

Save time and money by using PC Technologies for embedded system development

Typical embedded and real-time system development is a multistep process that includes code programming, debugging, compiling, downloading, and deploying. Traditionally, these separate tasks are performed in separate environments rather than integrated with each other. But this often is a time-intensive process that works against the ultimate goal when developing an embedded real-time system - reducing time-to-market. A definite way to save time is using versatile tools that are platform independent, tightly integrated and easy to use. They will save time, and thus money. In addition, code reuse will preserve capital investment over long periods of time. Traditional PC development environments have solved time-to-market problems and are now bringing rapid development tools to embedded systems.

Platform independence gives developers a strong competitive edge. Developers have the versatility to move easily between platforms, and to create scalable solutions so that precious time does not have to be spent learning new software programming for each new hardware target.

As for code reuse, tools that allow one to reuse existing or new code can save months of development time for engineers, and thousands of dollars in development costs.

Furthermore, when software and hardware tools are tightly integrated, time is not wasted dealing with intermediate compiler steps and compatibility issues.

When development tools take advantage of industry standards, it makes them easier to use. Not only are they familiar to the user and easier to integrate, standard development tools have a backing of an abundance of industry expertise when problems arise.

PC TECHNOLOGIES GROWING IN THE EMBEDDED WORLD

As the possibilities of embedded solutions expand, PC technologies are entering the embedded world. One benefit of PC technologies is that powerful software application development environments like National Instruments LabVIEW RT can now be used to create real-time and embedded systems. PC technologies provide integrated hardware and software development environments that promote code reuse and cross platform compatibility. Therefore, embedded developers are no longer forced to develop applications with closed, proprietary programming tools that offer little or no ability to port or scale to new applications. For example, Microsoft Windows CE is a modular, scalable operating system used for small footprint, embedded system platforms. The benefit of Windows CE is that users can leverage their experience with the industry-standard Microsoft Windows operating system application programming interfaces (APIs) to create applications that can "compute everywhere and connect anything." Example applications include net-

worked industrial controllers and the Auto PC. As Windows CE extends to new markets and device categories, there is tremendous potential for embedded developers to create applications faster and at less cost, for a variety of hardware targets.

CHALLENGES FACING DESKTOP SIMILAR TO EMBEDDED

The faster time-to-market issues that embedded developers face are the same issues desktop computer programmers have faced for years. For computer-based measurement and automation applications, National Instruments LabVIEW has emerged as the defacto industry standard. LabVIEW is a highly productive graphical programming environment that combines easy-to-use graphical development with the flexibility of a powerful programming language. It offers an intuitive environment tightly integrated with measurement hardware for engineers to quickly produce solutions for data acquisition, data analysis, and data presentation. The reason LabVIEW is so successful is that it has saved tens of thousands of engineers and scientists countless hours of development time while producing cost-effective solutions.

LABVIEW EXTENDS TO REAL-TIME AND EMBEDDED SOLUTIONS

National Instruments LabVIEW RT is a superset of LabVIEW and brings real-time performance to Windows measurement and automation applications by tight integration with real-time hardware. LabVIEW RT creates embedded programs that run in real-time on dedicated processors on National Instruments RT Series DAQ boards. LabVIEW RT runs directly on the DAQ board's processor and independent from the processor of the host computer. Therefore, LabVIEW RT users achieve deterministic, true real-time control. An application continues to run reliably even when the mouse is moved, the host system switches applications, or disk cache occurs. Because LabVIEW RT code is run on the intelligent DAQ board, control loop



Figure 1. When creating a multi-threaded application with LabVIEW, simply choose an execution thread and assign a priority level.

applications continue to work even if the host PC crashes.

LabVIEW RT is unique because it creates traditional real-time applications, but takes advantage of all the benefits of LabVIEW for Windows graphical programming development environment. Therefore, embedded developers can use more than ten years of LabVIEW advancements for increased productivity and faster time-to-market solutions. The most important features of LabVIEW RT are listed below:

1. LabVIEW RT uses the LabVIEW graphical programming environment that has proven to increase productivity by a factor of 4 to 10. LabVIEW RT graphical programming is easy to learn and master, and provides the ability to rapidly prototype, design, and modify systems in a shorter amount of time than text-based environments. Because the LabVIEW programming environment is similar to standard flow chart notation, the program is intuitive and easy to use.
2. LabVIEW is a fully compiled and multi-threaded application. More importantly, these features are built into the language so that developers use them directly as benefits. There is no extra coding that needs to be learned and performed. For example, to prioritize a LabVIEW program (a virtual instrument, or VI) over another, simply select the priority level and execution system in the VI set-up box.
3. LabVIEW RT code is reusable and will work with the latest versions of LabVIEW so that users can easily take advantage of the latest technologies available. LabVIEW is a compiled application that is also platform independent and can be developed on a number of operating systems like Windows, Macintosh, Sun, and Linux.
4. LabVIEW RT not only reduces development costs, but preserves capital investment as well. Personal computers combined with application software and plug-in boards provide a cost-effective, versatile solution for embedded system development. Since LabVIEW RT is just LabVIEW graphical programming, companies benefit from having a unified development environment that scales application development from desktop to embedded real-time systems. Engineers can develop solutions for embedded applications using the same tools they are familiar with on desktop computer-based data acquisition systems. For existing LabVIEW and data acquisition users, creating real-time embedded

applications is no different from desktop programming. There are no new real-time APIs to learn. More importantly, customers do not have to program any interrupt handling because the data acquisition drivers take care of that automatically.

5. With LabVIEW RT one can complete entire applications. But LabVIEW RT is an open environment so if users want to use external or existing code, that is possible. With LabVIEW RT, users can import custom C code or call external dynamic link libraries (dlls).
6. LabVIEW RT creates embedded executables automatically. To download applications to embedded processors, the user pulls up a simple menu item selection. All compiling and downloading takes place seamlessly. This is possible because the hardware and software are tightly integrated with each other. No extra time is spent learning different hardware compilers or worrying about how to transfer the code into the embedded system.
7. Because of the tight integration of hardware and LabVIEW RT software, communication between the target system and the host LabVIEW RT development system takes place automatically. A benefit of this built-in communication is that programs can be debugged in real-time while running embedded, making it easier to find and correct mistakes.
8. LabVIEW RT is a superset of the industry standard graphical programming of LabVIEW. There are tens of thousands of LabVIEW users worldwide. This huge installed base provides an unlimited supply of expertise in programming that spans across endless applications.

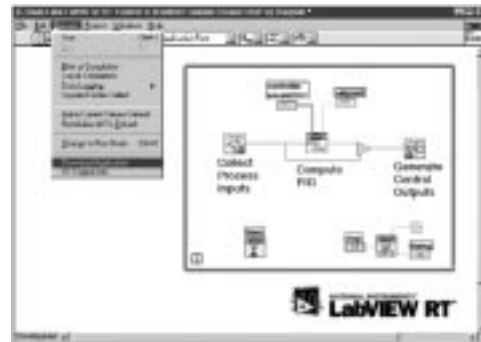


Figure 2. LabVIEW graphical programming code is compiled and downloaded automatically at run-time, or with a simple menu selection. The actual compiling and downloading steps are hidden from the user.

EASIER EMBEDDED SOLUTIONS

As the lines between real-time, embedded, and desktop development continue to blur, one thing remains clear - embedded and real-time system developers can now take advantage of highly powerful and productive desktop environments. Users can now program in a unified development environment and reuse the same code for a variety of hardware targets. Therefore, developers can create code with common PC technologies and then target any hardware plat-

form that will serve the end application goal. With National Instruments LabVIEW RT, engineers develop, download, debug, and deploy faster. This is done through easy-to-use tools that leverage industry standard graphical programming. More time can be allocated to developing the most important part of the system: developing in the actual application software. There is no need to spend valuable time worrying how to embed the code in hardware to achieve the desired performance. With real-time technologies like LabVIEW RT, developers ultimately achieve their faster time-to-market goal ■

Jim Balent is the real-time systems product manager at National Instruments. Previously, he worked as a district sales manager and prior to that as an applications engineer with National Instruments. Balent holds a B.A. in physics from Franklin and Marshall College and an M.S. in electrical engineering from the University of Utah.

REAL TIME MAGAZINE

EDITORIAL CALENDAR

- 1Q00 NETWORKS
(incl. RTOS Section)
- 2Q00 INSTRUMENTATION & AUTOMATION
- 3Q00 BUS TECHNOLOGY
- 4Q00 EMBEDDED INTERNET

previews on the web at:
<http://www.realtime-info.com>

ADVERTISEMENT INDEX

QNX SOFTWARE SYSTEMS	2
RTS2000	13
EMBEDDED SYSTEMS 2000	17
LOGIC TECHNOLOGY	27
VMETRO	31
SONY	43
VMIC	47
SYSTRAN CORP.	59
GESPAC	63
NATIONAL INSTRUMENTS	77
OBJECTIME	91
FORCE COMPUTERS	92

For more information on advertisement opportunities, call 32-2-520.55.77 or send an e-mail to info@realtime-info.com

EDITOR'S NOTE

Inclusion in this Company Directory about Development Tools for Real-Time Systems should not be taken as a Real-Time Magazine endorsement or recommendation. Likewise, omission from the company directory should not be taken negatively. The information here was believed to be accurate at the time of writing, but Real-Time Magazine cannot be responsible for omissions, errors or changes that occur after compilation.