The Embedded Muse 110

Editor: Jack Ganssle (jack@ganssle.com) Feb. 17, 2005

EDITOR: Jack Ganssle, jack@ganssle.com

CONTENTS:

Editor's Notes

Want to increase your team's productivity? Reduce bugs? Meet deadlines? Take my one day Better Firmware Faster seminar. You'll learn how to estimate a schedule accurately, thwart schedule-killing bugs, manage reuse, build predictable real-time code, better ways to deal with uniquely embedded problems like reentrancy, heaps, stacks and hardware drivers, and much, much more.

I'm presenting this:
- Austin, TX on April 18
- Baltimore, MD on April 20

Want to be your company's embedded guru? Join us! There's more info at http://www.ganssle.com/classes.htm, including cheap flights to these cities from around the USA.

If your outfit has a dozen or more engineers who can benefit from this training I can present the seminar on-site.

Thoughts on Consulting

A lot of developers work full or part-time as consultants, developing embedded projects for hire. Some augment their day-job salaries this way while others consult full time. It

can be an interesting way to delve into lots of different sorts of projects and learn many new things... or to lose a lot of money fast.

Steve Friedl's "So you want to be a consultant...? (Or: Why work 8 hours/day for someone else when you can work 16 hours/day for yourself?)" (available at http://www.unixwiz.net/techtips/be-consultant.html) is an excellent paper about many of the issues in the consulting racket. I especially admire his high moral stance in billing and admitting mistakes.

I've written quite a bit about the subject, and have recently put all my consulting thoughts together in a special report, which you can find at http://www.ganssle.com/articles/iconsul1.htm .

A Fiendishly Clever Circular Buffer

Phil Ouellette sent along a circular buffer routine that at first looked horribly wrong. It doesn't even look for the end of the buffer to wrap the pointers!

Then I looked deeper and saw the hidden cleverness. This routine is very efficient.

Phil tells me the code is derived from a Keil routine (http://www.keil.com/download/docs/intsio2.zip.asp), though his implementation is more compact and easier to read. Keil graciously put their version into the public domain... but did caution me to say that they do not provide technical support for the code!

So the code that follows is my slight changes to Phil's code which is based on Keil's. Any errors are mine alone. But check it out:

```
/*********************************************/
// Fifo.h
// Public header file containing Definitions
// and function prototypes.

#ifndef _FIFO_H_
#define _FIFO_H_

// FIFO_SIZE Must be a power of 2 (2, 4, 8, 16, 32, 64, or 128).
#define FIFO_SIZE   32

// FifoPut Return values.
#define FIFO_ADD_OK 0  // Indicates that FifoPut Succeeded
#define FIFO_FULL   1  // Indicates that FifoPut Failed
                       //  (no room for more entries).
```

```
// FifoGet Return Value.
#define FIFO_EMPTY  -1 // Indicates that FifoGet Failed
                       //  (Fifo is empty).

// Function Prototypes
void FifoInit(void);
bit FifoPut(unsigned char);
int FifoGet(void);

#endif

/***********************************************/
// Fifo.c
// Source level module

#include "FIFO.H"

// Macro to calculate the current number of entries in
// Fifo, saving the overhead of function call.
#define FifoLength   (FifoHead - FifoTail)

// Private, Module Level Variables
unsigned char FifoBuffer[FIFO_SIZE];
unsigned char FifoHead = 0;
unsigned char FifoTail = 0;

// Function to flush the Fifo
void FifoInit(void)
{
   FifoHead = 0;
   FifoTail = 0;
}

// Function to add an entry to Fifo
// If Fifo if full then function returns FIFO_FULL
// else function adds entry to FifoBuffer, increments
// FifoHead and returns FIFO_ADD_OK.
int FifoPut(unsigned char Entry)
{
  if (FifoLength >= FIFO_SIZE)
  {
     return FIFO_FULL;
  }
   FifoBuffer[FifoHead++ & (FIFO_SIZE - 1)] = Entry;
  return FIFO_ADD_OK;
}

// Function to get an entry from Fifo
// If Fifo is empty then function returns FIFO_EMPTY
// else function returns oldest entry in FifoBuffer
// and increments FifoTail.
int FifoGet(void)
{
```

```
  if (FifoLength == 0)
  {
     return FIFO_EMPTY;
  }
  return (FifoBuffer[FifoTail++ & (FIFO_SIZE - 1)]);
}
```

The nice thing about this implementation is that you never have to concern yourself with checking for the indexes wrapping or pointing past the end of the FIFO. The only limitations are that your FIFO buffer size is limited to a max of 128 entries and must be a power of 2. You can use larger FIFO buffers if you change the FifoHead and FifoTail to larger variable types (with a corresponding code space and run time impact).

The code uses an AND to wrap the pointers around. The modulo operator might be faster on some processors. `FifoTail++ & (FIFO_SIZE-1)` is, in this case, equivalent to `FifoTail++ % FIFO_SIZE`.

Fear of Editing

I was asked to look at a troubled project recently. It wasn't particularly big, but the developers were absolutely terrified of changing the code. Each change seemed to break something else. They wrote extensive wrappers rather than digging deep into a function to clean it up.

In most systems a little bit of the code causes most of the problems. We've all had that nasty bit of code that breaks every time someone changes a lousy comment. When you're afraid to change something, when only Joe is allowed to work on a particular snippet of software, we know for sure the code is BAD. We try to beat the beast into submission but it's a never-tamed hydra. One change leads to another, and another, ad infinitum.

5% of the functions consume 80% of debugging time. I've observed that most projects wallow in the debug cycle, which often accounts for half of the entire schedule. Clearly, if we can do something about those few functions that represent most of our troubles, the project will get out the door that much sooner.

Barry Boehm observed that these few functions that create so much trouble cost four times as much as any other function. That suggests it's much cheaper to toss the junk and recode than to reactively remove the never-ending stream of bugs. Perhaps we really blew it when first writing the code, but if we can identify these crummy routines, toss them out, and start over, we'll save big bucks.

Many agile methods practitioners believe in aggressive refactoring. If the code can be improved it *must* be improved. That's a bit extreme for me. We are, after all, required to ship something as soon as possible. But wise developers will be alert for code that

scares people. That's the stuff that needs rewriting. As Boehm showed, you'll save money in the long run.

Jobs!

Let me know if you're hiring firmware or embedded designers. I'll continue to run notices for embedded developers as long as the job situation stays in the dumper. No recruiters please.

Z-world is looking for embedded hardware/software engineers at their Ann Arbor, Michigan, facility. And they need both hardware/software engineers and embedded firmware engineers in Davis, CA. See http://zworld.com/company/careers/listings/engineering.shtml for more information.

Vadatech needs several people in their Henderson, Nevada site. Senior Kernel Software Engineers are needed to port and enhance a Linux kernel for use with various embedded processors along with their supporting hardware. Basic driver modules will be designed, documented, implemented, integrated and tested for stability, reliability and performance. Much of the work will be low level hardware access code including memory (ccNUMA) and interrupt handlers. A degree, plus strong knowledge of the Linux kernel are required, as well as a working knowledge of chipset and CPU internals, CPU schedulers, ccNUMA optimizations, MMU, SMP locking, interrupt routing, etc.

They also are long for Senior Driver Software Engineers to design, document, implement, integrate and test various driver components in a variety of OS environments. Participate in the creation and review of architecture, design and the code review of peer software and hardware components. Perform specification compliance testing in an embedded environment. Evaluate system performance on a variety of hardware platforms and under various loads. A degree, with expert knowledge of the Linux kernel, NETDEV and kernel module driver architecture, TCP/IP networking and the PCI bus interface are required.
See http://www.vadatech.com/career.html for info on both jobs.

Boulder, Colorado-based Spectralink is seeking multiple Embedded C Programmers. You will work with 802.11, Ethernet, TCP/IP and numerous standard and proprietary VoIP protocols, with exposure to Wireless LAN's and PBX Integration. A solid background in operating system development including Linux kernel or DSP in a C and Assembly real-time and embedded environment required. You will translate operating system requirements into design specifications on new and existing product lines. On the job experience with logic analyzers and oscilloscopes is highly desired. A thorough understanding of hardware architecture is required and an ability to interpret schematics of digital hardware. BSCS or equivalent degree. Knowledge of wireless applications, networking protocols, signal processing, microprocessor architectures, and PBX

technology or voice applications preferred but will not be given preference in lieu of basic real-time fundamentals.
See http://spectralink.com/careers/index.html .

YORK International Corporation has an immediate opening for a Technical Manager, Software Development. Follow the link to http://www.york.com/employment/postings . This position is located in York, Pennsylvania.

Panasonic Automotive Systems Company of America in Peachtree City, Georgia are looking for Software Engineers to do design work for the multimedia automotive industry.

Responsibilities include: Analyze requirements, design, code in C, C++, integrate, release and test software products for embedded processors (V850 etc.,) Products include radios, DVD players, amplifiers, etc for the automotive OEM market.

A degree plus 5-8 years real-time embedded software design in C is required. Desired, but not required, experience includes work in an automotive environment, experience with multimedia products, vehicle networks, and software architecture design. Relocation provided. Send a resume to: irvins@us.panasonic.com or jobs@panasonic-mcusa.com

Joke for the Week

In honor of Valentine's Day, Scott Fahringer sent the following:

How do I love thee,
More ways than I can calculate.

When I first gazed upon you from across the room,
You caused my registers to overflow.

But now when I look into your eyes,
my routines become defect-free,
and my exception handler never activates.

This evening with you is a CMM level 5 experience,
and I desire to hold you in my peripherals.

My program execution for you will continue forever.

The ring you wear is on my symbol table,
for all external references are resolved and
fade away when I'm linked with you.

I look forward to making an enhanced
category 5 connection with you tonight,
for we truly complement each other.

Yes, when we Google tonight - Yahoo!!

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.