expresslogic

Industrial Grade Software:
What It Is and Why You Need It

The growth of commercial real-time operating systems (RTOSes) has enabled the revolution we've all enjoyed in snazzy leading-edge consumer electronics products, informative and entertaining automotive electronics, and sophisticated, life-saving medical devices. Leading manufacturers are enhancing our quality of life with product after product that use modern, commercial microprocessor and software technology. Figure-1 shows the HP Officejet Printer, one of HP's market leading printer families, all of which use Express Logic's ThreadX® RTOS.

Figure-1

Years ago, such products were less common and often ill-fated, as the electronics industry – particularly the software sector – tinkered with free, low-cost, open-source, or Do It Yourself (DIY) RTOS solutions.

The result was unreliable products, overdue development schedules, and costly staffing that struggled with unproven software.

Fortunately, over the last 20 years, software development has made dramatic advances that have skyrocketed productivity and propelled the electronics industry into a key role in everyday life. Paramount among these advances are the tools available today for software developers that enable them to be much more productive.

Industrial Grade Software

But all commercial RTOSes are not alike. They vary in features, performance, and use in successful products. At the high-end of commercial RTOS quality, we find "Industrial Grade" RTOSes, which dominate those used in electronic products that have achieved widespread market success. Examples abound – including HP Inkjet Printers, GoPro Cameras, FitBit Wearables, GE Smart Meters, Honeywell Home Security Systems, Sony Televisions, Avidyne Flight Control Systems, Welch Allyn Medical Devices, Renesas Synergy Platform for IoT, and many more. Most of these highly successful products employ Express Logic's X-Ware (ThreadX RTOS, FileX FAT 16/32/exFAT File System, NetX TCP/IP Stack, USBX USB Host/Device Stack, GUIX GUI Development Framework, TraceX Event Analysis Tool). In fact, Express Logic's ThreadX RTOS (and other X-Ware software) is deployed in over 5.5 billion electronic products worldwide. That's more deployments than any other commercial RTOS! Express Logic's X-Ware is designed to cover all the needs of IoT product development with a broad, high-quality, Industrial Grade software suite, and this accounts for its overwhelming success.

Here are some general characteristics of Industrial Grade Software:

- **Widely Used Across Industries**

  Industrial Grade software is found in market-leading, highly-successful products, in consumer electronics, medical devices, industrial automation, transportation, and anywhere else where high-quality, robustness, and proven reliability are paramount. Such products are produced in very high volume, so any failure would be extremely expensive and damaging to the manufacturer's reputation. The success of such microprocessor-based products requires solid system software that has been field-proven.

- **Pre-Certified**
  Pre-certifications of an RTOS serve to assure the user that the RTOS has met the rigorous design, test, quality, and documentation requirements of the certifying agency. This benefits developers of all applications, whether their particular products require certification or not. Common safety certifications include IEC 61508 SIL 4, IEC 62304 Class C, ISO 26262 ASIL D, UL/IEC 60730, UL/IEC 60335, UL 1998, and EN 50128 SW-SIL 4. It's important to note that SIL 4 and ASIL D are the highest level of safety certification within those standards. Express Logic's ThreadX RTOS, and NetX Duo IPv4/IPv6 TCP/IP Stack are certified to these standards. NetX Duo is the only commercial TCP/IP offering that has this level of certification.

- **Availability of Full Source Code**
  The availability of full source code for an RTOS enables developers to better understand how the RTOS operates, and enables build-time setting of various configuration parameters and features for specific application needs. For an in-depth analysis of the value of source code, please see our white paper, "Consider the Source." (see www.rtos.com Downloads -> White Paper Downloads)

- **Quality Source Code and Documentation**
  As discussed in our "Consider the Source" white paper, simple availability of source code is not enough. That code should be clean, clear, and consistent in order to achieve the full benefits of that access. Poorly written, undocumented source code is of lesser value than well commented, clear code. A high-quality RTOS is written in code that meets these criteria, and offers users a greater benefit than an RTOS that does not. Express Logic's X-Ware is all coded to the same consistent coding standards, with User Guide documentation written before any code, assuring a developer's point of view with regard to ease of use.

- **Validated Source Code**
  Third party static analysis applications exist that analyze source code and evaluate it for consistency, correctness, and adherence to modern coding standards, such as ANSI and MISRA. Examples of such analysis programs are Synopsys' Coverity and IAR Systems' C-STAT. Achieving a clean build with these tools assures the developer that the RTOS code is properly constructed and free from structural defects, and that it meets all requirements of the standard against which it's being tested. By utilizing Gcov, IBM's PureCoverage, and LDRA's LDRAcover X-Ware product testing achieves 100% statement and 100% branch coverage, which is a necessary component for safety certification.

- **No Open Source Code**

An RTOS that includes open source code, or software of unknown pedigree (SOUP), presents the developer with a liability issue regarding infringement on the Intellectual Property of another party, or the requirement to adhere to a restrictive license such as GPL. A high-quality RTOS should be developed from scratch by a responsible company, without reliance on any open source code or SOUP. Many product manufacturers require all code that they license to be 100% free from open source or SOUP.

- **Indemnification**
  Indemnification is a commitment by the RTOS provider to defend against any claims of IP rights infringement by the RTOS. By indemnifying its users, an RTOS company guarantees that there is no IP infringement, and backs up that guarantee with full financial responsibility to defend against any claims of infringement. Without such a blanket indemnification, a user would be at risk of some unintentional infringement that comes to light down the road, causing expensive litigation and/or payment to the holder of those rights, and possibly a major product recall.
- **Support**
  Good support is essential to help developers resolve issues they encounter with the use of an RTOS. Whether it's an unlikely bug, or the more common user issue, an RTOS supplier who stands behind its product with full support for its users provides a resource of immense value.
- **Small Size**
  An RTOS should be designed and coded to be as small as possible, while still implementing the features necessary to meet the needs of embedded system applications. Small code footprint is a must for today's small memory microcontrollers, and even in larger memory applications, a small RTOS leaves more space for application code to provide useful functionality that makes the product more successful.
- **High-Performance**
  RTOS real-time performance is a critical factor in achieving efficient operation. The time taken to provide RTOS services, such as interrupt response, context switching, and message passing is overhead that should be minimized. In a demanding real-time environment with rapid interrupt activity, the RTOS service time must be kept small to avoid system thrashing and the inability to keep up with events, leading to system crash.
- **Advanced Technology**
  Premium quality RTOSes offer more than the basic functionality of commodity RTOSes. Useful capabilities such as Event Trace, Event Chaining, Preemption-Threshold Scheduling, Downloadable Modules, Execution Profiling, SMP support, Run-Time Stack Analysis, and Priority Inheritance, distinguish high-quality RTOSes from less capable ones.
- **Breadth of Products**
  As beneficial as an Industrial Grade RTOS might be for IoT product development, many such projects require more than just an RTOS. In fact, most, if not all, would require some sort of TCP/IP network stack, Cloud Protocols, and a Graphical User Interface. Some might also require USB and a File System. Each of these products should be developed to the same standards as the RTOS, and ideally be consistent in source code design, structure, style, and API.
- **Adam Smith's "Invisible Hand"**

The market force that helps the demand and supply of goods in a free market to reach equilibrium. A successful company has the revenue (from sales of its products in demand) to invest in its product development, which in turn, generates profits to fuel further growth. An open source development team has no such profit motivation or benefit. In all areas of commerce, products that do not generate revenue are not able to justify further investment from their developers. By choosing an RTOS that is in demand, and that generates life-sustaining revenue, developers can be assured that the RTOS provider is highly motivated to help its users succeed, since only through their success can the company succeed as well.

- **Easy to Use**
  It's important for software tools (RTOSes, IDEs, Middleware, and Application Development Tools) to be easy to use, so that developers can learn to use them more quickly, make fewer mistakes, and be more productive. Industrial Grade software has great documentation, examples, wide processor support, and intuitive API's. The result is greater productivity, faster product development, and better products.

- **Picokernel Architecture**
  A nanokernel architecture is one in which only basic RTOS services are required in the kernel, but additional services may be called from application software and added automatically to the image at link time. Automatic kernel configuration based on what is being used keeps RTOS code size to a minimum, while giving developers full control over what is included in the kernel executable image. This requires each function to be self-sufficient, packaged in its own source code file, and not dependent on and optional kernel services. Industrial Grade RTOSes offer this flexibility and sophistication, while other types require complex configuration and may result in pockets of dead code that get linked but are never called.

- **Developer Success**
  Embedded Market Forecasters of Framingham, Massachusetts, have surveyed hundreds of embedded developers for over 10 years, and their findings show that developers using commercial RTOSes outperform those using open source and low-cost RTOSes in achieving on-time schedule completion and ROI. Further, the data shows that not all commercial RTOSes fared the same. One in particular, ThreadX, distinguished itself from the pack. See the referenced white paper, "*Shootout At The RTOS Corral*," and excerpts below.

- **Fast Time to Market**
  Time to Market is critical to the success of any product. By using a high-quality RTOS, developers can increase the likelihood that they will complete their project on time, and achieve faster time to market. Because this is so important, we will now explore this critical issue further, and show how fast time to market can enable a product to achieve maximum success.

The Critical Importance of Time To Market

Developers know that they need to get a quality product to market as soon as possible, but delays that slip in as a result of adding new features, losing track of requirements, and unforeseen difficulty implementing design features render that product far less profitable and successful than it would otherwise be.

"*In the electronics industry, introducing a product nine to 12 months late can cost it fifty percent of its potential revenues*," say George Stalk and Thomas Hout in *Competing Against Time*.[1] Fifty percent. That's a significant impact. But delays in time to market create other, even more sobering problems. They point out: "*A long development process exposes the project to the risk of changes in the market and environment*.".

In part, delays can make the difference between holding first-mover advantage versus launching a product in a market that's already partially subscribed. The challenges go beyond that, however—consumer needs and expectations may have moved on. A 3G phone, however good, launched after 4G models have been released simply will not accrue the same amount of revenues. "*Reducing elapsed time can make the critical difference between success and failure*," says the synopsis. "*Give customers what they want when they want it—or the competition will*."

Being late erodes the addressable market for a product. If a manufacturer initially targets a market segment of ten million units with a market lifetime of 18 months but delivers the product six months late, the addressable market will shrink. Not only will the market be smaller, the competition will be much more intense because competitors will have an established base from which to sell. Finally, being late, by definition, means that the development process took longer than anticipated, which makes the cost of designing and commercializing the product more costly as a result of the additional developer salaries and overhead that result. But the biggest financial consequence of such delays actually is not the extra cost of development, but the cost of lost revenue.

The chart shown in Figure-2, produced by Kim Rowe and published in an EETimes article in 2010 (sorry – article no longer online), illustrates the effect of time to market on aggregate sales volume by showing a product-adoption curve over time. The green and orange areas under the curve represent the total number of product units that can be sold. Being late erodes the market (green portion) in two ways. First, you don't get to sell the product for as long a period of time. Second, you lose market share early and can't regain it later, so the number of available prospects is smaller. The result is lost revenue, compared to what would have been possible with an earlier product introduction.

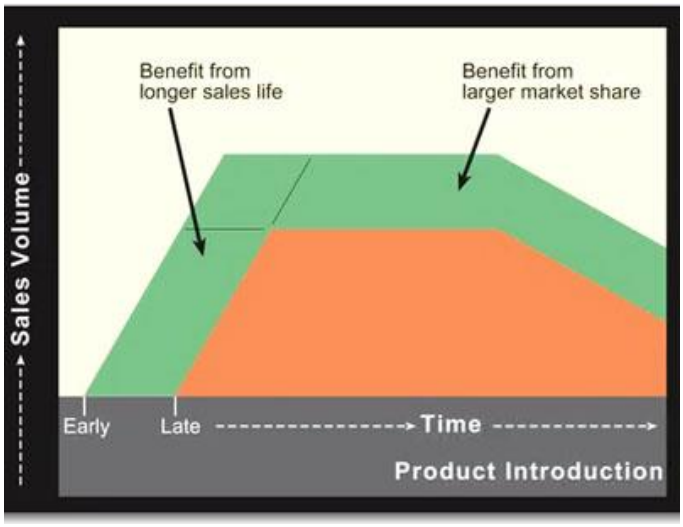

Figure-2: A product brought to market late (orange) accrues lower sales volume (green band) over the entire commercial lifecycle.[2]

[1] G. Stalk and T. Hout, *Competing Against Time*, Simon & Schuster (1990).

The Role of the RTOS in Achieving Fast Time To Market

What can developers do to help bring better products to market faster? Design teams are in control of one critical aspect of time to market, and that is completing their development project on time. Once a project has been defined and a schedule prepared, many aspects of product launch swing into action, all coordinated with project completion. If the project is delayed, the entire development plan suffers, and the product will be late to market.

Development projects for embedded systems products often include selection and use of a real-time operating system (RTOS) for the product. The RTOS is then embedded into the final product for production. Does the RTOS have an effect on the project completion, and on the ultimate product quality?

Industry analysts at Embedded Market Forecasters (EMF) (www.embeddedforecast.com), Framingham, Massachusetts, have surveyed over 500 embedded developers for the past several years. They have found that Commercial RTOSes dominate ROI results compared with open-source RTOSes.

In their 2010 white paper, "Shoot-Out at the RTOS Corral," EMF shows the ROI advantages of using Commercial RTOSes over others. According to the EMF survey, Commercial RTOSes better enable developers to meet schedule and get products to market on-time, with 60% of projects completed "on time or ahead of schedule." Readers are encouraged to read the full white paper at: http://www.rtos.com/PDFs/ShootoutRTOSCorral-03-2009.pdf.

Notably, not all commercial RTOSes faired the same. EMF asked, "What operating system did you use in your last project? And, "Was your last project completed on-time, ahead-of-schedule, or behind-schedule?"

A tabulation of recent (2016) data in the latest EMF white paper, "*Comparing Time To Market of Open Source Software with Time to Market of Commercial RTOSes*," revealed that, on average, 38% of their projects were completed **behind schedule**. Projects that used Express Logic's ThreadX® RTOS performed the best, with a 70% rate of completion **on-or-ahead-of-schedule**. Clearly, using the right RTOS can significantly benefit a project's likelihood of finishing on time (see Figure-3)

"*In our 2016 analysis of RTOS developers, those using ThreadX completed 70% of their projects on or ahead of schedule, while all other RTOS vendors averaged just 62%, and embedded Linux vendors averaged 61%, giving ThreadX a higher ranking in time to market*," says EMF president Jerry Krasner. "*What is of even greater interest in looking at this data is that ThreadX has achieved the best 'on or ahead of schedule' results for 10 years running*."

–Dr. Jerry Krasner, Chief Analyst, EMF

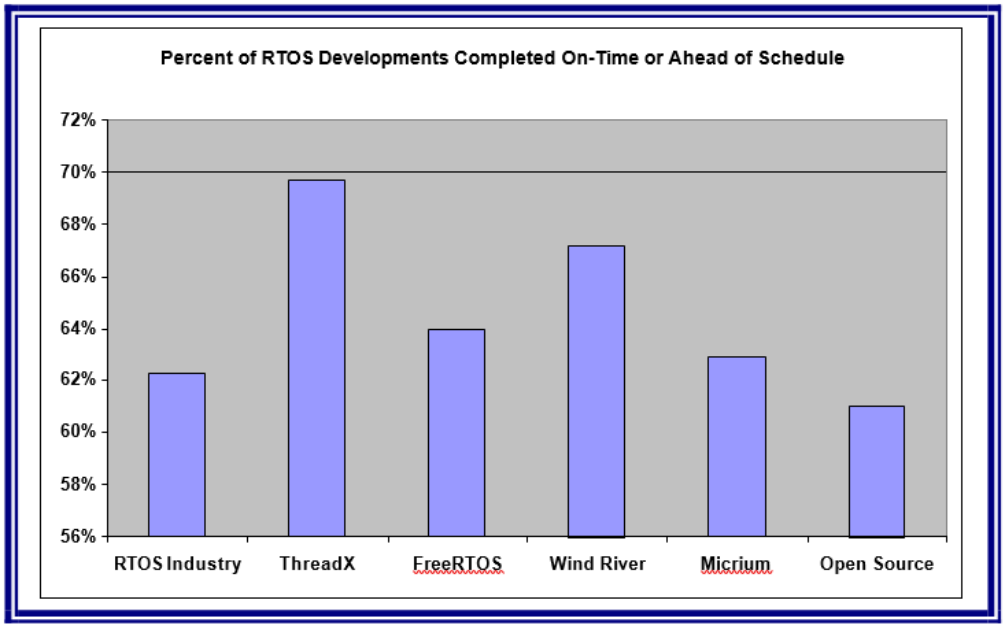

Figure-3: Development Projects Completed On or Ahead of Schedule[2]

Ensuring product quality

Of course, getting to market quickly is meaningless unless you deliver a quality, reliable product. Latent defects can result in recalls, excessive warranty costs, and a degraded brand. The drawbacks don't have to be that extreme to impact success, though. All it takes to erode your market share is the release of a competitive product that offers better performance than yours. Here, too, the right RTOS can help you develop a better product that's more profitable. Development teams always make trade-offs among factors including features, performance, development time, and budget. It's tempting to spend as little as possible for development, but that can be "penny wise and pound foolish." Using the best tools, whether the RTOS, compiler, or other development tools, might cost more than using "free" or less costly alternatives, but the data shows that the nominal savings typically represents only a fraction of the resulting cost of being late to market or delivering a product with noncompetitive performance or quality. Developers who achieve fast schedule completion by jettisoning product performance, functionality, or features jeopardize product success. It's important to see whether achieving on-time project completion was done without sacrificing design expectations. Again, in this area, Express Logic's ThreadX RTOS showed better results than the Industry Average (see Figure-4).

Within 10% for:	RTOS Industry	ThreadX	Free RTOS	Wind River	Micrium	Open Source
Performance	46.0%	68.4%	49.7%	62.8%	45.7%	52.7%
Systems Functionality	45.0%	57.9%	37.6%	70.8%	52.9%	48.5%

Figure-4: comparison of product quality design results for different RTOSes[2]

[2] *Comparing Time To Market of Open Source Software with Time to Market of Commercial RTOSes,* EMF, 2016.

Conclusion
Industrial Grade is the highest standard of commercial software available today. It is difficult to achieve, but offers unique, powerful benefits to developers using it. Here is a summary of the characteristics of Industrial Grade software, and how Express Logic's X-Ware delivers each one:

Feature	Benefit	Express Logic's X-Ware
Widely Used Across Industries	Field-proven, both for reliability and for success in use in market-leading products.	Used in market-leading products from HP, Sony, GE, Honeywell, Welch Allyn, NASA, and many others, with over 5.5 billion deployments
Pre-Certifications	Off-the-shelf approval for use in safety-critical systems. Saves time and cost.	IEC 61508 SIL 4, IEC 62304 Class C, ISO 26262 ASIL D, UL/IEC 60730, UL/IEC 60335, UL 1998, and EN 50128 SW-SIL 4. First pre-certified TCP/IP Stack (IPv4/IPv6 NetX Duo)
Ease of Use	Increases development quality and speed	EMF Survey Data shows ThreadX is used in more successful products. The reason for this is largely due to the outstanding Development Tools that are included In X-Ware, that help developers implement successful solutions quickly and efficiently. These tools include Express Logic's TraceX real-time event analysis visualization tool, and its GUIX WYSIWYG PC-based GUI Design Studio tool
Picokernel Design	Automatically scales to minimum required code size	All X-Ware designed with picokernel architecture
Developer Success	Proven success using the software in the development of commercial products	EMF survey data shows developer success using the ThreadX RTOS, achieving high scores in timely project completion and product functionality
Fast Time to Market	Proven results meeting development schedule and product quality	EMF survey data quantifies results in meeting schedule, speeding time to market, and increasing ROI
Availability of Full Source Code	Enables better understanding and configuration	100% source code provided with all X-Ware products
Quality Source Code and Documentation	Clean, Clear, Consistent for ease of use and maximizing the benefits of having source code	Static analysis tools such as Coverity and C-STAT help ensure code quality. Free Evaluations are available for developers to see firsthand

Validated Source Code	Every statement and branch analyzed for correctness	100% statement and branch coverage is achieved with Gcov, PureCoverage, LDRAcover
No Open Source Code	All code developed by vendor	No open source; no SOUP
Indemnification	Free from IP rights infringement risk	Full indemnification for all X-Ware products
Support	Vendor stands behind all code	Full support for all X-Ware by the authors
Small Size	Fits in smallest microcontrollers	ThreadX less than 2KB, depending on services used
High-Performance	Reduces overhead	Benchmark results show significant performance advantage over competition
Advanced Technology	Gives developers tools for making better products	Event Trace, Event Chaining, Preemption-Threshold Scheduling, Downloadable Modules, Execution Profiling, SMP support, Run-Time Stack Analysis, and Priority Inheritance
Adam Smith's "Invisible Hand"	Product demand motivates vendors to invest in development and support. Revenue funds both.	Express Logic has been profitable and growing for 20 years, with 5.5 Billion deployments and 4,000 licensees
Breadth of Products	Provides tools and solutions for the wide-ranging needs of IoT product development	Express Logic's X-Ware Platform includes an RTOS, IPv4/IPv6 TCP/IP stack, Cloud Communication Protocols, USB Host/Device/OTG, GUI Development Framework, and Run-Time Analysis Tool for event trace. X-Ware Platform covers the full breadth of Industrial Grade functionality needed for IoT product development.

The decision regarding where to spend your limited development budget—and indeed, how large a development budget you need— must be made in the context of the overall success of the product and the enterprise. Use the software products and tools that will enable you to bring your product to market as quickly as possible, and to make your product as successful as possible – not necessarily the tools that cost the least. If that "free" RTOS led to a slower development project and a delayed time to market, even if it saved you $100,000 in development cost, would you still consider it a bargain?

Another way to look at this issue is to compare against other industries. Take construction, for example. Would a contractor building a skyscraper outfit his workers with $50 shovels rather than use a $250,000 earthmover to dig the foundation? Would he buy them $10 hammers instead of $50 nail guns? In cases like these, success is aided by using the best tools available, not necessarily the least expensive. Embedded developers would be well served to follow similar practices when choosing an RTOS - and other development tools as well. **Choose an**

Industrial Grade RTOS that best enables fast time-to-market and high performance, not just one that might be less costly or "free."