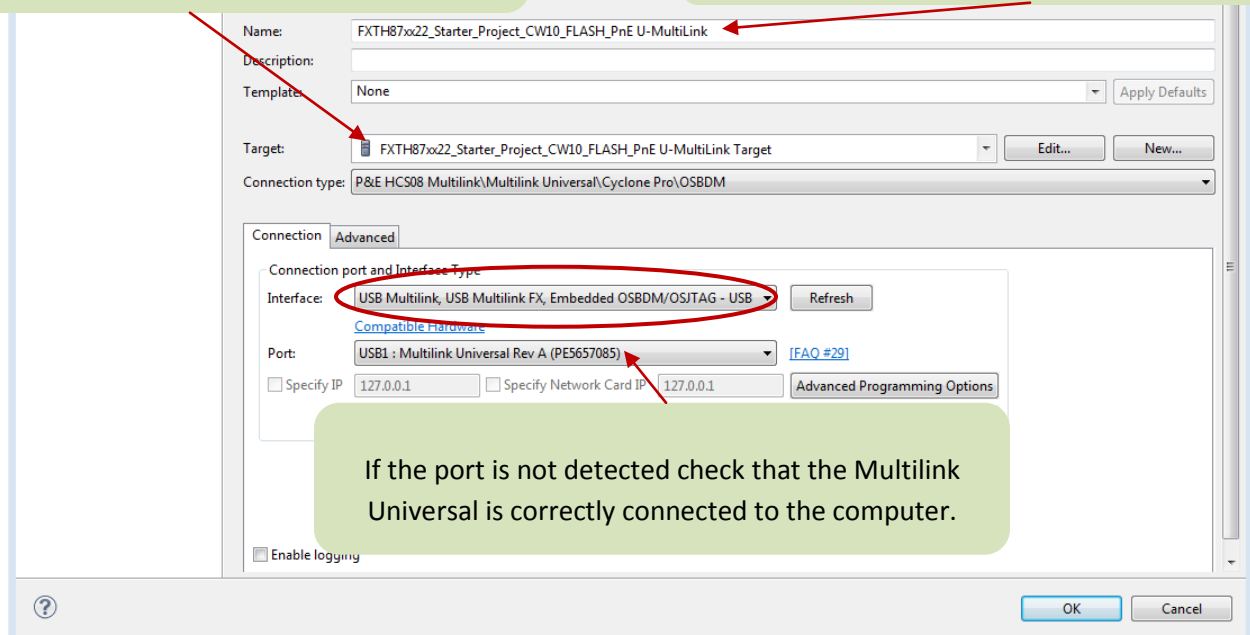


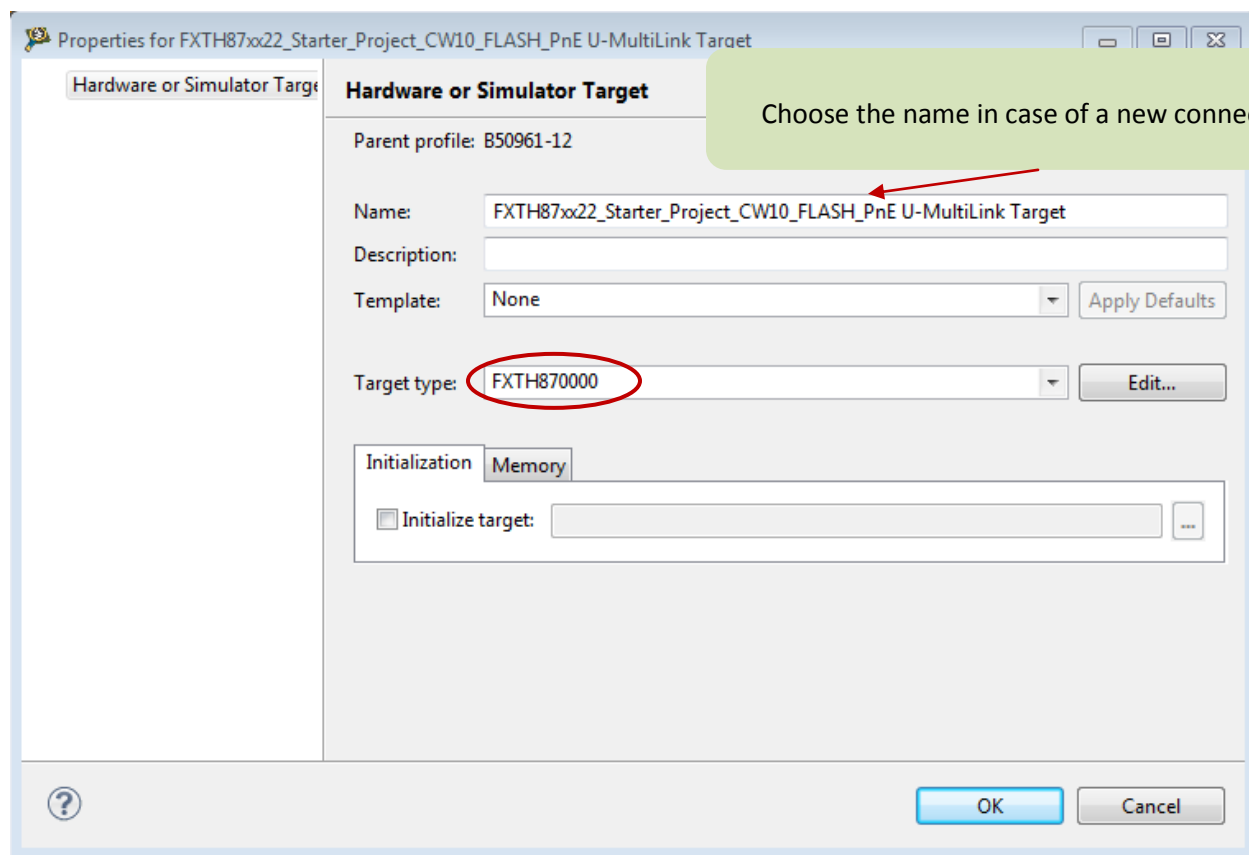
Once the executable file has been selected, choose the connection.



Edit the target to verify that the right one has been selected or click on *New* in case of a new connection

Choose the name in case of a new connection





With that the debug environment configuration is complete.
The debug configuration can now be selected to program the device.

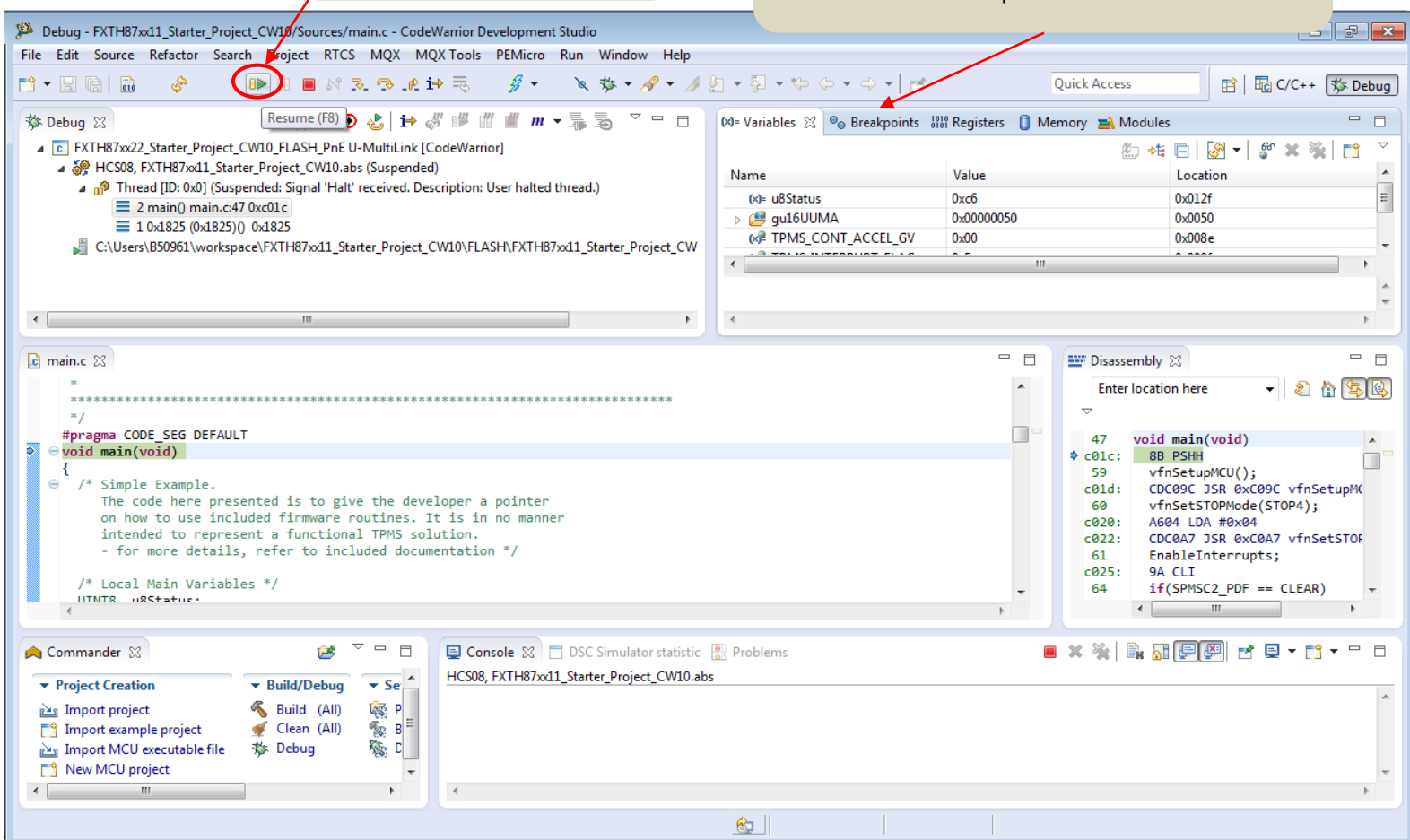
d. Running the program

When the debugger has been launched the following window is displayed. In order to run the program and leave the debugger click on *Run* then on *Terminate* and leave the *Debug* panel. Then reset the TPMS emitter. No frames will be sent if the module has not been reseted.

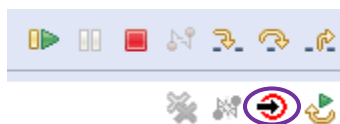
In order to work in debug mode do the following:

Click here to run the program
after having reseted the
device (see below)

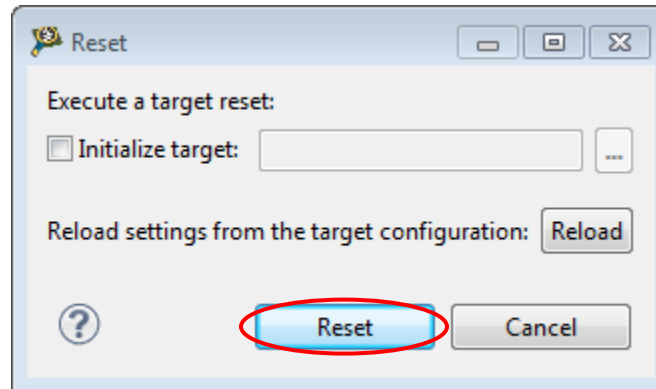
In this window the breakpoints can be managed.
Look at the Important note concerning the
breakpoints below.



Before being used the device needs to be reseted. It is necessary to reset it each time it has been reprogrammed. For that click on *Reset*.



Then click *Reset* again.

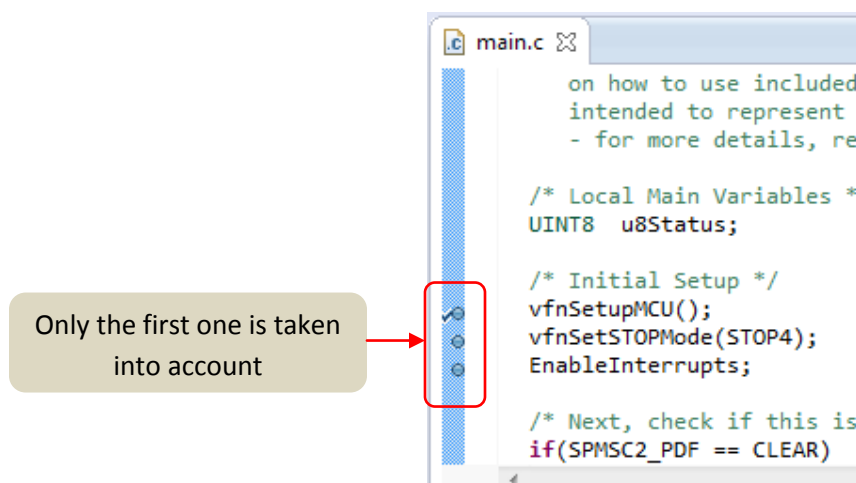


It is only after this reset that the device can be used. Do not run the program before having reseted the device.

Important note – breakpoints:

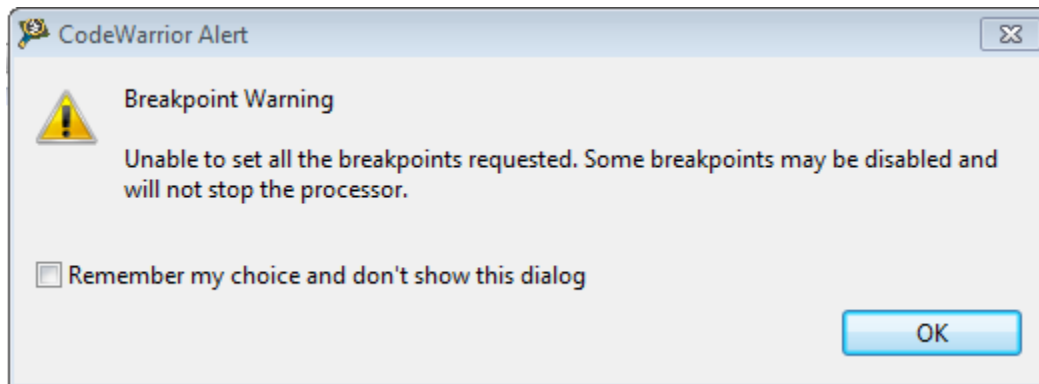
With TPMS devices it can be used only **ONE** breakpoint at a time. By default one breakpoint is set by the debugger at the beginning of the main. That is why the program is stopped at the beginning of the main after the debugger has been launched.

Once the program has been stopped at the beginning of the main, another (and only one) breakpoint at a time can be set elsewhere. If several breakpoints are set, only the first one will be taken into account.



If the user wants to have several breakpoints **it is not possible to have them at the same time**. The only possible thing to do is to set one breakpoint, then when the program stops on this breakpoint the user can delete it (by double clicking on it) and set a new one at another line. Then when the program reaches this other breakpoint the user can delete it, set a new one...

If a breakpoint is set somewhere **before** entering the debugger, the following warning will appear.

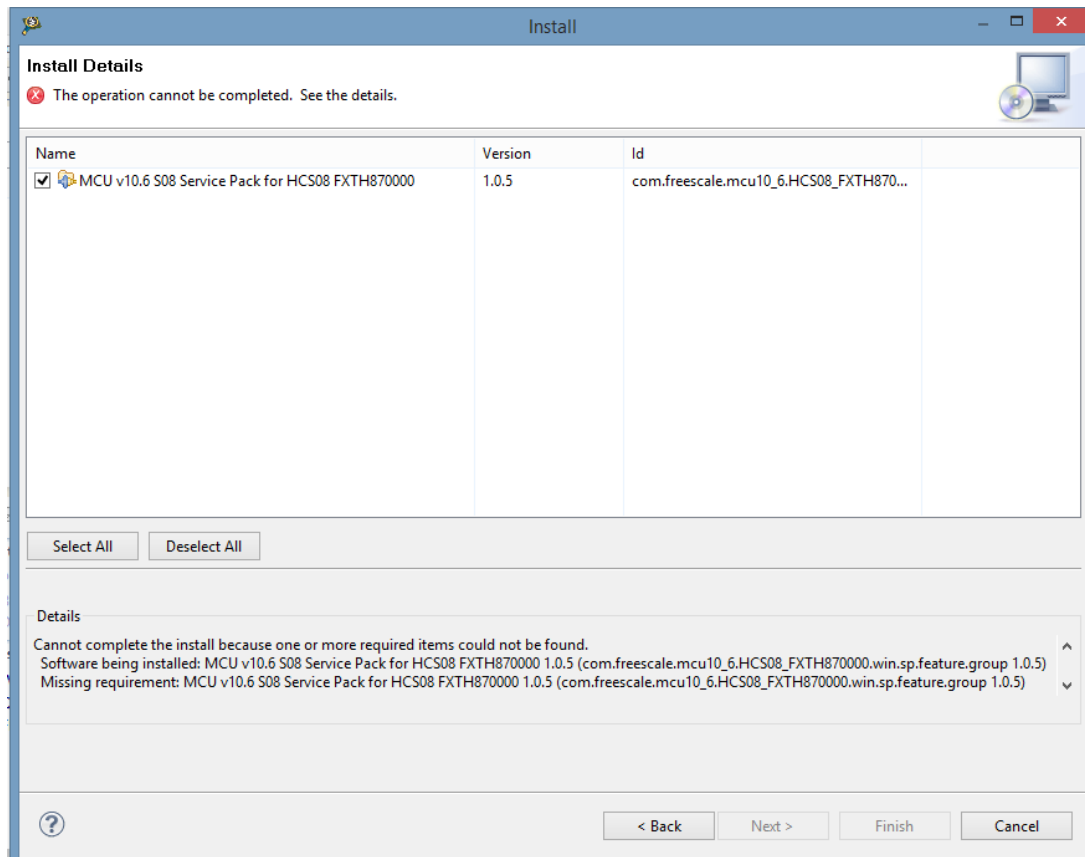


In this case the program will stop at the breakpoint set by the user and not at the one set by default (at the beginning of the main). Again, if several breakpoints have been set, only the first one will be taken into account.

4. Troubleshooting

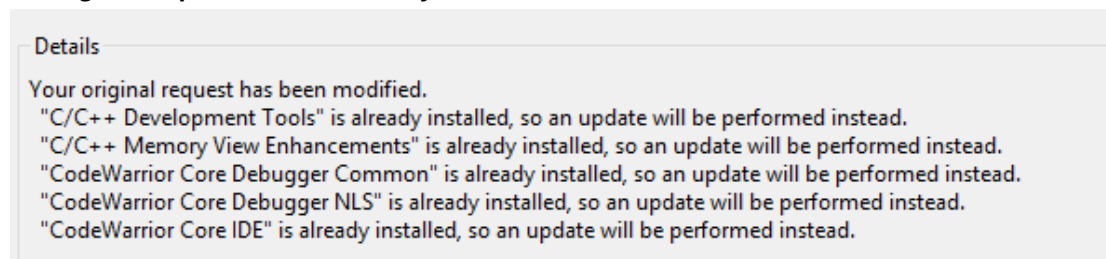
a. During the installation of the update

⇒ ***The operation cannot be completed: one or more required items could not be found***



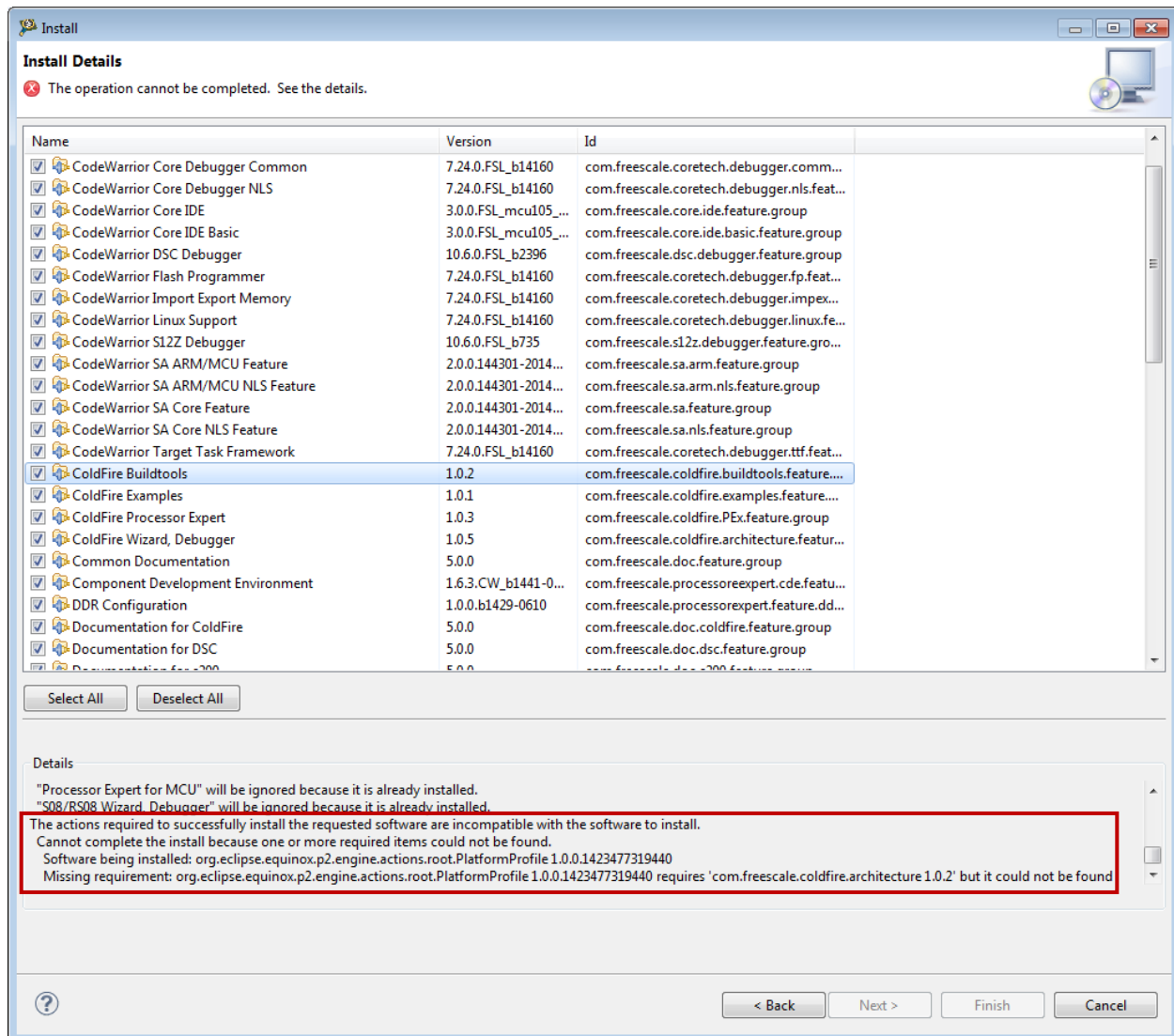
This error means that some necessary complementary packages have not been selected. So go back to the previous window and select additional packages. To know the list of the necessary packages, c.f. to the end of Part 2: Note 2 – General information about update installation.

⇒ ***Your original request has been modified***



This is not a problem, so continue the installation.

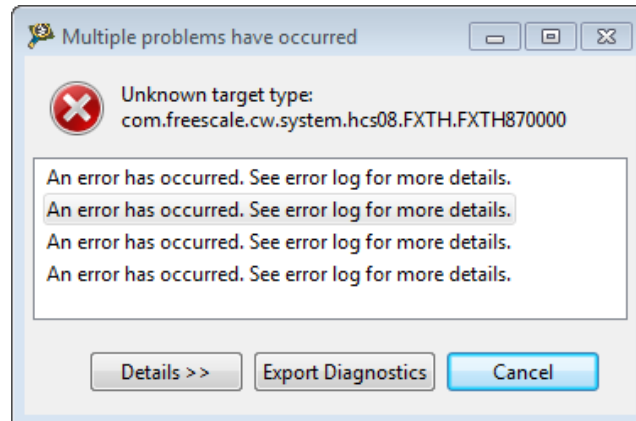
⇒ **Cannot complete the install because one or more required items could not be found**



This means that some packages related to components that have not been installed have been selected. In this example, updates for coldfire components have been selected whereas the coldfire component has not been installed (it has not been selected during CW 10.6 installation).

b. When opening a TPMS project with CW 10.6

⇒ **Unknown target type**



The necessary Service Pack has not been installed.

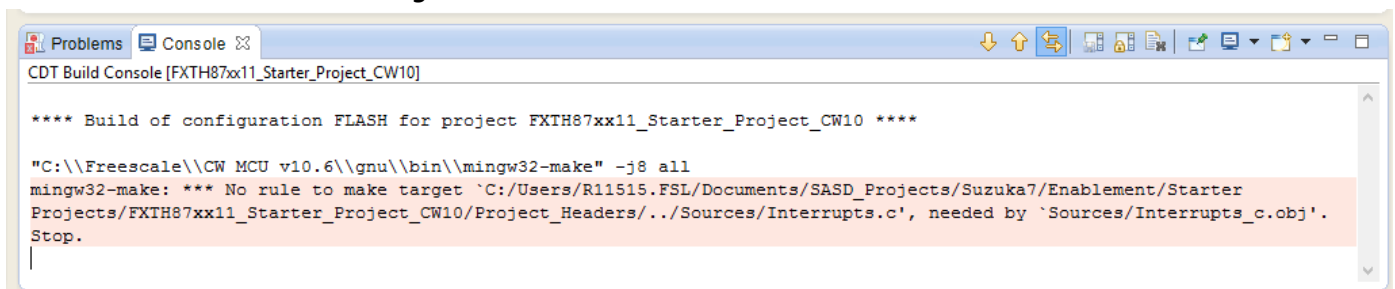
Verify the content of the following folder:

(CW Installation Path) > CW MCU v10.6 > eclipse > p2 > org.eclipse.equinox.p2.core > cache > binary

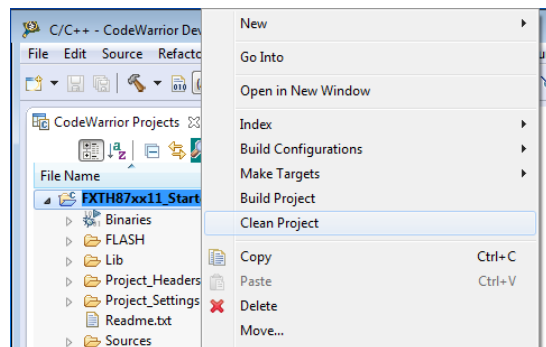
The `com.freescale.mcu10_6.HCS08_FXTH870000.win.sp_root_1.0` file should be here. If it is, close the project, close CW 10.6 and then reopen everything. If the file is not in the folder follow the whole procedure explained in Part 2 of this document.

c. When building a TPMS project with CW 10.6

⇒ **No rule to make target**

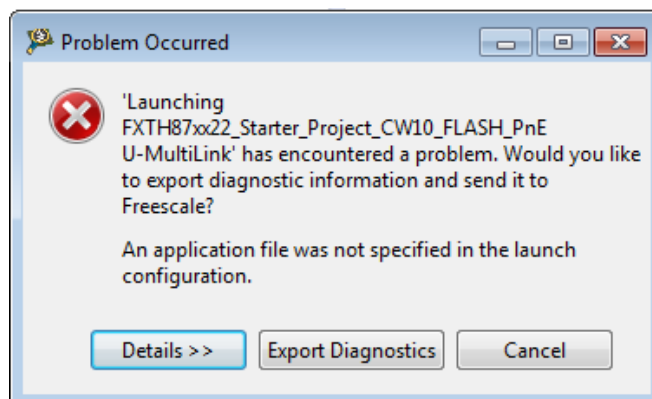


The project needs to be cleaned before being built. For that, right click on the project and select *Clean Project*. Then try to build it again.

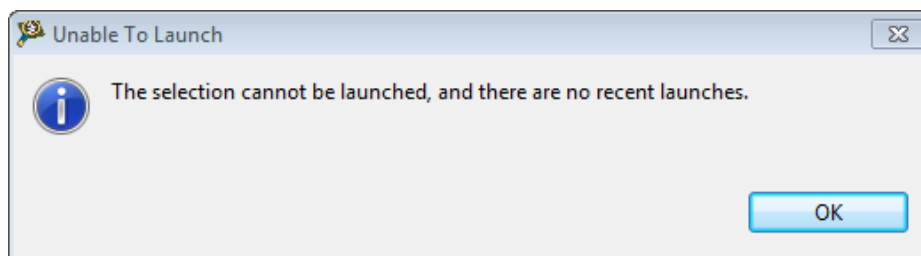


d. When entering the debugger

⇒ **Problem Occurred or Unable To Launch**



Or

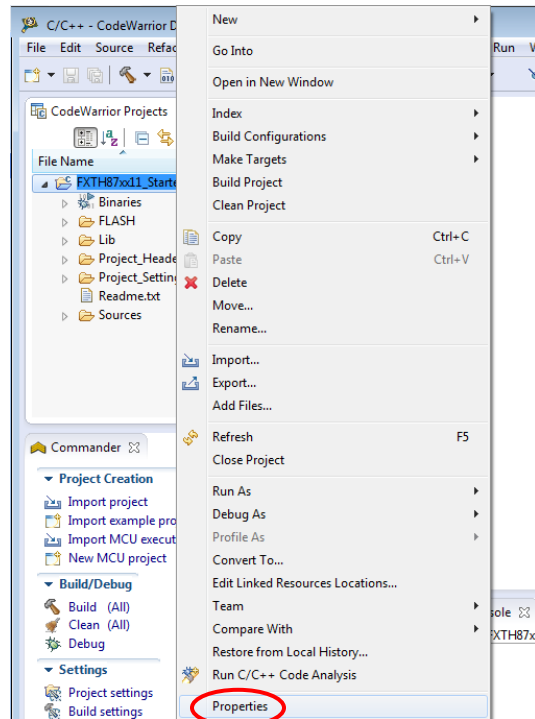


Before programming the device for the first time some Debug Configurations may need to be (re)selected. To do that the project needs to be built first.

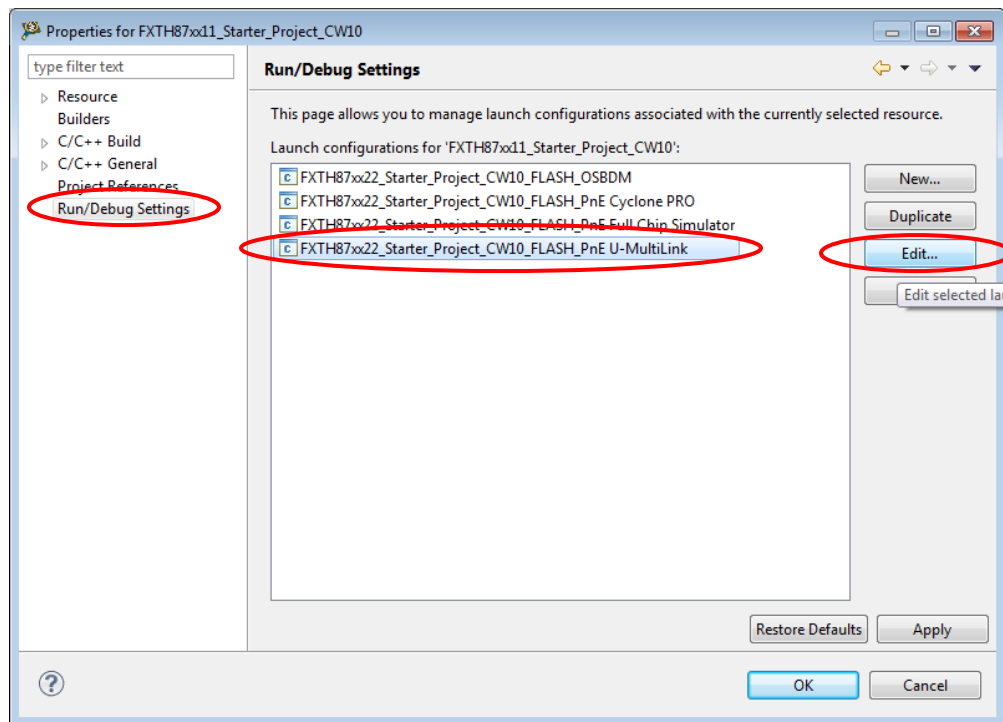
Leave the debugger if it has been launched. To leave the debugger click on C/C++ at the top right part of the CW window.



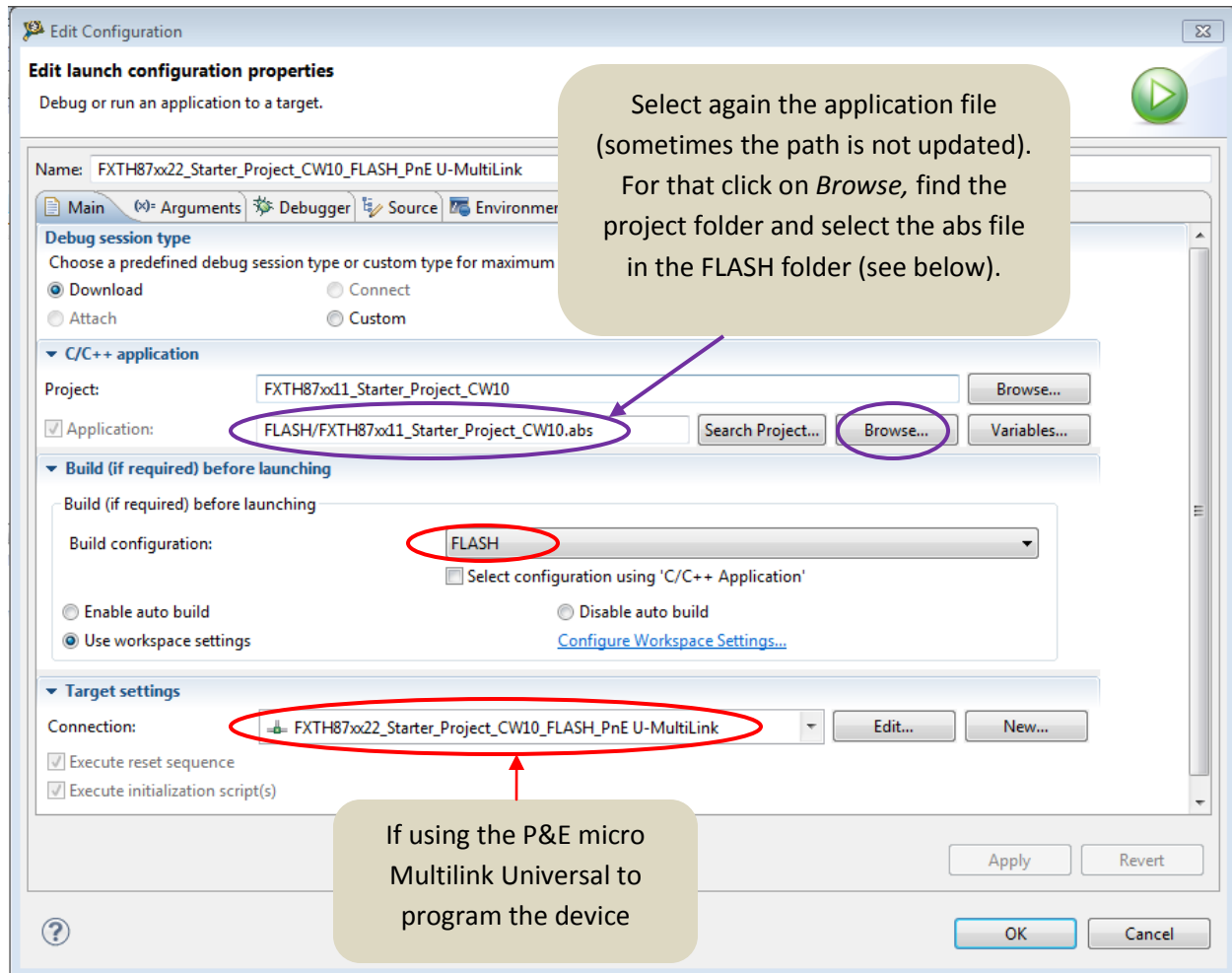
Then build again the project if necessary, right click on the project and select *Properties*.



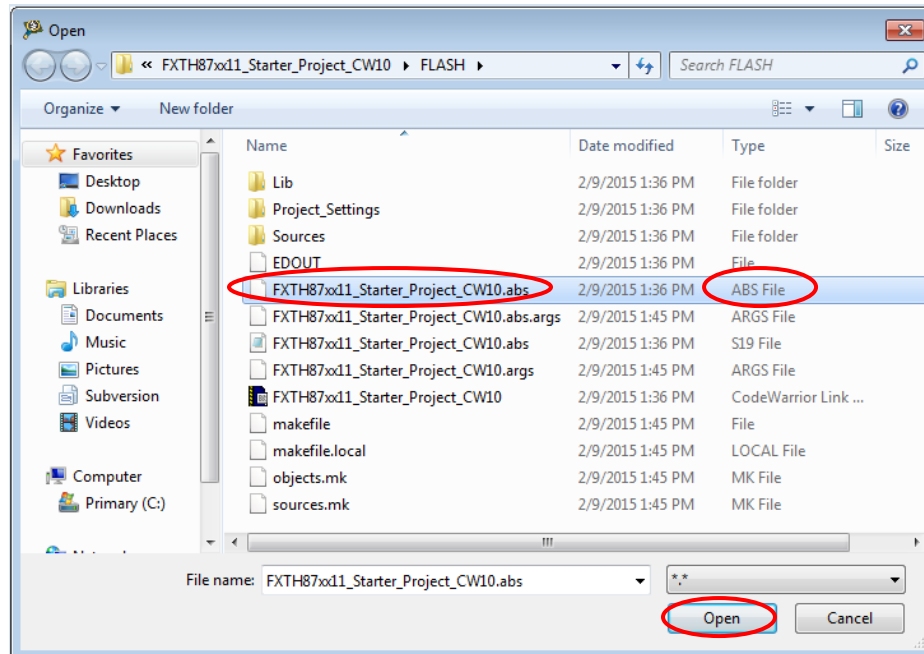
Go to *Run/Debug Settings* and edit the configuration that is used. Here it is the *FLASH_PnE U-Multilink* configuration (the P&E micro Multilink Universal is used to program the device).



The Configuration window must look like the following.

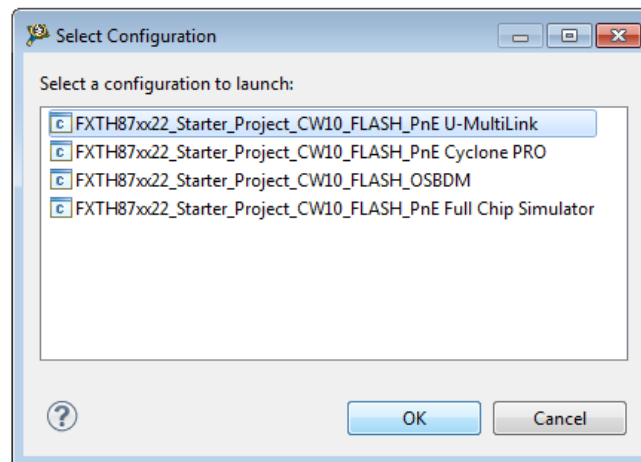


To select the application file go into the FLASH folder (it is in the project's folder) and choose the abs file (be careful: do not select the s19 file that has the same name).



Once the configurations are done, click *Apply* and OK.

In Code Warrior, click on *Run* and then *Debug*. If the following window appears, choose again the correct configuration.



e. When running the program after having reprogrammed the device

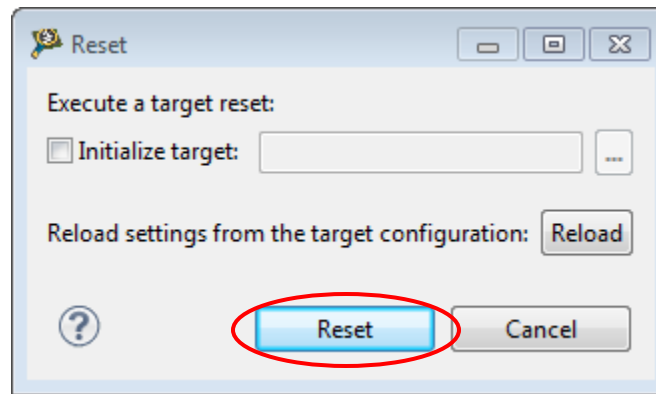
⇒ If the program does not stop at breakpoints

Verify that only one breakpoint is set. If it is the case the target has probably not been reset. The device must always be reseted after having been reprogrammed.

In the debugger window click on *Reset*.



Then click on *Reset* again.



After this reset a new breakpoint can be set.