The Embedded Muse 40

Editor: Jack Ganssle (jack@ganssle.com) December 1, 1999

New Embedded Books

Jean LaBrosse is back with an update of his “Embedded Systems Building Blocks” (R&D Books, ISBN 0-87930-604-1), hot off the press within the last week or two. I’ve always been a fan of his articles and books, and think this latest is a valuable addition to any embedded library.

The book is a collection of drivers for some of the more common embedded problems. It starts with a 40 page introduction to managing real time problems, largely in the context of using a RTOS. This section alone is worthwhile for anyone trying to learn about using an RTOS, though I’d also recommend getting his “uC/OS-II, The Real Time Kernel”

Included code (also on the companion CD-ROM) covers the following:

- Keyboard handler, for keypads arranged in matrices that are software scanned.
- Seven segment LED driver, for multiplexed arrays of LEDs
- A complete LCD driver package for units based on Hitachi’s HD44780 chip
- A time of day clock package, which manages time in an year:month:day and hours:minutes:seconds format. It’s Y2K compliant, to boot!
- If your real time code works with multiple delay and timeout issues, his timer manager is a useful chunk of code. It manages up to 250 software timers; all are driven off a single timer interrupt source. His example shows this running from a 1/10 second interrupt rate; I’d be interested to see if it can be scaled to higher rates on reasonably small CPUs.
- Discrete I/O drivers for inputs and outputs, with edge detection code. I like the way he abstracts the hardware to “logical channels”, which makes it so much easier to change things, and to create software stubs for testing before hardware is available.
- Fixed point math – one of the best discussions I’ve seen on this subject, which is critical to many smaller embedded apps. Fixed point is a sort of poor man’s floating point: much smaller code that runs very fast, but you sacrifice resolution and range.
- ADC and DAC drivers, with a good discussion of managing these beasts in engineering units rather than un-scaled bits.

An appendix includes Jean’s programming conventions, a firmware standard. I passionately feel that firmware standards are the starting point of writing decent code. An alternative standard is available at www.ganssle.com/misc/fsm.doc.

Much of the code is targeted at applications using the uC/OS RTOS. It's rather easy to port the code to any RTOS, and in many cases even to use it with no RTOS (though you'll have to delete some of the OS function calls).

When I read the first version of his book my gut reaction was "well, I could write this stuff easily myself." That's true; most of this code is not terribly complex. But why bother? Why re-invent the wheel? The best developers find ways to buy, recycle, and borrow code, rather than write every last routine.

Another new book is one I'll avoid reviewing… as it's my latest! "The Art of Designing Embedded Systems" (Butterworth-Heinemann, ISBN 0-7506-9869-1) is finally out, and is available from Amazon.com. The book is not an update to my earlier work; the two volumes are quite distinct and cover very different subjects.

Thought for the Week

Klingon Programming Proverbs:

Specifications are for the weak and timid!

This machine is a piece of GAGH! I need dual Pentium processors if I am to do battle with this code!

You cannot really appreciate Dilbert unless you've read it in the original Klingon.

What is this talk of 'release'? Klingons do not make software 'releases'. Our software 'escapes' leaving a bloody trail of designers and quality assurance people in its wake.

Debugging? Klingons do not debug. Our software does not coddle the weak.

I have challenged the entire quality assurance team to a Bat-Leth contest. They will not concern us again.

A TRUE Klingon Warrior does not comment his code!

By filing this bug report you have challenged the honor of my family. Prepare to die!

Our users will know fear and cower before our software! Ship it! Ship it and let them flee like the dogs they are!

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.