

Better Firmware... *Faster!*

A One-Day Seminar

Presented at

Your Company

Does your schedule prevent you from traveling?

This doesn't mean you have to pass this great opportunity by.

Presented by **Jack Ganssle**, technical editor of *Embedded Systems Programming Magazine*, author of 6 books and over 600 articles

More information at www.ganssle.com

The Ganssle Group
PO Box 38346
Baltimore, MD 21231
(410) 504-6660
fax: (647) 439-1454
info@ganssle.com
www.ganssle.com

**For Engineers
and
Programmers**

This seminar will teach you new ways to build higher quality products in half the time.

80% of all embedded systems are delivered late...

Sure, you can put in more hours. Be a hero. But *working harder is not a sustainable way to meet schedules.* We'll show you how to plug productivity leaks. How to manage creeping featurism. And ways to balance the conflicting forces of schedules, quality and functionality.

... yet it's not hard to double development productivity

Firmware is the most expensive thing in the universe, yet we do little to control its costs. Most teams deliver late, take the heat for missing the deadline, and start the next project having learned nothing from the last. Strangely, *experience* is not correlated with *fast*. But *knowledge* is, and we'll give you the information you need to build code more efficiently, gleaned from hundreds of embedded projects around the world.

Bugs are the #1 cause of late projects...

New code generally has *50 to 100 bugs* per thousand lines. Traditional debugging is the *slowest* way to find bugs. We'll teach you better techniques proven to be up to 20 times more efficient. And show simple tools that find the nightmarish real-time problems unique to embedded systems.

... followed by poor scheduling

Though capricious schedules assigned without regard for the workload are common, even developers who make an honest effort usually fail. We'll show you how to decompose a product into schedulable units, and how to use killer techniques like Wideband Delphi to create more accurate estimates.

Learn From The Industry's Guru

Spend a day with Jack Ganssle, well-known author of the most popular books on embedded systems, technical editor and columnist for *Embedded Systems Programming*, and designer of over 100 embedded products. You'll learn new ways to produce projects *fast* without sacrificing quality. This seminar is the only non-vendor training event that shows you *practical* solutions that you can implement *immediately*. We'll cover technical issues – like how to write embedded drivers and isolate performance problems – as well as practical process ideas, including how to manage your people and projects. *Contact us to learn how we can award each of the attendees 0.7 Continuing Education Units.!*

Seminar Leader



Jack Ganssle has written over 600 articles in Embedded Systems Programming, EDN, and other magazines. His five books, **The Art of Programming Embedded Systems**, **The Art of Developing Embedded Systems**, **The Embedded Systems Dictionary**, **The Firmware Handbook**, and **Embedded Systems, World Class Designs** are the industry's standard reference works

Jack lectures internationally at conferences and to businesses, and was this year's keynote speaker at the Embedded Systems Conference. He founded three companies, including one of the largest embedded tool providers. His extensive product development experience forged his unique approach to building better firmware faster.

Jack has helped over 600 companies and thousands of developers improve their firmware and consistently deliver better products on-time and on-budget.

Course Outline

Languages

- C, C++ or Java?
- Code reuse – a myth? How can you benefit?
- Controlling stacks and heaps.

Structuring Embedded Systems

- *Manage* features... or miss the schedule!
- Using multiple CPUs.
- Five design schemes for faster development.

Overcoming Deadline Madness

- Negotiate realistic deadlines... or deliver late.
- Scheduling – the science versus the art.
- Overcoming the biggest productivity busters.

Stamp Out Bugs!

- Unhappy truths of ICEs, BDMs, and debuggers.
- *Managing* bugs to get good code fast.
- *Quick* code inspections that keep the schedule on-track.
- Cool ways to find hardware/software glitches.

Managing Real-Time Code

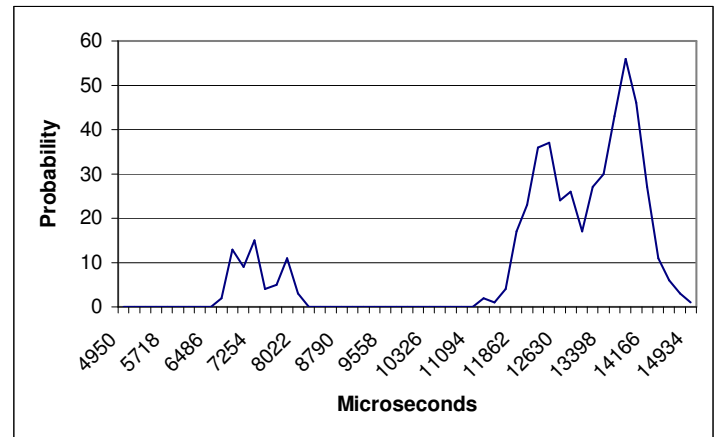
- Design *predictable* real-time code.
- Managing reentrancy.
- Troubleshooting and eliminating *erratic crashes*.
- Build better interrupt handlers.

Interfacing to Hardware

- Understanding high-speed signal problems.
- Building peripheral drivers faster.
- Inexpensive performance analyzers.

How to Learn from Failures... and Successes

- Embedded disasters, and *what we must learn*.
- Using postmortems to accelerate the product delivery.
- Seven step plan to firmware success.



Do those C/C++ runtime routines execute in a usec or a week? This trig function is all over the map, from 6 to 15 msec. You'll learn to rewrite real-time code proactively, anticipation timing issues before debugging.

Why Take This Course?

Frustrated with schedule slippages? Bugs driving you batty? Product quality sub-par? **Can you afford not to take this class?**

We'll teach you how to get your products to market faster with fewer defects. Our recommendations are *practical, useful today, and tightly focused* on embedded system development. Don't expect to hear another clever but ultimately discarded software methodology. You'll also take home a 150-page handbook with algorithms, ideas and solutions to common embedded problems.

If you can't take the time to travel, we can present this seminar at your facility. We will train **all** of your developers and focus on the challenges unique to your products and team.

Here is what some of our attendees have said:

Thanks for the terrific seminar here at ALSTROM yesterday!
It got rave reviews from a pretty tough crowd.

Cheryl Saks, ALSTROM

Thanks for a valuable, pragmatic, and informative lesson in embedded systems design.
All the attendees thought it was well worth their time.

Craig DeFilippo, Pitney Bowes

I just wanted to thank you again for the great class last week. With no exceptions, all of the feedback from the participants was extremely positive. We look forward to incorporating many of the suggestions and observations into making our work here more efficient and higher quality.

Carol Bateman, INDesign LLC

Here are just a few of the companies where Jack has presented this seminar:

Sony-Ericsson, Northrup Grumman, Dell, Western Digital, Bayer, Seagate, Whirlpool, Cutler Hammer, Symbol, Visteon, Honeywell, Kodak and Western Digital.

Did you know that...

- ... doubling the size of the code results in much more than twice the work?*** In this seminar you'll learn ways unique to embedded systems to partition your firmware to keep schedules from skyrocketing out of control.
- ... you can reduce bugs by an order of magnitude before starting debugging?*** Most firmware starts off with a 5-10% error rate – 500 or more bugs in a little 10k LOC program. Imagine the impact finding all those has on the schedule! Learn simple solutions that don't require revolutionizing the engineering department.
- ... you can create a predictable real-time design?*** This class will show you how to measure the system's performance, manage reentrancy, and implement ISRs with the least amount of pain. You'll even study real timing data for common C constructs on various CPUs.
- ... a 20% reduction in processor loading slashes development time?*** Learn to keep loading low while simplifying overall system design.
- ... reuse is usually a waste of time?*** Most companies fail miserably at it. Though promoted as the solution to the software crisis, real reuse is much tougher than advertised. You'll learn the ingredients of successful reuse.

What are you doing to upgrade your skills? What are you doing to help your engineers succeed? Do you consistently produce quality firmware on schedule? *If not . . . what are you doing about it?*

Contact us for info on how we can bring this seminar to your company.
e-mail: info@ganssle.com or call us at 410-504-6660.