

# Using SEGGER Embedded Studio emSim Simulator

## Abstract

This document describes the usage of the SEGGER Embedded Studio emSim simulator as target debugger. At the time of writing this document version 7.32a of the simulator was tested. The SEGGER emSim simulator supports multiple module tests in parallel. The communication between TESSY and SEGGER emSim uses files only. No breakpoint is required. In order to interactively debug your test object, SEGGER Embedded Studio's debug binary needs to be replaced by the built-in data binary compiled by using TESSY and the SEGGER CC compiler. The replacement is done automatically by TESSY. Refer to chapter 4 for further notes.

**Important Note:** A *functional SEGGER Embedded Studio project which can build a target binary and successfully launch a debug session is required as a prerequisite!*

**Please note:** The SEGGER emSim debugger adaption does not support interactive debugging features when executing tests with TESSY. (See 4 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 TESSY Environment Settings.....	2
3 Parallel Built and Parallel Execution .....	3
4 Interactive Debugging.....	4
5 Troubleshooting.....	7

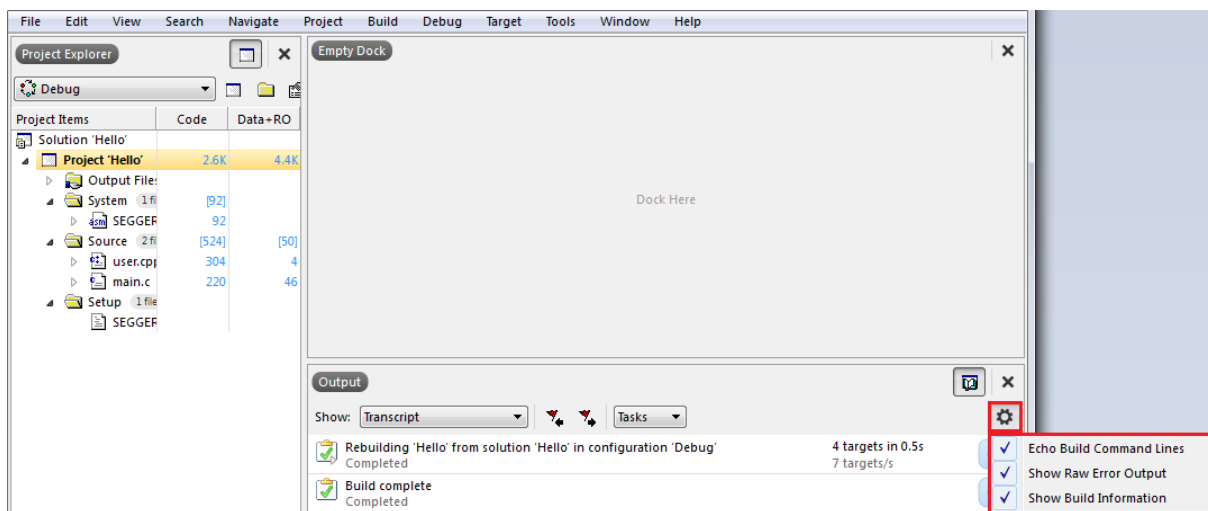
# 1 Introduction

The communication between TESSY and the SEGGER emSim debugger is based on `emSim.exe` which exchanges the test data and test results via files. The `emSim.exe` is started by TESSY by executing the command line found in TEE attribute `Slave Call`. Please do not alter this attribute.

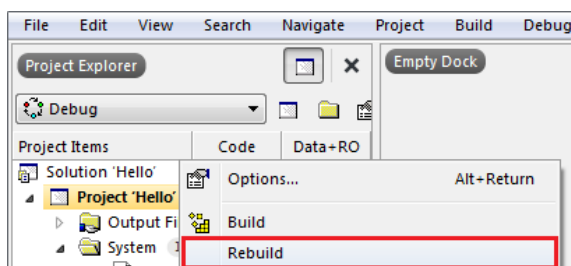
In order to debug the test application interactively with the test case data provided from within TDE you need to rebuild the test application in a special mode, i.e. the input values will be compiled into the application. The application binary is copied automatically into your SEGGER Embedded Studio project. TEE attribute `Project Binary Path` has to be set appropriately. Please refer to chapter 4 for further details on how to generate the binary.

# 2 TESSY Environment Settings

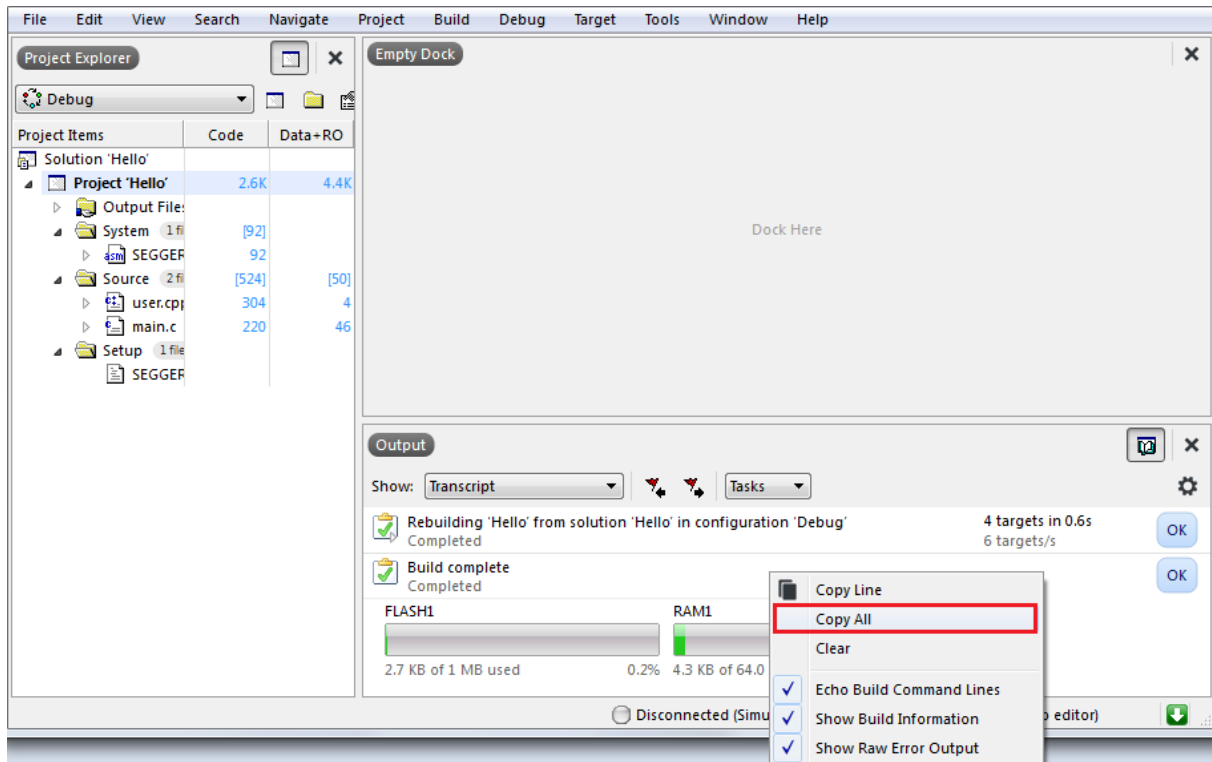
At first, set the `Compiler Install Path` and the `Target Install Path` and check which errors remain after toggling the `Show Errors/Warnings` toggle button, which is found in the `Attributes` view's toolbar. Please adjust all remaining unresolved paths being displayed. Next, check the TEE attributes `SEGGER Compiler Options` and `SEGGER Linker Options`. The default setting for the RISC-V processor matches the `rv32`. The following steps will help you to find the compiler and linker option. Open SEGGER Embedded Studio and activate the built log messages.



Rebuild the project.



Copy all build messages and paste them into an editor.



Remove all dependency arguments and the dependency file argument, all object file arguments, and the source file argument from the copied command lines. Substitute the SEGGER installation directory of the `-isystem` option for `$(Compiler Install Path)`. Finally, copy the resulting arguments and paste them into the respective **SEGGER Compiler Options** and **SEGGER Linker Options** attributes of the TEE.

If you are using a different processor you will also have to use your own startup code, which is found under the `Obj` folder of your SEGGER project folder. It is strongly recommended to collect the startup code objects in a new folder of your TESSY project. Let TEE attribute `InitObjDir` point to that folder path and let TEE attribute `Linker File` point to the new linker file, which should be copied into your TESSY project `config` folder.

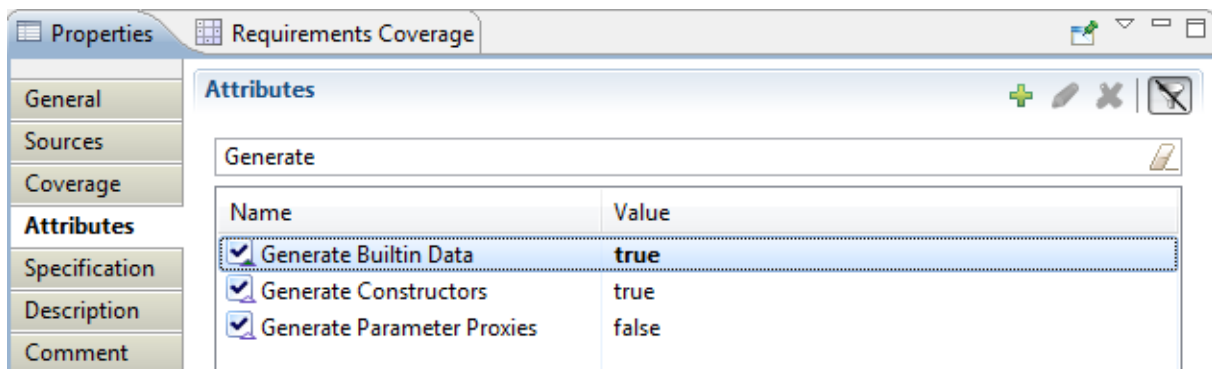
### 3 Parallel Built and Parallel Execution

The amount of test objects built in parallel is given by **Compiler Concurrency**, which is by default set to 4, while 20 source files per test object are compiled in parallel. So, no more than 80 processes run by default in parallel for the built. You can disable the parallel built by setting TEE attribute **Compiler Concurrency** to 1 and by clearing the TEE attribute **Make Options**. The parallel execution is set to 10 module tests by default. Too many compiler or debugger processes in parallel may slow down the test execution. It depends on your computer system. So, feel free to test different values for **Compiler Concurrency** and **Target Concurrency**. You have to **Enable Expert Mode** from the **Attributes** view's toolbar to alter these attributes.

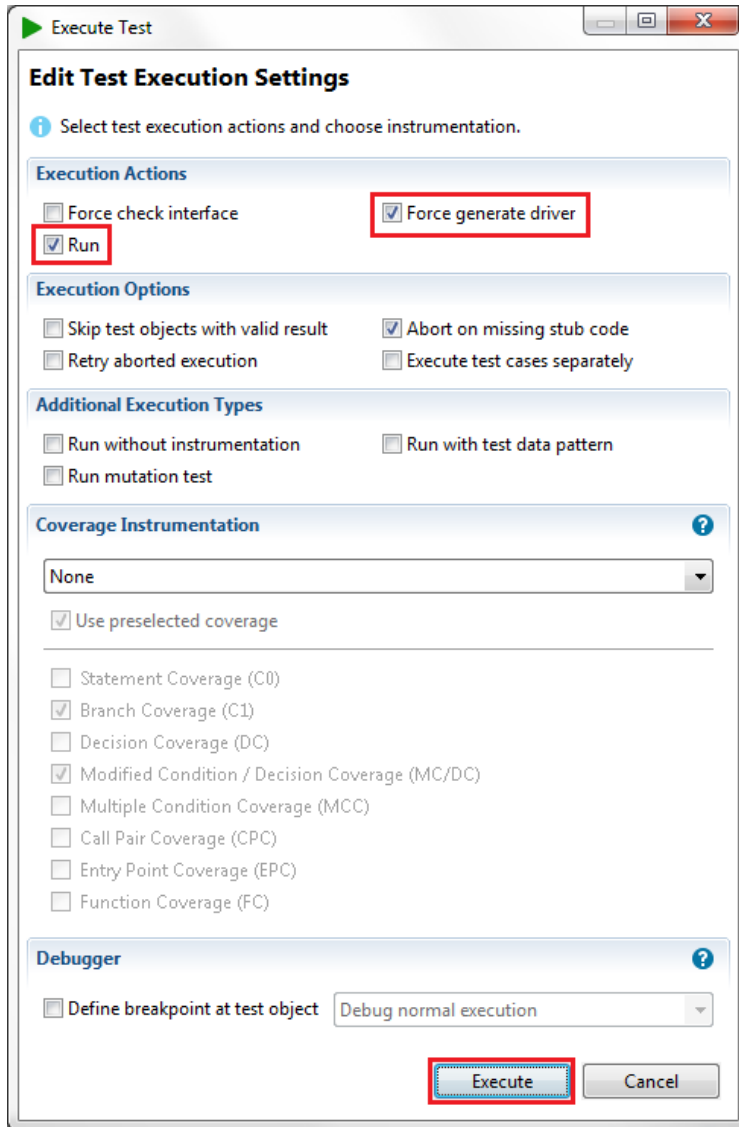
## 4 Interactive Debugging

**Important Note:** You need a **functional** project which can successfully launch a debug session! Set TEE attribute **Project Binary Path** to your SEGGER Embedded Studio project's executable path, which will be loaded into the debugger. **TESSY will replace the project's executable with the generated built-in data executable.** After the interactive debugging session, you will have to rebuilt your SEGGER project before you can debug the original executable again.

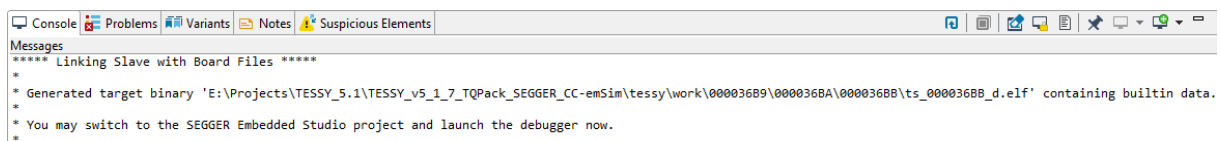
In order to debug the test application interactively with the test case values provided from within TDE you need to rebuild the test application in a special mode, i.e. the input values will be compiled into the application. TESSY provides the TEE attribute **Generate Builtin Data** to facilitate this task. 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.

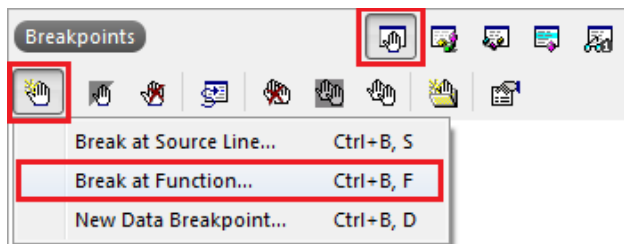


Since a test driver was built containing the test data but a test was not executed, TESSY will complain about it. Please ignore this complaint.

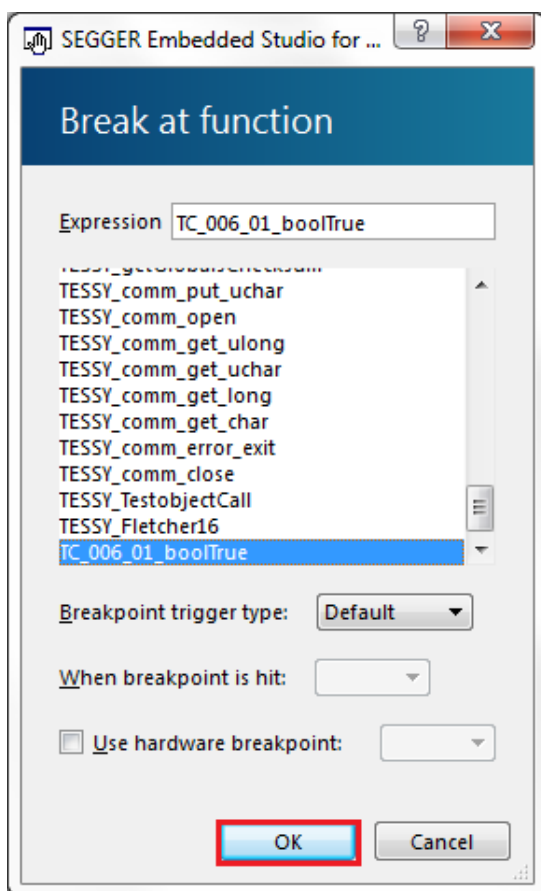
As mentioned in the console output, switch to SEGGER Embedded Studio and launch the debugger.



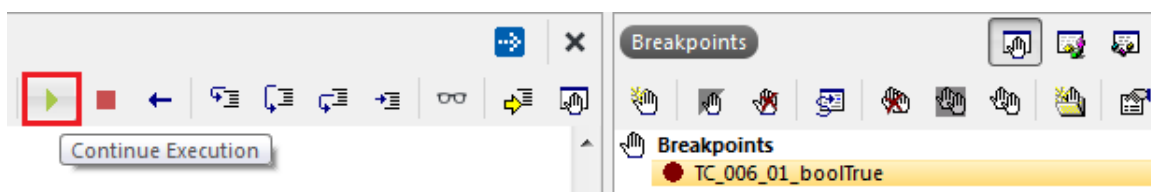
Open the **Break at function** dialog.



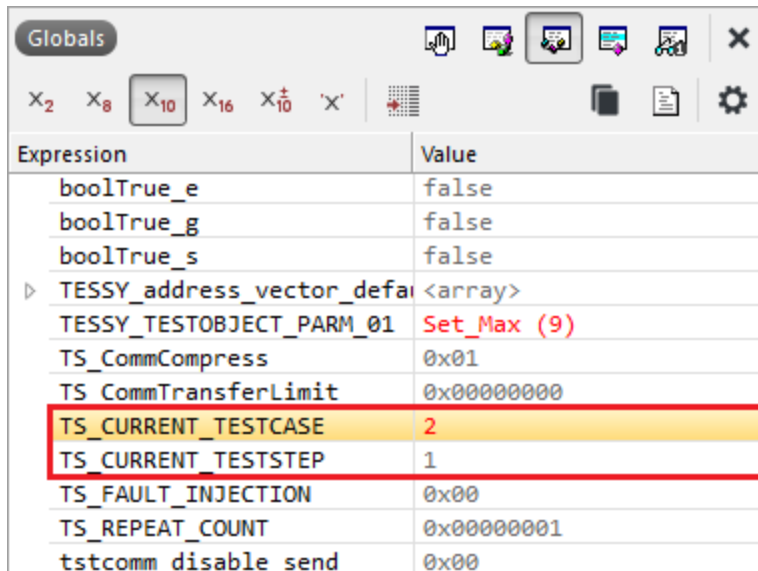
Select your test function from the list, click OK,



and click **Continue Execution**.

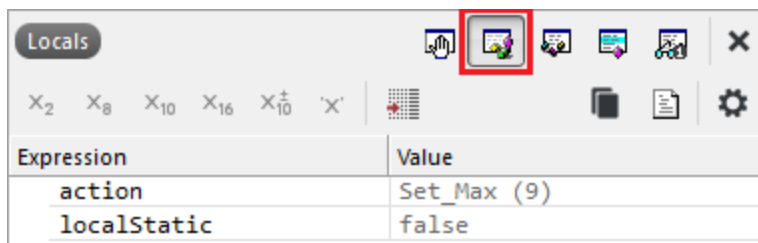


It may be useful to select the **Globals** view of SEGGER Embedded Studio to track the current test case or test step.



Expression	Value
boolTrue_e	false
boolTrue_g	false
boolTrue_s	false
TESSY_address_vector_defa	<array>
TESSY_TESTOBJECT_PARM_01	Set_Max (9)
TS_CommCompress	0x01
TS_CommTransferLimit	0x00000000
TS_CURRENT_TESTCASE	2
TS_CURRENT_TESTSTEP	1
TS_FAULT_INJECTION	0x00
TS_REPEAT_COUNT	0x00000001
tstcomm_disable_send	0x00

Select the **Locals** view of SEGGER Embedded Studio to track the local variables of your test function.



Expression	Value
action	Set_Max (9)
localStatic	false

## 5 Troubleshooting

Please contact [support@razorcat.com](mailto:support@razorcat.com) if you encounter any unsolvable problems.