

Using Melexis Debugger

Abstract

This document describes the setup and handling for the Melexis simulator/emulator and the Melexis Interactive Debugger.

Important Note: You need a **functional** development project which can successfully build a target binary and launch a debug session.

Please note: The Melexis simulator/emulator adaption does not support interactive debugging features when executing tests with TESSY. (See 5 to learn how to debug interactively having your test data statically built into the target binary.)

Table of Contents

Abstract	1
1 Introduction.....	2
2 Prerequisites	2
3 Melexis Configuration.....	2
4 TESSY Environment Settings.....	3
5 Interactive Debugging.....	4
6 Troubleshooting.....	6

1 Introduction

TESSY supports the Melexis simulator/emulator for automatic test execution using an interface DLL provided by Melexis. During test execution, there is no interactive debugging possible, i.e. the “Define Breakpoint” feature of the TESSY execution dialog has no functionality within this target environment.

In order to debug the test application interactively with the test case values provided from within TDE you need to rebuild the test application in a special mode, i.e. the input values will be compiled into the application (see chapter 5).

2 Prerequisites

Running tests with the Melexis simulator from TESSY requires the installation of both the simulator **and** the emulator software with the minimum versions listed below.

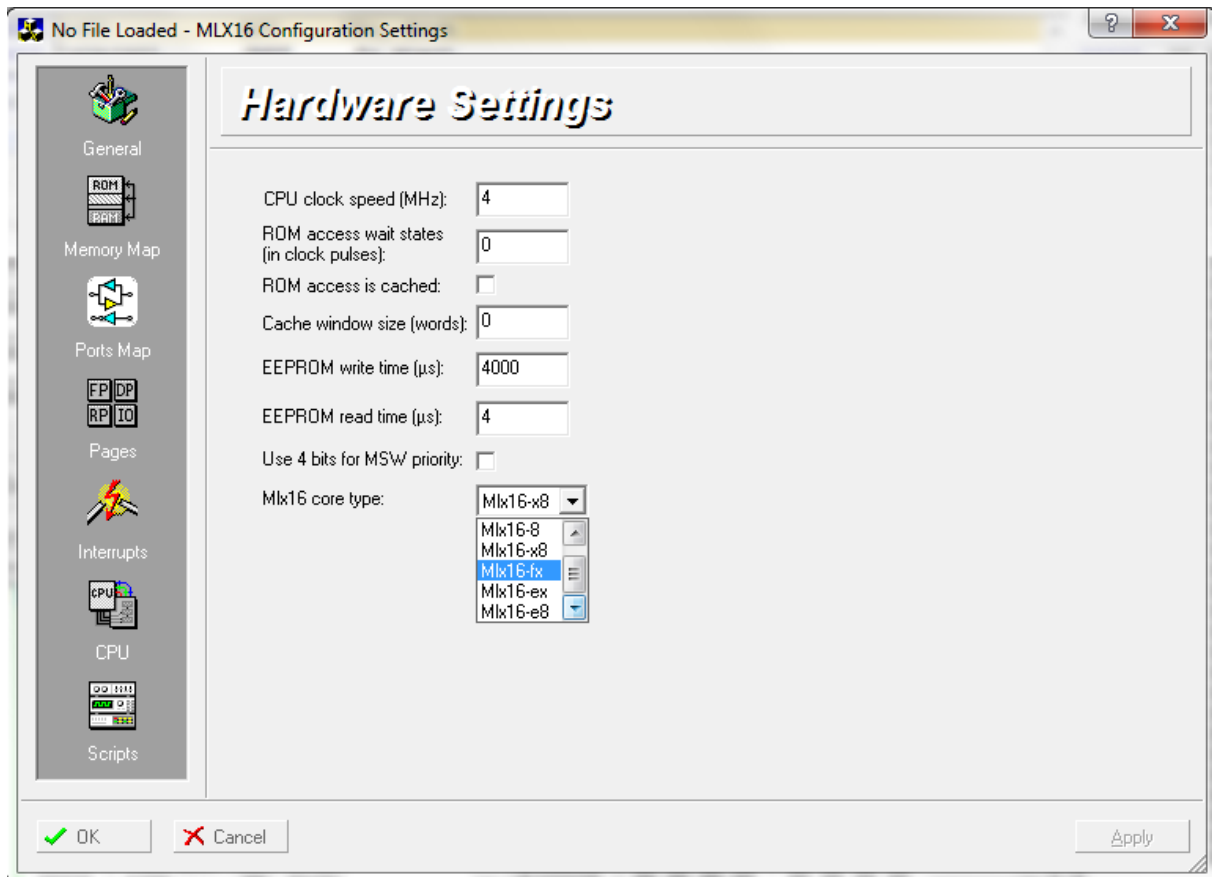
You need to have a Melexis configuration readily prepared for the simulator. The table below lists the supported Melexis controllers and the required minimum version of the Melexis tool chain.

Controller	Required Melexis Simulator Version
Mlx16	2.60
Mlx16-8	
Mlx16-x8	
Mlx16-FX	
Mlx16-ex	
Mlx16-e8	

The predefined setup within TESSY is prepared for the Mlx16-fx controller, if you are using another controller, you need to adapt the TESSY environment settings and the Makefile template accordingly.

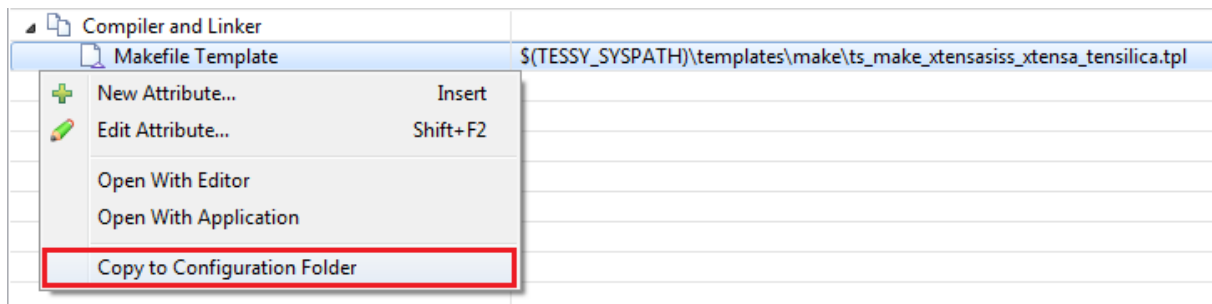
3 Melexis Configuration

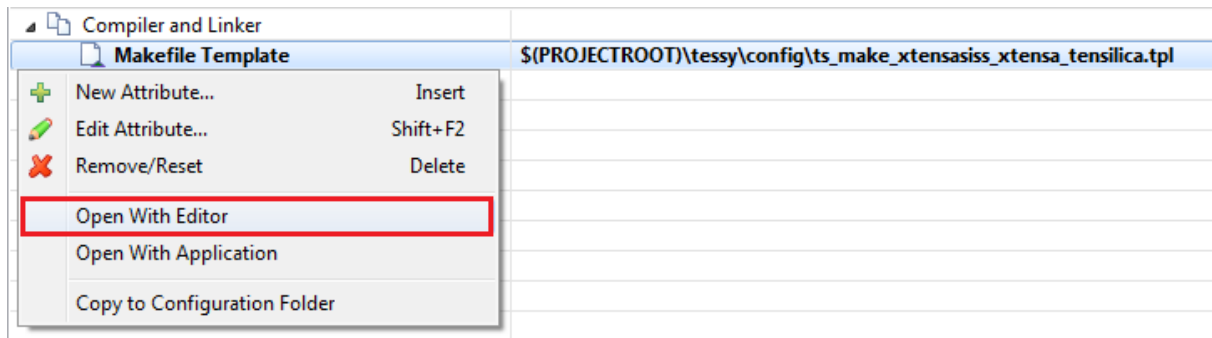
The Melexis **CfgXplorer** provides the configuration of the simulator/emulator. You need to select the appropriate CPU and save your configuration. The screenshot below shows the CPU tab of the **CfgXplorer** with a list of the available Mlx16 controllers.



4 TESSY Environment Settings

The best way to start is to set the **Compiler Install Path** and the **Target Install Path** and check what errors remain after toggling the **Show Errors/Warnings** toggle button, which is found in the **Attributes** view's toolbar. Please adjust all remaining unresolved paths being displayed. If you are using a different CPU than the default from the TESSY configuration, you will have to create a new folder within your TESSY project and save the corresponding startup code into the folder. Let TEE attribute `InitObjDir` point to the folder. Finally, check **Linker** and **Linker Options** attributes within the TEE. For more advanced compiler and linker settings copy the Makefile template into your TESSY project's config folder, open the file, and adjust the variables `S_COMP_OPTIONS` and `S_LINK_OPTIONS` accordingly to your local settings.

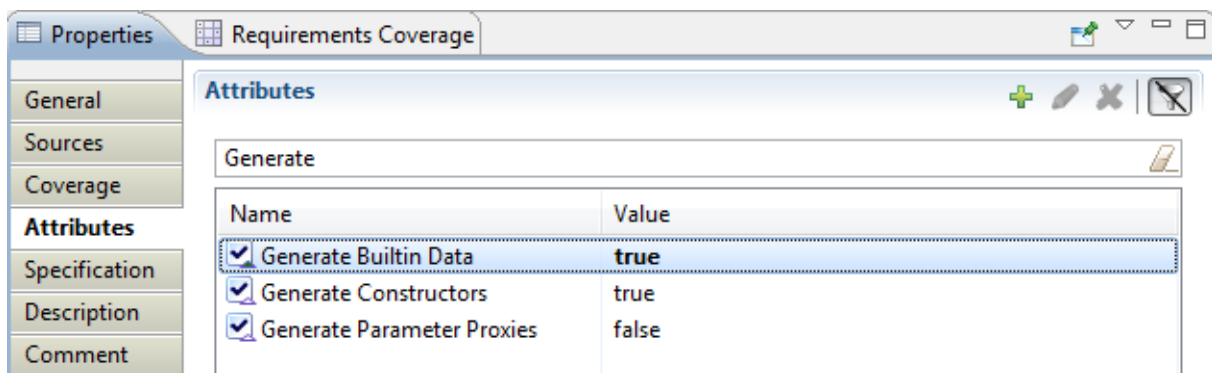




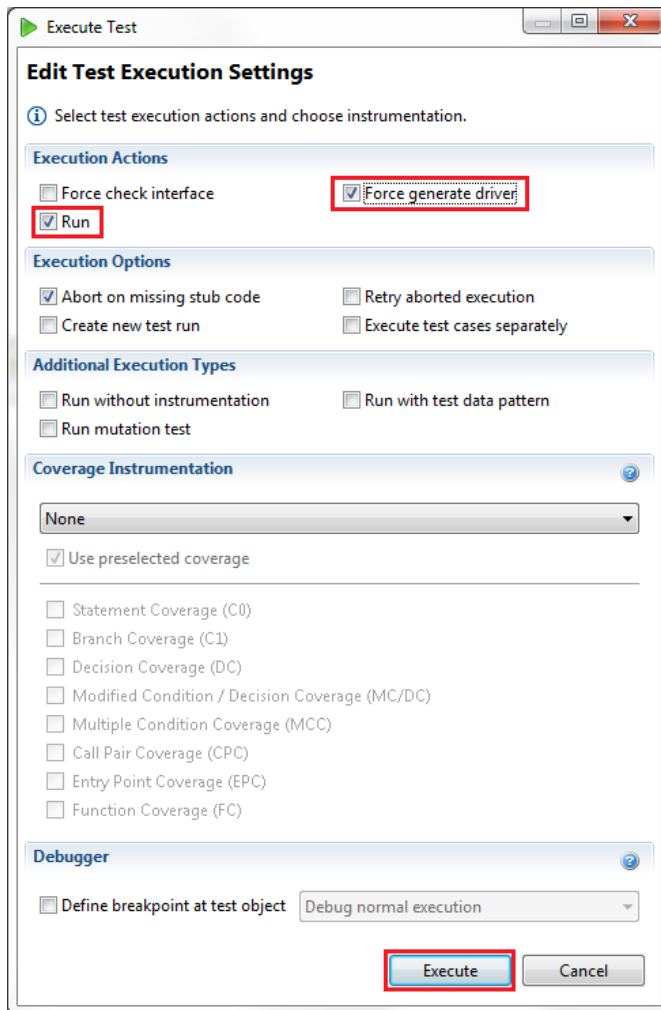
5 Interactive Debugging

Important Note: The test object has to be compiled with option **-O0**. If compiled with **-Os**, the source code of your test object cannot be displayed within the Melexis debugger.

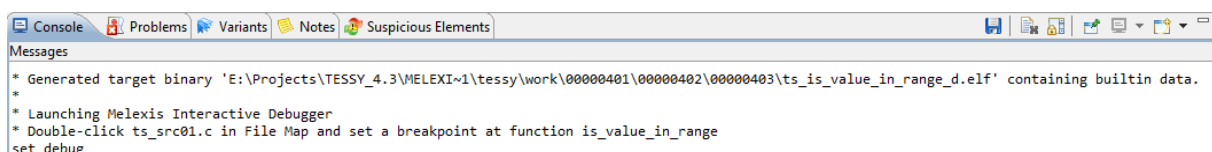
The TESSY Melexis debugger adaption does not support interactive debugging during a test run. But it is possible to debug your test object interactively having the test data built-in which might be useful in case of errors during a test run. So, in order to debug the test object interactively TESSY provides the **Generate Builtin Data** attribute. The attribute is of type Boolean and, if set to **true**, TESSY will rebuild your target binary during the next test run having the selected test data built-in, i.e. TESSY will not actually perform the test run but instead create the target binary with test data built-in to it. To disable this feature, you have to set the attribute to **false**.



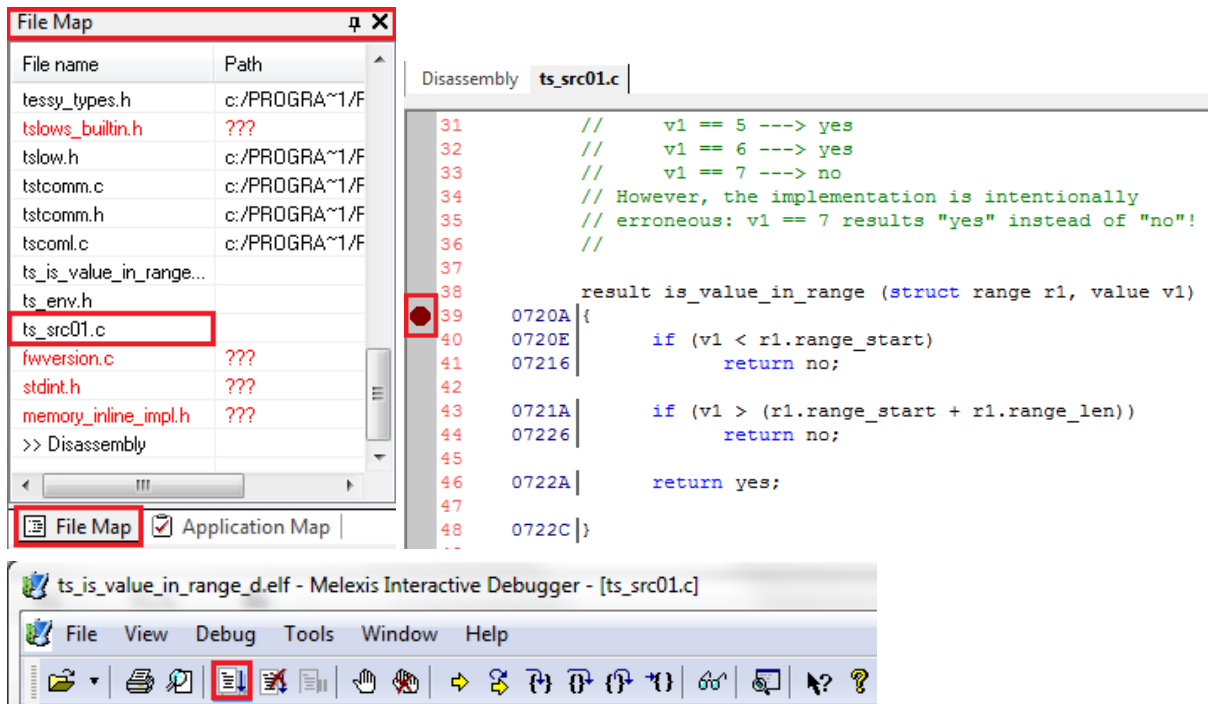
Open the **Execute Test** dialog and make sure **Force Generate Driver** is selected.



Now execute the test by pressing the **Execute** button. TESSY displays the path to the generated built-in target binary in the **Console** view.



If the **Target Install Path** is properly setup the Makefile will launch the Melexis debugger with the path to the generated binary. When the Melexis debugger is running, find the **File Map** view within the Melexis debugger and search for file *ts_src01.c*. Double-click the file name to open the file. Search for your test object. The name of your test object is also shown in TESSY'S **Console** view. Create a breakpoint at your test object and press the **Go** button from the toolbar of the Melexis debugger.



6 Troubleshooting

By now, there are no known problems. Please contact support@razorcat.com if you encounter any unsolvable problems.