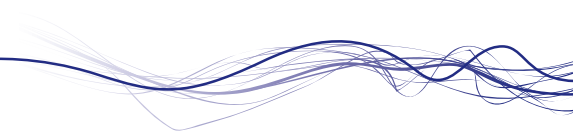




Rapi**Time** Explained



Contents

Summary	3
1. How the world of real-time embedded systems is changing	4
2. What is the Rapi Time solution?	5
Execution time measurement	6
Worst-case calculation	6
Performance optimization	7
Code coverage tracking	8
“Rewind” debugging	8
3. The top ten Rapi Time benefits	9
4. Discover what Rapi Time can do for you	11

Summary

This White Paper explains:

- How the world of critical real-time embedded software systems is changing and what these changes might mean for manufacturers and engineers;
- How, as demands for new and more complex functionality in critical real-time embedded systems increase, the job of testing for correct functional and correct timing behaviour will become ever more difficult;
- And why a major challenge for the next generation of software analysts is to ensure testing and verification procedures keep pace with developments in embedded real-time systems.

top ten RapiTime benefits

1 *Identify timing problems when they are introduced, not at the end of the project.*

1. How the world of real-time embedded systems is changing

The amount of code that is typically included in embedded devices is doubling about every two years, according to a recent Wind River study*. The rate of increase, says the study, “is making it increasingly difficult for QA and test teams to keep up with traditional tools and processes”.

The trend towards advanced architectural complexity, represented by a move from ‘16-bit to 32- and 64-bit architectures, greater utilization of multi-core technologies, ongoing operating systems changes’ and other hardware acceleration features, has long been an accepted part of the embedded world.

One of the major consequences of this trend is that it makes the job of testing for correct functional and correct timing properties ever more difficult because the time and effort required for verification increases as the functionality provided becomes more complex.

As a result there is a growing demand for engineering approaches and tools that can be used to detect potential timing problems during the early stages of product development.

The demand for increased functionality is most obvious in the avionics and automotive electronics industries. It is here that we find significantly more pressure on manufacturers to develop more complex systems in less time and at lower cost without compromising their reputations for quality and reliability.

Some of the predicted demands on future platforms in, for example, the avionics industry include:

- the increased use of automatic dependent surveillance-broadcast (ADS-B) systems, which allow aircraft to determine their own position using a global navigation satellite system and to periodically broadcast this position and other relevant information to potential ground stations and other aircraft equipped with ADS-B;
- “pilot optional” aircraft designed to bridge the gap between UAVs (Unmanned Autonomous Vehicles) and conventional aircraft;
- the demand for diagnostic/prognostic systems that can monitor aircraft components in flight and determine their “health”;
- the need to reduce carbon footprint levels through greener engines and lighter composites will require significant changes to existing systems.

Whether it is in the avionics, automotive electronics or other industries, the challenge for the next generation of engineers is to ensure testing and verification procedures keep pace with developments in embedded real-time systems.

* The Crisis of Complexity, A Wind River Market Survey of Device Software Testing Trends and Quality Concerns in the Embedded Industry, June 2010

top ten RapiTime benefits

2 Reduce the effort required to improve timing performance.

2. What is the RapiTime solution?

RapiTime is an automated performance measurement on-target timing analysis tool. Aimed at real-time, embedded applications, RapiTime collects execution traces to provide you with *code coverage metrics* and *execution time measurement statistics*. RapiTime also analyses worst-case execution time and guides your optimization efforts.



RapiTime reduces the cost and effort required to conduct timing verification, optimize software, update legacy systems and integrate critical real-time embedded systems. In this section we introduce the cornerstones of RapiTime before explaining how they work in greater detail on pages 6 and 7.

Execution Time Measurement

If you want in-depth measurements at a far quicker rate than normal and with substantially less effort, RapiTime automates execution time measurement and rapidly highlights potential problem areas.

Worst-Case Calculation

Go beyond conventional code profiling techniques which only identify the lines of code that execute the most on *average*. RapiTime quickly identifies worst-case hotspots from the point of view of their contribution to the overall worst-case execution time.

Performance Optimization

When you need to reduce worst-case execution times but aren't sure where to look, RapiTime identifies worst-case hot spots and helps you to direct your optimization efforts to where they will have the greatest benefit.

Additional Features

RapiTime also includes the collection of on-target **code coverage** – which gives you an accurate picture of the parts of the source code investigated during testing – and **debugging support** through the Rewind facility, which gives users the ability to move backwards and forwards – “Rewind” – through source code.

top ten RapiTime benefits

3 As well as identifying targets for optimization, RapiTime enables engineers to answer “what-if” questions.

top ten RapiTime benefits

4 Minimize system integration pain.

top ten RapiTime benefits

5 Avoid or postpone unnecessary, costly upgrades for legacy systems.

Here are the cornerstones of RapiTime :

Execution time measurement

Our on-target profiling software automates execution time measurement, highlights potential problems, and accurately determines how long real-time software takes to run. This gives you in-depth measurements more quickly and with substantially less effort.

Worst-case calculation

RapiTime accurately determines the worst-case execution time for deadline-critical applications. Unlike measuring high water marks, which tells us what is the longest execution we've observed, worst-case calculation tells us what is the longest execution that could take place (see the dotted line in Fig 3).

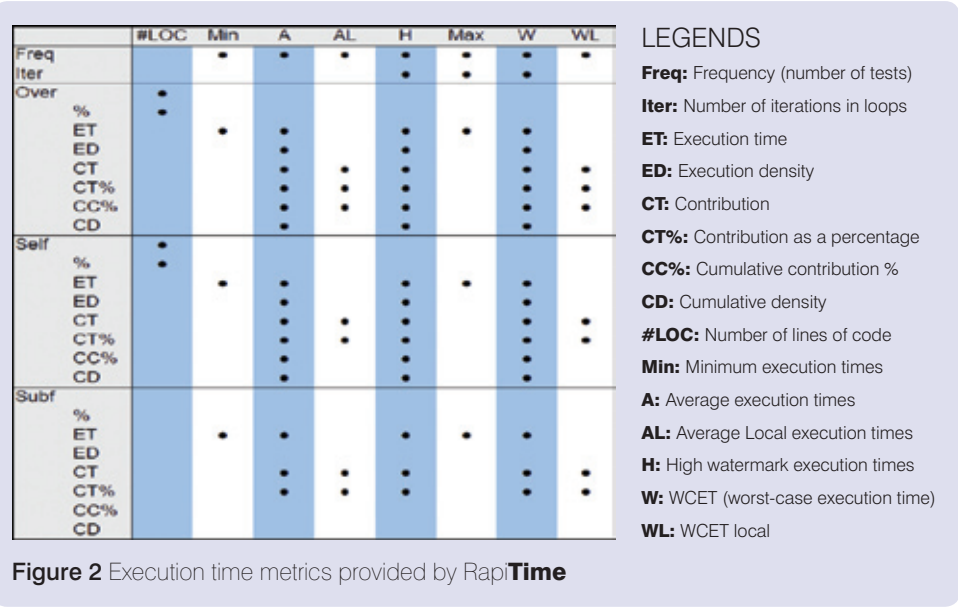


Figure 2 Execution time metrics provided by RapiTime



“Not only did we reduce our effort requirements for the testing, but we could use our results in ways that were infeasible before. It is now significantly faster for us to identify a timing issue, update the software to resolve the issue, test the updated software and verify that it’s fixed”

WAYNE KING, ENGINEERING FELLOW

Performance optimization

Optimizing code with RapiTime can mean less effort spent on optimizing the wrong code. First we collect trace execution data and look for suitable candidates for optimization. This will include code which takes unusually long or short times to run (worst-case hot spots).

Once we know which code makes the biggest contribution to the worst-case execution time, performance optimization can begin. Removing or amending code can dramatically reduce worst-case execution times on large real-time software systems, as we demonstrated in our work on the Hawk Mission Computer Operational Flight Program for BAE Systems.

Asked to reduce the overall execution time of the software, RapiTime used RapiTime to first identify and then optimise the appropriate code. This work achieved a 23% reduction in the worst-case execution time of the analysed partitions with less than 10% of the effort of previous approaches.

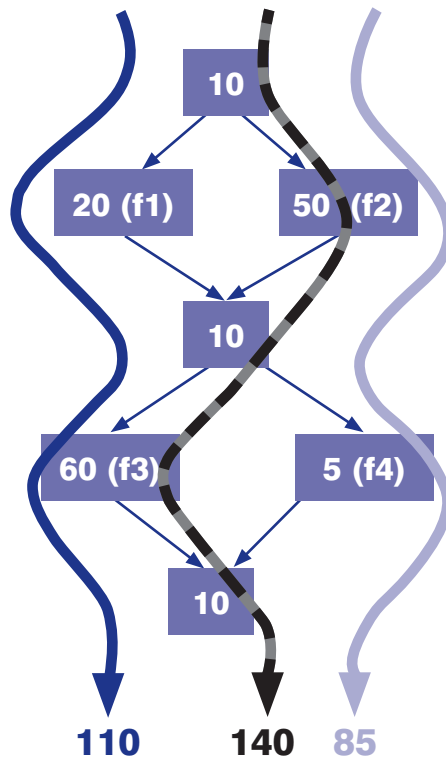


Figure 3 Determining WCET from measurements and structure

Example in Fig 3 is two “if” statements:

Two test cases are performed. One takes 110ms, the other 85ms. The longest observed execution is 110ms.

However, from the information generated by these tests, RapiTime determines that the worst-case execution time (WCET) is actually 140ms (see dotted line).

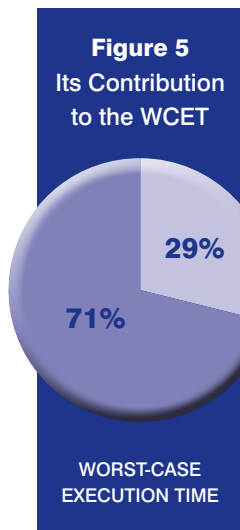
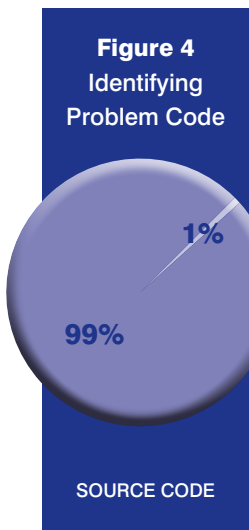
top ten RapiTime benefits

6 Ensure that your timing tests are complete.

Using a typical example, Figures 4 and 5 illustrate a common outcome achieved by optimizing code with RapiTime.

In our example, 1% of the source code has been identified as “a worst-case hot spot”. Our analysis shows that this 1% is actually responsible for 29% of the worst-case execution time.

The 1% is a worst-case hot spot. Quickly identifying and optimizing code in hot spot areas dramatically reduces the time and effort usually required to address performance issues.



“We are delighted with the schedule reductions achieved in phase 1”,

SAID DEAN ARMSTRONG, SOFTWARE ENGINEERING MANAGER AT BAE SYSTEMS BROUGH.

“The integration team are now looking at how RapiTime can be used as an integral part of the software development process for future Hawk developments”.

top ten RapiTime benefits

7 Move backwards through code to highlight problems and take appropriate action with “Rewind”.

Code coverage tracking

Using RapiTime can significantly improve your code coverage analysis and reporting. It will give you a precise percentage of the code measured during tests.

You can also learn about the number of times each function/block has been executed and how many times each has been called (see Figure 1 on the previous page).

“Rewind” debugging

Embedded developers often find themselves in a situation where the target just stops working. Stepping backwards from the point where the application stopped to see the sequence of events leading up to the problem helps developers to identify the cause.

Our solution is a specialised debug support facility which gives users the ability to move backwards and forwards

through source code. Then the task of examining why specific runs take unusually short or long amounts of time can begin.

To summarise, in general terms RapiTime works as follows:

- Instrumentation code which highlights the execution of a specific section of code is automatically added during your build process;
- As the application is executed on the target, an execution trace is collected. This trace is a sequence of time-stamped values that show when the instrumentation code is executed;
- Using the trace RapiTime identifies the parts of the source code which have and have not been executed (code coverage), and performance metrics for each part of executed code;
- Rewind allows you to move backward and forward through source code.

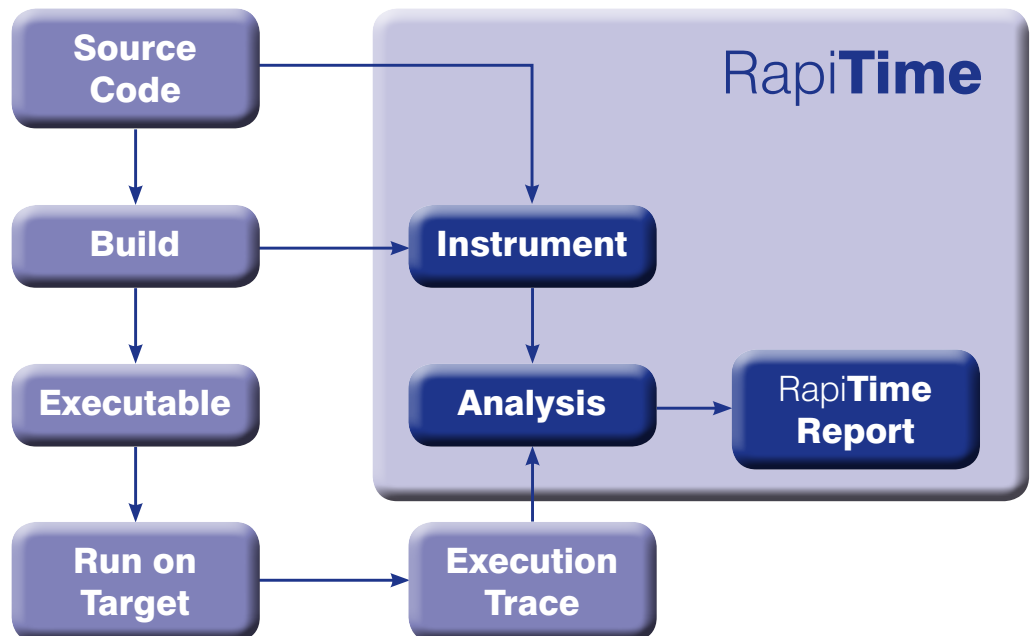


Figure 6 How RapiTime works

3. The top ten Rapi**Time** benefits

Now we know how Rapi**Time** works, let's look at the top ten Rapi**Time** benefits.

1 Identify timing problems when they are introduced, not at the end of the project.

Once Rapi**Time** is integrated into your development chain, the effort required to measure the performance of your software is close to the cost of running a test. This means that timing measurements can be made throughout the development process.

2 Reduce the effort required to improve timing performance.

When you need to reduce worst-case execution times but aren't sure where to look, Rapi**Time** identifies worst-case hot spots, meaning you can direct your optimization efforts accordingly.

3 As well as identifying targets for optimization, Rapi**Time** enables engineers to answer "what-if" questions.

The rapid modify-build-measure cycle that Rapi**Time** supports will allow engineers to quickly try alternative solutions and to assess their impact on software timing.

4 Minimize system integration pain.

Integrating multiple software modules onto a single target can cause difficult to pinpoint performance problems to emerge. Rapi**Time** allows the exact performance cost of each module to be measured, avoiding the problem of overruns occurring at integration time.

5 Avoid or postpone unnecessary, costly upgrades for legacy systems.

Large embedded applications can require significant functionality enhancements without corresponding hardware updates. Rapi**Time** can help you to make the optimizations necessary to add new features without exceeding peak CPU load requirements.

6 Ensure that your timing tests are complete.

Using Rapita's on-target code coverage allows identification of the parts of the source code that have been exercised during testing. This tells you whether you have a complete picture of the timing measurements, or whether some parts have not been measured.

top ten Rapi**Time** benefits

8 Re-use the same approach across all your embedded platforms.

3. The top ten Rapi**Time** benefits, *continued*

top ten Rapi**Time** benefits

9 Deploy systems with the confidence that worst-case execution time is bounded.

7 Move backwards through code to highlight problems and take appropriate action

with “Rewind”. Identifying the sequence of events which led up to the problem helps developers to identify the cause and to take the appropriate action to remedy the situation.

8 Re-use the same approach across all your embedded platforms. Rapi**Time** is

not restricted to work with particular processors, compilers or debuggers. This means that it is now possible to use a single technology to perform timing analysis across all of your projects.

9 Deploy systems with the confidence that worst-case execution time is bounded.

Rapi**Time**'s ability to determine worst-case execution time based upon measurements of your target gives you evidence that your system will meet its performance targets.

10 Take advantage of excellent customer support services.

Rapita provides expert telephone / on-site / email support for anyone using Rapi**Time**. Our standard package gives you access to an experienced Field Engineer who will discuss any issues you have when installing or using Rapi**Time**, providing valuable advice and guidance. Alternatively, you can arrange for a site visit which is charged at our typical commercial rate.



4. Discover what RapiTime can do for you

RapiTime addresses one of the core issues at the heart of reliable real-time embedded software development on advanced microprocessors: the need to understand the execution time performance of your various software components.



By providing this information, RapiTime enables you to use a systematic and scientific approach to ensuring that time constraints are met, allowing you to engineer timing correctness into the system rather than spending a great deal of time and effort trying to get timing bugs out.

And for embedded technology manufacturers, the potential return on investment from utilizing the technology that RapiTime brings is huge.

This toolset offers the prospect of far fewer timing bugs going undetected through unit test and integration phases.

Identifying timing issues early in development has the proven impact of reducing development cost, reducing time to market and enhancing the company's reputation through higher quality and more reliable products.

Try it for yourself. Visit www.rapitasystems.com/evaluation and follow the link to request the RapiTime Trial Version.

top ten RapiTime benefits

10 Take advantage of excellent customer support services.



IT Centre
York Science Park
Heslington
York YO10 5NP
United Kingdom

Tel No: +44 (0)1904 567747
Fax No: +44 (0) 1904 567719
Email: enquiries@rapitasystems.com
Website: www.rapitasystems.com

Registered in England & Wales
Company Number 5011090