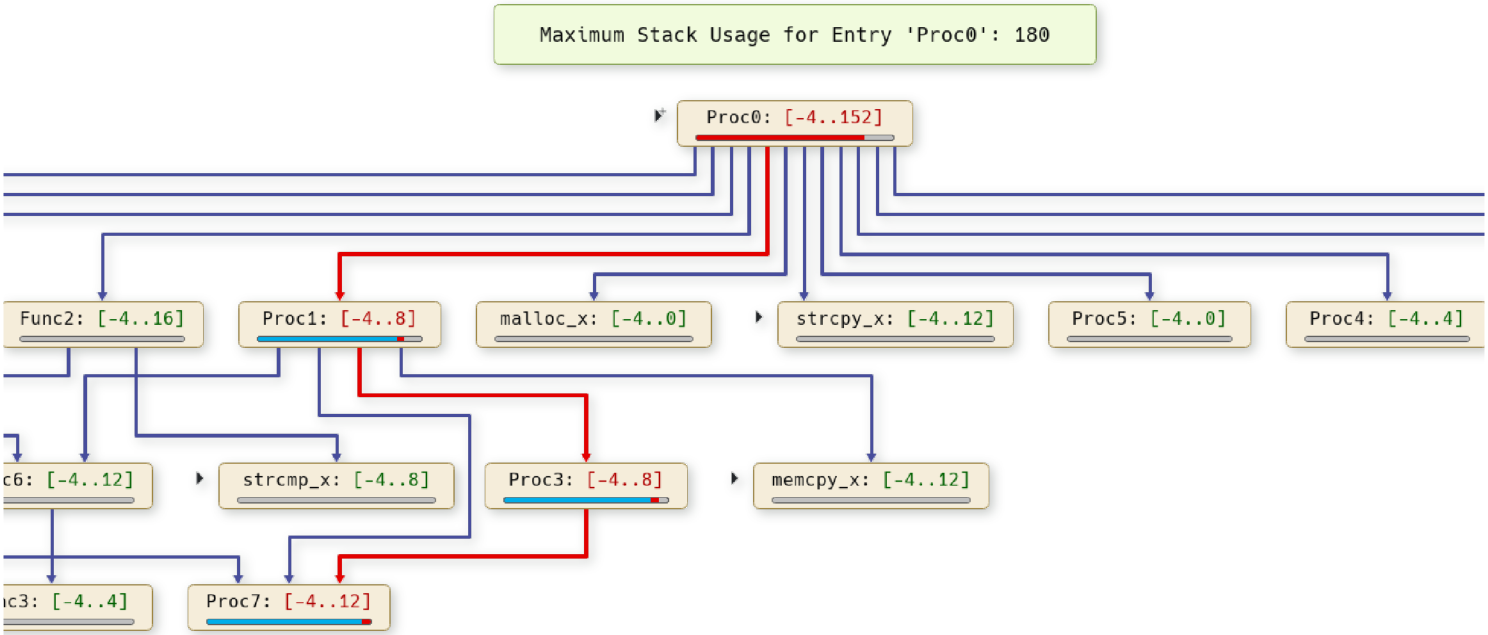
Static stack-usage analysis for C28x

StackAnalyzer for C28x statically determines the worst-case stack usage of tasks in safety-critical applications written in C or C++ and compiled to run on TMS320F28x processors.



Maximum Stack Usage for Entry 'Proc0': 180

Your benefits

- StackAnalyzer lets you prevent stack overflow for all possible inputs and task executions under any circumstances — without wasting hardware resources.
- StackAnalyzer requires no code instrumentation, no testing, no measuring, no modification of your system, no modification of your toolchain, and it will not be misled by potential flaws in debug information.
- Using StackAnalyzer is essential in meeting current safety standards such as ISO 26262, DO-178B/C, IEC-61508, and EN-50128, where statically analyzing your stack usage is part of the architectural safety requirements.

Key features

- Static analysis of binary files, exactly as they are executed in the final system.

- Clear and precise information on the worst-case stack usage by tasks, procedures, basic blocks, and individual instructions.
- Recursions, function pointers, inline assembly code, and library-function calls are all taken into account.
- Automatic recognition of dead code.
- Exceptionally fast analysis of complex real-world software.
- Freely selectable entry points for the analysis, so you can focus on the worst-case path or other areas of interest, and speed up the analysis even further.
- Difference analysis for comparing the stack usage before and after making changes to your system.
- Customizable XML reports for documentation and certification.
- Feature-rich GUI with graphical and textual views for analysis results, statistics, control flow, source code, assembly code, DWARF debug info, symbol tables, configuration files, and more.
- Command-line mode for easy integration into automated build processes.
- Plugins for TargetLink and Jenkins.
- Seamless integration with other analysis tools from AbsInt — e.g. aiT for static worst-case execution time analysis.

Supported processor derivates

- Generic TMS320F28x
- TMS320F2801x
- TMS320F2802
- TMS320F2806
- TMS320F2808
- TMS320F2809
- TMS320F2810
- TMS320F2811
- TMS320F2812
- TMS320F28035
- TMS320F28335
- TMS320F28069
- TMS320F28377D

All license variations always cover all of the above.

Supported compilers

- Texas Instruments C/C++ compiler

Also available for C28x

- Static WCET analysis
- Timing profiling
- Memory safety analysis

Qualification support

Your usage of StackAnalyzer for C28x can be qualified according to ISO 26262, DO-178B/C, and other safety standards. We offer Qualification Support Kits that help you simplify and automate your qualification process:

- Base QSK
- Optional compiler-specific add-on QSKs for
  - TI 6.2.0
  - TI 18.2.3.lts

Additional compiler-specific QSKs can be developed on request.

Recent improvements

21.04 21.10 23.04 23.10 24.10 25.04

- Support for C++ virtual member function calls
- Support for the instructions `PREDIVF64`, `POSTDIVF64`, and `SUBC3F64`
- Improved automatic switch table decoding for the TI compiler

- See the complete release notes

System requirements

- Windows: 64-bit Windows 10 or 11
- Linux: 64-bit RHEL 9 or compatible
- 4 GB of RAM (16 GB recommended)
- 4 GB of disk space
- The Linux version requires the `libxcb-*` family of libraries to be installed
- Support for macOS is possible on request for a surcharge

Download
this factsheet
Download
StackAnalyzer flyer

Free trial

You can try StackAnalyzer for free, on your own applications, for a period of 30 days. Your free-trial package includes online training and tech support.

Get started today