

The Embedded Muse 158

Editor: Jack Ganssle (jack@ganssle.com)

March 25, 2008

You may redistribute this newsletter for noncommercial purposes. For commercial use contact info@ganssle.com.

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

- Editor's Notes
- Tin Whiskers
- Response to Great Engineers
- War Story
- Tools and Tips
- Styles
- Free Stuff
- Jobs!
- Joke for the Week
- About The Embedded 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> .

Are you in the Chicago or Denver area? I'll present a public version of the Better Firmware Faster class in Chicago and Denver, on April 23rd and 25th. Registration and other info here: <http://www.ganssle.com/classes.htm> . You'll earn 0.7 Continuing Education Units, learn a lot, and have more than a little fun.

I'll be teaching a public version of this class in London, UK, May 19. 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.

Tin Whiskers

The EU's RoHS standards have caused a wholesale retreat from the use of lead in solder. While their intentions are noble, the electronics industry is likely to suffer mightily since alternative solders spontaneously grow tin whiskers that can – and have – create short circuits. Much has been written about this subject and there's little for me to add. I do recommend Howard Johnson's discussion of the problem (<http://www.edn.com/article/CA6477864.html>) in which, among other things, he makes the claim that alternative solders are actually more harmful to the environment than lead. And there's great technical information about tin whiskers here: <http://nepp.nasa.gov/whisker>.

For those interested in this problem, Bob Landman sent information about a new resource: As a result of much concern on the subject of Tin Whiskers (being a much more severe problem in some applications than expected) there has been much dialogue on email between a number of interested and informed parties such as NASA, CALCE, as well as many military and telecommunications companies.

As a result of these various emails the group requested that a list server be put up to discuss areas of tin whisker issues and research.

The list server is now up and can be subscribed using the following information:

- Users can subscribe to the list by sending email to tinwhiskers-request@freelists.org with 'subscribe' in the Subject field.

- Once subscribed, to post to the mailing list, simply send email to tinwhiskers@freelists.org

- To unsubscribe from the list by send an email to tinwhiskers-request@freelists.org with 'unsubscribe' in the Subject field

List moderator is John Burke at <http://www.RoHSUSA.com>

Response to Great Engineers

Steve Litt of troubleshooters.com had interesting comments in response to my comments about great engineers last issue:

I contend that the guru is using a troubleshooting process. Sure, it looks to *us* like he sniffs, licks his finger and touches a node, and immediately discovers the problem, but

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.

what he really did was use a process to troubleshoot, including sophisticated use of divide and conquer. Either that or he's seen the symptom before and remembers the corresponding fix.

I'm not saying he isn't hyperintelligent -- he probably is. That's how he can, in his head, quickly devise diagnostic tests and keep a real-time mental map of what he's ruled out and what is still within the root cause scope. I'm not saying he doesn't have spectacular knowledge of the underlying technology -- he probably does. That's how he deduces what diagnostic tests prove what.

What I'm saying is this -- I've seen lots of brilliant expert builders and designers who couldn't troubleshoot their way out of a paper bag because they used no troubleshooting process. You show me a guy who can sniff, lick his finger and touch a node, and immediately discovers the problem, and I'll show you a guy using a troubleshooting process. He may deny he's using a troubleshooting process. He might be unaware that he's using a troubleshooting process. But he's using one.

This is a vital distinction because a significant chunk of technical people, and most of the nontechnical people who hire them, believe that troubleshooting productivity depends on intelligence and system knowledge alone. Belief in this myth causes ballooned MTTR (Mean Time To Repair), choosing wrong training, and bad employee/job fits.

The guru technologist, like the Olympic ice skating champion, makes the difficult look easy. But like the ice skating champion, he still needs to use the right techniques to produce the desired results. In the case of troubleshooting, that means following a process.

War Story

Andy Faithfull has his own war story:

I delayed starting my bachelors course for a year to go and work for an aerospace company and get some industrial experience. It certainly paid off when it came to practical and project work at college. Anyway, I spent most of that year building cards for a 6100 based system. The 6100 was chosen because it was available in a radiation hard form I think. The system was basically designed from the data sheets. Everything was built on prototyping boards, painstakingly wired the old fashioned way using single strand wire and soldered connections. Another project team in the company was using wire-wrap techniques to do something similar, but this was considered "new-fangled" in the 70s!

We built board after board like this. The ram boards (256 bytes x 12) had so many chips

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.

on them that we had to drill some extra holes in them and glue on a couple of DIL sockets. I also built a keypad and display unit for the beast, but it never got used I don't think. Eventually we had built all the boards that the engineer in charge had designed and the grand day came;- what would we do with the thing? Entering code was achieved with thumbwheels to set up the address and data, and a button to program the selected location. Only then did it dawn on us that some software tools might be useful as programming directly in machine code was tedious to say the least.

The enormous box of boards (now christened "Ernie" after the random number generator used by the government to draw the number of the winning government bond holder each month) sat gathering dust until one enthusiastic staff member started playing around with it. He wrote a program to play music on the thing by toggling the overflow flag, which came out to a pin. He laboriously coded up "The Entertainer", and everyone was impressed. Bearing in mind the effort that had gone into programming it, the machine was not powered down for some days. One night after everyone else had gone home, the lab cleaner asked me what the big blue box with all the lights on did. I very showily demonstrated Ernie's party piece, but something went wrong and as each note was played it seemed to get corrupted. After a couple of bars the only sound coming out was an unpleasant rasping sound. The cleaner left unimpressed, and I slunk away and pretended it never happened. So I would finally like to put the record straight after nearly 30 years. Trevor, I am really sorry. I don't know how I broke it, but it was me!

Tools and Tips

Dave Kellogg sent a veritable treasure-load of resources and links. He's big on podcasts; being one who spends little time in the car (and less exercising!) I have few opportunities to listen to them. But his list is great:

Here are some of the free education sources that I find very helpful. A lot of these are down-loadable podcasts. (Any software engineer that is not listening to a cast while commuting or exercising is missing a chance at self-improvement and increased professionalism.) Much of this is from the broader world of software engineering, rather than just embedded. However, the skilled embedded practitioner will be able to mine lots of valuable information. Several of these sources have an Agile/Lean slant, since I find lots of benefits in that method of software development.

Agile Toolkit Podcasts <http://agiletoolkit.libsyn.com/>
Developer Works Podcasts (IBM) <http://www-128.ibm.com/developerworks/podcast/>
IT Conversations <http://itc.conversationsnetwork.org/index.html>
Manager Tools <http://www.manager-tools.com/> (This is a MUST for me each week)
Software Engineering Radio <http://www.se-radio.net/> (academic, occasionally useful)
Lean Enterprise Institute <http://www.lean.org/Events/WebinarHome.cfm>

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.

IEEE Spectrum Radio Online

<http://www.spectrum.ieee.org/radio?date=01.07.07&segStart=1>

Software Quality Engineering <http://www.stickyminds.com/podcasts/>

Agile Project Management

http://community.featureplan.com/community/webinar_archive/

Net Objectives <http://www.netobjectives.com/resources/webinars>

CM Crossroads <http://www.cmcrossroads.com/>

Atomic Object <http://spin.atomicobject.com/embedded-corner/> (Interesting embedded work)

Methods & Tools <http://www.methodsandtools.com/> (A quarterly software development e-publication)

Quantum Leaps <http://www.quantum-leaps.com/index.htm>

Mary Poppendieck <http://www.poppendieck.com/>

Steve McConnell <http://www.construx.com/> (of Code Complete fame)

Applied Software Project Management <http://www.stellman-greene.com/aspm/>

Agile Management Science for Software Engineering (David Anderson)

<http://www.agilemanagement.net/index.html>

Dan Saks - <http://www.dansaks.com/articles.htm>

Embedded.com <http://www.embedded.com/columns/pp>

Jack Ganssle <http://www.ganssle.com/>

Software development using Theory of Constraints.

<http://finance.groups.yahoo.com/group/TOCSoftware/>

Lean Software Development <http://tech.groups.yahoo.com/group/leandevlopment/>

Lean Software Development and/or Scrum

<http://tech.groups.yahoo.com/group/leanagilescrum/>

Agile Embedded <http://tech.groups.yahoo.com/group/AgileEmbedded/> (Agile techniques are low overhead best practices used to develop software. Many of the agile techniques apply to embedded software.)

Google Video <http://video.google.com/videosearch?q=Google%20engedu> and search on EngEdu

IT Measurement and Productivity Institute (ITMPI) <http://www.itmpi.org/webinars/> (Lots of good topics about s/w development)

Zohair Ahmad sent: Another is at <http://www.eng-tips.com/index.cfm> and is definitely worth a visit.

Joost Leeuwesteijn contributed: A really good forum for embedded stuff:

<http://www.keil.com/forum/threads.asp> . Not just Keil specific info, and it has great info about C166, C51 and ARM architectures. T

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.

The Ganssle Group, www.ganssle.com

Gary Lynch added another ton of resources: I routinely bookmark tutorials and forums, primarily, because I find the best ones serendipitously and then can't remember them when I need them. Forums appear below. Items bulleted with 'G' are in German.

1.0 Embedded hardware

On my current project I use exclusively 8051-family parts from Silicon Labs. They offer a good forum for hardware questions:

<http://www.cygna1.org/scripts/Ultimate.cgi?action=intro>

... and a German, user-sponsored forum: G

http://www.c51.de/c51.de/Kommunik/Forum_frm.php?UIN=

I sometimes have to do schematic capture with Altium Designer, and rely on their forum:

<http://forums.altium.com/forums/> to sort out stuff that isn't in the manual.

2.0 Embedded software

My IDE is by Keil, and I sometimes consult their forum to understand its eccentricities:

<http://www.keil.com/forum/threads.asp>

Embedded control HW/SW/etc G <http://www.mikrocontroller.net/forum/>

3.0 Windows, PCs, Internet, etc.

MS-Access <http://www.mvps.org/access/>

MS-Excel G <http://spotlight.de/cgi-bin/newmessages.pl?maxcount=10>

General Wintel questions <http://www.computing.net/forums/>

Even broader PC questions <http://www.computerhope.com/forum/>

On-line security, spyware <http://forums.spywareinfo.com/index.php>

Everything up to here is on the world-wide web. Then there is the grand-daddy of all forums--UseNet. Most useful to our cause:

Hardware:

- <news://alt.engineering.electrical>
- <news://sci.electronics.components>
- <news://sci.electronics.design>
- <news://sci.electronics.misc>
- <news://sci.energy>
- <news://sci.engr.electrical.compliance>

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.

- news://sci.engr.electrical.sys-protection

Software:

- news://comp.arch.embedded

- news://comp.lang.awk

- news://comp.lang.c

- news://comp.lang.c.moderated

- news://comp.lang.c++

No employer of the Internet age has ever allowed me to access UseNet from the company LAN, but I can e-mail questions home, get answers and mail them back to myself. This may seem slow, but it's often faster than my next best option.

Styles

Jean LaBrosse, well known author of the uC/OS RTOS and several embedded books, tells me that no one can identify individual authors of his company's code. At first I thought he meant the programmers maintain anonymity out of embarrassment of their spaghetti mess, the usual situation in this industry. Sure, we beat the stuff into submission and finally shipped a product, but we slink away from the source whose original beautiful structure devolved into anarchy and chaos.

Everyone thinks they write pretty firmware. But it's worth reading the source to Jean's operating system. Calling it pretty is like saying Leonardo seemed to know something about art. Meaning fairly jumps off the page; it's not buried into the obscure naming non-conventions and mishmash of wandering spacing that's more typical of this business.

It's consistently beautiful partly because everyone writes code the same way. You can't identify the developer by coding style, simply because all use an identical style, one enforced by a comprehensive firmware standard. There's no variation in indentation, brace placement, or any of the other little differences that C encourages.

"Do you guys have the code police, then?" I asked. Turns out that it's mostly a matter of education. Many of their people are newbies recently out of college. Once exposed to this correct-think they accept it as apparently the industry-standard way to create programs.

Surely these folks will move on to other outfits over their career. Some will succumb to the pressure to forgo any styles in the mania to ship now. Like evangelists, though, surely at least a few will espouse the benefits and bring others into the fold of standardized styles.

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.

I've long believed in enforced coding standards. Yet in talking to thousands of developers over the years I've found that only a few percent of companies enforce any such standards. Though many do have a standard, few conform to it.

Why such resistance? In large part it's due to our self-image. We're like knights of old, girded for combat with the evil aggressor (the specification). Our tools are as crude as the knight's lance; usually nothing more than an editor and compiler. We win based on our ability to outwit, out-think, and out-perform the enemy. So our final product by its nature is a reflection of our superior abilities.

Using a standard, especially one as effective as Jean's, strips our software of all signs of individuality. The result is code that's absent of any sign of the author's quirks and prejudices. That sounds much like being just a coder in a vast software machine, but the challenge is solving the problem, not advocating some particular spacing convention.

One wag turned the tables on me. He made an analogy between coding and writing, and asked if authors (meaning me) should write equally "sterile" prose. Sputtering I protested that these were too very different things and, ah, style is in fact part of the nature of writing. Later I realized one difference between English prose and firmware is the longevity of the code. My writing is but a trifle which will soon disappear into obscurity. Firmware, though, lasts for decades. Writing is not held to any functional standard – it doesn't have to work, like software does.

For style guides, see:

- Jean Labrosse's guide is in his very worthwhile book *Embedded Systems Building Blocks*.
- http://www.codingstandard.com/HICPP_MANUAL_REQUEST.htm
- Chris Lott's site has many different standards and style guides, at <http://www.chris-lott.org/resources/cstyle/>
- [cs.washington.edu](http://cs.washington.edu/pub/cstyle.tar.Z) in `pub/cstyle.tar.Z` (the updated Indian Hill guide)
- [ftp.cs.toronto.edu](ftp://ftp.cs.toronto.edu/doc/programming) in `doc/programming` (including Henry Spencer's fun "10 Commandments for C Programmers:")
- [ftp.cs.umd.edu](ftp://ftp.cs.umd.edu/pub/style-guide) in `pub/style-guide`
- <http://www.ganssle.com/misc/fsm.doc>

Free Stuff

The last contest was so much fun I've decided to run another. Answer this puzzler correctly for a chance to win a free copy of *The Embedded Systems Dictionary* by Mike Barr and myself: What was the first C compiler used to develop a microprocessor-based embedded system, and when was it available? I'm not entirely sure of the correct answer so please include a link or other substantiation if you can.

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.

Contest rules: one winner only, who will be selected at random on or about April 4 from all of the correct entries; by "correct" I mean the answer that appears to be correct (i.e., fun is more important than ultimate Truth). Send your entry to marybeth@ganssle.com. Sorry, we can't acknowledge entries. I'll post the winner's name (but no other contact info) in the next Muse and any funny or interesting commentary.

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.

Haas Automation, in Ventura, CA, is looking for an experienced Electrical Engineer to design embedded products for CNC machine tools.

Provides technical assistance and training to others. Must be able to design analog and digital circuits, debug and prove them out. Must have programming experience including Assembly, C, and HDL. Must be able to design a circuit board layout with EDA tools. PCB layout experience a plus. Must be able to prepare detailed documentation of design. Must be able to work within planned schedules. Must be able to choose components based on extensive knowledge of tradeoffs in quality, cost, availability, and performance. Must be able to create systems level designs. B.S.E.E. or equivalent and 3-8 years experience.

Email "Scott Patchett" spatchett@haascnc.com

BJ Services Company in Tomball, Texas is looking for a Senior Firmware Engineer. Check us out at: WWW.BJSERVICES.COM

The primary role of this position is to develop embedded software to satisfy product functional and performance requirements and to develop and improve the systems, methods and processes used to accomplish the task. The Senior Firmware Engineer has technical responsibility for firmware functional requirements through the full product development life cycle from detailed analysis and design through verification and release. Must identify, analyze and resolve firmware and system functional deficiencies, and develop and implement corrective actions.

Desired Skills/Experience:

- * EE/CE/CS bachelor's degree
- * 3-6 years experience

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.

- * Experience with embedded C software development
- * RTOS concepts and low level programming
- * Knowledge of embedded digital hardware architectures
- * General knowledge of communication protocols
- * Software development on Windows and Linux platforms
- * Ability to port Linux to new platforms and produce required drivers
- * Experience with ARM & Coldfire processor development

Only candidates with experience in developing firmware using C and C++ with RTOS experience will qualify for the position. In addition, those candidates with digital hardware design experience will be given special consideration. Send resume to embedded@bjservices.com or fax to (281) 357-2716 Principals only no recruiters.

Draper Laboratory is an independent, not-for-profit engineering research and development organization. Its mission is to serve the national interest in applied research, engineering development, education and technology transfer. We are located in Cambridge, Ma.

We have an opening for a Software Engineer with 3 years embedded systems experience. Projects include working on micro-electronics systems, soldier portable systems and unmanned air and ground vehicles. Specific exposure to any of the following is a plus: PowerPC, TI DSP, GPS, audio signal processing.

To apply send a resume to nbeaumont@draper.com

Phoenix International, a John Deere Company, is seeking 6 to 8 embedded software engineers to contribute to the development of our product line of agricultural and construction equipment. Product lines consist of engine controllers, hydraulic controllers, navigation, electric motor control and displays. The ideal candidate will have experience in developing embedded C/C++ code on a variety of processor platforms (i.e. C167, ARMs, DSPs).

Desired Skills/Experience:

- * EE/CE bachelor's or master's degree
- * 3-8 years work related experience
- * Experience with embedded C/C++ software development
- * RTOS concepts and low level programming
- * Knowledge of embedded digital hardware architectures
- * Knowledge of communication protocols
- * Modeling tool experience is a plus but not required

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.

We are also looking for individuals with WinCE Display and BSP experience. Send resume to Phoenix International, Human Resources, at RadloffRenaeL@johndeere.com.

Joke for the Week

Did you ever want machine instructions like:
BRO (BRanch to Oblivion)
BRO (Branch on Power Off)
DMPE (Decide to Major in Physical Education)
EMSL (Entire Memory Shift Left)
HCF (Halt and Catch Fire)
JAA (Jump Almost Always)
OPP (Order Pizza for Programmer)
PON (Power ON)

Well, there's a pretty complete list here: <http://www.physics.ohio-state.edu/~bcd/humor/instruction.set.html>

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.

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.