Using R**VS** with iSYSTEM® trace-enabled debuggers

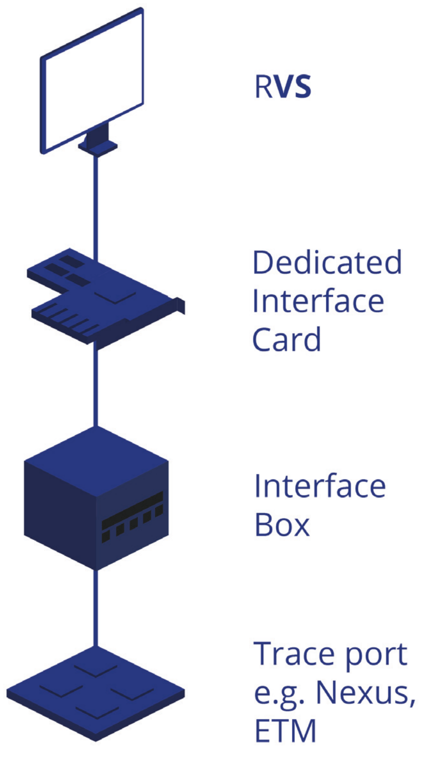
Rapita **Verification Suite** (R**VS**) tools are for software engineers who conduct on-target verification, optimization and code coverage analysis for critical real-time embedded systems. R**VS** consists of dedicated on-target aerospace and automotive optimization and verification products:

- Rapi**Time** – on-target timing analysis, verification and optimization;
- Rapi**Cover** – on-target code coverage analysis.

Rapi**Time** is an automated performance measurement and timing analysis tool. Targeted at real-time, embedded applications, Rapi**Time** measures performance, determines worst-case execution times, and guides your optimization efforts.

Rapi**Cover** is an on-target code coverage measurement tool that supports system and integration testing processes. Rapi**Cover** maps code coverage to specific test cases, involves minimal instrumentation, supports justification of missing coverage and provides an adaptable approach to on-target code coverage measurement.

The R**VS** toolsuite provides a streamlined and cost-effective way to conduct on-target timing verification, optimization and code coverage analysis.



*R**VS** data collection using iSYSTEM trace-enabled debuggers*

Technical Details

Tool requirements:

- iSYSTEM iC3000GT and iTRACEGT or iSYSTEM iC5000

Target requirements:

- CPU with support for:
  - NEXUS Class 3 (e.g. Freescale PowerPC®, National Semiconductor® CRX)
  - ETM data trace (e.g. ARM® Cortex®-A)

R**VS** and iSYSTEM® debuggers

Rapi**Time** obtains results by examining the timing of instrumentation points inserted into the application's source code. An efficient integration of Rapi**Time** can be achieved using the iSYSTEM ic3000GT with the iTRACE GT as the mechanism for collecting the trace of instrumentation points.

Using iSYSTEM debuggers for trace capture

Using a trace port (NEXUS or ARM-ETM), the iSYSTEM iC3000GT/iTRACE GT or iSYSTEM iC5000 can collect traces by monitoring writes made to a specific register or memory location. As each write is made, the debugger timestamps the value. Consequently, the implementation of an instrumentation point consists of writing a constant value (the Ipoint ID) to a specific location or register. Typically, this can be implemented in a single machine instruction, as so is inherently thread-safe.

The R**VS** user documentation gives examples of how to implement a suitable instrumentation point routine as a macro or inline function.

Configuring iSYSTEM debuggers for trace capture

Configuration of the iSYSTEM tools for timing trace acquisition requires the following steps:

- **Hardware connection:** the iSYSTEM debugger needs to be attached to the target board's trace port using the provided connector.
- **winIDEA Configuration:** The iSYSTEM IDE (winIDEA) needs to be configured to capture writes of the Ipoint

ID to the allocated variable or register. Setting a value for "Deep Trace File Size" will cause the trace data to be spooled directly to disk.

When the source code has been instrumented, compiled, and linked, it can be downloaded onto the target ready for testing. A trace of the software's timing behavior can now be obtained by running a series of tests on the target and capturing the trace data using winIDEA. Trace collection ends either when the target CPU is manually stopped or the trace collection is stopped and the CPU left running.

Once trace data has been captured, it needs to be exported by winIDEA to the host for processing. This is done by performing a binary export selecting only timestamp and data fields.

The trace file then needs to be pre-processed using Rapi**Time**'s **`traceutils`** utility to convert it into the Rapi**Time** native format. This is done using the `byte_reader` filter to unpack the exported binary data. The filter to use depends on the endian-ness of the processor.

For big-endian devices such as PowerPCs, use:

```
byte_reader( B8 B9 B10 B11, B7 B6 B5 B4 B3
B2 B1 B0 );
```

For little-endian devices like the ARM use:

```
byte_reader( B11 B10 B9 B8, B7 B6 B5 B4
B3 B2 B1 B0 );
```

Summary

iSYSTEM tools provide a simple and effective means of capturing timing trace data. This solution minimizes measurement overheads by supporting minimal instrumentation points (typically a single instruction) via the use of external time-stamping. To find out more or to arrange a demo, contact info@rapitasystems.com or visit rapitasystems.com.

About Rapita Systems

Rapita Systems provides customized on-target verification solutions which reduce the cost of measuring and optimizing the timing performance of large real-time software systems in the avionics and automotive electronics markets.

Rapita **Verification Suite** (R**VS**), which includes Rapi**Time** and Rapi**Cover**, is the essential collection of on-target timing verification, optimization and code coverage measurement tools for real-time embedded systems. It is the only product on the market that can tell users exactly where to focus optimization effort to minimize worst-case execution time.

Using R**VS**, customers have cut the worst-case execution time of large scale, legacy applications by up to 50% with only a few days effort, and significantly reduced unnecessary testing and instrumentation overheads.

Rapita's software supports Windows and Linux and Ada, C and C++ projects.

About iSYSTEM

iSYSTEM offers expert knowledge based on more than 23 years of experience with embedded systems. They are pioneers in emulation and debug technology for 8-/16-/32-bit microcontroller architectures and FPGA-based emulator hardware tools.

The modular system provided by iSYSTEM contains in-circuit and on-chip emulation hardware for more than 50 microcontroller families and their derivatives. iSYSTEM tools support all common trace ports such as NEXUS and ARM-ETM allowing conventional debug capabilities to be augmented with a rich trace of timed events leading up to and following key behaviours.