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 it is the tool that generates the 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 Software Development Kit (SDK)**, 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. That process is well-defined and with the SuperTest validation suite it is within reach of every safety-critical product development.

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 provides two routes to library qualification. Under Part 6, library code is treated the same as other application code that has to run on the target. In fact, that part does not mention libraries as a separate category of code at all, which may be the reason that the qualification of the standard library is often overlooked. Part 8 Section 12 of ISO 26262 is about the "Qualification of software components." This section does specifically mention libraries as pre-existing software components and can be applied to Commercial Of The Shelf (COTS) software such as the library that comes with the SDK.

As critical step in both Part 6 and Part 8 is the validation of the library with a test suite that matches the requirements of the library definition.

One of the best reasons for the popularity of C and C++ is that there exist ISO language specifications for these languages. This is not true for many alternatives programming languages. These specifications have a long history and are well understood. They include also the definition of the standard library. Thanks to this we also have a very good test suite for the C and C++ standard libraries: **SuperTest.**

SuperTest has been more than 35 years in the making. Its development started when the first contours of the ANSI-C definition started to appear. 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++, and for all versions.

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.