By Nigel Street,
Project Leader,
Wind River Systems.

Meeting the Challenge of Developing Today's Real-Time Embedded Applications

The embedded software designers are faced with increasingly complex systems and reducing development schedules. Traditional development tools based on source-level debuggers don't allow developers to debug at higher levels of abstraction and can't deliver the system visualisation required to enhance productivity. A new class of highly graphical, software logic analyser development tools are now emerging. These tools can provide the developer with execution flow visualisation, comprehensive triggering and links for custom data processing and presentation.

The challenge of embedded system development today is to meet the conflicting demands of increasingly complex systems and time-to-market pressure. Fortunately, new software tools now enable designers to reduce development and test times dramatically - even for the most complex real-time applications.

THE CHALLENGE

Development and test schedules can be crucial bottlenecks in bringing today's embedded systems to market. To compound the problem, competitive new designs involve increasingly complex software, which can be difficult to develop and sometimes impossible to troubleshoot and proof-test.

Software engineers have traditionally used combinations of tools such as source-level debuggers, in-circuit emulators combined with their real-time expertise. These staple tools may have some task awareness, but can only provide debugging assistance at the source, assembler and register level.

For complex multi-threaded applications with hard real-time requirements specialised tools are required to provide debugging at the task level so that the development engineer can design and debug in the same paradigm. In addition, the complexity of the execution path requires an effective and productive user interface with a comprehensive graphical presentation to the developer, so as to give maximum visibility of system behaviour.

These requirements have precipitated an entirely new breed of tools that can often help engineers find in minutes the problems that formerly may have taken hours, if not days, to discover.

THE SOFTWARE "LOGIC ANALYSER"

A prime example of the new generation of these visual debugging environments is WindView from Wind River Systems - a tool that is tightly integrated with the Tornado development environment and the VxWorks real-time operating system.

Working in effect as a software logic analyser, WindView captures data on the target, which is then presented graphically to show the interaction of each of the tasks and interrupt service routines over time. Context switches are clearly shown in the display as well as system events such as semaphores, message queues, and signal tasks timers. Developers can also define their own events to be logged and represented by the tool. Using WindView, multi-tasking and inter-process communication can be tracked and measured. Deadlocks and race conditions can be quickly isolated and identified.

In figure 1 the task or ISR, which is executing at any point in time is displayed in green, those which are not executing (ready, pending etc) are shown in other colours. Viewable events are predefined by the real-time operating system or defined by users to trace application specific events. The level of detail of events, which are captured, can be configured before each run.

This allows WindView to be used even where target memory is limited - developers can tune the buffers used to hold the event data, specifying the parameters within which the buffer may dynamically resize.

Time intervals are simple to measure by using the mouse pointer to select the relevant area in one of the display windows. Here, for example, the execution time for ISR of interrupt INT10 is measured as 44 microseconds. The precision of this measurement depends on the timer used, but is typically microseconds. WindView supports this high level of precision with the use of hardware timers, giving an order of magnitude better resolution than could be achieved using the system clock alone.

The WindView GUI has been refined, enabling developers to quickly isolate critical areas of the application. The user can hide selected tasks and interrupts - tasks and events that are not relevant to the analysis can be filtered out. In this way the user can quickly isolate real-time problems, which become readily apparent once the distracting information is suppressed.

Users will benefit from the online help, which is available at any time. The tool features context sensitive help linking to relevant sections of the online manual.

AD WIND RIVER SYSTEMS

ADVANCED DATA COLLECTION

Experienced developers will recognise that the capture of the target event data itself presents a significant problem, particularly given the diverse set of targets being developed upon.

To address this problem, WindView offers flexible data collection options, which are configurable before each operation. Data collection is configurable in three key aspects.

Firstly, the level of event logging that is performed is selectable, from minimal logging, allowing presentation of just context switch information, through to detailed selection of individual VxWorks libraries.

Secondly, the buffer used to hold the event log data while it is stored on the target is reconfigurable.

The event data buffer can be configured via the GUI to provide the optimum configuration dependent upon the characteristics of the target.

The event data buffer space required on the target is dependent upon the data capture rate, memory space available on the target and the rate at which this data can be transferred to the host.

The buffer can be configured by specifying the initial number of buffers, the maximum number of buffers and the size of each buffer. The data is written cyclically to these buffers which are held in a chained ring.

This allows WindView to be used even where target memory is limited - developers can tune the buffers used to hold the event data, specifying the parameters within which the buffer may dynamically resize.

Thirdly, the upload mechanism, which specifies when and how the data is transferred to the host, is also configurable.

Selecting continuous upload mode specifies that event data be uploaded continually as it is logged. Of course, one side effect of this is that the upload activity itself is captured in the log, represented as additional interaction with the host. Alternatively, the upload of the event data may be deferred until event data collection has stopped.

Selecting deferred upload mode is preferred as it gives the clearest representation of the target activity.

The combination of these logging, buffering and upload configuration options ensures that highly constrained targets can configure a viable buffering and upload path, and that larger targets do not dedicate more resources to the tool than is absolutely necessary.

TRIGGERING - A POWERFUL DEVELOPMENT TOOL

In the same way as hardware developers using logic analysers, software developers also need event triggers so they can zoom in quickly on the problem area. The integrated WindView triggering facility allows users to define actions that are performed following specific events or sequences of events.

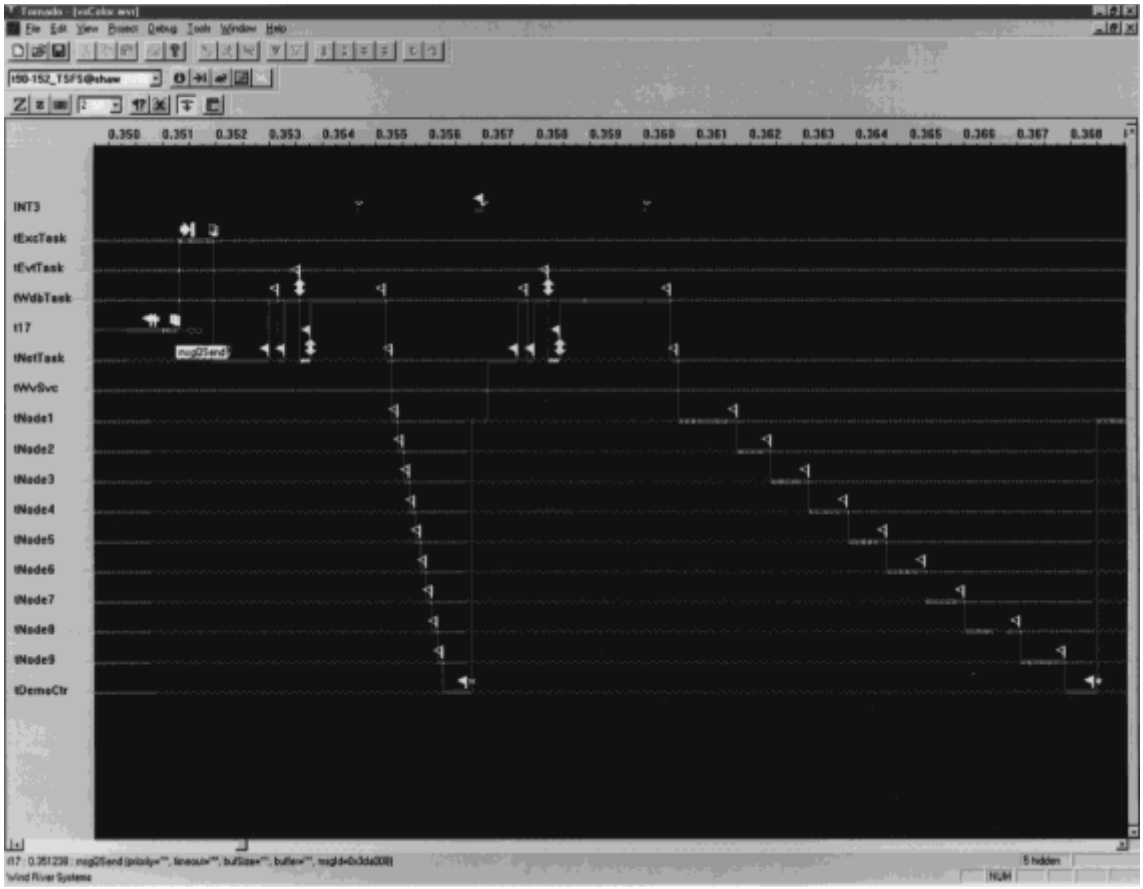

Figure 1. WindView II

Developers can program one or more triggers by specifying the event of interest, the context of the event (e.g. specific task or ISR) and the specific object to which it applies. A trigger may be further defined by optionally making the trigger conditional upon a user specified function or variable that is evaluated against a specific value or range of values.

One powerful technique is to use a trigger to invoke a specified function. This could be a user-supplied function which collects some diagnostic data such as reading hardware device register values, for example.

Triggering to be used to start and/or stop WindView event collection. This gives the developer precise control over the WindView logging: allowing capture of an execution sequence which may only take a few microseconds, but which might not occur for several hours - or even several days.

ANALYSIS OF DATA

Once a log has been captured in the target system's event log and transferred to the host computer it is available for processing and graphical analysis.

The WindView GUI allows users to precisely control the tasks and events displayed - those that are not relevant to the analysis can be filtered out. The tool can give a summary of the state of a particular task or interrupt context with precise timings. Users can choose to have this summary over the entire event log, screen view or a particular selection.

Multiprocessor applications are catered for by enabling multiple event logs to be merged together and viewed simultaneously. This feature can also be used to compare different logs captured from the same target, allowing assessment of some modification, for instance.

Post-mortem mode is extremely useful in those situations where a target problem results in an application failure. Post-mortem mode logs the event data to a region on the target that is preserved during a target reboot. The WindView tool then supports upload of this event log and therefore provides a log of the events leading up to the failure. This is evidence can be invaluable in diagnosing this type of problem.

OPEN ARCHITECTURE - CUSTOMISATION AND INTEGRATION

Embedded developers will have differing requirements and tools for data analysis. As a fully integrated tool within the Tornado development environment, the flexibility and open API of WindView creates additional opportunities for the developer, including custom analysis. Open C++ and Tcl (Tool command language) APIs provide direct access to the event database for 3rd party tools integration (hardware or software) and/or custom analysis.

An eventBase API is provided (in both C++ and Tcl) which allows developers to implement their own tools to analyse event logs. For instance, a custom tool could be implemented to detect whether a certain hard-real time requirement is achieved throughout the sequence described by the log, or to generate a set of statistics specific to a particular application.

Wind River can also provide WindView analysis packs to perform specific analysis of the event log - which may be output to an analogue overlay graph which is part of the WindView tool, or directly and automatically into MS Excel on the host system. In addition, newly instrumented memory and I/O libraries make possible further calculations including memory and CPU usage per task, or for all tasks.

TARGET CONNECTION

As WindView is fully integrated into the Tornado development environment, a user can log-on remotely to any VxWorks target on the network. Tornado's open structure allows WindView to connect independently of the host-connection, which can be Ethernet, in-circuit emulator or serial. This further enhances the productivity boost offered by such a powerful development tool, since any design team member can attach to the target system from any location, even from a remote site. No more need to transport (and re-configure) a shared target board - just share the connection across the network!

SUMMARY OF KEY WINDVIEW CAPABILITIES

- Triggers on events (start/stop measurement or user-defined function)
- Hide tasks or interrupts to focus on real-time problems
- Filter events
- Deferred data upload to minimise the impact on system
- Analysis packs enable viewing of memory and CPU usage
- APIs provide integration into Excel spreadsheets
- C++ and Tcl command language APIs to event database
- Target system file server - development link from target server to debugging agent can be used for upload
- Dynamic configuration (ring-of-buffers) of memory on the target
- Supports multiprocessor applications
- Analyses in deferred upload, continuous upload or post-mortem modes ■

Nigel Street is Project Leader for WindView based in Swindon, UK and is part of Wind River's Visualisation Tools Group. Nigel began his career at ICL as a Software Engineer where he gained considerable experience as a user of Wind River's software development tools. He selected VxWorks for several developments in mainframe computer communications and disk peripherals. Nigel subsequently joined Radstone Technology developing software for single board computers. Nigel has nine years experience in developing real-time software and has BSc(Hons) in Computing Science.