LibmCS: Mathematical Library for Critical Systems Standards compliant and pre-qualified to ECSS Category B

Fabian Schriever, Thomas Wucher, Christoph Weiß, Joan Roig, and Andoni Arregi



## Why Your Important Project Needs the LibmCS

Mathematical libraries are used in nearly any flight software, in particular in the AOCS/GNC and scientific algorithms. As computing power increases and more calculations are done on OBCs and payloads, well characterized mathematical functions become more and more important. Previously used libraries might have flight heritage but do have known bugs, which will show up when doing more calculations, and are not qualified to any standard, thus having no sensible, standardized test-set and documentation. This may lead to several issues when integrated into application software, e.g:

not easily detectable accuracy or timing issues, e.g. due to compiler issues/misconfiguration

# The Test-Suite (MAXI Toolbox)

The provided Test-Suite enables through its GUI:

- the assessment (numerical and timing behavior) of the mathematical procedures and
- their fully automated qualification on target through GDB
- including generation of Qualification Reports
- ► if necessary, interactive analysis of test results with Jupyter Notebooks
- ► and, easily extensible for custom mathematical functions

MAXI Orchestrator			MAXI Orchestrator		
Configuration About this Software			Configuration Run Evaluate Ad-hoc test About this Software		
Load configuration preset:	Linux on localhost with libmcs	Switch to Expert Mode	Coverage evaluation		
General settings					
Workspace Directory:	workspace	Load config from Workspace	Generate GCOV coverage report       Open in default browser		
Specification Directory:	test-specification		Accuracy evaluation		
Runner settings					
Runner Make target:	linux	List possible runner targets	timing_tests/mathd/acos/acos-acos-pos	Select all	
Extra CFLAGS for runner:	-mlong-double-64		timing_tests/mathd/acos/acos-acos-zoom timing_tests/mathd/acin/acin-acin-acin-acin-	Select none	
Mathlib settings			timing_tests/mathd/asin/asin/asin/asin/asin/asin/asin/asin		
Mathlib Source Directory (libmcs root):	/libmcs/		timing_tests/mathd/asin/asin-asin-zoom	Update test list	
Path to static libm binary:	/libmcs/build-x86_64-pc-linux-gnu/bin/libm.a		timing_tests/mathd/atan/atan-atan-neg		

application software developers might not be aware of known issues in mathematical functions, because of missing documentation

Both issues can lead to **mission-critical errors**.

### **LibmCS** Main Attributes

The Mathematical Library for Critical Systems (LibmCS) provides an open source, standard compliant (IEEE-754, ISO C18, and POSIX) mathematical library (libm), pre-qualified to ECSS E-ST-40 and Q-ST-80 Category B, so that project specific integration and delta-qualification efforts for this building block are minimized and its long term maintenance guaranteed.

- ► Standard compliance (IEEE-754-2019, ISO C18, POSIX, MISRA C:2012)
- ECSS Category B qualification evidence including ISVV
- Designed to support many processor architectures: x86-64, SPARC V8 (all LEON), ARM, RISC-V...
- ► Fully compatible with qualified RTEMS 6 SMP and EDISOFT RTEMS 4.8
- Provides step by step Qualification Guideline and qualification template
- Minimized project specific delta-qualification effort (below 2 days of test execution)
- ► Freely available (Qualification Kit also free for ESA Missions)

Optional settings for cross-compilation Architecture Name (defined by GCC if empty): x8 Toolchain Prefix (host GCC if empty): Optional settings for debugging GDB target (direct, local execution if empty): Debug monitor command (no monitor if empty):	Enable DAZ and FTZ for GRFPUv4 and earlier Patch libmcs for 100% unit test coverage 6_64-pc-linux-gnu Use gdb-multiarch instead of toolchain provided GDB		<pre>timing_tests/mathd/atan/atan-atan-zoom timing_tests/mathd/atan2/atan2-atan2-neg_neg timing_tests/mathd/atan2/atan2-atan2-neg_pos timing_tests/mathd/atan2/atan2-atan2-pos_neg timing_tests/mathd/atan2/atan2-atan2-zoom timing_tests/mathd/ceil/ceil-ceil-neg timing_tests/mathd/ceil/ceil-ceil-neg timing_tests/mathd/copysign/copysign-copys-neg_neg timing_tests/mathd/copysign/copysign-copys-neg_neg timing_tests/mathd/copysign/copysign-copys-pos_neg timing_tests/mathd/copysign/copysign-copys-pos_neg timing_tests/mathd/copysign/copysign-copys-pos_neg timing_tests/mathd/copysign/copysign-copys-pos_neg timing_tests/mathd/copysign/copysign-copys-zoom timing_tests/mathd/cos/cos-cos-neg timing_tests/mathd/cos/cos-cos-pos timing_tests/mathd/cos/cos-cos-pos</pre>		Delete selected
Validate and apply configuration	rly braces {} are replaced by the runner file name           Run Unit Tests         Run Validation Tests	Run Timing Tests			
			timing_tests/mathd/exp/exp-exp-zoom timing_tests/mathd/fabs/fabs-fabs-neg timing_tests/mathd/fabs/fabs-fabs-pos timing_tests/mathd/floor/floor-floor-neg	Step 2	
Qualification Test Status: Idle	167003-maxi log):	Clear Command Log	Recreate up-to-date test reports	Show workspace results evention	
Timing Test Specification is available: test-specification/timing_tests exists Timing Test Specification for complex float is available: test-specification/timing_tests/complexd exists Timing Test Specification for complex double is available: test-specification/timing_tests/complexd exists Timing Test Specification for float is available: test-specification/timing_tests/mathf exists Timing Test Specification for double is available: test-specification/timing_tests/mathf exists Timing Test Specification for long double is available: test-specification/timing_tests/mathf exists Build Information is available:/libmcs/build-x86_64-pc-linux-gnu/build_info.yml exists User Makefile is available:/libmcs/build-x86_64-pc-linux-gnu/bin/libm.a exists Header file math.h is available:/libmcs/libm/include/math.h exists yaml is available: Yes yamale is available: Yes numpy is available: Yes nbformat is available: Yes The configuration is looking good!				Create HTML report for workspace	
			<pre>Clear Command Log Command Log (abbreviated for perfomance reasons, see maxilog for all messages): [W 12:48:50.34b NOTEDOOKAPD] NOTEDOOK MAX1.1pynD 1s not trusted [I 12:48:50.780 NotebookApD] Kernel started: 8820a662-dabb-4e2b-9c71-8b6addd6d021, name: python3 [I 12:49:06.633 NotebookApD] Saving file at /maxi.ipynb II 12:49:06.631 NotebookApD] Starting for 8820a662-dabb-4e2b-9c71-8b6addd6d021:cf48f1acb99b4c248dc3474840fc6363 [I 12:49:11.926 NotebookApD] Kernel shutdown: 8820a662-dabb-4e2b-9c71-8b6addd6d021 Process has exited . \$ jupyter notebook workspace/reports/maxi.ipynb or http://127.0.0.1:8888/7token=52bd18457c654476a5ecb7782ac4ac0e3e9588eb51f66ec5 [W 12:49:16.629 NotebookApp] Notebook maxi.ipynb is not trusted</pre>		

# **The Test Specification**

- Over 4.000 Unit-Tests achieving full statement, decision, and even 100% MC/DC coverage
- Over 1.000 single value Validation-Tests
- Over 250.000.000 floating-point accuracy tests

#### **Qualification Reports**

The Test-Suite creates a detailed Qualification Report including:

- Much extended functionality and improved test-suite over the precursor MLFS library
- Improved accuracy compared to the Newlib libm
- Clearly characterized accuracy and execution time behavior

#### Provided math.h Procedures in 32 bit and 64 bit Precision

- Classification macros: fpclassify, isfinite, isinf, isnan, isnormal, signbit
- Trigonometric functions: acos, asin, atan, atan2, cos, sin, tan
- Hyperbolic functions: acosh, asinh, atanh, cosh, sinh, tanh
   Exponential and logarithmic functions: exp, exp2, expm1, frexp, ilogb, ldexp, log, log10, log1p, log2, logb, modf, scalbn, scalbln
- Power and absolute-value functions: cbrt, fabs, hypot, pow, sqrt
- Error and gamma functions: erf, erfc, lgamma, tgamma
- Nearest integer functions: ceil, floor, nearbyint, rint, lrint, llrint, round, lround, llround, trunc
- Remainder functions: fmod, remainder, remquo
- Manipulation functions: copysign, nan, nextafter, nexttoward

- Test Environment Meta Data
- Numerical results overview
- Accuracy and Timing Plots
- Source Code coverage on target CPU using GCov



#### **Provided Data Package**

- The complete Qualification-Kit includes:
- Full ECSS engineering documentation in compliance with E-ST-40 and Q-ST-80
- Maximum, minimum, and positive difference functions: fdim, fmax, fmin
- Fused multiply-add: fma
- Comparison macros: isgreater, isgreaterequal, isless, islessequal, islessgreater, isunordered
- Bessel functions (only in 64 bits): j0, j1, jn, y0, y1, yn
  - In addition all required standard complex functions of complex.h are provided for integration purposes.

# Acknowledgements

Development of this project has been funded by the European Space Agency under contract 4000130278/20/NL/AS

- Full qualification evidence on x86-64, LEON2 (AT697), and LEON4 (N2X) platforms
- Qualification Guideline and qualification template

## **Availability and Contact**

- ESA ESSR: Attps://essr.esa.int/project/ libmcs-mathematical-library-for-critical-systems
- LibmCS on GitLab:
   https://gitlab.com/gtd-gmbh/libmcs
   GTD GmbH Information and Support:
   https://gtd-gmbh.de/libmcs
   libmcs@gtd-gmbh.de



libmcs@gtd-gmbh.de

http://www.gtd-gmbh.de