



*Embedded software in medical devices will be required to meet new guidelines, not only for the software itself, but also for the process and methods used to develop it*

# Operating under more pressure

The challenges involved in developing high integrity applications for the new generation of powerful, low cost microcontrollers. By **David Brook**.

For many years, the embedded microcontroller industry was neatly segmented into 8, 16 and 32bit architectures. The cost of the mcus, as well as the associated software and tools, increased in line with the complexity of the processor.

The landscape has changed dramatically in recent years and engineers now have access to powerful 32bit microcontrollers at prices comparable to older 8 and 16 bit devices. But, typically, the budget available to buy software and tools has not grown. This means that embedded engineers and suppliers alike have to deal with a world increasingly dominated by software, where factors such as cost, regulation, safety and power management are converging to create a much more difficult engineering challenge.

## Challenges

Some of the implications are obvious. Creating a next generation product using a processor like a Cortex-M3 with 256kbyte of flash, when previous generations used a 16bit processor and an in house scheduler, requires a new approach to deal with the increased volume and complexity. Using an off the shelf real time operating system and peripheral drivers can provide a cost effective way of reducing risk and shortening time to market.

The growth and importance of software content has presented other challenges, which are now being addressed by regulatory authorities. For example, the US Food and Drug Administration (FDA) has recently stipulated that embedded software in medical devices will be required to meet new guidelines, not only for

the software itself, but also for the process and methods used to develop it. This clearly puts embedded software development at the heart of cost, legal, and safety issues. Importantly, the software also represents the perceived value of products and this presents a tremendous opportunity for those companies who get the approach to software right.

## Changing the software business model

Interestingly, the role of software vendors is now also being redefined by industry momentum. The falling cost of 32bit microcontrollers also means the high prices traditionally associated with 32bit tools are no longer viable and many companies are filling the void with low cost or free tools. Indeed, according to a recent major industry survey, the most popular operating

system in use by professionals is FreeRTOS. Clearly, the business models used by software and tool suppliers are being modified and there is now a need for new business models, which include free software and tools to be part of the same development ecosystem as high integrity software.

**Considerations for choosing software**

For engineering managers, the challenges at the software level seem, at first, to be reasonably straightforward. If the software has reached a size and complexity which justifies the use of an operating system, network stacks and drivers, is it simply a case of defining the technical criteria and calling the suppliers to find the one that is right for you? If only it were that simple.

• *Lifetime costs*

Since cost is a major factor, the price of tools and middleware can often be high. Many software companies offer a low price of entry – knowing that when you are committed to their platform, they will have the opportunity to earn more revenue. It is critically important to understand the long term cost implications, not only for your current project, but also for the foreseeable future. It would be very good advice to ask suppliers for quotations for more complex projects, bigger teams, higher production volumes and additional software modules. It can often be surprising how quickly these costs can rise. Remember that switching to another tool or software vendor can be expensive, risky and technically difficult.

• *Meeting certification and safety needs*

Certification is an area which is often not properly considered in context. Regulation and



**BROOK: “FACTORS SUCH AS COST, REGULATION, SAFETY AND POWER MANAGEMENT ARE CONVERGING TO CREATE A MUCH MORE DIFFICULT ENGINEERING CHALLENGE.”**

certification requirements relating to embedded software are gradually being extended to more markets and products. The challenges of certification have to be examined closely, since a number of software and chip companies have released precertified versions of their products.

There seems to be a trend emerging for companies to ‘tick boxes’. Safety certification? Box ticked! The problem which engineers have to face is that, despite the growing availability of precertified hardware and software, it is only the final product or application which is truly certifiable. Using pre-certified components can

be an excellent way to reduce risk, and save time and money, but only when it is clear who is really responsible for safety. Regardless of the certification status, the developer still assumes full responsibility unless the supplier has the capability and competence to support their software through the development lifecycle.

• *Long term support*

This has fairly wide ranging implications. For example, how will a supplier respond if an assessor challenges an aspect of the certification claim being made for the third party software? This is not a trivial question. If the supplier has externally contracted the certification process and does not have an in house mature, process oriented culture, then the potential risk to the project could be high.

Supporting complex projects which require certification should be undertaken with a full assessment of the capabilities of all participants, including the suppliers of third party software. The FDA, for example, is proposing to require that code coverage analysis is provided for all (or at least for a high percentage) of the software contained in the application. Engineers should ask their suppliers to share in providing a high degree of validation and support for their products and not just a certificate and a safety manual. A number of companies are emerging who clearly take this approach and have that capability. SafeRTOS from WITTENSTEIN, for example, comes with a fully documented lifecycle and a support contract which extends not just to the source code, but also to the certification evidence as well.

The influence of software, regulation and low cost, powerful processor architectures will continue to grow in the foreseeable future. Engineers will increasingly have to build relationships with suppliers and ensure they find partners who understand their needs. The technical performance of software is important but, on powerful microcontrollers with more cpu performance and memory, the softer issues such as support capability, domain knowledge and long term costs of ownership, become just as important as the technical features of an RTOS, compiler or network stack.

**Author profile:**

David Brook is sales and marketing director with WITTENSTEIN High Integrity Systems.

