SELECTING AN EMBEDDED RTOS

Prepared by:
eg3.com
Jason McDonald, Senior Editor

eg3.com
tel : 510.713.2150
email : info@eg3.com
web : http//www.eg3.com

Excerpted from "Insiders' Guide: Selecting an Embedded RTOS," at http://www.rtos-report.org, with permission from eg3.com.

MICRO DIGITAL INC.: RTOS LICENSING MODELS

9 November 2007: RTOS Licensing Models

INTERVIEWEE. RALPH MOORE
PRESIDENT
TEL. 800-366-2491, 714-437-7333
EMAIL. ralphm@smxrtos.com
COMPANY. MICRO DIGITAL INC.
WEB. http://www.smxrtos.com/

Q. **First of all, tell us a little bit about yourself and your responsibilities at Micro Digital.**

A. A partner and I founded Micro Digital in 1975 as a consulting company. Our objective was to design microprocessor based systems for customers. Our first project used the 8080 in a gas pipeline metering system. In 1988, I decided to go into the RTOS business. *smx* was released and first sold in 1989. Today, I am the President and chief salesman of the company and my son, David, is Director of Development. Between us, we handle marketing.

Q. **What are Micro Digital's offerings in the RTOS and embedded software space? Please give us a very brief run-down of the products.**

A. We offer the *SMX RTOS* consisting of the smx hard real time multitasking kernel, TCP/IP stack and networking protocols, FAT file system and flash file system, USB host, device, OTG stacks with class and function drivers, and other products. We focus on *ARM*, *ColdFire*, and *x86* processors and we offer out-of-the box solutions that include BSP code and drivers.

Q. **We touched base briefly before the interview, and you indicated that you wanted to explore "licensing" issues in terms of the RTOS choice. For those who don't know, can you explain briefly the basic models for licensing an RTOS?**

A. The traditional model is the royalty model where there is a moderate up-front charge per development seat, then a royalty for every unit shipped. Microsoft, Wind River, QNX, and others still use this model. The dominant model, today, for embedded software is the no-royalty, one-product license. For this, there is a relatively large up-front charge for one developed product and no royalty per unit shipped. My feeling is that the royalty model makes more sense. However, by the time that I and contemporary RTOS vendors (Accelerated Technology, A.T. Barret, U.S. Software, and others) entered the market we needed a way to beat entrenched competition, so we independently decided to follow the lead of Kadak and we each offered no-royalty pricing with full source code.

Q. **What is your opinion of the "royalty free" movement? Most - if not almost all - RTOSes these days are "royalty free." What do you think of that?**

A. I think the royalty model makes more sense because what the customer pays is proportionate to the value of the software to him – i.e. the more he sells of his product, the more he pays. However, most customers do not want to keep track of shipments and royalties. Hence, even among royalty companies, royalty buyouts are common. With the royalty-free method, we obviously cannot sell one license to a customer to use forever. We would soon be out of business. Hence, our competitors and we limit the license to one developed product.

Defining one product is difficult. If there are two versions, one with an industrial hardened case and the other with a normal case, and everything else is identical, are there two products or one? If some versions have more inputs, but everything else is identical, is there just one product? I think you can see that defining what is one product leads to a lot of negotiating. I had one prospect argue with me that their refrigeration and heating units were all one product because they all moved heat from one place to another! What do you do if there is single "platform" used in diverse products such as parking systems and data acquisition systems? Many large companies like to do this to keep costs down. What do you do if your products are part of a technology used in diverse moderate-cost products of which only 10 or 20 are made per year?

Q. **Micro Digital is a representative RTOS vendor. Can you explain for us your own licensing and purchase models?**

A. We offer the one product royalty-free license as well as other licenses. About 2 years ago, I decided to introduce a 50,000-unit production limit on our least expensive one product license. Most true embedded systems have lifetime productions of less than 50,000 units, so this is no problem for them. Products with greater than 50,000 lifetime productions tend to be consumer or quasi-consumer products and I think they should pay more. Most people agree with this. There is a scale of percentage increases for higher limits, up to 50% increase for no limit. Some people have not liked this plan, but generally, it has been accepted well. I find that it allows me to be more liberal on the one-product definition. Many of our competitors deal with this problem by sharpening the one product definition when they find out that large production quantities are involved. Needless to say, they don't make many friends this way.

When one product does not fit, then customers often want to negotiate license buyouts so they can develop many products with one license. I would prefer not to do this, because it is very hard to have a stable cash flow if you must start every sale from scratch. However, the customer is always right. Hence, we offer multi-product licenses. However, as with the single product license we cannot sell one license forever after to each customer or we will soon be out of business. Hence, we and other RTOS vendors limit multi-product licenses to one processor. However, this too can be tricky to define. Is it the same processor if it just has more memory? What if it has an additional peripheral? One processor is not good enough for many customers – they want to use an *ARM7* for low-end products and an *ARM9* for high-end products.

We have addressed this range of requirements with a range of licenses: Product Line (2:1), Product Family (3:1), and site license (4:1). Each, in succession, is more liberal. For example, the product line is limited to one processor; the product family allows two processors; and the site license allows 3 processors. This is a new licensing plan and it is too early to tell how well it will work. I like it because it provides a smooth range of prices to customers and precise definitions of each license.

Q. **Microsoft argues that royalty free is actually not beneficial to developers as it puts 100% of the risk and cost on the early design phase. Microsoft argues instead that "royalties" (as it charges for XPe and CE) create a "shared success" model in which the RTOS vendor is motivated to ensure that the developer actually ships real products quickly. What is your response to that?**

A. The main statement is correct. In general, 1/3 of all projects never make it to market. Another 1/3 are marginal successes due to being late to market, too expensive, not meeting the market's requirements, etc. Hence, only 1/3 of the projects pay the bills for royalty companies. Thus, the RTOS vendor who charges royalties is definitely sharing risk with the customer and is motivated to help that customer succeed. Many times, small companies, startups, and others have asked me if we will offer them a royalty. I tell them it is possible, but if they later want to do a buyout, it will cost 3X our list prices. They don't like this, and so far I have not had any takers.

There is a dark side to royalties, also. I think the royalty RTOS companies tend to provide better support to the well-funded, well-staffed projects that look like big winners. It is sort of like betting on a horse race. The small guys and probable losers get neglected. One reason for this is that there often is not enough money in the up-front payment to provide much support. It is ironic that you cite Microsoft, because Microsoft does not provide good RTOS support – it charges $200-300 per "incident" and it relies on newsgroups to provide most support – no better than Open Source.

The other side of the coin – to say that the no-royalty companies provide poor support is false. Micro Digital provides very good support and most of our no-royalty competitors provide at least adequate support. In general, the no-royalty RTOS vendors are small and the programmers who wrote the code provide support. Obviously, this is better than providing support with go-betweens. We are motivated to provide good support because happy customers come back for more products.

Q. ***Linux*, in contrast to Microsoft, is both "open source" / GPL licensed and "royalty free." What is your take on the "open source" license such as the GPL?**

A. I think the whole Open Source movement is based upon false premises and is harmful. In as much as *smx* was focused on the x86, when *Linux* appeared, it probably caused us more harm than it did our competitors. In general, I think it has suppressed growth of RTOS companies and of innovation in the RTOS market. I have seen studies that show that the using *Linux* on large projects actually costs much more than using the most expensive commercial RTOSs. Yet, people persist in thinking that it is free.

I am starting to see a little bit of sense returning to the market. Some people I talk to have tried Open Source, found it wanting, and have returned to the commercial fold. However, may others sill consider it to be the best solution. Many engineers seem to think that they work for free and the man-months necessary to learn *Linux* and provide their own support matter little. If these people were paying their own salaries, what decision would they make? I think many of the engineers who choose Open Source have the same motivation as the ones who chose to design in-house RTOSs, prior to *Linux*. Designing an in-house RTOS is no longer politically correct, but picking Open Source is.

Q. **QNX recently introduced a dual-source license with non-commercial uses having full access to source and uninhibited use of the RTOS. Commercial users must license. What is your opinion of this "hybrid" open source sort of license?**

A. This "hybrid" is not a hybrid at all – it is practiced by most Open Source projects and to me is another hypocritical action in the Open Source community. Code is either free or it is not free. How can Open Source project leaders use the free work of community members then turn around and sell it for a profit? This is the moral high ground? Of course, QNX is a commercial company and they have a different motivation to introduce this licensing. Many other commercial companies have done the same thing. I think QNX's objective is to get their RTOS in greater use and hopefully that will lead to more sales. I do not know whether this is working for them. I tend to find that the people who want free lunches will never pay for them, so I have so far avoided offering *smx* this way.

Q. **Have you considered much of the legal issues in terms of licenses? We hear, for example, that many are worried that Linux contains commercial products illegally introduced into it, and/or that Linux might force software code to be "open sourced" itself. What is your take on the legalisms around RTOS licensing?**

A. This is the reason that many attorneys and corporate leads consider GPL to be pollution. Many big companies practically insist that we swear on a stack of Bibles that there is no GPL pollution in our code – they do not want to end up in messy lawsuits. I understand that the *Linux* Community has allowed commercial code to be linked to the *Linux* kernel,

without requiring it to be Open Source, but the more radical members want to end this practice as of January 2008. Our position has been to steer clear of *Linux*. I refuse to allow any of our code to be linked with the *Linux* kernel.

Q. **Many vendors offer a "per seat" license, a "per project" license, and even a "per processor" license. Do you think any of these are especially advantageous? What factors would you recommend a developer pay attention to in terms of potential license "gotchas?"**

A. Some of the compiler companies have introduced per seat RTOS licenses, probably because this is how they sell their other software. Customers are allowed to do any number of projects without paying more. I think these vendors will find that this is a poor business plan – there is not enough money to fund supporting new processors, to provide good support, and to keep an RTOS team together. After all, those of us who have been in the RTOS business much longer are not getting fabulously wealthy. There are good reasons why we charge as we do. I expect to see these plans fade away, in time. I have not studied any RTOS per seat licenses, but I would think that the purchaser should expect to write his own BSP code and drivers and should not expect much support. Also, as with chip vendor software, the code may not be well written or well debugged. Probably the programmers who wrote it have already been moved off to other projects – that is the problem with something that is not financially viable.

The per project and per processor licenses are akin to what we offer.

Q. **Thank you for this interview.**