

Virtual round table on **software for IoT embedded system design**

*In today's design environment, the software content of projects typically outweighs the hardware aspect – often by a factor of several times. In the embedded system design space, the relationship between “design content” and the often-quoted “team size” is significant. Especially in smaller enterprises, different roles are not necessarily taken by different engineers. Flexibility and adaptability are essential qualities for the design team, just as they are powerful attributes of the support on offer from the suppliers of the silicon on which the embedded designs will run. In Part II of its focus on embedded system design, **Cliff Ortmeier, Global Head of Technology Product Marketing & Solutions Development at electronic component & system distributor, Premier Farnell** talks us through the virtual round table **Farnell** hosted with **STMicroelectronics, Renesas, NXP, Silicon Labs & Texas Instruments** to review some of the software options available...*

Re-use is a key term in embedded system design. On the hardware side, the same circuit blocks will appear across a family of devices, adding or omitting a range of features, and may be adapted substantially, or slightly – or even not at all – when employed in the next generation of product. Re-use in the software dimension can range from the simple act of coding a function the same way as the time before – because it worked – through to inclusion of off-the-shelf operating systems, and adaptation of generic application-example packages.

We asked several makers of microcontrollers to pick out some of the key aspects about the support they offer in this respect, to ease the design-in process for their particular silicon devices – often at the device-driver and useful-function level, but also extending to the involvement of engineering teams at the supplier who will guide device choices and advise on software strategies.

At **STMicroelectronics**, Laurent Vera, EMEA Marketing Director, describes the functionality it offers in its STM32Cube software tool: “It consists of two main components:



STM32CubeMX and STM32Cube embedded software libraries.” The former bundles elements such as: configuration C code generation for pin multiplexing, clock tree, peripherals and middleware setup with graphical wizards; generation of IDE-ready projects for integrated

development environment toolchains; power consumption calculation for a user-defined application sequence; direct import of STM32Cube embedded software libraries from st.com; and integrated updating features.

The STM32Cube embedded software libraries include, Vera continues: “The HAL hardware abstraction layer, enabling portability between different STM32 devices via standardised API calls; the low-layer (LL) APIs, a lightweight, optimised, expert-oriented set of APIs designed for both performance and runtime efficiency; and a collection of middleware components, such as RTOS, USB library, file system, TCP/IP stack, touch sensing library or graphics library (depending on the MCU series).”

One semiconductor vendor that has invested heavily in an integrated, silicon-plus-software approach is **Renesas**, with its Synergy offering. It aims to enable design teams to begin software development at the API level, with capable lower-level code fully in place. “Software development,” says Renesas’ Andy Harding, Director, Broad-based Solutions, “is the raison d’être of the Synergy platform. We recognise that customers have more software engineers these days than hardware. And if we can get them programming at a higher level (API), not spending time on non-differentiating code (drivers, middleware, and so on), then they can save time, money and focus on differentiating their products in the market.” He adds, “if you employ people with [silicon IC design] expertise [together] with good software architects... there are some additional benefits of software being written by software



Cliff Ortmeier, Global Head of Technology Product Marketing & Solutions Development, Premier Farnell

Re-use is a key term in embedded system design: in the software dimension, it can range from coding a function the same way as before – because it worked – through to inclusion of off-the-shelf operating systems, and adaptation of generic application-example packages.



engineers in the device manufacturer... and that is that you can design software with silicon in mind, deriving much more performance."

At **NXP**, Geoff Lees, SVP & General Manager, Microcontrollers, comments on the depth of support NXP has assembled in its engineering teams: "We've always been at the higher performance end of spectrum... applications such as energy conversion, motor control and wireless connectivity. Our system engineering and applications engineering team... understand both our products and our customers' applications, so they can be a huge help to customers. They often test by building reference designs, demos or almost complete products, using good-to-go software, electronics and so on. Sometimes customers can use the systems we build, and we think of our systems team as the 'first customer'. In the last five years, we have focused a great deal on software. Today, it's not about providing broad enablement, such as Android, it's important to provide product application layer solutions... half of our engineering team are software engineers."

Lees also comments on the importance of driver software for peripheral functions. "You can have best hardware, but if software can't exploit it, the customer can't access it... Drivers allow peripherals to be dealt with by higher level system software. Our engineering teams

optimise drivers. Sometimes, we can get access to high performance features and bring them to open source software, creating 'rich drivers' for Android, Linux and more standardised RTOSs, such as FreeRTOS, ARM Embed OS and UCOS."

Silicon Labs has a particular focus on microcontrollers with integrated wireless connectivity, and makes available a suite of software that majors on that functionality. Tom Pannell, Silicon Labs' Senior Director of Marketing, outlines how the company offers, "... software-wireless stack – as well as hardware. In fact, we have gone from a chip-focused company to being vertically orientated, providing ICs, software and tools. This allows us to expose more value in the ICs. The value is truly found in the interconnection of software and hardware. When we develop products, our software teams collaborate with hardware, and feeds back to hardware engineers. So, software engineers, as well as hardware engineers, can influence product definition. Tools are also very important; the world isn't perfect. If a customer wants to deploy 200 mesh nodes, they need to have tools that show them how mesh is working/forming. They need tools that give them insight at packet level."

The skill set associated with transferability, and with re-using software across product ranges

and device families, is termed "nimbleness" by **Texas Instruments**. TI's Henry Wiechman, Embedded Processor Software Marketing & Strategy: "Support for [TI's ARM-core-based] SimpleLink platform is provided through a suite of tools and shared SDK [software development kit]. The SDK provides software components with code portability across the family. The ability to port from one member to another is important for nimbleness... Nimbleness is important, [though] interestingly, customers may not be aware they need it. They just know they need to develop product A. But, when they release product A, sometime down the road they see the market changing. If customers don't think about nimbleness, they are at a disadvantage when the market changes. Nimbleness allows them to quickly pivot; to grow and change with the market."

Wiechman's colleague, Adrian Fernandez, Microcontroller Development Experience Manager, nonetheless makes the point that commonly-used functions can, with advantage, migrate to hardware form, "[rebalancing] hardware and software; to have more hardware be available for particular tasks. We already have multi-core devices that use a combination of hardware accelerators," Fernandez concludes.

The field-engineering teams at major distributors [such as Farnell] can likewise be a valuable resource: not only can they – with due regard for confidentiality – bring to bear experience gained assisting a wide variety of projects, they are also familiar with where all of each maker's software resources are to be found.

Nimbleness – the ability to port from one embedded device to another – is important, though interestingly, customers may not be aware they need it.