The Embedded Muse 99

Editor: Jack Ganssle (jack@ganssle.com) July 12, 2004

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

Editor's Notes

There are no current plans to host a public Better Firmware Faster seminar.
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. for more details.

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.

When the economy took a dive and engineering employment tanked I started running (free) employment ads as a service to developers looking for work. Last issue I asked if the upturning economy meant these ads no longer had value for you readers. The response was overwhelming; of 419 replies all but one asked to keep the ads running.

Numerous readers did suggest that ads from recruiters be discontinued. So that will indeed be the official policy.

An article on embedded.com (http://embedded.com/showArticle.jhtml;jsessionid=124V2KDHLP5RKQSNDBNSKHY?articleID=21400577) suggests that many managers are reluctant to add costs despite the improving economic outlook. So hiring is expected to remain soft.

The IEEE is reporting that the number of EEs continues to decline. In the US 327,000 EEs were employed in the first quarter of 2004 versus 386,000 in Q1 2003. Unemployment for EEs went from 4.5% in Q3 2003 to 5.3% Q1 this year.

Unemployment rates for computer programmers rose from 4.6% to an unbelievable 9.0%.

Computer hardware engineers (whatever that means – are these EEs who work in embedded systems or folks who design PCs?) did much better. Unemployment fell from 9.0% to 4.9%.

It's still a tough market out there.

Debouncing

The last of my three part series on debouncing switches has run in Embedded Systems Programming. Check the articles out at:

http://www.embedded.com/showArticle.jhtml?articleID=18400810
http://www.embedded.com/showArticle.jhtml?articleID=18902552
http://www.embedded.com/showArticle.jhtml?articleID=22100235

Thanks to everyone for their ideas and thoughts.

Therac 25, Revisited

Most of us are aware of the Therac 25 disasters. A radiotherapy device suffered from a series of software errors that led to the deaths of 3 patients. Software can, and does, kill. The code was a nasty spaghetti mess and a home-brew RTOS was subject to timing errors. The classic report on this incident is Nancy Leveson's http://sunnyday.mit.edu/accidents/therac.pdf, which is well worth a read.

But that was 20 years ago and can't happen again… right? Maybe it can. In 2001 another radiotherapy device overexposed patients in Panama, leading to 20+ deaths. Debbie Gage and John McCormick wrote "We Did Nothing Wrong" (http://www.baselinemag.com/article2/0,1397,1543564,00.asp), a fascinating description of that failure. Their article is short on technical specifics but outlines the sometimes fatal consequences of software problems.

We're entering a new era where code kills. More of our systems are utterly reliant on software; in some of them an error can lead to death. I hear that some cars, like Toyota's Prius, have a brake-by-wire control. Hopefully that code is perfect.

My September column for Embedded Systems Programming (which may run earlier on embedded.com) looks at the how fire safety codes evolved in response to too many tragic deaths from blazes. I'm struck that the fire community looks at safety from two different but complementary perspectives. Fire Prevention comes first – designing a safe environment that's unlikely to burn, using fireproof materials and the like.

Then there's Fire Mitigation, making sure the people can escape, and keeping the flames from spreading rapidly. Sprinklers may stop the inferno outright. Panic bars that open locked doors help get people out of the burning structure and so reduce the impact of a fire.

Fire Prevention is analogous to careful software design, as well as a software engineering process that insures bugs don't make it into the code in the first place.

Exception handlers and "safe" modes are the software equivalents of Fire Mitigation. So if your system can beam dangerous radiation at a patient, there better be fast-acting mechanisms to shut the system down when the code goes bonkers.

Some studies suggest 2/3 of all system crashes are either due to lousy exception handlers, or could have easily been handled by them. Consider the Ariane 5 incident. A bug shut down the two redundant inertial navigation units. Both continued transmitting data to the main steering computer with diagnostic bits set, meaning "don't use this data, we're confused." The steering unit ignored the diagnostic bits, used the wildly incorrect data, and swiveled the nozzles hard to one side. Half a billion dollars of high tech garbage rained down over the Atlantic Ocean that day.

Spacecraft often have an entire sequence of safe modes that are invoked in case of trouble. If comm with Earth disappears, or the battery voltage drops unexpectedly, or any of a number of odd things occurs, the satellite might autonomously seek out the sun to insure the solar panels keep cranking out power. When things go bad it's not unusual for a spacecraft to drop the wide bandwidth link and try communicating via a low gain omnidirectional antenna that doesn't need precision pointing.

It's a brilliant – and costly – way to insure a mission isn't lost even if something totally unexpected happens. But firmware is the most expensive thing in the universe. Shortcut the safe modes, the awesome exception handlers, the parameter checking, and your code just may give you far too much fame on a future episode of 60 Minutes.

eXtreme Programming

Long time correspondent Nancy Van Schooenderwoert is one of the few folks I know who has had great success using a doctrinaire version of eXtreme Programming. I hear from plenty of people using an XP subset, but unmodified XP as preached by the experts seems, in the embedded field at least, largely unused.

What's your experience? How many of you are using XP on embedded work? Has it been successful? Are you using the version exactly as promoted by Kent Beck, or some variation?

I'll run interesting stories in a future version of the Muse.

In some circles XP has been elevated to the same religious status of Linux. Take a swipe at it and the flames roar. Though I find some of the XP ideas simply brilliant, others leave me cold for the bulk of embedded applications. What are your thoughts?

Nancy does maintain a list for people interested in discussing embedded agile issues. Signup at http://www.agilerules.com/mailinglists.phtml at the link "XP-Embedded Discuss".

Ron Morsicato and Nancy will speak at XP/Agile Universe Conference in Calgary, Alberta, Canada, August 15 - 18, 2004. Their talk is "Agile Methods for Safety-Critical Software Development." See http://www.agileuniverse.com/schedule/index.

Nancy will also present "Embedded Extreme Programming Experience Report and Clinic" at the Boston Embedded Systems Conference, September 13th to the 16th, 2004. See http://www.esconline.com/boston/. I'll be at the show, and hope to sit in on her talk.

Quantum Trap

Dave Kellogg sent along a link to an interesting datasheet for a unique kind of non-volatile memory. Simtek calls the STK14CA8 a "Quantum Trap," probably because the name looks so good on marketing documents. Regardless, it's a 128K X 8 conventional fast SRAM with a shadow 128K X 8 device made using some sort of non-volatile technology sharing the same address space.

Flash is a lousy place to store parameters that change frequently because it has a limited number of write cycles. The Quantum Trap uses the SRAM and then, only upon a power-down or a software initiated command, copies the SRAM to non-volatile memory. So your code can read and write all day long to the part, but will incur a write to non-volatile cycle only when power disappears.

What's unique about the STK14CA8 is the backup and restore operations all happen automatically. A 10 uF capacitor powers the device throughout the power-down save operation. Even if your 3 volt supply crashes to zero in a nanosecond the part stays alive for the 10 msec or so needed to completely save everything.

It's not cheap, listing for $17 to $24 in small quantities. But for some applications, this would be a cool part to use.

For more info see: http://www.simtek.com/product-information/datasheets/1M-PDF/Stk14cA8.pdf

Jobs!

Let me know if you're hiring firmware or embedded designers. I'll continue to run notices for embedded developers as long as the job situation stays in the dumper. No recruiters please.

Gentex Corporation in Zeeland, MI has an opening for an Embedded Software Test Engineer. They develop and manufacture advanced electro-optical products for the automotive industry in C using 8 and 16 bit microprocessors. They are looking for someone with experience testing embedded products. Job responsibilities include defining requirements, writing test plans, tracking defects, and developing automated tests. View www.gentex.com for more information or send a resume to debdv@gentex.com.

L-3 Communications Systems-EAST located in Camden, New Jersey, currently has several opportunities available:

- **Software** (Member/Senior Member Engineering Staff)
    - Embedded Software Intelligence
    - DSP Intelligence Applications
    - Secure Communications Applications
    - Key Management Applications
- **Hardware** (Member/Senior Member Engineering Staff)
- **Systems** INFOSEC(Information Security) Systems Engineers
- Program Management

Please send resumes to renee.hill@L-3com.com.

YORK International Corporation has an immediate opening for a Software Development Engineer IV. Follow the link to http://www.york.com/employment/postings, and scroll down until you find job number 04-059. This position is located in York, Pennsylvania, situated one hour west of Philadelphia and 45 minutes north of beautiful Baltimore, Md.

YORK International Corporation is a full-line, global provider of heating, ventilating, air conditioning and refrigeration (HVAC&R) products and services.

MOTO Development Group is looking for a Senior Embedded Systems Engineer/Project Lead/Technical lead engineering involving: architecting embedded systems, researching and selecting components, firmware development in assembly and C on a variety of 8-bit, micros and in C/C++ on ARM processors, implementation of TCP/IP communication protocols, board bring-up and systems, reviewing and modifying Orcad schematics, reverse engineering embedded systems, writing, implementing, and updating specifications. Project Lead responsibilities include: client and vendor communication, management of project / program schedule and budget, co-ordination of embedded systems projects and management of software, firmware, electronic and mechanical engineers, tracking and resolving issues to move production forward, new project proposal writing. BSEE/CS or equivalent with minimum of 10 years relevant technical experience and 5 years project management experience required. Experience in project/program management and embedded systems development.

Joke for the Week

We know about as much about software quality problems as they knew about the Black Plague in the 1600s. We've seen the victims' agonies and helped burn the corpses. We don't know what causes it; we don't really know if there is only one disease. We just suffer -- and keep pouring our sewage into our water supply.

- Tom Van Vleck

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.