## Contents

1. Installing Code Warrior for MCUs v10.x ................................................................. 3  
   a. General information........................................................................................................ 3  
   b. Downloading the IDE .................................................................................................... 4  
   c. Installing CW 10.x......................................................................................................... 5  

2. Programming the FXTH reference project ........................................................................... 6  
   a. Downloading the project ............................................................................................... 6  
   b. Importing the project under CW 10.x .......................................................................... 7  
   c. Programming the device ............................................................................................... 10  
      i. Hardware set up ........................................................................................................ 10  
      ii. Debug Configuration ............................................................................................... 11  
   d. Running the program .................................................................................................. 14  

3. Troubleshooting ................................................................................................................ 16  
   a. Compilation error ‘No rule to make target’ ................................................................. 16  
   b. When entering the debug session: ‘Problem occurred’ or ‘Unable To Launch’ ............ 17  
   c. Debug configuration: creating a target ....................................................................... 20  
   d. Debug configuration: no binaries/application file available ....................................... 23  
   e. In debug mode: the program does not stop at breakpoints ........................................ 24
1. Installing Code Warrior for MCUs v10.x

a. General information

The TPMS projects provided by NXP are compatible with all CodeWarrior versions greater than 10.6.3 (included). Versions below 10.6.3 can be used but require an additional update package installation. This is why we will focus on versions above 10.6.3. The screenshots refer to version 10.7 but the whole procedure is applicable to all versions above 10.6.3.


An evaluation version can be downloaded from this page (need to sign in).
b. Downloading the IDE

The evaluation version is free. On this page choose the Eclipse IDE for MCUs v10.7. It is also possible to download version 10.6.4 in the panel Previous:

After reading the license agreement, select the desired executable (Offline or Online) and download it either directly by clicking on the file or via the installer:
c. Installing CW 10.x

Once downloaded the application can be executed.

After having read and accepted 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 continue and finish the installation.
2. Programming the FXTH reference project

a. Downloading the project

A TPMS reference project can be found on the FXTH87 webpage: www.nxp.com > Sensors > Pressure Sensors > Tire Pressure Monitoring Sensors > FXTH87 > Software & Tools. Under Lab and Test Software, the TPMS reference projects and Graphical user interface can be found.

Download the FXTH reference project TPMS_FXTH87_MKW01_LF_RF_Communication_CW10_RevX and unzip it. Documentation on the TPMS demo setup can be found in the Documentation folder of the project.
b. Importing the project under CW10.x

Start CodeWarrior v10.x. Select the workspace i.e. the folder in which your project will be stored.

When the IDE is started click on *Import Project* (bottom left panel):
Select the unzipped folder of the project.

For the FXTH reference project select the `TPMS_FXTH87_MKW01_LF_RF_Communication_CW10_RevX` folder

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

When the project’s folder is selected click *Finish*
Once the project is opened, build it by clicking on the hammer. Four compilation targets are available, select the appropriate one (here 434MHz, 2-axis device).

If the compilation error ‘No rule to make target’ occurs, clean the project 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 (search for USBMULTILINKBDM on www.nxp.com).

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

![Connector board and connection between Multilink and TPMS emitter](image)

- Blue LED indicates that the Multilink is connected to the computer
- Yellow LED indicates that the emitter is supplied
- Both LEDs must be turned on to start programming
- 3V power supply
- To the computer
- To the emitter
ii. **Debug Configuration**

Open the *Debug Configuration* window:

![Debug Configuration dialog]

- Double click on **CodeWarrior**
To select the abs file click on Search Project and select the frequency used at compilation (here 434MHz). If no binary is available, it means the project has not been built. Build the project in order to select and application file (binary):

Name of the configuration. This can be modified if needed.

Name of the CW project as it appears in the workspace

The executable file (abs file) needs to be selected. See below.

The target needs to be selected. See below.
To specify a target, click on the little arrow of the empty connection field and the TPMS_BDM connection will be available. Select this connection.

If there is no target available, go to the section Troubleshooting – Debug configuration: creating a target.

If the following message appears during programming, turn off then on the FXTH power supply without disconnecting the BDM then click OK.
Installing and using CW 10.x for TPMS applications

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. Disconnect the BDM from the FXTH, then do a hardware reset of the TPMS emitter (turn off then on the FXTH after the BDM has been disconnected). The FXTH may not start correctly if the hardware reset is not done after disconnecting the BDM.

- In order to work in debug mode do the following:

1. Click here to run the program
2. In this window the breakpoints can be managed. Look at the important note concerning the breakpoints below.
3. Click here to terminate the debug session
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...
3. Troubleshooting

a. Compilation error ‘No rule to make target’

⇒ No rule to make target

It means 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.
b. When entering the debug session: ‘Problem occurred’ or ‘Unable To Launch’

⇒ **Problem Occurred or Unable To Launch**

Before programming the device some Debug Configurations need to be (re)selected. To do that the project first needs to be built (be careful of the compilation target selected!).

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 should look like the following (it can also be directly accessed via Run > Debug Configurations or via the little arrow near the debug button):

![Debug Configurations Window](image)

When configuring for the first time, make sure the selected Build configuration is Use Active:
c. Debug configuration: creating a target

If no target is available when clicking on the little arrow then a new connection needs to be created. For that click on New.

Select Hardware or Simulator Connection:
Enter a name and click on New:

Enter a name and select the target type (click on the little arrow and select FXTH870000 under hcs08.FXTH:)

Do not fill Initialization and Memory

Then click Finish.
In **Connection type** select *P&E HCS08 Multilink\Multilink Universal\Cyclone Pro\OSBDM*:

![Graphical representation of the connection type selection](image)

If the Multilink is connected to the computer, the port should show up automatically. If not, click **Refresh**.

Then click **Finish**.

![Graphical representation of the configuration settings](image)
d. Debug configuration: no binaries/application file available

If no binary/application file is available when clicking on Search Project it means the project has not been built. The project needs to be built in order to generate an application file.
e. **In debug mode: the program does not stop at breakpoints**

- **If the program does not stop at breakpoints**

Verify that only one breakpoint is set. If it is the case the target probably needs to be reseted.

In the debug window click on *Reset*.

Then click on *Reset* again.

After this the program should stop at the breakpoint. Remove and replace the breakpoint if necessary.