

The Embedded Muse 121

Editor: Jack Ganssle (jack@ganssle.com)

Dec. 19th, 2005

You may redistribute this newsletter for noncommercial purposes. For commercial use contact info@ganssle.com.

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

- Editor's Notes
- Tools
- Unmaintainable Code
- Jobs!
- Joke for the Week
- About The Embedded Muse

Editor's Notes

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> .

I find the history of this industry endlessly fascinating. Ars Technica has an interesting history about the development of the personal computer at <http://arstechnica.com/articles/culture/total-share.ars/> .

Thanks to everyone on this list (over 18,000 of you!) for your thoughts and discussion over the last year. I wish you health and prosperity in 2006... and a bug-free year!

Tools

I mentioned Beyond Compare as a fantastic compare tool. Readers had a few comments:

Vlad Pambucol wrote:

We use Beyond Compare here in our development and we absolutely love it! It has great diff capabilities (for instance I like that they show you where your diff is on the line not just the line that is different), you can re-sync manually your compare (we had to do this

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

for #ifdef branches of the code that we compared with original no ifdef-ed code), you can compare folders and many many more useful features.

... And speaking of tools (since I like this one a lot too for playing with hex or mot files) there is a software tool called Hexplorer that can open all sort of bin/hex formats and you can look in your files, you can do crc and checksums on block of bytes, you have ascii view and hex view of the file ... very useful tool.

See <http://sourceforge.net/projects/hexplorer/> .

Andrew Lin said:

Not a bad diff tool, but I like Araxis Merge better.

<http://www.araxis.com/> .

Admittedly, it costs a lot more, but it can do a three-way merge, which is very useful in a multi-developer environment when the VCS diff tool can't figure things out.

John Johnson contributed:

On tools I suggest that you look at the following:

1: One tool I have been using of late is a text editor: <http://www.vim.org/> developed under the GPL (hence free and user supported with one champion, a chap named Bram Moolenaar). vim (or gvim) has color coded code sense for C, C++, HTML, etc. and a difference utility. vim (gvim) is said to support regular expressions, through a user supplied plug-in though I have not yet enabled this capability. vim (gim) users have developed a host of other plug-ins for other purposes.

2: Another tool I have used, but not recently, is ProjeX, an inexpensive (\$US20) project management tool from <http://www.waa-inc.com/projex/index.htm>. ProjeX is an Excel plug-in, and in my opinion, does most of the work of the other project management tool that people actually need for a fraction of the cost.

3: I am looking for a good and inexpensive (free?) project management tool that produces simple PERT network diagrams. While ProjeX has PERT capabilities in the data it generates it does not (based on a look at the web site) generate the network diagrams. A Google search for PERT brought up the following tutorial (among other references) http://www.mindtools.com/pages/article/newPPM_04.htm

Any thoughts on that request anyone?

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Unmaintainable Code

A reader who preferred to remain anonymous reacted to the note about unmaintainable code. His story is pretty compelling:

Thank you for that link about unmaintainable code, it really made my day. The code fragment in "Jude the Obscure" was one the funniest things I've seen in a while. Replacing switch break statements with `if(0)` statements is truly demented.

The sad truth is that I've run into stuff like this for real. I was involved in the tail end of a fairly large (at least for us) embedded development project for an OEM customer. The design was architected and initially implemented by a group of contractors (we were staffing up for the project and lacked manpower and expertise at the beginning). Just about every good software development guideline was ignored or perverted in this project. I'm just guessing, but unless they were just totally clueless, the point of this seemed to be to create a gravy train of never ending work for the contractors. The actual result was to harden our attitudes (perhaps unjustly) against ever using contractors again. Here are some examples of what I mean.

- No RTOS was used although there were two, simultaneous, processes with hard, real time requirements being handled. The overhead of an RTOS was judged to be too expensive (in memory space and execution time), so instead each process in the project had to implement a state machine to only execute a portion of what it needed to do on each trip through the round robin workloop. We essentially implemented a manually managed, time-sliced multi-threaded process, except since it wasn't really time-sliced, it had to be hand tuned to meet timing requirements. Imagine how much effort it took to make that work reliably. Also try to imagine how difficult it was to add new functionality without bringing down the whole house of cards. To be fair, this decision may have been foisted on the contractors (I wasn't part of the group when this decision was made so I can't say who drove it).
- Modules looked at (and changed!) other module's state variables as a way of sequencing logic. One of the developers took undue liberties (even by these lax standards) with other modules to make his own job easier. This created bugs when those other module were modified. A big part of the refactoring that happened later was to remove these unnecessary linkages by just having a module ask via a function call if something was OK to do or to request the other module do something. What a concept!
- Just about all the variables were global and declared in one file named...you guessed it, `glodef.h`. Variables were grouped by function (Display, Com Port etc) and extern declarations were conditionally accessed by any module that wished, by including a

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

`#define` for that variable groups `#ifdef`. This file contained both assembly and 'C' variable definitions and was included in every module in the project (both assembler and 'C'). Tricks were played with comment blocks to prevent the compiler from seeing the assembly statements and the assembler from seeing the 'C' statements.

- All assembly code variable and function names were 6 characters long even though the assembler supported long names. The claim was that it made the code easier to read if everything lined up. This was from the same guy who thought variable names like `wbnpzspt` and `wbnpzfpt` are a good idea (neatly formatted, but totally incomprehensible).

- Hungarian naming convention was seriously misused. This project had variations that ran on different processors (8 and 16 bit) and the desire was to have a single set of source code conditionally compiled for each version (not a bad idea in itself). So all variable declarations were run through a secret decoder ring to deal with hardware and compiler differences. One interesting outcome of this was the `FUCBYTE` definition (an 8 bit, far unsigned char). I think this particular definition was just some passive-aggressiveness leaking out from the contractor involved (he seemed like a deeply unhappy man and I guess he was taking his pleasures where he could).

Here is a real, if trivial example from the code to demonstrate how deeply the commitment to proper code obfuscation ran in these guys:

```
for (wbnpzcnt = OTBCCNT, wbnpzfpt = nvNVbuff + NVPAGSZ, wbnpzspt =
nvNVbuff;
    wbnpzcnt != 0 ;
    wbnpzcnt--, wbnpzfpt++, wbnpzspt++ )
    *wbnpzspt = *wbnpzfpt ;
```

Note: The `wbnpzcnt`, `wbnpzfpt` and `wbnpzspt` variables are all LOCAL and are not used for any other purpose in the function, my eyes glaze when I recall it. This (along with a number of similar code fragments) was replaced with calls to that *<sarcasm>incredibly esoteric</sarcasm>* lib function `memcpy`.

P.S. Despite (perhaps as a result of) significant turnover in the developers mid-project, the project actually ended up going very well. We launched the product a month ahead of schedule and it was one of the most profitable products in my company's history. Software maintenance was pure hell for about a year, but we had a great software testing group and decided to bite the bullet and refactor mercilessly at every opportunity which greatly reduced the pain level.

Jobs!

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

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

MITEQ Incorporated located in Hauppauge, New York has a unique opportunity available for an experienced Embedded Systems Engineer.

To contribute to our dynamic, professional environment, you will need a BSEE, with 5+ years experience in Firmware design. Only candidates with experience in developing firmware using C and C++ with RTOS experience will qualify for the position. Experience developing SNMP, Telnet and embedded web servers is preferred. In addition, those candidates with digital hardware design experience will be given special consideration.

We offer a competitive compensation and benefits package. To explore this opportunity, please email your resume to: amarinozzi@miteq.com or fax to (631) 439-9216

Offshore electronic design company in the Caribbean is looking for an experienced power electronics designer with preferred knowledge of embedded programming in the area of power factor correction. This would be a consulting job for this project specifically with the possibility of more future work. Contact patrick.yeppez@avatargroupstgo.com

That one sounds pretty interesting to me on this cold December day!

Joke for the Week

Here's a site of funny T-shirts for that programmer in your life: <http://www.computergear.com/codetshirts.html> . It's a hoot!

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

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

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.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

The Ganssle Group, www.ganssle.com