The Embedded Muse 89

Editor: Jack Ganssle (jack@ganssle.com) November 24, 2003

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

Editor's Notes

Want to learn to design better firmware faster? Join me for a one-day course in San Jose on December 5. 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 there for folks coming from out-of-town.

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.

Even More on Watchdogs

I thought I'd written the definitive treatise on watchdog timers (see http://www.ganssle.com/watchdogs.pdf), till Steve Ciricillo corresponded last week with yet another vulnerability.

Amongst other approaches, the paper referenced above suggests building a state machine to tickle the WDT. Instead of simply calling a tickle_wdt() routine, generate code that looks something like:

```
main(){
   state=0x5555;
   wdt_a();
```

```
   .
   .
   .
   state+=0x2222;
   wdt_b();
}

wdt_a(){
   if (state!= 0x5555) halt;
   state+=0x1111;
}

wdt_b(){
   if (state!= 0x8888) halt;
   state=0;
   kick dog;
}
```

A crashed program that wanders into the tickle routine will fail, since the state machine will be in the wrong condition.

But Steve raised another interesting failure mode. Suppose the code crashes and for inscrutable reasons probably having to do with Murphy's Law and the perversity of nature vectors into wdt_b() just before the kick_dog command. The protection mechanism of the state machine won't help.

Perhaps it's safe to assume that the code will again crash when wdt_b() returns, so the system will miss the next watchdog tickle. But… perhaps not – who knows what evil lurks in the mind of runaway software?

Is this fear paranoid? You bet. But the WDT might be the last line of defense between deflecting the Earth-bound asteroid and utter disaster, or at least in rebooting the pacemaker before grandpa collapses. Assuming that crashed code will operate in any benign mode is naïve.

Perhaps a better approach is to modify the wdt_b() routine as follows:

```
wdt_b(){
   if (state!= 0x8888) halt;
   kick dog;
   if (state!= 0x8888) halt;
   state=0;
}
```

The double-check of variable "state" insures the system halts (so the watchdog can issue a reset) even if rogue code wandered into wdt_b() just before issuing the "kick dog" command.

The interesting thing about watchdog timers is that a proper design requires very little code. And yet they're surprisingly tricky, as a decent watchdog has to fire despite ANY sort of software failure.

Backups

I'm always astonished how badly so many of us manage our backups, especially in these days of continuous attacks against our computer systems. In recent Muses (http://www.ganssle.com/tem/tem88.pdf and http://www.ganssle.com/tem/tem87.pdf) I've discussed handling embedded projects over the course of the decades. One vital aspect of that is, of course, guaranteeing that the data stays intact.

A recent Microsoft study (http://www.itsecurity.com/tecsnews/oct2003/oct168.htm) suggests that over a quarter of small companies don't back up at all, and 40% do so less than once a month.

Even the Electronic Frontier Foundation was recently attacked (http://ftp.gnu.org/MISSING-FILES.README), and have suggested their FTP archives may have been compromised.

Do you backup regularly? Are your backups readable? Is *everything* saved? A glib "yes" isn't enough. Evaluate your backup strategy a couple of times a year. The alternative – losing source code – is too horrible and expensive to contemplate.

Steve Litt's site has some good info on backing up. See http://www.troubleshooters.com/tpromag/9807.htm, and (for Linux users) http://www.troubleshooters.com/lpm/200208/200208.htm.

We do have many backup media options today. CDs are cheap, but their capacity isn't all that great. DVDs are an interesting alternative - the media price isn't too bad, and it's cheap to mail the burned DVD to a friend in a different state for off-site storage.

Recently I've started to use an external 200Gb Maxtor disk drive for daily backups, coupled with weekly DVD burns. The Firewire connection provides is dramatically faster than burning a DVD. A number of contacts have claimed that the Firewire drives are as fast as internal disks, but my tests (using the nice Sandra benchmarks at http://www.sisoftware.net/) demonstrate that an internal ATA100 disk is about twice as fast as the external Firewire unit.

Joke for the Week

What should you do if your Internet connection goes down? Here's a belly-laugh full of advice: http://www.thetoque.net/031118/internetdown.htm

My favorite: "Spend time with your spouse - Communicating with your wife or girlfriend (sic) may seem like a radical suggestion, but the time investment may offer long-term rewards. Spending any amount of time talking about your "relationship" may free up more Internet time for you later on, when your ADSL or Cable link to the World Wide Web has been restored."

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.