The Embedded Muse 142

Editor: Jack Ganssle (jack@ganssle.com) March 5th, 2007

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

Editor's Notes

Did you know it IS possible to create accurate schedules? Or that most projects consume 50% of the development time in debug and test, and that it's not hard to slash that number drastically? Or that we know how to manage the quantitative relationship between complexity and bugs? Learn all this and much more at my Better Firmware Faster class, presented at your facility. See http://www.ganssle.com/brochure-onsite.pdf.

Or, come to **Boston May 4**, where I'll present this class at the Sheraton Braintree hotel. Registration and other info here: http://www.ganssle.com/classes.htm . You'll earn 0.7 Continuing Education Units, learn a lot, and hopefully have a bit of fun, too.

The Embedded Systems Conference is coming to San Jose April 1-5. It'll be a whirlwind. I'm giving 5 talks, and am really looking forward to the teardown of a Toyota Prius, which couldn't exist without an awful lot of embedded smarts. The currently super-controversial Al Gore is giving the keynote, a very interesting move on the part of conference management, considering how conservative engineers are. But no one will complain about Orange County Choppers being there… though what they have to do with embedded is beyond me. Fun? You betcha. If you come to the conference do stop me and say hello.

I’ve created a video titled “Develop Firmware in Half the Time” that distills the basic ideas and processes needed to efficiently crank out great firmware. There’s more information available at http://www.ganssle.com/video.htm.

Ralf Holly commented on my reference to an article: “Regarding Dan Saks compile-time assertions article. I wrote an article about compile-time assertions for C/C++ User's Journal; it shows different implementations and various examples on how and when to use them. It is available online at http://www.ddj.com/dept/cpp/184401873?pgno=1 .” Worth reading.

I’m trying to tune an application that preprocesses C programs to extract contracts (a la Bertrand Meyer’s Design by Contract ™). But it’s written in Ruby and racc. If anyone knows those programs and is willing to give a couple of hours of thought, please contact me. No fortune, not much fame, but it might be fun.

English as a First Language

I’m always astonished at how many EEs write firmware full-time. All of those years in college studying electronics, transistor circuits, Kirchoff’s Law, and Electromagnetics, only to be doing essentially the work of a computer science major. Of course, in the embedded world so much of what we do is deep inside the hardware, so an EE background doesn’t hurt.

In fact many colleges now recognize the mixed skills needed in our industry with the Computer Engineering degree program, a sort of watered down EE with serious software emphasis. It takes a lot more code than hardware to build an embedded system, so a software focus just makes sense. Most surveys peg firmware at 70-80% of a product’s development effort.

When hiring firmware developers I’ve found little difference between each of these majors as long as the candidates have practical experience. In some particular industries, though, which degree might be important. One company I know almost exclusively hires mechanical engineers. They, too, get quite a bit of programming experience in college. Their sound understanding of machine control is what’s important to this company that makes material handling equipment.

ME, EE, CS, CE – all are engineering or engineering-like programs that stress practical problem-solving abilities. We’re taught to build things, to decompose big problems into solvable chunks. That in a large sense differentiates us from our friends studying the liberal arts.

Donald Knuth promoted the idea of Literate Programming, an approach that intermixed code and description, the description being expressed using TeX, a graphical word-processor-like formatting scheme. Postprocessors separated the description from the code. But central to his idea was that of very carefully expressing one's intentions, using all the richness of the English language instead of the abbreviated cryptic comments so typical in most code.

I've found that some of the best developers of all are English majors. They'll often graduate with no programming experience at all, and certainly without a clue about the difference between DRAM and EPROM.

But they can write. That's the art of conveying information concisely and clearly. Software development and writing are both the art of knowing what you're going to do, and then lucidly expressing your ideas.

The worst developers, regardless of background, fail due to their inability to be clear. Their thoughts and code tend to ramble rather than zero-in on the goal.

It's easier to train someone in a new language than to teach them to think clearly. C really isn't that hard to learn; it has but a handful of constructs. Most folks can learn the fundamentals quickly. Debugging takes longer, but all new programmers find themselves at sea when first faced with bugs.

Too many engineering-trained developers have a total disregard for stylistic issues in programming. Anything goes. Firmware is the most expensive thing in the universe, so it makes sense to craft it carefully and in accordance with a standard style guide. Make sure it clearly communicates its intent. This is where the English majors shine; they've spent 4 years learning everything there is to know about styles and communication.

In fact, the best programming book ever written was first published in 1914 and never mentions computers. It's "The Elements of Style", by William Strunk and E. B. White (Allyn & Bacon; ISBN: 020530902X). It's a slender volume of 105 pages that tells us how to write well. Just five bucks from Amazon.com. My observation is that most programmers are in dire need of such help. Unless a program's comments are in well-formed English that unambiguously express the developer's intent, the code is junk. Just as we've got to conform to C's syntax to make a useful routine, our comments must be in solid English to aid future maintenance. And maintenance, as we all know, usually costs much more than the initial development.

A more recent, shorter and less pedantic style guide is "The Technical Writer's Checklist", by unspecified people at iTrain.org (http://itrain.org/ttwc/). Its 15 pages list 38 rules essential to effective writing. Then there's Jane Straus's The Blue Book of

Grammar and Punctuation (2004, ISBN 0-9667221-7-5) is a large-format (8.5 x 11) volume of just 104 pages which is a practical and useful guide to improving one's use of English.

Today it's all about communication. Write clearly, especially when crafting comments and program documentation.

Last Buy

Intel, the company that launched the embedded systems industry with their four-bit 4004 in 1971, has announced the end-of-life for most of their embedded processors. The MCS51, MCS251, MCS96, 80X18X, 80X386, 80X486, and i960 will no longer be available. Last buy is March 30, 2007, with final shipments made by September 28.

A lot of folks use these parts. The good news is that this obsolescence will lead to more design work and full employment for embedded developers!

Tools & Tips

Thor Johnson wrote: "You mentioned SciTools TrackBack in the last Embedded Muse; I found one that is free-as-in-beer that seems to do the same thing: FileHampster http://www.mogware.com/filehamster/ .

"It seems simple enough; Not as "robust" as SubVersion or a real version control software (no tags/branches/etc), but easy to use to make sure you don't obliterate the wrong file at the wrong time.

"Works OK on WinXP Pro."

Grant Sargent likes Unison: "For keeping various files in sync (home, work, removable drives etc), I recommend the Unison file sync application (http://www.cis.upenn.edu/~bcpierce/unison/). Great if you don't need the revision control of subversion. Available for Linux and Windows. Open Source, works as advertised."

Stephen Pelc responded to this statement: "Please, PLEASE, recommend to your readers that they always use at least a 4-digit year (when naming files)" with: "Please also could USAnians use unambiguous dates! 5/7/2007 is a menace, whereas 05 July 2007 or 07 May 2007 are not.

“A practical date coding scheme for change notes in source code
we have found is to use:
“ date marker comment
“e.g.
“ 20070219 SFP003 Sent off to JG for newsletter
“These are in reverse chronological order at the top of the file.

“Because the date code is numeric it can be easily sorted. The marker is used in comments in the source and identifies the author (three letters) and change number (three digits).”

Oliver Bachmann writes: “If you are looking for a good search tool, try InfoRapid Search And Replace. It has one of the fastest search engines I have seen so far, and I use it regularly for searching strings in files even in subdirectories where other search tools do not generate a hit.”

Book Review

The numbers are impressive: 45% of developers working on 32 bit processors report using Linux in an embedded app. And Wind River recently bought FSMLab’s IP for the embedded space, giving WRI a “hard” real-time Linux distribution to complement VxWorks and their soft real-time Linux. I put “hard” in quotes as the various Linux vendors continue to slug out the notion of real-time in the Linux environment. See “Hard–real-time Linux deal under scrutiny” in the Feb 26, 2007 issue of EE Times for more on the imbroglio.

Now there’s a book about putting the OS into firmware. “Embedded Linux Primer, A Practical Real-World Approach” by Christopher Hallinan is one of those few books whose contents exactly match the title. It makes no attempt to introduce Linux to newbies, and is entirely focused on getting Linux running on an embedded system.

The chapter on building the kernel is interesting by itself as it gives some insight into how Linux is structured. Out of general curiosity I would have liked more information, from a higher-level perspective, but this is how-to tome instead presents the details needed to do the work.

A long section on processors was not particularly useful. A lot of CPUs will run the OS, and the author doesn’t point out particular advantages or downsides to any of the choices presented.

Bootloaders, including details of the ever-more-popular GRUB get good coverage, as does Linux's initialization sequence so one can understand what's going on as it spits out those 100+ lines of sometimes cryptic startup dialog.

The chapter on writing device drivers is short and could have been fleshed out in more detail, but does work through a useful example.

Most of the chapter on tools is too abbreviated to be of serious use to serious users of gdb and the like. But the chapter on debugging the kernel is worth the price of the book. Better, there's a section about what to do if your gleaming new OS doesn't boot.

Real-time issues are a real concern to a lot of firmware folks, and the author does a great job explaining both the problems, and the solutions, including some profiling data to get a bit of tantalizing insight into what sort of latencies are possible. Unfortunately, the author doesn't list the platform, so one is left wondering if a 78 usec worst-case latency applies to a 3 GHz Pentium… or a 100 MHz ARM.

This book is not a beach read. It's complex stuff. You'll have a hard time finding many pages that aren't crammed with directory listings, code, make file extracts, and the arcane argot of those delving deep into Linux. But the author writes well and informatively. If you're planning to stuff Linux into your embedded system, get the book.

Jobs!

Let me know if you're hiring firmware or embedded designers. No recruiters please, and I reserve the right to edit ads to fit the format and intents of this newsletter.

Nordic Semiconductor is looking for several people for work in Trondheim, Norway:

Software Architect -We seek an experienced Software Architect to take a leading role in the development of the Wibree protocol stack. You will have the opportunity to shape the development of the next big standard for wireless communication and be central in the definition and development of the first complete protocol stack for Wibree.

Software Developers - We seek both newly qualified and experienced Software Developers. The tasks include development of applications and protocol stacks for Wibree and for Wireless Desktop.

Project Managers - We seek experienced Project Managers to direct and lead the activities of the development team and ensure timely delivery. You will be responsible for providing leadership to the project team and technical guidance to the engineers.

More info: http://www.nordicsemi.no/index.cfm?obj=menu&act=displayMenu&men=11 . Fill in your application at: http://www.nordicsemi.no/com/index.php Application deadline is 16 March, but applications are evaluated as they arrive. All applications will be treated in confidentiality.

Crestron Electronics is actively seeking experienced and fresh-out embedded developers. We are located in New Jersey, close to Manhattan, between the Tappan Zee and George Washington bridges. We are a private company with over 1000 employees and are growing 20-25% per year!

We are finalizing the construction of a completely NEW Engineering facility that will open here in Rockleigh this spring. Our Engineering Department will now have it's OWN building.

There is a special section regarding engineering opportunities on our website – see http://www.crestron.com/features/careers/engineering/default.asp .

Qstreams Networks is seeking a Senior Embedded Software Engineer. We are an early stage, well-funded network systems company located in Glen Burnie, MD. Our product includes a variety of exciting technologies, from high-speed optics to LAN switching.

Required Skills

* Expert-level C programming, shell scripting and Linux system administration
* Experience with resource-constrained systems
* Strong knowledge of LAN technologies (Ethernet, TCP/IP)
* Strong knowledge of digital electronics. Must be able to read schematics and data sheets and work directly with hardware engineers during design and debug stages
* Ability to work independently and as part of a small team
* BSEE/BSCS with 5-7 years of embedded software experience

Preferred Skills

* Linux device driver development
* Experience with multiple processor families
* Python, C++ and Perl programming expertise
* Knowledge of optical telecom technologies
* Network management agent development (SNMP, HTTP)

Benefits include health care, dental, vision and a flexible work environment. Send resumes to careers@qstreams.com

Joke for the Week

.
From Slashdot. NOT (I think) a joke, but pretty darn funny.

"The new US stealth fighter, the F-22 Raptor, was deployed for the first time to Asia earlier this month. On Feb. 11, twelve Raptors flying from Hawaii to Japan were forced to turn back when a software glitch crashed all of the F-22s' on-board computers as they crossed the international date line. The delay in arrival in Japan was previously reported, with rumors of problems with the software. CNN television, however, this morning reported that every fighter completely lost all navigation and communications when they crossed the international date line. They reportedly had to turn around and follow their tankers by visual contact back to Hawaii. According to the CNN story, if they had not been with their tankers, or the weather had been bad, this would have been serious."
http://www.dailytech.com/Lockheeds+F22+Raptor+Gets+Zapped+by+International+Date+Line/article6225.htm

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the words “subscribe embedded *your-email-address*” in the body. To unsubscribe, change the message to “unsubscribe embedded *your-email-address*”. ". BUT - please use YOUR email address in place of “email-address”.

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.