

You may redistribute this newsletter for noncommercial purposes. For commercial use contact [info@ganssle.com](mailto:info@ganssle.com). To subscribe go to <http://www.jackganssle.com/lists/?p=subscribe&id=1> ; to unsubscribe see the end of this email.

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

CONTENTS:

- Editor's Notes
- Quotes and Thoughts
- Readers Write
- Jobs!
- Joke for the Week
- About The Embedded Muse

Editor's Notes  
-----

Did you know it IS possible to create accurate schedules? Or that most projects consume 50% of the development time in debug and test, and that it's not hard to slash that number drastically? Or that we know how to manage the quantitative relationship between complexity and bugs? Learn this and far more at my Better Firmware Faster class, presented at your facility. See <http://www.ganssle.com/classes.htm> .

The Embedded Muse will go on hiatus for the (northern hemisphere) summer after this issue.

Quotes and Thoughts  
-----

"There is only one boss: The Customer. And he can fire everybody in the company, from the chairman on down, simply by spending his money somewhere else." -Sam Walton

Readers Write  
-----

John Carter wrote: "An error, but a certainly not an impossible value. From the glibc docs...

```
"-- Function: int open (const char *FILENAME, int FLAGS[, mode_t MODE])
The `open' function creates and returns a new file descriptor for the
file named by FILENAME. Initially, the file position indicator for the
file is at the beginning of the file. The argument MODE is used only
when a file is created, but it doesn't hurt to supply the argument in
any case.
```

"The normal return value from `open' is a non-negative integer file descriptor. In the case of an error, a value of -1 is returned instead. In addition to the usual file name errors (\*note File Name Errors:), the following `errno' error conditions are defined for this

function:

```
"Perhaps a saner API would have been
"Function: errorCode_t open (const char *FILENAME, int FLAGS,
                           unsigned int * fileHandle[, mode_t MODE])
```

"The sad thing is I have yet to meet a C programmer who knows and understands all the "Usual Arithmetic Conversions" as specified in the C99 standard.

"I highly recommend stashing a pdf of the "The New C Standard: An Economic and Cultural Commentary" by Derek Jones on your disk. It's a most curious book and an invaluable aid when a touch of "Language Lawyering" is required: <http://www.knosof.co.uk/cbook/cbook.html>

"Suppose you have a "Real World (TM)" always and forever positive value. Should you represent it as unsigned?

"Well, that's actually a bit of a step that we tend to gloss over...

"As Jones points out in section 6.2.5 the real differences as far as C is concerned between unsigned and signed are...

" \* unsigned has a larger range.

" \* unsigned does modulo arithmetic on overflow (which is hardly ever what you intend)

" \* mixing signed and unsigned operands in an expression involves arithmetic conversions you probably don't quite understand.

"For example I have a bit of code that generates code ... and uses `__LINE__` to tweak things so compiler error messages refer to the file and line of the source code, not the generated code.

"Thus I must do integer arithmetic with `__LINE__` include subtraction of offsets and multiplication.

\*\* I do not care if my intermediate values go negative.

\*\* It's hard to debug (and frightening) if they suddenly go huge.

\*\* the constraint is the final values must be positive.

"Either I must be `_very_` careful to code and test for underflows `_before_` each operation to ensure intermediate results do not underflow. Or I can say tough, convert to 32bit signed int's and it all just works. I.e. Line numbers are constrained to be positive, but that has nothing to do representation. Use the most convenient representation.

"C's "unsigned" representation is useless as a "constrain this value to be positive" tool. E.g. A device that can only go faster or slower, never backwards:

```
unsigned int speed; // Must be positive.

unsigned int brake(void)
{
```

```
--speed;  
}
```

"Was using "unsigned" above any help to creating robust error free code? NO! "speed" may now always be positive... but not necessarily meaningful!

"The main decider in using "unsigned" is storage. Am I going to double my storage requirements by using int16\_t's or pack them all in an array of uint8\_t's?

"My recommendation is this...

```
" * For scalars use a large enough signed value. eg. int_fast32_t  
" * Treat "unsigned" purely as a storage optimization.  
" * Use typedef's (and splint (or C++)) for type safety and accessor  
functions to ensure constraints like strictly positive. E.g.  
typedef int_fast32_t velocity; // Can be negative  
typedef int_fast32_t speed;    // Must be positive.
```

```
typedef uint8_t dopplerSpeedImage_t[MAX_X][MAX_Y]; // Storage  
optimization
```

Jonathan Graham Harston responded to this: "

```
"Yes and no, unfortunately bureaucrats get involved, so they will  
> stipulate Comp Sci qualifications, and to make their life easier to  
> get this agreed to by bodies like lawyers, insurance companies  
> (even your professional indemnity), it would become mandatory.
```

"On this side of the pond we have a thing called a European Computer Driving License that more and more organisations are insisting employees have.

"I have lots of issues with it. Firstly, its not a license. A license is contractual or statutory permission to perform an act, in the absence of which performing the act is illegal or breaks the contract. The ECDL is /not/ a license, it is a certificate of competency. I have no idea how it managed to get past various trading standards laws as its very name is fraudulent misrepresentation.

"Additionally, it doesn't test IT literacy, or computer use competency. It tests the ability to use certain Microsoft products in very restricted ways. It doesn't test your understanding or word processing or using a spreadsheet, it tests your use of MSWord and MSExcel, as though driving lessons teach you to drive a Ford Escort instead of how to drive a car.

"There is a standardised test application that I played with and I got as far as "How would you save this document?" I duly pressed Ctrl-S. "Wrong!" it berated me. The only way to get the question correct is to move the mouse pointer to the "File" menu, open the menu, drop down to the "Save" command, click on it.....

"And the multiple choice questions with no correct answer: How many bytes in a kilobyte? a) 1000 b) 10000 c) 100000 d) 1000000. Argh!"

Asbjorn Saebo wrote:

> LccWin32 is a DOS C/C++ compiler that works on WinXP is available  
> from a number of sources.

"lcc-win32 has been subject to a few discussions on the comp.lang.c  
usenet group. Searching the archives for this group might turn up  
useful information for those interested in trying out this compiler."

Ebbe Kristensen took issue with the following statement:

> LccWin32 is a DOS C/C++ compiler that works on WinXP is available  
> from a number of sources.

"While Lcc-Win32 probably is an excellent C compiler it is nowhere near  
being a C++ compiler. It does support a few C++ like constructs such as  
operator overloading and generic functions but that's about it.

> "The C compiler in Cygwin does a good job as well but you have work  
> in a Cygwin window outside of the DOS/Windows environment making  
> printing difficult.

"Not true. The Cygwin C/C++ compilers can easily be used in a normal  
Windows command shell such as cmd.exe or be invoked from a capable  
editor such as EMACS or SlickEdit. If the "printing" problem, John  
refers to is "printing on a printer", then yes, this is not easy.  
Printing to the console (using printf etc.) works as expected in  
cmd.exe.

> And Cygwin come with its own baggage, i.e. it is a  
> full Unix implementation."

"It is that. The executables need access to the Cygwin DLLs for any  
kind of OS access. But for QaD programs that are only run on the same  
PC they are created on, this is not really a problem. On the other  
hand, Cygwin may be used for commercial use."

Paul Carpenter wrote: "Recently having struggled with stupidities of  
various IDEs and environments, some enforced behaviours made me extend  
what we corresponded about after the last musings..."

"Use spaces to make things more readable:  
file=open ("file.txt",r);

"is LESS readable than:

```
file = open( "file.txt", r );
```

"Please - Put spaces around operators, then you can distinguish  
between:

```
- 1      Subtract 1  
-1      negative 1  
  
&a      Address of A  
& a     And a
```

\*b Contents pointed to by b  
\* b Multiplied by b

"Put function parenthesis, array square brackets next to the name, to distinguish variables from functions and arrays of variables, e.g:

open( ... ) not open ( ... )  
array[ index ] not array [index]

"You do this for structures (hopefully):

a.name not a .name  
a->name not a -> name

"Also if your function has a void list use:

do\_something( )

"put a space between the parenthesis, so in whatever font at whatever size it is OBVIOUSLY a function with empty parameter list.

"With some fonts ( ) can be misinterpreted as 'O' Letter O or '0' zero.

"(Yes I know that some integrated environments like VB6 have the silly method of not knowing at a glance what a variable and a function is).

"Put spaces around VARIABLES and after commas, e.g:

x = y \* floor( a, b );  
  
not x=y\*floor(a,b);

"Then it is easy to determine operators, functions and variables/parameters in ANY font at ANY size.

"Use spaces, not tabs, since different tools/platforms have different default or minimum size for tabs, so what looked Ok printed or viewed on the screen of one system is a complete mess on another.

"Spaces at end of lines: Use editors/tools that either automatically or manually do this. Printing can often get messed up as these extra spaces are wrapped onto more lines.

"Your possible one function per page coding standard gets screwed up. Also makes file comparison for changes (diff) easier.

"Choice of letters: Avoid using -

3	Three	B	Letter B
8	Eight	D	Letter D
5	Five	l	Letter L
0	zero	i	Letter I
1	One	S	Letter S
2	Two	Z	letter Z

"...in names as they become confusing as some combinations are VERY confusing in some fonts consider 'ill' (also means unwell) in many fonts this can look like THREE lower case letter L.

"For an example read the license number label for Microsoft OS on

computers from at least 0.45m (approx 18 inches), without magnifiers.

"Avoid Capital 'I' in long variable names as it is often missed or interpreted as an 'L'. Often easy to forget and not notice the difference in case sensitive languages. e.g.

OpenTapStateHotWhenIFeelLikeIt is NOT obviously different to OpenTapStateHotWhenFeelLikeIt or OpenTapStateHotWheniFeelLikeIt.

"DO not end variables or functions in NUMBERS. This is partly font related ending variables or functions in numbers can cause confusion with Letters 'l' Lowercase L, Upper or Lower case 'I', 'S' Uppercase S, 'Z' Upper case Z.

"International Spellings: Avoid names that can be spelt in different dialects/languages differently for the same thing. Examples being US and non-US English for:

color	colour
capitalize	capitalise
authorize	authorise

"Expect your code to be printed!!! Think what you code looks like printed as plain text! Keep it if possible at 80 columns (start code on LEFT HAND MARGIN):

```
int main( argc, *argv )
{
char a[] = "hello";
char b[] = "world";
char end[] = "\n";

printf( a );
printf( b );
printf( end );
}
```

"NOT

```
int main( argc, *argv ) {char a[] = "hello"; char b[] =
"world";

char end[] = "\n";
printf( a );
printf( b );
printf( end );}
```

"Assume a FIXED WIDTH font (non-proportional).

"Functional Blocks: When you have a functional block of code make all lines INCLUDING COMMENTS line up. Do NOT rely on visual tools in a particular editor or IDE to so this all for you. Even a version change on the tools could change how easy this is to do.

"Only put comments to line up on LEFT hand margin after several indentations as MARKERS for debugging!

"Please also line up the block marker characters with the code:

```
for( a = 0; a < 10; a++ )
{
  b = b + ( a * b );
  c += d[ a ];
}
```

"Is easier to quickly read than:

```
for(a=0;a<10;a++) {b=b+(a*b);
                  c+=d[a];
}
```

"Use Signed and Unsigned Correctly: Too many languages and systems expect everything to be signed, as an indication of LAZY development, and reminds me of tales of bad data models that only worked for modeling the wildlife of Africa if certain areas were allowed to have negative numbers of Antelope. Nothing can have NEGATIVE mass, it can have negative weight.

"Understand the difference between absolute and relative units. Even temperature is relative as a unit, Celsius and Kelvin are compared to the triple point of water."

Jobs!

-----

Let me know if you're hiring firmware or embedded designers. No recruiters please, and I reserve the right to edit ads to fit the format and intents of this newsletter. Please keep it to 100 words.

Circuitlink International (Sydney, Australia) is looking for an experienced embedded C/C++ developer to join our growing team. We build a range of automotive related products and the role will involve working on new projects as well as adding new features to existing ones. Please contact Peter McConaghy - [peterm@circuitlink.com.au](mailto:peterm@circuitlink.com.au) - for more information.

Joke for the Week

-----

A song to program by:

Eleanor Rigby  
Sits at the keyboard  
And waits for a line on the screen  
Lives in a dream  
Waits for a signal  
Finding some code  
That will make the machine do some more.  
What is it for?

All the lonely users, where do they all come from?  
All the lonely users, why does it take so long?

Guru MacKenzie  
Typing the lines of a program that no one will run;  
Isn't it fun?

Look at him working,  
Munching some chips as he waits for the code to compile;  
It takes a while...

All the lonely users, where do they all come from?  
All the lonely users, why does it take so long?

Eleanor Rigby  
Crashes the system and loses 6 hours of work;  
Feels like a jerk.  
Guru MacKenzie  
Wiping the crumbs off the keys as he types in the code;  
Nothing will load.

All the lonely users, where do they all come from?  
All the lonely users, why does it take so long?

About The Embedded Muse

-----

The Embedded Muse is a newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to me at [jack@ganssle.com](mailto:jack@ganssle.com).

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.