

The Embedded Muse 88

Editor: Jack Ganssle (jack@ganssle.com)

November 11, 2003

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
- Toolchains Aren't Forever (Part 2)
- Jobs
- Joke for the Week
- About The Embedded Muse

Editor's Notes

Want to learn to design better firmware faster? Join me for a one-day course in San Jose on December 5. This is the only non-vendor class that shows practical, hard-hitting ways to get your products out much faster with fewer bugs. See <http://www.ganssle.com/classes.htm> for more details.

There's also cheap fly-in options listed there for folks coming from out-of-town.

I often do this seminar on-site, for companies with a dozen or more embedded folks who'd like to learn more efficient ways to build firmware. See <http://www.ganssle.com/onsite.htm>

Electronic voting is getting a lot of press lately; I've run a few columns on [embedded.com](http://www.embedded.com) about it, and plenty of other folks continue to badger vendors and government about the subject. I wrote a lighthearted piece about this a year ago (<http://www.ganssle.com/articles/thevote.htm>) which seems totally appropriate today.

Toolchains Aren't Forever (Part 2)

Last issue I wondered about the problem of maintaining old embedded systems. An awful lot of what we build stays in service for years and decades. Surprisingly often firmware

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.

changes are needed even in these ancient systems – how does one go about preserving toolchains for long periods?

That elicited a flood of responses. Several readers objected to my statement that CD-ROMs aren't forever; they feel that optical media will be readable for a very long time. Web searches seemed to confirm lifetimes of decades to 100 years or more. Maybe the media itself won't go bad... but will CD-ROM drives still exist in 10 years? The industry has shown a total lack of interest in backwards support of legacy media, such as 8" disks, 5.25 inch disks, paper tape, punched cards...

<http://www.nature.com/nsu/010628/010628-11.html> shows that CD-ROMs are subject to attacks by fungus, so save them in sealed plastic bags. I find salt sea air destroys the media in months. And

<http://www.informationweek.com/story/showArticle.jhtml?articleID=15800263&pgno=1> suggests that many failures of CD-ROMs can be attributed to the glue used to attach labels. Better: label the disks with a permanent marker.

Perhaps the ultimate backup is cuneiform on clay tablets. History has proven they remain readable after thousands of years.

Dave Ruske pointed out a paper written by Steven and Michale Gemeny (<http://www.cosmacelf.com/A%20Tale%20of%20Two%20Processors.pdf>) about their efforts to resurrect a 25 year old COSMAC ELF, 1802-based, microcomputer system. It's quite an interesting read (at least for geeks like me), which raises the question: what should NASA do to manage long-duration space missions? The New Horizons Pluto flight may require 20 years of support. We know all of the development tools will be consigned to the dust bin well before the spacecraft even encounters Pluto.

Another paper (not on the web) by one of the two authors is a more formal appraisal of supporting these long duration missions. The short summary: save everything, from engineering notebooks to all product documentation. Be wary of binary file formats (.txt is best). The most thought-provoking comment in the paper is "ask yourself, in 10 years, when I'm called out of retirement to fix this system, what will I wish I had saved?"

Robert Japenga had this to say about my pushing the use of a VCS:
"As much as I believe in version control, I have seen too much code lost in obsolete version control systems. Our company works with systems that still have running code that I wrote in 1975. And they are still maintained. We make changes to these systems. But at this one customer, there is a ton of code that is stuck on mag tapes readable only by Vax's and their version control. When Y2K hit, we couldn't find a way to extract most of the source code for most of those systems.

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.

“Our own version control system is simple, straightforward and DOES NOT touch the format of the source code. It does what it needs to do and nothing more. 30 years later, when the IT group has forgotten to port or cannot port the VCS to the new HW and OS - what do you do? Can you convince management to convert? I favor VCS as long as it DOES NOT change the basic format of the code.”

Jaime H. Chavez wrote:

”Being in field applications with customers and in my own development experience with IBM and Lucent over the last fourteen years, I have experienced headaches not only with the obsoleted software, but differences in functionality of tools as newer versions are released both in hardware and software. We're always scurrying for older versions of compilers, for example, and occasionally get requests for resurrecting and supporting old hardware. It almost leads me to believe that designing more generic hardware (less likely to become obsolete or easier to re-create/build) and use more of software to create features might be desirable for the long-term. In this manner, the active version control system can archive both the "soft-hardware" and software required for the project. Recommendations to archive ALL software required for build follow suit. All software would not only include the project source code, but, potentially, the development environment, compiler, etc as well as the "soft-hardware", if separate from the IDE.”

John Neubert hit on some toolchain issues that have always irked me:

”I am a Sr. embedded engineer for a large medical instrumentation firm. Since medical instrumentation must go through all the rigors of the FDA, we are required to be able to build ANY fielded version of any code at any time during the entire life of the instrument. This illustrates perfectly the need to maintain the toolchain. We have done everything you listed, from storing CDs to storing away entire development systems.

”I would like to bring to light a trend in the IDE suppliers' world which is making the maintenance of the toolchain much more difficult if not impossible. One example (and one very near to me right now) is Metrowerks. They offer two licensing options - node locked (to your NIC or hard drive), or networked. Networked licensing is a problem for embedded development because the embedded hardware may not always be near a network connection. I have mostly opted for the node locked flavor; however, this also has huge problems. First, I have had a hard drive crash, and needed to contact the vendor for a new license, I have upgraded to a new PC - same thing. And 10 years from now when I need to prove to the FDA that I can build version 1.0.0, do you think that there is any chance at all that I will be able to get a license file that will let me run? I certainly doubt it. I have asked for dongle license support but have gotten no response. And then there is the case where I move on to a new project using a different micro, and another engineer is designing with the chip that I have the IDE locked to my machine, now we

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.

have to again contact Metrowerks to get the license transferred.

”Furthermore, I should be able to load the program on as many machines as I like, as long as only one is in use at any time. I should be able to have it on my desktop for most of my development, and on a laptop for those field debug sessions.

”I certainly understand the need for the IDE folks to get paid for all their hard work, but I don't think that there is a large group of folks who would steal the IDE for the DSP56F807.

”I am not certain what the ultimate solution is, maybe your readers have other ideas, and maybe if enough folks complain about this push to node lock everything, industry will find a new way.

”Unfortunately, I just bought a new laptop, and now I get to transfer the 11 node locked licenses to the new machine - each transfer requiring me to contact the vendor in order to get a program that I own to work on my machine.”

He also noted that Orcad, Protel and others switched to a less restrictive licensing arrangement without apparent impact on their sales.

Bill Gatliff, well known GNU/Linux advocate for the embedded world, weighed in with a comment that the use of open source tools will increase long-term system maintainability. Clearly it solves the licensing problem. We corresponded a bit – here's some of our dialog:

Jack: “Will Linux 5.0 be anything like the current distribution? I'm wondering how much that product will evolve. We know Windows-of-the-future will be totally unlike Windows-today. Do you think future Linux will be more or less static?”

Bill: “Linux definitely won't be static. But it's based on a very mature API, so programs will probably run for a long time.”

”A widespread change in Linux that breaks tools isn't unprecedented. A recent change affected how shared libraries worked, and a lot of programs had to be recompiled/relinked to work right in the new OS (the change was in a critical runtime library, not the Linux kernel). Ditto for gcc's latest support of c++, which changed the ABI used by the compiler.

”That's where companies like Red Hat come in. You never saw the change with them; their Red Hat Linux 8 was based on the "old" way, and they rolled all the changes up into their Red Hat Linux 9 product. Same user functionality, very different "under the hood".

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.

”In all these cases, users weren't forced to switch to the new environment, and many of them probably haven't. Having the source code and redistribution rights for everything means you're in the driver's seat all the time!”

Jack: “I know you're thinking that saving the source means you can rebuild an old Linux, but that's sort of true even of a proprietary OS: save the binaries.”

Bill: “No, it isn't at all.

”A binary gets irreparably broken if the hardware or ABI it depends on changes. Source code can get fixed. I could run the GNU compiler on my PDA today (yep, done it!), for example, so I can only hope that today's PC architecture goes away as quickly as possible to remove the limitations of its legacy design. I wouldn't dare wish that on someone that depends on Win32 and a binary-only environment!

”If a distributor of a binary-only product goes away, users are left with "abandonware". If Linus or anyone else in the Free Software community gets hit by a truck, I pull out my CD and pick up where they left off. Huge, HUGE difference!”

Jack: “Philosophically, an environment designed to be stable over decades does solve a lot of problems. But I'm not sure if anything in this insane field can resist market pressures to "improve".”

Bill: “Stability creates problems, too. Look at just how messed up the PC hardware architecture is! With open platforms and Free software, the system evolves as its users see the need, because they aren't dependent on the attentions of Microsoft and Intel to do the "heavy lifting" for them!”

Chris Pitcher made a final comment that appealed to me. Old tools do go bad, capacitors fail, and tapes become unreadable. Chris said “All tools lose their edge when not in use. The longer you leave them on the shelf, the duller their edge.”

Amen to that!

Jobs

Plexus has openings for some digital and firmware folks in North Carolina and Colorado. See http://www.plexus.com/cgi-bin/r.cgi/frame_careers.html and then click on “current openings.”

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.

L-3 Communications Systems-East in Camden, New Jersey needs Hardware engineers, Software Engineers, and System Engineers. The domain of the company projects is Real-time, highly reliable embedded applications in minicomputers, microprocessors and digital signal processors.

Interested individuals that have questions or would like to submit their resume can email renee.hill@L-3com.com.

Echostar Communications in Denver has a couple of embedded positions open. Check out:

<http://jobsearch.monster.com/getjob.asp?JobID=19796538>

<http://jobsearch.monster.com/getjob.asp?JobID=19796461>

Let me know if you're hiring firmware or embedded designers. I'll continue to run ads for embedded developers as long as the job situation stays in the dumper.

Joke for the Week

Lazza wrote regarding a use for the Write Only Memory (data sheet at <http://www.ganssle.com/misc/wom.html>): It is the ultimate device for copy protection. The RIAA and MPAA will love it!

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*".

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.