How to evaluate TESSY

Frank Büchner, July 2016, rev. 013

1 Objective

The objective of an evaluation of TESSY is to figure out if the functionality provided fits to your needs.

Because TESSY always includes the Classification Tree Editor (CTE) for test case specification, you should also get at least a basic understanding about the method behind that editor and how this editor is integrated with TESSY. However, TESSY can be used without this editor.

2 Available Material

Hitex provides various documents and other material about TESSY, including a movie and evaluation software. This material is available on CD or can be downloaded from the Hitex web site (www.hitex.com/tessy). Some of the material, e.g. brochures and white papers, is also available in printed form.

3 Evaluation Route

The following gives you an overview about the available material and the proposed procedure of an evaluation of TESSY.

You should begin by studying some of the material (= “document evaluation”) and then proceed by exercising TESSY’s evaluation software.

3.1 Document Evaluation

An evaluation should always start by looking through the documents available from Hitex. You can get a good impression of the functionality of TESSY from these documents. It prevents you from putting effort in technical details (e.g. installing the evaluation software and getting a license) without knowing what you can expect afterwards.

The material presents TESSY from various perspectives.

3.1.1 Brochure

The brochure from Hitex about TESSY gives you an overview about the features of TESSY.

3.1.2 Video Tutorials

With ease you will obtain comprehensive information by watching the TESSY video tutorials. Within some minutes you get a good impression of a test with TESSY.

The source file used for the videos is installed with TESSY: <TESSY installation directory>\Examples\IsValueInRange\is_val_in_range.c

3.1.3 Recorded Webinars

About 10 recorded webinars on the operation of TESSY are available. A lot of topics are covered.

3.1.4 White Papers

Various white papers explain the theory behind unit testing, its benefits, and how unit testing is accomplished by TESSY. Also covered are the Classification Tree Method and the Classification Tree Editor.

3.1.5 FAQs

Glancing through the Frequently Asked Questions may answer some of your issues.

3.1.6 Testimonials

Maybe you want to read what TESSY users from different companies say about working with TESSY.

3.1.7 Tutorials

Several tutorials explain working with TESSY step-by-step using simple examples.

3.2 Practical Evaluation

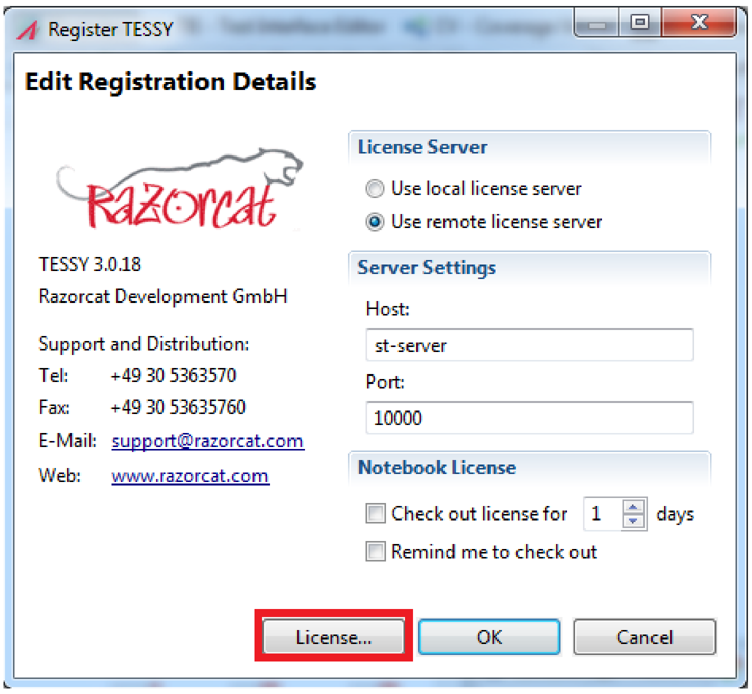
After a satisfying document evaluation, you probably want to give TESSY a try.

For that, you need to install TESSY and to get an evaluation license.

Afterwards, you can explore TESSY at various levels.

3.2.1 Install and Get an Evaluation License

- Get the TESSY installation software, either from http://www2.hitex.com/tessy-download-e or from the TESSY distribution media from Hitex.
- Install TESSY and all subcomponents on a PC running Windows. TCP/IP must be installed, but the host PC doesn't have to be connected to a network. Administrator rights and the right to write to the registry are needed during the installation of TESSY.
- Start TESSY. TESSY will automatically open the registration dialog when no valid license is found.
- Please request your evaluation license by following these steps:
  - In the TESSY registration dialog, first click on "License..."


Register TESSY

Edit Registration Details

TESSY 3.0.18
Razorcat Development GmbH

Support and Distribution:
Tel: +49 30 5363570
Fax: +49 30 53635760
E-Mail: support@razorcat.com
Web: www.razorcat.com

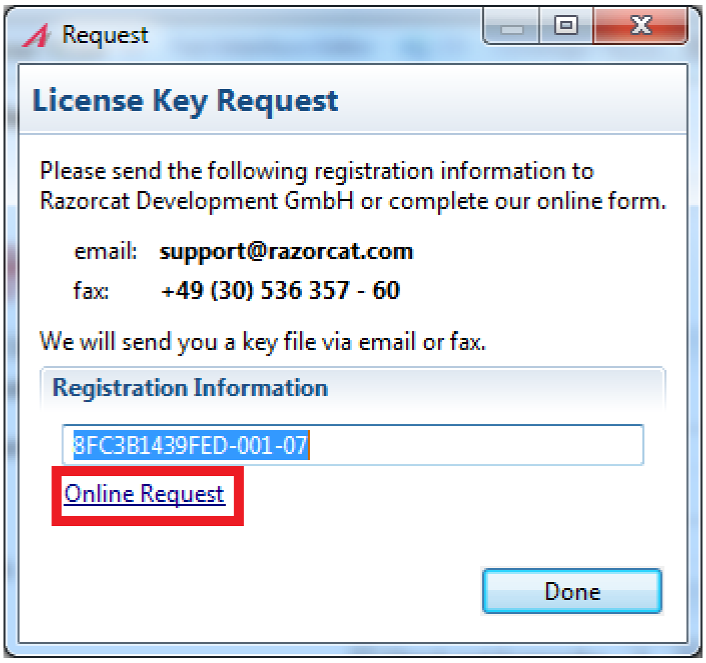
License Server
Use local license server
[✓ Use remote license server]

Server Settings
Host: st-server
Port: 10000

Notebook License
Check out license for 1 days
Remind me to check out

Registration dialogue for TESSY

- In the following dialogue click on "Online Request".



Please send the following registration information to Razorcat Development GmbH or complete our online form.

email: **support@razorcat.com**
fax: **+49 (30) 536 357 - 60**

We will send you a key file via email or fax.

Registration Information

8FC3B1439FED-001-07

License Key Request for TESSY

- Fill in the form and submit it.
- You will get your evaluation license file by e-mail. The evaluation license is created in a manual process by a human. I.e. you should not expect to receive the evaluation license immediately, especially if your request was sent outside the normal working hours in Germany. Sometimes additional information (e.g. the microcontroller you want to use; the background of your evaluation) may be necessary to get an evaluation license.
- Follow the instructions you get with the license key file to activate the license.

The evaluation license is valid from the point in time you receive it. The evaluation license expires after some days (usually 10 days). This is independent if resp. how often you use the evaluation license.

The evaluation license will be a time-limited full-featured license for TESSY (and CTE).

3.2.2 Levels of Practical Evaluation

3.2.2.1 Using the Gnu C Compiler for Windows

This level allows you to explore almost all features of TESSY.

For this kind of evaluation, all necessary software components are already installed and all necessary settings are already done during the installation of TESSY. The debugger used is Eclipse/CDT or VisualGDB for Windows.

Remark

Be aware of the following issues if you use the Gnu C compiler for Windows to test your code instead of the cross compiler for your embedded system (refer also to the following section 3.2.2.2).

- All source code to be tested must be ANSI-C compatible, because it has to be compiled by the Gnu C compiler for Windows. I.e. it must not contain special keywords normally used in embedded programs like "__interrupt", "sfr", "bit", etc.
- There may be different behavior of the test object resulting from e.g. different integer sizes, little vs. big endian, etc.
- Because the test is run under Windows and not on the real microcontroller, there is no access to peripheral registers and/or external hardware.

First Steps

Use the video tutorials, webinars, and other documentation as introduction.

Advanced Topics

You may now exercise the more advanced features of TESSY:

- Export / import of test data
- Generate test reports
- Check the code coverage of your tests
- Run tests without user interaction
- Create various stubs for functions called by your test object
- Accept a numeric result with a deviation
- …

We recommend writing your own small functions (20 lines approx.) to exercise and experiment with TESSY:

- How is the path to header files set?
- How are #defines set for the test object?
- How can you direct TESSY to generate a stub function for you?
- How are pointers handled?
- How is a state-machine tested?
- What happens, if you change your sample code, e.g. rename a variable or change its type?
- …

Valuable documentation to important topics is available in TESSY by using the menu Help > Documentation…

3.2.2.2 Using a Cross Compiler

Using TESSY with a cross compiler for a certain microcontroller allows you to test source code which is not ANSI-C compliant (e.g. because it contains special keywords like "interrupt") and to run your tests eventually on the real microcontroller / target system.

However, you must dispose of such a cross compiler and additionally you must dispose of an execution environment, which allows TESSY to run a binary created for a specific microcontroller. Normally, a debugger for that microcontroller is used for that purpose. Often, such a debugger is able to act either as interface to an instruction set simulator, or as an interface to an in-circuit emulator, or as interface to a JTAG / DBM debugger.

Please read the documents about your cross compiler and your target system from the TESSY main menu Help > Documentation > Compiler and . Help > Documentation > Targets

Please note:

Target Software Sample Packages which use a cross compiler /debugger and an appropriate instruction set simulator are available from

http://www.razorcat.com/downloads/samples/

Please check if an appropriate package for your intended cross compiler / debugger is available. If yes, please download the package and use this package as a starting point. (Download the package, unzip it, double-click on the *.pdb resp. *.pdbx file, …)

If an appropriate package is not available, follow the steps below.

Step 1: Enabling the Cross Compiler and the Debugger

To be able to use a cross compiler, you must first enable this compiler and the appropriate debugger using TESSY's Environment Editor (TEE). In TESSY, use the menu "File > Edit Environment …"

Step 2: Use Cross Compiler and Instruction Set Simulator

If your debugger of choice is able to act as instruction set simulator, we highly recommend making use of this possibility. We also recommend using initially a very simple test object, e.g. is_value_in_range() from is_val_in_range.c.

This source file is installed with TESSY: <TESSY installation directory>\Examples\IsValueInRange\is_val_in_range.c

As an intermediate goal, TESSY should compile is_value_in_range() using your cross compiler and then execute some tests using your debugger acting as instruction set simulator.

Then you can go ahead and try your application code instead of is_value_in_range().

Step 3: Use Cross compiler and JTAG Debugger

Adapt the default settings for compiling and linking of the test application to your target hardware. This is primarily done by using TESSY's Environment Editor (TEE).

Some Background Information

If you want to run (any) program using a JTAG or BDM debugger, that program must run on the real target hardware. I.e. the program must be loaded in the program memory of that target hardware and probably some hardware specific settings have to be made by the program during startup. Hence, the program must be linked appropriately to the specific target hardware (i.e. the program code must be located in program memory; variables, stack, and heap must be located in RAM) and probably specific startup code has to be used. [The situation is not so bad when an in-circuit emulator is used, because an in-circuit emulator is able to provide memory for program execution in a much more flexible way than a JTAG debugger.]

How should TESSY know about your specific target hardware by default? This is actually not possible! By default, TESSY compiles and links the program used for testing ("test application") according to some default settings. To enable TESSY to generate the test application appropriate to your specific target hardware, you have do adapt TESSY's standard settings to your target hardware.

TESSY's default settings normally are appropriate to execute the test application using a debugger acting as instruction set simulator. Therefore, you should always start by using an instruction set simulator. This makes it much easier to sort out problems like wrong or missing paths, etc.

Important for the adaptation is the makefile template used by TESSY. It normally allows customizing things like compiler and linker options, link addresses, alternative startup files, etc. See help > Documentation > Customizing > Makefile Templates.

TESSY uses a makefile template as basis for the generating the test application. It needs to be changed only in rare cases.

However, the most challenging task is not to direct TESSY to use different settings, but to find out, what the settings for a specific hardware have to be. Furthermore, this information has to be given to the cross compiler and cross linker that you use, i.e. you maybe have to change linker command files or compiler options.

Therefore: You need some familiarity with your compiler, and your linker, and your target hardware to be able to run the tests on your target hardware.