The Embedded Muse 101

Editor: Jack Ganssle   (jack@ganssle.com)   Sept. 10, 2004

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
- Driving a Hybrid
- More on Debouncing
- Elvis Explains
- A Codewright Replacement?
- 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 Boston on November 1 or in Las Vegas December 10. 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 on the web site for folks coming from out-of-town.

But register soon - my able assistant cleverly negotiated free rooms for the first 15 people to sign up at each venue.

Las Vegas's casinos hold little interest for me but the city's gambling imperative seems awfully analogous to the state of too many embedded projects!

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.

I'll be presenting at the Embedded Systems Conference in Boston next week. Monday's all-day tutorial is called "Managing Embedded Projects." Tuesday, in a new talk, I'll discuss embedded disasters. There's lots to learn from the dramatic failures this industry has experienced. As George Santayana said, "Those who cannot remember the past are

destined to repeat it." Tuesday morning at the dreadful hour of 7:30 I'm hosting a Shoptalk (an informal discussion) about strategies for building reliable systems, and in the afternoon will talk about building real-time systems.

While not talking I'll mostly be wandering the show floor to see what new cool products are available. Access to the exhibits floor is free (more info here: http://www.esconline.com/boston/).

Stop by and say "hi!"

Driving a Hybrid

A few weeks ago we drove a Toyota Prius to Rhode Island and back. People have complained that these cars get closer to mid-40s MPGs rather than the 55 highway/60 city advertised, so I tracked the data.

Three of us with not much gear filled the car. We drove both ways with the air conditioner on most of the time, and we left Baltimore and Newport with a full tank.

The car averaged 54.9 MPG on the 385 miles north. The traditional way to measured MPG verified the dashboard computer's number within 2%. That drive was mostly at highway speeds of 55 to 70 MPH, though as usual Connecticut had some stop and go. It was sort of eerie arriving in Newport with nearly a half tank still left.

The return voyage yielded 53.1 MPG. Driving conditions were much different than Saturday's trip, though. RI and CT were both parking lots, adding 2.5 hours to our time. Speeds ranged from 0 to 70, with never-ending speedups and slowdowns. I expected the constant acceleration/deceleration to destroy gas mileage and was surprised - and puzzled - to see just numbers very much like those on the previous day. When crawling at under 15 MPH for one 15 minute interval the gas engine never came on at all, pegging the display at 100 MPG.

So why did we get such high numbers? I'm guessing it's either the radically improved car Toyota developed for the 2004 model year, or our driving styles. We're both gentle drivers and don't stomp either pedal. Marybeth averaged about 1 MPG better than I did, suggesting that driving styles do count.

Apparently the regenerative braking was so much improved in 2004 that most drivers will never actuate the hydraulic system; nearly all braking occurs by converting the car's kinetic energy into electricity.

Toyota replaced all of the older car's 8 and 16 bit microprocessors with 32 bitters to handle a much higher computing load. The braking, in particular, requires rather massive calculations.

I collect embedded disaster stories, and sometimes the collection grows morbid. The hybrid trip was a story of a different sort, one of a spectacular embedded success. It pleases me tremendously that a few square centimeters of silicon aided by brilliant firmware can save tanker-loads of fuel. My hat's off to hybrid designers everywhere. And we embedded designers should take pride in our industry, one that can offer tremendous environmental benefits. I believe that many of the global problems facing this stressed planet will be at least partly solved by a maze of embedded computers, quietly computing, efficiently using smart algorithms instead of brute force to reduce waste.

More on Debouncing

Steve Litt of Troubleshooters.com wrote about his experiences with debouncing, and makes some interesting comments about lubricants for electronics:

Over a year ago, on two of my motherboards, increasingly over time the boot process produced short, clicking, almost static-y beeps. As time went on, intermittently the boot would fail with a keyboard error. Finally, it would always fail with a keyboard error.

Replacing the keyboard would correct the problem, but soon enough would come the static-y beeps, then the intermittent failures, then the hard failures.

I banged my head against the wall trying to figure out the root cause. My first breakthrough came while discussing the problem with the vendor from whom I got the computer. I said "it's almost like the computer is somehow damaging the keyboards".

The second breakthrough came while listening to the static-y bell. I noticed it sounded like what you'd hear through your stereo switches while pressing a dirty tape monitor switch. That, in turn, reminded me that when I was a stereo repair man, I cured such problems by spraying lubricant into the switch.

Keeping the above in mind, I sprayed WD-40 into the keyboard and mouse holes on the computer, and the keyboard and mouse plugs that plug into those holes. The problem abated.

At first I attributed this solely to corrosion on the mouse and keyboard contacts, and explained the fact that it happened only on certain brand/model keyboards to the metal used in the mouse and keyboard connectors. Voluminous subsequent email discussions

and thought led me to believe the computer's debounce circuits also played a role. Certain brands and/or chipsets' debouncing circuits/algorithms were less able to handle a resistive connection to the keyboard.

All that happened over a year ago. Since then, I've incorporated electronic contact lubrication as preventive maintenance on all the computers in my business and household. There have been no further instances of keyboard caused boot failures. There's still the occasional static-y beep, but it never escalates to boot failure. I've also used lubrication to cure static-y sound on computer speakers by lubricating the mini-din plug with which they plug into the sound card.

Contact lubrication is now standard preventive maintenance here at Troubleshooters.Com. Whenever I install a mouse, keyboard, sound, video, network or printer cable, I lubricate. I lubricate every floppy cable, IDE cable, daughtercard and RAM stick. It's hard to say how many intermittents I've prevented, due to the nature of intermittents, but I'll continue to use this as preventive maintenance.

Not too long after the discovery I switched to Lube Job electronics lubricant, available at http://www.blowoff.com/lubejob/electlube.html. Lube Job is made specifically for electronics, whereas WD40 is paraffin (kerosene) based and therefore has lots of problems, including the low resistance mentioned.

I've also used or heard of the following being used for electronic lubrication:

- Stabilant 22
- Automatic Transmission Fluid (Dexron III)
- Break Free CLP
- ProGold G100
- Contact/Control Cleaner & Lubricant Radio Shack #64-4315

I recently used ProGold G100 to fix a nonfunctional L key on one of those incredibly high quality old IBM keyboards.

This link explains everything I know about contact lubrication:
http://www.troubleshooters.com/tpromag/200310/200310.htm

Thanks, Steve! In high school I worked as an electronics technician. One day a customer returned one of our instruments for repair. He had sprayed WD-40 all over the excruciatingly-designed low-leakage analog circuits. The WD-40 provided a low-impedance path to ground that no amount of cleaning could completely eradicate. We had to replace the PCBs. Since then I've always been wary of spraying this goop into circuits.

Elvis Explains

More than a few readers wrote about last issue's Elvis story, wondering why an oscillating databus might crash a system. The story is based on an experience I had (without Elvis). A rather slow CPU, maybe 10 MHz if I recall correctly, crashed occasionally due to these oscillations. The system was designed for these rather low frequencies. 2 layer PCB, no terminations, casual impedance matching. The oscillations ran at hundreds of MHz, dumping high frequency signals all over the board. They bounced around, slowly damping out, but meanwhile coupling into other signals, causing flops to clock when they shouldn't have.

I later saw a system self-destruct from the same cause! The oscillations caused parts to go into SCR latchup.

I have no idea where the original oscillations originated, but since the bus wasn't driven during the tri-state period it could have come from anything like shot noise, coupling from the outside world, or some chaotic combination of events on the board.

Since then I've terminated all databusses as a matter of course.

But the experiences were certainly eye-openers.

A Codewright Replacement

A reader wrote:

I'm a subscriber to both your Embedded Muse and the Embedded Systems magazine, and I have a question that I would love to ask the readers of the Muse.

We have a company policy that software we buy has to have a service contract, and with Borland buying and then dropping Codewright support (smacks of what happened with Brief...), we will no longer have access to it. Currently, we are using TextPad for editing (mostly due to cost of a site license), and the loss of features is annoying.

We don't use MS Visual Studio since we don't build Windows apps, and the cost of buying copies for everyone to use as an editor is not in management's plan.

What else is left and comparable in features to Codewright? I'd like to hear alternatives of what is out there and in the same price range.

What do y'all think? I'll reprint suggestions in the next Muse.

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.

Boeing has a number of entry-level openings for embedded software engineers. The programs hiring include Joint Unmanned Combat Air System, (J-UCAS), AWACS Block 40/45 (the modernization of the Air Force's mission processing system), Airborne Laser (ABL), Connexion by Boeing, 737 AEW&C (our surveillance platform being developed for Australia and Turkey) and Multi-mission Maritime Aircraft (MMA). See https://jobs.boeing.com/JobSeeker/JobSearch

Santa Cruz-based Plantronics, the world leader in communications headsets, is seeking Embedded Software Engineers for our Core Technology development group who have Firmware development and Wireless, Bluetooth, DECT or 802.11 experience. All interested persons must apply directly at Plantronics website at http://www.plantronics.com/careers. Completing the short questionnaire takes 2-5 minutes. Plantronics is an AA/EEO Employer.

Cerritos, CA-based Corelis, a worldwide provider of high-tech electronics and software tools, is seeking a highly skilled and experienced software engineer who will be an integral member of the team responsible for real-time firmware, drivers, and GUI software development for the entire CORELIS product line. The successful candidate will be writing C and C++ code for both Windows and Linux tools as well as firmware, and will undertake product responsibility from conception to development to production (including post-sale maintenance and support). A BSEE, BSCS or BSCE engineering degree and relevant experience of 5-10 years is required. Please send all resumes to Jobs3@corelis.com.

Joke for the Week

The acronym "TEAM" continues to inspire readers. Pearce Risto wrote in with this:

There is no "I" in Team, but it does contain a silent "scapegoat!"

And a true story from Jack – last week my wife and I were driving near Annapolis where we spotted one of those giant signs that display the temperature. It was a cool day for summer, but not quite what the sign showed: -196 degrees! I don't know what went wrong, but surely a wiser developer would have set reasonable bounds for what can be displayed.

If only I'd had a camera.

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.

The Ganssle Group, www.ganssle.com