The Embedded Muse 133

Editor: Jack Ganssle (jack@ganssle.com) Sept. 6, 2006

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

Editor's Notes

Want to become your company's embedded guru? I'll present my "Better Firmware Faster" seminar December 1 in Santa Clara, California. Check out http://www.ganssle.com/classes.htm for details. The site is just a few miles from San Jose airport, which is inexpensively served by Southwest Airlines and others. Come for the seminar on Friday and explore San Francisco over the weekend!

TV chef Emeril is always "kicking things up a notch" by adding garlic and other zestiness to his recipes. How are your firm's engineering recipes? Want to kick up your development processes by more than a notch? I can present my Better Firmware Faster class at your facility. See http://www.ganssle.com/classes.htm .

The Embedded Systems Conference will be in Boston September 25-28. It's always an interesting and fun event. I'm giving three talks. Some 150 exhibitors will be showing off their latest offerings. I'll be wandering the show floor; stop me and say "hello."

Preserving Design Decisions

"What the hell does this mean?" yelled Bill, his usual complacency violated by the peculiar behavior of a brand-new VHF radio. I looked at the LCD screen and saw the hand-held was in some sort of diagnostic mode, displaying "err 4C" whenever he pressed the transmit button.

I know he had agonized over the $300 purchase, his engineering attitude wanting to insure the radio was the best possible value for the money. We were with our families well outside of the USA with no email access and phone service priced out of sight for support calls. The manual documented quite a few error messages, but strangely not this one.

Weeks later, back home, Bill started wending his weary way through the frustrations of telephone support. “Your call is very important to us, please hold” pales after 30, 40, 60 minutes of Muzak. Finally a technician responded. Clearly baffled, after calling in his supervisor, the tech announced this meant the radio was “broken”. Under pressure he admitted that no one there really knew what the message meant, but was unusual so the radio required repair.

Sound familiar? A year earlier my autopilot stopped steering. It sounded three short beeps, paused, and then one more before repeating the pattern over and over. Clearly this was some sort of diagnostic code, one not listed in the manual, and one that tech support couldn’t identify. “Send it in for repair” was the oh-so-standard response to all such unknown embedded systems behaviors.

Yet surely the firmware engineers went to a lot of trouble to put these exception conditions into the code. They mean something; but that meaning was lost due to poor documentation. Why bother detecting and reporting the condition when users will be baffled?

It’s easy to tell the programmers to log everything they do, and to hope that the documentation and customer support folks will turn these notes into useful web FAQs and manual addendums. But that’s tough. Developers are busy enough trying to fulfill requirements and work out system bugs. Creating a support doc in parallel just ain’t gonna happen, perhaps unless one has an always-available tool where it’s easy to make notations. In a collaborative environment the best tool is probably a wiki, coupled with training and an iron will to log everything we learn.

Maybe we need a documentation scavenger. Someone empowered to dig through the code looking for things that impact the customer. A person that bridges the development, support, and doc teams.

Or perhaps, if we’re wise enough to be doing code inspections, one of the inspectors is always a customer/documentation representative. If you’re using eXtreme Programming, where a pair of developers share one computer, the person not typing could capture each of these new exceptions and insure they’re emailed to the doc team.

Some might insist that at requirements time we know all of the error codes and other intricacies of the system. In my experience, though, during coding we discover – and write code to trap - odd cases that should never happen, or that we'd hope would never happen. A three level deep IF can resolve into an awful lot of possible conditions, far more than we usually anticipate until we're deep into the unit design or code. That's when a lot of the error messages and exception handlers get written, and that's where they get lost to the customers.

I do remember committing the same sin early in my career. Our instrument solved a very complex series of simultaneous polynomials using an iterative algorithm. (And this was on an 8008!). At times no solution was possible, so I'd programmed it to give up after 20 minutes and display "Help" on the 7 segment LEDs. Years later, long after the device was obsolete and most of the original staff dispersed to other companies, a unit came back for repair. The young technician appeared at my office door, ashen. He had been adjusting internal pots when the machine started flashing "help" messages. He thought some intelligence inside the box was reacting to his repair attempts…

What do you think? How can we pass the critical error messages created by our exception handlers into useful customer artifacts?

Tools

Simon Large contributed this: If you want a short and sweet (and free) replacement for the Windows standard Notepad, try Notepad2 from http://www.flos-freeware.ch/notepad2.html. At 250k bytes it is bigger than Notepad, but still small enough to load quickly. Unlike Notepad it handles Unix line-endings (and Unicode/UTF-8), has syntax highlighting for quite a variety of formats, and generally behaves much better than the MS default tool.

Another tip for those who haven't found it yet is to add Notepad (or your favourite text editor) to the Send To list. Just create a shortcut to the editor in Documents and Settings\{USER}\SendTo. That way you can view any document type as text directly from a right click.

Christopher Meisenzahl wrote: I'm always on the lookout for a great, free text editor, and I've tried many. ;-)

My current favorite is this one. Pretty small, can be easily scripted, and allows column editing! http://www.context.cx/

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.

TI is looking for a Firmware Manager in Austin, Texas for our line of Mixed-Signal Microcontrollers tailored for fuel-gauging solutions. As Firmware Manager you will be responsible for the overall development as well as leading high caliber Firmware Engineers. Job responsibilities include:

- Overseeing Microcontroller Firmware architecture, design, & documentation

- Ensuring Firmware design reviews are held, and auditing verification plans

Qualifications:

- Background in Mixed-Signal Microcontroller Systems for embedded applications. Understanding of data acquisition technology.

- Experience in assembly language & C/C++

- Knowledge of compilers, debuggers and Integrated Development Environment (IDE)

- A BSEE with minimum of 8+ years of applicable Software/Firmware experience

Innovative Concepts (www.innocon.com) is looking for an Embedded Networking Software Developer. Work as a team member to develop real-time, embedded networking software with a high level of quality and reliability in a timely manner. Assist the team in developing cutting-edge mesh networking technology. Focus is on battery-powered ad-hoc networking devices. Collaborate to identify emerging technologies and synthesize them into the product roadmap.

Requirements: C & C++ development, IP and link-layer networking, Linux networking, kernel development, Object oriented design and analysis, Multi-threaded programming, Unit testing experience, Cross-platform development, Strong teamwork skills.

Please contact Heather OCallaghan- heathero@inclusioncs.com for more detailed information.

Joke for the Week

.

```
If architects had to work like programmers:
http://www.stsc.hill.af.mil/resources/tech_docs/gsam3/appene.pdf .
```

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.