

# The Embedded Muse 20

Editor: Jack Ganssle ([jack@ganssle.com](mailto:jack@ganssle.com))

April 9, 1998

## **Editor's Notes**

If your email in-box is like mine, you're overwhelmed by mail from recruiters looking for good people - or, ANY sort of people - for embedded work. A recent call caught me short when a fellow wondered if The Embedded Muse or the seminar series was a front for generating names to pass to recruiters. Perhaps it's time to state my policy: the mailing list is never sent to anyone under any circumstances. Further, commercial messages are never accepted. On those rare occasions I discuss a particular product, it's because I have found the product to be interesting, and wish to share my thoughts. The vendors are a wonderful source of information, but this forum is for rants and ramblings, not commercials.

TEM 19 talked about the issue of Discipline in firmware engineering, which prompted a lot of response from readers. Almost concurrently a couple of my articles in Embedded Systems Programming magazine discussed why a firmware development standard is so important. If you'd like a firmware standard, feel free to download one you can customize for your own needs from [www.ganssle.com/misc/ssm.zip](http://www.ganssle.com/misc/ssm.zip). Once unzipped it's a Word document.

## **An Embedded History - Part 2**

TEM issue 16 contained the start of a personal history of the embedded systems industry. Response was so favorable here's the next installment.

We left our story with the Intellec 8, an 8008-based development system that used nothing more than an ASR-33 as both "mass storage" (via the paper tape), and console.

I worked for a company that built grain analyzers - a scanning monochrometer beamed a spectrum of IR light at a sample of grain - wheat, barley, and the like. Detectors sampled the reflected IR, amplified it by numerous orders of magnitude, and digitized the signal before sending it to the 8008-based controller. The first computerized version had a 4k program - unbelievably tiny by today's standards, huge by those of the day. It took 16 ROMs (1702s, the only game in town at the time, had a whopping 256 byte capacity) to store the 4k of code. No kidding.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

Even with the dawn of the microprocessor age our code was as riddled with bugs as today - probably more so. The problems were much more tractable back then, as the limited address space and exclusive use of assembly language eliminated any possibility of building really complex systems, though we did have to implement all calculations in floating point, and even ran a least squares regression to calibrate the system, all within that 4k.

For debugging purposes, we cabled the Intellec 8's bus back to our own computer's backplane, and let the Intellec's CPU take charge of our peripherals. We edited, assembled, and linked on that Intellec, and then loaded and debugged the code.

Building a program was a nightmare, though in those days having exclusive access to ANY computer seemed pretty wonderful. Load the editor paper tape. Enter and edit the program. Punch a tape of the source program (all assembly, of course). Load the assembler tape. The assembler was a three pass nightmare - you had to run the source tape through three times before it spit out a binary tape. Then, load the linker and each binary tape.

The 4k program required 3 days to assemble and link. Three days. Needless to say, one reassembled only rarely.

The Intellec had a simple monitor (much like a scaled down version of the PC's DEBUG utility) that let you set two software breakpoints. We'd load the binary and debug, making changes to the machine code with each bug fix. The monitor didn't have a mini-assembler, so we quickly learned all of the machine codes for the 8008's instruction set - not much of a feat when you consider how limited that set was.

Sometimes we could patch right on top of the code. When the change was too big, we'd patch a jump to a "patch area", and hand assemble new code in there. With each change we'd make a careful record on the source listings.

After a day of debugging, we'd have lots of changes (hey - we were kids at the time, with no concept of software engineering!). We'd punch a tape of the current memory image and go home, reloading it the next day to pick up from the same spot. Only infrequently would we edit in changes and go through the pain of a total reassembly.

Development was incredibly slow. Every aspect of the process was buggy. At the time even Intel didn't understand how to program 1702s, so they'd frequently drop bits. We spent months perfecting the algorithm in concert with them.

Despite the troubles things worked, and we delivered hundreds of units over a couple of years. Time moved on and we later used better processors and algorithms; people left the company, replaced with others not familiar with the older products.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

The original units used an iterative regression, which, if it didn't converge within 20 minutes, displayed "HELP" in the seven segment LEDs. I'll never forget some years later a technician came in, ashen faced, and told me "I've been trying to repair this ancient unit, and after I fiddled with it for a while it started displaying a panicked HELP message!"

## Thought for the Week

### SILICON VALLEY LINGO:

"percussive maintenance" - the fine art of whacking a device to get it working

"prairie dogging" - in companies where everyone has a cubicle -- something happens and everyone pops up to look.

"blowing your buffer" - losing your train of thought.

"yuppie food coupons" - twenty dollar bills from an ATM.

"treeware" - manuals and documentation.

"beepilepsy" - afflicts those with vibrating pagers characterized by sudden spasms, goofy facial expressions and loss of speech.

"cobweb" - a WWW site that never changes.

"high dome" - egghead, scientist, PhD.

"elvis year" - the peak year of popularity, as in "1993 was Barney the dinosaur's elvis year".

"generica" - fast food joints, strip malls, sub-divisions, as in "we were so lost in generica that I couldn't remember what city it was".

"irritainment" - annoying but you can't stop watching, e.g. the O.J. trial.

"meatspace" - the physical world (as opposed to the virtual), also "carbon community".

"silliwood" also "hollywired" - the coming convergence of movies, interactive TV and computers

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

## **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](mailto:jack@ganssle.com).

To subscribe, send a message to [majordomo@ganssle.com](mailto: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](mailto:info@ganssle.com) for more information.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**