The Embedded Muse 124

Editor: Jack Ganssle   (jack@ganssle.com)   Feb. 9, 2006

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:
- Editor's Notes
- Transistor-Free Computing
- Firmware is Cheap
- Yet More on Tools
- Jobs!
- Joke for the Week
- About The Embedded Muse

Editor's Notes

Denver and Dallas – Learn to developer better firmware faster in my one-day seminar in Denver on April 26 and Dallas April 28. Details at: http://www.ganssle.com/classes.htm. With luck there might be some decent skiing in the Rockies that week!

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 .

Transistor-Free Computing

Today we can't build an electronic alarm clock without stuffing hundreds of thousands of transistors into the thing. Ditto for any consumer product. Consider radios – once upon a time ads blared "Six Transistors," even when in some two were there only to provide marketing hype, as if this grand number conveyed lots of capability.

I recently wrote about vacuum tubes (http://www.embedded.com/showArticle.jhtml?articleID=175803580) – in their heyday few products needed more than a handful of tubes.

Any sort of computational device needs a sea of transistors, of course. Or does it? Check out http://acarol.woz.org/ , which describes a working difference engine, analogous to those Babbage conceived, made from Legos! The site contains lots of pictures and a pretty good description.

On a similar retro theme, C|Net has a great story about the history of hard drives (http://news.com.com/2300-1010_3-6031405-1.html?tag=ne.gall.pg) with lots of pictures. To me, "hard drive" doesn't necessarily mean "disk," though the article seems to disagree. Once upon a time computers also used drum memories, rotating cylinders whose operation was very much like a disk. The Univac 1108 at my college had a pair of Fastrand drum systems, each of which stored about 90 MB… and each of which weighed over two tons! More at http://www.fourmilab.ch/documents/univac/fastrand.html .

Firmware is Cheap

One of my mantras is that firmware is the most expensive thing in the universe. Though it has no weight, no real substance, the cost to create reliable, maintainable and documented code is astronomical.

But why is software so expensive? Surely it's because we create intricately-complex applications that are almost unrealizable in hardware.

Consider the following bubble sort, lifted shamelessly from Wikipedia and not checked for accuracy:

```
void bubblesort(int * A, int n)
{
    for(int i(0); i < n; ++i)
        for(int j(0); j < n - i - 1; ++j)
            if(A[j] > A[j + 1])
                std::swap(A[j], A[j + 1]);
}
```

It's a mere 110 non-space characters, perhaps tossed off in an hour or two. Suppose we didn't have software and had to implement a sort using discrete ICs. The cost would be astronomical.

Suppose you were tasked with building an MP3 player, with an LCD screen, from logic chips… or Legos. The circuitry/Lego blocks required to implement scrolling menus might fill a building.

A software version is comparatively cheap.

Every time I start to sketch out a circuit I'm sorely tempted to plop in a CPU. It's the natural solution, but breaks the rules of this game.

In an hour I managed a rough block diagram, one above the chip level (blocks have names like "adder," "16 bit latch" and the like). But the sequencing logic is clearly pretty messy so I've just tossed in a PLD, assuming at some point it wouldn't be too hard to write the appropriate equations. And, yes, perhaps that breaks the no-programmable-logic rule, but to design and debug all that logic using gates in any reasonable amount of time is as unlikely as buck-a-gallon gas.

Last year, for a school project my son was told to build a very simple scrambler that translated switch inputs to peculiar patterns of lights. The teacher expected a board made with lots of wires and diodes. I suggested he toss in a simple computer, in this case a Rabbit board, and write 10 lines of C. The result worked great, was more versatile than a hardwired version, and was done early. As a special benefit it totally ticked off his instructor who didn't expect such an approach!

Firmware *is* the most expensive thing in the universe, but only because of the unimaginable complexity of the problems it solves. It's just vastly cheaper than any alternative. So when your boss irritably asks why the software takes so long, answer "Compared to what?"

Yet More on Tools

The tool discussion that started a couple of issues ago is still generating a lot of response. Thanks, all!

Phil Schmidt wrote: "Here's yet another tool that people might find useful. Leo (http://webpages.charter.net/edreamleo/front.html) is a literate programming editor. It is difficult to describe what Leo can do without actually trying it out. I have used it for several different tasks, including code development, code reverse-engineering, a general notebook, and structured document creation. Its single most useful feature is the ability to clone nodes, and again, without trying it out, this concept is hard to describe. I recommend that folks take a look at it."

From Gennady Palitsky: Another addition to your great tools collection. As we all know, web research constitutes a significant part of any development. And information capturing and gathering is a big part of this process. I don't remember how many bookmark managers I tried - with no big success - until I've found the ContentSaver (www.macropool.com). It saves complete web pages or clippings on a fly, edits and annotates, organizes and categorizes, has great search options and many many more.

After I've tried it, I will never go to any bookmark managers. Not free, but saves so much time that easily pays for itself. Highly recommend it to anyone who spends any significant time for internet research (and I am not affiliated in any way with Macropool !). It does have a free trial (I used it for one day and decided to buy)."

Wil Blake contributed: "Embedded Systems Programming used to have link to a hex editor. This one works well. http://www.hexworkshop.com/ .

"If you must use Forth, a source code browser will help. http://www.mtechelectronics.com/pages/downloads/ ."

Vinay Patil likes Splint: "Here is another very good, and free tool called splint for checking C Syntax, Memory leak, Buffer overflow, Naming conventions checking(even customizable), etc... http://www.splint.org/ ."

Jan PleskaŠ says: "Our team use JEDIT, a Java based open-source editor which is very extendable and very supportive for development. We used it for C, VHDL, latex... More at www.jedit.org ."

Jobs!

Let me know if you're hiring firmware or embedded designers. No recruiters please.

Larson Davis, a manufacturer of sound/vibration measurement systems, seeks an electrical design engineer to join our team. Responsibilities include:
* Develop firmware for noise & vibration equipment. Design hardware and or test engineering design validation. Assist other engineers as part of a project team, and maintain existing designs. Program using a high level language
* Knowledge of embedded system design.
* Ability to design for EMI/EMC compliance.
* Digital signal processing and electrical design knowledge.

Applicant needs MSEE or PHD EE or equivalent with 5 years experience preferred. Full-Time with benefits. Qualified applicants respond to HR@LarsonDavis.com or fax 801-354-0109.

Embedded Software Engineer: Design and develop device drivers and board support packages around the Xilinx PowerPC and Microblaze processor offerings in the Xilinx

Embedded Development Kit (EDK). Integrate device drivers into Linux and VxWorks and develop hardware/software integration test applications. Work location in Albuquerque NM, San Jose CA, or Longmont CO.

Required Position Qualifications
* BSCS / BSCE / BSEE
* 5+ years of work-related experience designing and developing embedded software
* 3+ years of experience developing device drivers in RTOS environments (Linux and VxWorks preferred)
* Excellent communication (verbal and written) and teamwork skills

Desired Position Qualifications
* MSCS / MSCE / MSEE
* Experience in networking, communications, or other target application areas
* Experience developing Perl, Tcl, and UNIX shell scripts
* FPGA architecture knowledge and design experience

Contact: abq_epd_jobs@xilinx.com

Joke for the Week

From Catherine French:

Subject: Computer language problems

DIVORCE.BAT found....deleting C:\MARRIAGE\SPOUSE

hAS ANYONE SEEN MY cAPSLOCK KEY?

H lp! S m b d st l ll th v w ls fr m m k yb rd!

COMMAND: A suggestion made to a computer.

Hidden DOS secret: add BUGS=OFF to your CONFIG.SYS

ASCII a stupid question, get a stupid ANSI!

I am logged in, therefore I am.

I are Pentium of Borg. Division is futile. You will be approximated.

I came, I saw, I deleted all your files.

I was going to switch her to DOS, but she had a gun

I'll give up my Windows 3.1 when they pry my cold, dead fingers off the mouse.

If at first you don't succeed, call it version 1.0

In my next life I'm going to have more memory installed

It said, "Insert disk #3," but only two will fit!

Jesus saves! The rest of us better make backups.

Klingon prompt: strike any user when ready.

Manual Writer's Creed: Garbage in, gospel out.

Mental backup in progress - Do Not Disturb!

Multitasking - screwing up several things at once.

My software never has bugs. It just develops random features.

Nerd: someone on the Dork Side of the Farce.

Network: anything reticulated or decussated at equal intervals, with interstices between the intersections.

Never trust a computer you can't lift. - Stan Masor

Never trust a computer you can't throw out the window. - S. Hunt

Old programmers never die; they just give up their resources.

Programming Department: Mistakes made while you wait.

Programming is an art form that fights back.

Real programs don't eat cache.

Some programming languages manage to absorb change but withstand progress.

System has violated data integrity. Delete all data? Y/y __

The determined programmer can write a FORTRAN program in any language.

The generation of random numbers is too important to be left to chance.

The name is Baud... James Baud.

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.