

The Embedded Muse 170

Copyright 2008 TGG

December 1, 2008

You may redistribute this newsletter for noncommercial purposes. For commercial use contact info@ganssle.com. Subscribe and unsubscribe info is at the end of this email.

EDITOR: Jack Ganssle (jack@ganssle.com)

CONTENTS:

- Editor's Notes
- More on Reuse
- Dot Com Redux?
- Jobs!
- Joke for the Week
- About The Embedded Muse

This issue of The Embedded Muse is sponsored by Netrino.

You know it is possible to write reliable and maintainable embedded software. So why are there so many bugs in your company's firmware?

It turns out there a few dozen coding best practices that dramatically reduce firmware bug counts, in programs with or without an RTOS. Improve the bug reducing skills of your team at the hands-on Embedded Software Boot Camp. The next public session runs January 26-30, 2009. Full details are online at <http://www.netrino.com/Embedded-Systems/Training-Courses/Boot-Camp-Muse>

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 this and far more at my Better Firmware Faster class, presented at YOUR facility. See <http://www.ganssle.com/classes.htm> .

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Computerworld's annual salary survey is out:

<http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=326024> . It's for IT workers rather than embedded engineers, but the data is interesting nonetheless.

Last issue I reviewed Joseph Yiu's book "The Definitive Guide to the ARM Cortex-M3," and mentioned the various typos. Turns out there's an errata list available at <http://www.arm.com/miscPDFs/21948.pdf> .

My good friend and fellow ESC instructor Bill Gatliff has updated his embedded Linux training curriculum again. Now, instead of merely learning about using Linux in embedded applications, you spend a week adapting a Linux kernel and user applications to control a model train layout--- and you keep the hardware after class is over. Bill still covers device drivers, real-time scheduling, interrupts, web interfaces and other equally important details, but you get the added benefit of being able to apply that knowledge in a "real life" application before class even ends. For more info see <http://billgatliff.com> .

Dot Com Redux?

Nearly 100 issues of the Muse ago, back in number 76 in the dismal dot-year of 2002, I started running free job ads in an effort to help connect out-of-work engineers with employment opportunities. When the economy recovered readers asked me to continue these, so they remain a regular Muse feature.

There's only one posting in this issue for jobs. The embedded industry seems to have weathered the storm pretty well, so far, but it's hard to believe we're immune from the meltdown. What are you seeing? Will 2009 be a year of embedded layoffs?

More on Reuse

Greg Nelson responded to the comments in Muse 167 about reuse: I was out of town when EM 167 came in, but want to chime in two quick thoughts on the reuse issue that I think are very specific to embedded systems programming.

I've noticed that the modules that truly get reused without modification are often the sorts of things we would think of as library functions -- generalized hash tables, sorting routines, basic domain-specific math functions. When I try to think with the other parts do **not** get reused, it comes down to this: code reuse becomes a lot easier when you have commonalities in your hardware and operating system/development platform, a

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

luxury that is rare in the embedded market.

The area in which I've had the greatest struggles to reuse code has been porting across operating systems. In our company, reused or nearly-reused code has been used in systems ranging between OS-free control loop systems, roll-our-own OSes, VxWorks, vendor microkernels, Linux, and Windows. We've encountered some of our greatest reuse problems in the different ways the OS factors things.

For instance, `open()` and `socket()` API calls might return interchangeable handles on one system, and incompatible ones on another, and how you close them at the end-of-life might be the same or different. Is an RS232 port handled through an `open()` call, or another mechanism? How do you do a `select()` across your network and serial ports when they're not in the same name space? How do you deal with `sleep()` taking an argument in seconds on one kernel but milliseconds on another? Do you need to invoke a different set of functions for a threads-based implementation?

Compounding this (but clearly related) is the issue of working with a wide range of hardware. Parts of this shared code are running on everything from an 8051 to a 1GHz Pentium class processor, with big and little endian variations thrown in, etc. If everything had the same size of integers, lots of memory, and plenty of compute power, reuse would be easier.

Is there floating point support, or do you do tricks with integers to avoid the performance penalty of software emulation? Do you allocate 32 bits to every number and not worry about how they are passed around? But using too many 32-bit arguments on a 8051 processor may turn into an unacceptable performance drain; using 8- or 16-bit aligned data on an ARM may require special code to work around its de-facto alignment rules; and counting clock ticks on a PowerPC really needs 64 bits for even modest time intervals. Can you afford extra layers of wrappers which unify the OS interface, when some of your processors have very limited stack space?

Even with planning for reuse, some technologies change so fast that they outpace the lifetime of embedded products, like the display API that included an underutilized "color" field on a monochrome display "for future expansion." But it was built on a 16-bit system, and never considered the possibility of a >16 BPP color display in the next generation product.

Despite all this, we've been able to make *use* of a lot of existing code and interfaces, even though in many cases this consists of a framework that needs to get filled in with device-specific implementation.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

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. Please keep it to 100 words.

Despite the downturn, embedded systems consultancy Netrino (<http://www.netrino.com>) is still growing fast. The company currently has open positions for embedded software developers as well as digital and analog electronics designers in many parts of the U.S. The company's mix of development, consulting, and training work will entertain your brain and makes for a collegial learning environment. If you may be interested, please send your current resume to careers@netrino.com.

Joke for the Week

Nov 28: Moved in to my new digitally maxed-out Hermosa Beach house at last. Finally, we live in the smartest house in the neighborhood. Everything's networked. The cable TV is connected to our phone, which is connected to my personal computer, which is connected to the power lines, all the appliances and the security system. Everything runs off a universal remote with the friendliest interface I've ever used. Programming is a snap. I'm like, totally wired.

Nov 30: Hot Stuff! Programmed my VCR from the office, turned up the thermostat and switched on the lights with the car phone, remotely tweaked the oven a few degrees for my pizza. Everything nice & cozy when I arrived. Maybe I should get the universal remote surgically attached.

Dec 1: Had to call the Smart-House people today about bandwidth problems. The TV drops to about 2 frames/second when I'm talking on the phone. They insist it's a problem with the cable company's compression algorithms. How do they expect me to order things from the Home Shopping Channel?

Dec 8: Got my first Smart-House invoice today and was unpleasantly surprised. I suspect the cleaning woman of reading Usenet from the washing machine interface when I'm not here. She must be downloading one hell of a lot of GIFs from the binary groups, because packet charges were through the roof on the invoice.

Dec 3: Yesterday, the kitchen CRASHED. Freak event. As I opened the refrigerator door, the light-bulb blew. Immediately, everything else electrical shut down - lights, microwave, coffee-maker - everything. Carefully unplugged and replugged all the

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

appliances. Nothing. Call the cable company (but not from the kitchen phone). They refer me to the utility. The utility insists that the problem is in the software. So the software company runs some remote tele-diagnostics via my house processor. Their expert system claims it has to be the utility's fault. I don't care, I just want my kitchen back. More phone calls; more remote diag's. Turns out the problem was "unanticipated failure mode": The network had never seen a refrigerator bulb failure while the door was open. So the fuzzy logic interpreted the burnout as a power surge and shut down the entire kitchen. But because sensor memory confirmed that there hadn't actually been a power surge, the kitchen logic sequence was confused and it couldn't do a standard restart. The utility guy swears this was the first time this has ever happened. Rebooting the kitchen took over an hour.

Dec 7: The police are not happy. Our house keeps calling them for help. We discover that whenever we play the TV or stereo above 25 decibels, it creates patterns of micro-vibrations that get amplified when they hit the window. When these vibrations mix with a gust of wind, the security sensors are actuated, and the police computer concludes that someone is trying to break in. Go figure. Another glitch: Whenever the basement is in self-diagnostic mode, the universal remote won't let me change the channels on my TV. That means I actually have to get up off the couch and change the channels by hand. The software and the utility people say this flaw will be fixed in the next upgrade - Smart-House 2.1. But it's not ready yet. Finally, I'm starting to suspect that the microwave is secretly tuning into the cable system to watch Baywatch. The unit is completely inoperable during that same hour. I guess I can live with that. At least the blender is not tuning in to old I Love Lucy episodes.

Dec 9: I just bought the new Microsoft Home. Took 93 Giga-bytes of storage, but it will be worth it, I think. The house should be much easier to use and should really do everything. I had to sign a second mortgage over to Microsoft, but I don't mind: I don't really own my house now - it's really the bank. Let them deal with Microsoft.

Dec 10: I'm beginning to have doubts about Microsoft Home. I keep getting an hourglass symbol showing up when I want to run the dishwasher.

Dec 12: This is a nightmare. There's a virus in the house. My personal computer caught it while browsing on the public access network. I come home and the living room is a sauna, the bedroom windows are covered with ice, the refrigerator has defrosted, the washing machine has flooded the basement, the garage door is cycling up and down and the TV is stuck on the home shopping channel. Throughout the house, lights flicker like stroboscopes until they explode from the strain. Broken glass is everywhere. Of course, the security sensors detect nothing. I look at a message slowly throbbing on my personal computer screen: WELCOME TO Home-Wrecker!!! NOW THE FUN BEGINS ... (Be it ever so humble, there's no virus like the Home-Wrecker...).

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Dec 18: They think they've digitally disinfected the house, but the place is a shambles. Pipes have burst and we're not completely sure we've got the part of the virus that attacks toilets. Nevertheless, the Exorcists (as the anti-virus SWAT team members like to call themselves) are confident the worst is over. "Home-Wrecker is pretty bad" he tells me, "but consider yourself lucky you didn't get PolterGeist. That one is really evil."

Dec 19: Apparently, our house isn't insured for viruses. "Fires and mud-slides, yes," says the claims adjuster. "Viruses, no." My agreement with the Smart-House people explicitly states that all claims and warranties are null and void if any appliance or computer in my house networks in any way, shape or form with a non-certified on-line service. Everybody's very, very, sorry, but they can't be expected to anticipate every virus that might be created. We call our lawyer. He laughs. He's excited!

Dec 21: I get a call from a Smart-House sales rep. As a special holiday offer, we get the free opportunity to become a beta site for the company's new Smart-House 2.1 upgrade. He says I'll be able to meet the programmers personally. "Sure," I tell him.

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.

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.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.