The Embedded Muse 66

Editor: Jack Ganssle (jack@ganssle.com) October 1, 2001

A Dirty Word

Yet another email arrived from a frustrated developer last week about how their project schedule collapsed, leading to the abandonment of the project and the likely firing of a few people.

A recent survey showed that 17% of all embedded projects are cancelled, always after months of work and accompanying schedule slippages.

Disaster comes from a number of sources. One that's all too common is deferring test till lots of the system is built. It's easy and exciting to crank code, lots and lots of it. Maybe we do some unit testing to insure functions work as expected.

But too many schedules and project plans include a dirty word, one whose use is a sure indicator of problems. "Integration" is the word, and it's worse than any four letter expletive. For when our schedules contain this as a milestone we're doomed.

"Integration" is when we combine parts of the system for the first time (perhaps the hardware and the software), and is nearly always a disaster. It's when the hard part starts. It's the first time we really see how badly things are going. Prior to this point it's pretty easy to convince ourselves that the project is on-schedule and all is well. At "integration" time, though, for the first time we're engaged in the very difficult part of making everything work together.

I always advise developers to find a way to start testing the system – as a system – as early as possible. Don't defer system-wide tests till some arbitrary "integration" point late in the schedule.

The problem, of course, is that firmware development necessarily lags the hardware. How can we do real testing till the hardware is ready? That will be late (it always is), so our testing is even later.

Remember that engineering is not so much about building projects as it is about solving problems. We KNOW the hardware will be late; even it it's not, it's never early enough in the cycle to allow is to start "integration" up front. So, before writing code, before finalizing the schedule, recognize that you'll need a decent test platform and find a way to get one. Early.

Clearly, there are some systems that simply must have an integration phase. If you build F-16 avionics it's awfully hard for each developer to have their own F-16… though it sure would be cool!

Some ideas:
Buy an evaluation board from the CPU vendor. These always have at least a minimal development system. You can write code, download, and test it. In many cases you can test and awful lot of the system without your unique product's I/O.
Use a simulator. Some of these have gotten really good. I was running one for a 68HC11 recently that was breathtaking. Sure, you can't simulate real time events, but it's at least possible to test quite a bit of the system.
Cross develop on a PC. Write your code in a totally ANSI-compliant manner so it can run on the desktop environment. One friend enables compile-time switches to use the PC's I/O (timers, serial, etc) in place of those in the embedded system. Again, much testing is possible, without a shred of target hardware ready.
Insist, while creating the schedule (long before problems surface) that the hardware group provide prototypes of risky I/O or other portions of the product. Maybe these are nothing more than wire-wrapped boards that plug into the PC's parallel or serial port.

The moral is to test early and often.

A Very Cool Animation

Check out http://www.ucos-ii.com for Jean LaBrosse's awesome new Flash animation of how context switching works in an x86-based RTOS. Fancy graphics have finally come to the embedded world!

Thought for the Week

NOTE: Failure is not an option!

It comes bundled with the software…

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.