

==White Paper



Reduce wireless embedded software development time – and get your products to the market faster.

Author: Andrew Yen
For: Clarinox Technologies Pty Ltd
8 Bayside Crescent
Hampton, Victoria 3188
Australia

V2 Update Oct 2008

Copyright © Clarinox Technologies Pty Ltd, 2008
www.clarinox.com



Abstract:

With the development of wireless electronic devices, such as Bluetooth, WiFi, and GPRS and so on, it is important to bring new, developing concepts to the market quickly. By bringing products to the market faster you can increase their market share and profitability, and enable more profitable products to be developed in a given amount of time.

The development of embedded software is often a critical activity in wireless product development. Any opportunity to save time in embedded software development reduces the overall time to market.

An innovative company has developed an application development framework with a library of modules to assist in developing high-function embedded software for wireless devices. This library of modules allows programmers to write and simulate a product's embedded software in a conventional PC environment. This reduces learning time and provides the programmer with the expertise contained within the modules (IP).

This framework is portable to a range of operating environments - processors, operating systems and protocols. It contains expertise of pre-programmed blocks and so embedded software can be developed distinctly faster by less experienced programmers. There have been savings of up to 85% in programming effort reported.

The innovative technology reduces time and gives companies the competitive edge they need when entering new technologies onto a rapidly growing market.

Developing wireless products quickly:

Today's market is faster and more competitive than yesterday's. This is never truer than in the market for small electronic wireless devices. To market wireless electronic devices, the program manager is challenged to develop products with more features, in new and changing operating environments, and in less time. To be competitive, the program manager may be required to do all this with fewer and less experienced resources.

Fast commercialisation leads to market leadership:

First mover and fast follower strategies are utilised by market leaders to protect their leadership position. According to Ries and Trout's "law of leadership", it is better to be first than it is to be better⁵.

If companies are able to get their products from concept to market quicker they increase their domination of their segment and improve their industry leadership position.

Companies that are first to market their products then command higher prices and enjoy a greater market share³. This is especially true in rapidly changing markets, such as the wireless devices market. In this market, a company which has the ability to commercialise ideas quickly and effectively will be able to earn profits earlier, therefore making larger profits and obtain a competitive advantage⁴

The rapid development of a demonstration wireless system provided our business with new ways of looking at things and new opportunities for growth.

On the other hand, unfortunately, many opportunities can be lost or are not fully realised when products are introduced late into a market. In some cases the consequences of coming in late can be the loss of a major contract. Late product introduction gives market share to competitors and reduces the product's market lifetime.

In the laser printer market it has been estimated that introducing a product six months later can reduce the product's cumulative profits by one third⁶.



The challenges of fast commercialisation:

Product development has a number of inherent risks and is time consuming. There are many hurdles to overcome and unexpected delays can be numerous. The '*critical path*' includes many activities that are core to the project – such as developing embedded software, learning new technologies, testing and debugging.

In the wireless devices market, one area of product development often finds itself on the '*critical path*' – the development of embedded software.

The embedded software developed for wireless devices provides the application functionality, the middleware and the device drivers. The embedded software often runs on top of a 'real time operating system' (RTOS).

Embedded software for wireless devices is often a new technology for a company's engineers. The time it takes to learn and understand the protocol and then to develop a "Bluetooth stack", for instance, can eat into valuable project time. These challenges often put development of embedded software on the *critical path*. Each day that can be saved on this activity can bring the product to market a day earlier.

So how can a company reduce the time it takes to develop embedded software without sacrificing product features and specifications? How can a firm reduce its learning curve?

The solution to saving embedded software development time:

The answer lies in using the accumulated experience and knowledge of experts in the field. Product development systems that effectively capture and reuse best practice knowledge produce superior products faster². Buying the expertise and experience to develop product embedded software is a safe and highly productive option.

To access this experience a company can hire the experts themselves; however this comes with the associated resource hiring and management burden. Alternatively the company can access expertise contained in pre-written embedded software modules written by wireless embedded software specialists.

Years of accumulated wireless embedded software experience has been coded into a framework of pre-written embedded software modules. These modules can be selected, integrated and customised to create unique embedded software that offers high functionality. This modular programming approach is not limited by operating environments (processors, operating systems, drivers, protocols



etc.), as a wide range of target platforms are catered for. This becomes important as wireless products increasingly require new, high performance processors and other hardware components and RTOS to gain the competitive edge in the market.

This approach saves learning and development time because a high level of testing and expertise is already contained in the modules.

The principle of “not re-inventing the wheel” has been widely promoted in product development circles for its obvious time and cost benefits². This “object oriented” embedded framework based development approach allows the principle to be extended into the embedded software development field.



ClarinoxSoftFrame:

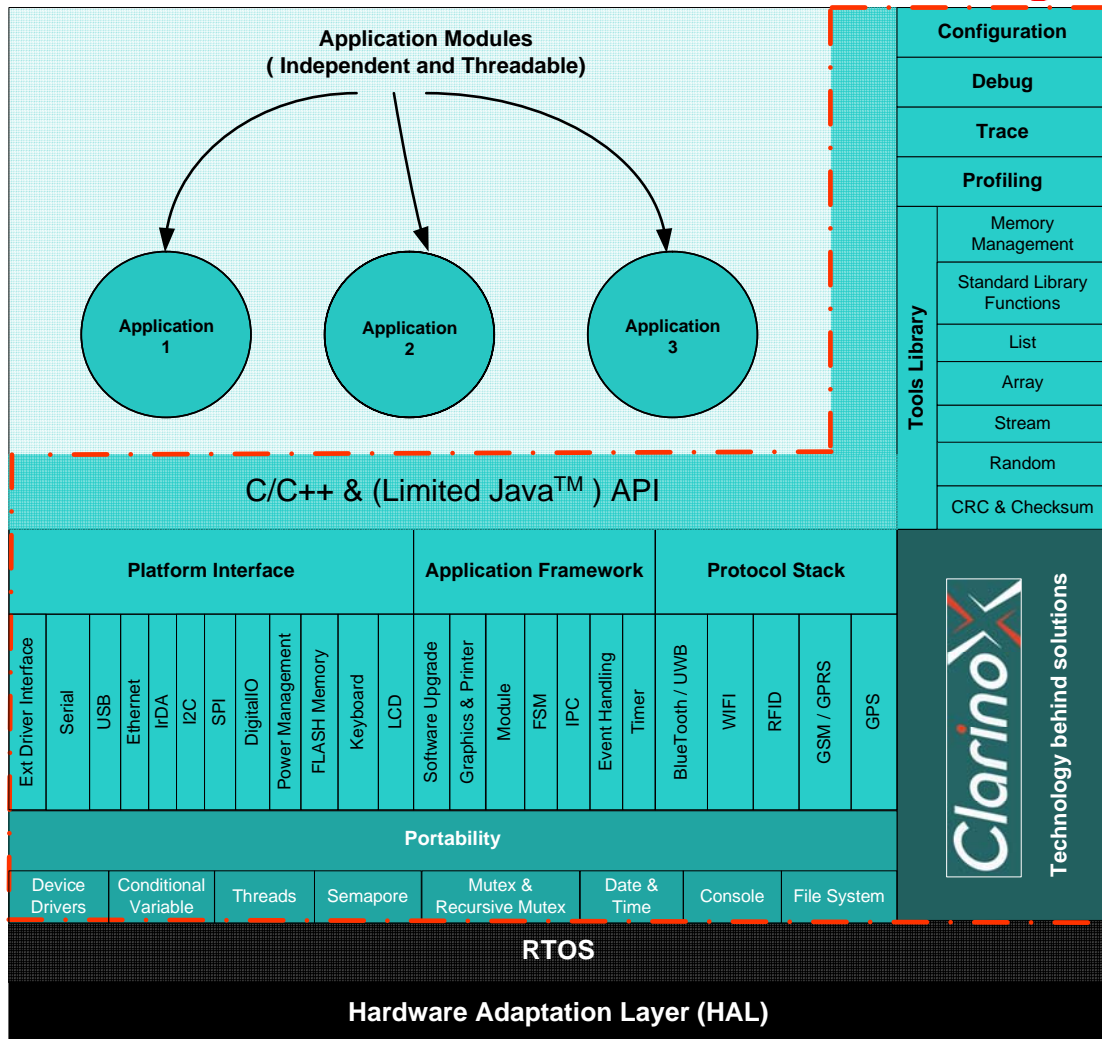
ClarinoxSoftFrame is an application building framework. It is an object oriented package that greatly assists the programmer to develop and integrate operating system, middleware and protocol stacks.

The conventional approach to wireless product software development is to select an operating system, select off-the-shelf protocol stack, select off-the-shelf middleware and then develop device drivers, BSP, and write an application in house that integrates the components together.

The conventional approach requires a good knowledge of the operating system, the device drivers and the protocol stacks. Debugging is time consuming and is performed at a low level. The development of middleware for protocols such as Bluetooth is often considered so complex as to be beyond many smaller firms⁴. This is because an understanding of many layers is required: radio, baseband, stack, profile, operating system driver and application. The interface and interaction between the layers is complex and creates many opportunities for bugs and potential crashes¹.

A typical development project for a common wireless communication product takes around 7 months using [the traditional] approach. Up to 6 programmers may be involved in this typical project at any one time.
--

The ClarinoxSoftFrame provides a range of pre-written software modules that allows the programmer to use any of a number of operating systems and processors. Platform interfaces, protocol stacks and other code modules are picked from a code library to create the base to develop the product application. The programmer works in an object oriented C++ environment.



The programmer is not required to have a detailed knowledge of the RTOS or the wireless protocols.

Because the modules have been developed as a suite their compatibility is assured. Debugging of low level code is completely eliminated. The product can even be prototyped and simulated with ClarinoxSoftFrame on a conventional PC then moved to the target platform with 100% compatibility.

Using ClarinoxSoftFrame a single programmer can reduce their learning requirements and quickly start developing the application. Typically a project that would have taken 7 months for a small team would now take a single programmer using ClarinoxSoftFrame only 2 months. Total effort savings of 85% have been reported.

Once the embedded software/application has been developed the ClarinoxSoftFrame provides further efficiency gains. Changes can easily be made to interfaces, target platform (processor and/or RTOS) or protocols by



adding or switching ClarinoxSoftFrame pre-built modules. In this way the product can be updated quickly to respond to market demands and opportunities.

A product developer using ClarinoxSoftFrame decided mid-project that a higher specification processor would be required in their product to meet emerging market needs. [Use of] ClarinoxSoftFrame meant that the software development completed so far could easily be ported to the new target platform without re-writing and with virtually no software development project delay.



How ClarinoXSoftFrame reduces embedded software development time:

ClarinoXSoftFrame is the result of many years of research and testing by experts in the wireless embedded software field. This experience is encapsulated in the pre written software modules that assist in developing and integrating applications, middleware and drivers.

SoftFrame reduces embedded software development time in three ways:

1. **Reduced learning required.** ClarinoXSoftFrame substantially reduces an embedded programmer's learning requirements. High end processors, RTOS and wireless technologies are all complex and require specialist knowledge, – ClarinoXSoftFrame reduces learning requirements by providing a PC based C++ object oriented environment that allows the programmer to use tried and tested modules to develop and test embedded software.

ClarinoXSoftFrame allowed us to rapidly develop a product even though there is not a lot of wireless expertise in-house

2. **Reduced testing effort required.** The pre-testing of ClarinoXSoftFrame modules ensures that they work with each other; in a range of operating system and hardware environments and that the programmer knows what performance to expect. This reduces and simplifies the overall testing effort. ClarinoXSoftFrame also provides a number of development tools to assist with trace and debugging.

We needed a FAST track approach to get a system running and get a product out to market as soon as possible. ClarinoX's off the shelf blocks (Bluetooth, TCP/IP) and development environment reduced our overall development effort.

3. **Embedded software can be adapted and reused on new products.** Companies wish to maximise the value they get for their investments. This applies to embedded software as well. Reusing the pre-written software modules leverages the investment and it enables the project team to deliver solutions faster. With ClarinoXSoftFrame, the time and effort put into developing embedded software for one product can be leveraged for the next. The portability of ClarinoXSoftFrame means that the next development project doesn't need to have the same hardware or operating system!



Conclusion:

The ClarinoxSoftFrame suite of embedded software modules contains expertise (IP) that reduces learning requirements and speeds development time for wireless embedded software. Programmers with relatively little experience can produce high functioning embedded software in much less time than in house development from scratch. A wide range of processors, hardware, RTOS, protocols and drivers are catered for.

The use of ClarinoxSoftFrame provides opportunities to bring innovative wireless electronic products from concept to market faster. In this fast moving market it is imperative that ideas are commercialised quickly and efficiently to maximise profits and market share.

References:

1. Cravotta, N, "Bluetooth Interoperability: It's all in the details", EDN, May 1 2003, pp 55 – 63
2. Dani S, Harding JA, Case K, Young RIM, Cochrane S, Gao J, Baxter D, "A methodology for best practice knowledge management", Proc IMechE Vol. 220 Part B: J. Engineering Manufacture, 2006, pp 1717 - 1728
3. Nevens, M T, Summe, G L and Uttal B, "Commercialising technology: What the best companies do", The McKinsey Quarterly, number 4, 1990, pp 3 – 22
4. Rice, J, "Industrial self-organisation in early technology emergence: evidence from Bluetooth application development", International Journal of Technology Management and Sustainable Development, Volume 4, Number 2, 2005
5. Ries, Al and Trout, Jack, "The 22 Immutable Laws of Marketing", (1993) Harper Business
6. Reinertsen, Don G, "Whodunnit? The search for New Product Killers", Electronic Business, July 1983, pp 62-66



For further information:

Contact Clarinox Technologies
8 Bayside Crescent, Hampton, Victoria 3188
PO Box 222, Sandringham, Victoria 3191
Phone +61 3 9502 0728
Fax +61 3 9502 0729
enquiries@clarinox.com