



Using SuperTest for C Standard Library Qualification in Safety Critical Applications

While SuperTest™ is traditionally positioned as a tool for compiler validation, it is important to understand that the C (and similar for C++) language specification defines both the C language, and the C standard library. **SuperTest validates both!**

The difference between the C language and C standard library is important in the light of safety critical applications. The compiler itself handles the implementation of the language, and is a tool in the production of target code. The library consists of code that is linked to the application and is actually loaded into the device. Both are provided in a single package: **the compiler toolkit**, so from the outside the difference is not so clear.

Functional safety standards such as ISO 26262 for automotive systems, and the more general IEC 61508, treat software tools differently from the code that is running on the target device. In particular, ISO 26262 has a section on the qualification of software tools such as the compiler. Although, using SuperTest, compiler qualification is not rocket science, some application developers still select to avoid it.

Library code is different.

Code from the C standard library is going to end up on the device and therefore it cannot go unqualified. ISO 26262 treats library code the same as other application code that has to run on the target. In fact, it does not mention libraries as a separate category of code at all. This is unfortunate because the construction of the library follows a different path than application code: it is supplied with the compiler toolkit and often the sources of the library are not even available.

There is however a very good specification of the C standard library: it is in the C language specification. And that in turn implies that there is a very good test-suite for the standard library: **SuperTest**.

SuperTest is organized according to its specification: the language standard (unlike open source test suites), and therefore it is ideal for standard library qualification, both for C and C++.



The cleanly structured nature of SuperTest means that for every library test it is easy to verify how it corresponds to the library specification. Also it is easy to verify the completeness of the test-suite because for every paragraph in the language standard there is a corresponding location for the test. This traceability from the test to the specification is an important requirement for safety critical testing.

SuperTest is easy to set up and easy to use, on any host platform.

SuperTest can be used for on-host (simulator) and on-target testing.

SuperTest can work with any compiler system and any target, independent of its data model.

SuperTest has been used in numerous functional safety qualification projects before.

Standard library validation has never been easier.

Contact Solid Sands for more information.

(c) Copyright 2017 by Solid Sands B.V., Amsterdam, the Netherlands
SuperTest™ is a trademark of Solid Sands B.V., Amsterdam, The Netherlands.