

Renesas Synergy™ Software Package (SSP) Redefines Software Development for the IoT Market

By Integrating All Major Software Functions in a Commercial-grade Package, SSP Offers Embedded Developers Faster Time to Market

In today's fast moving IoT market, developers should not be responsible for integrating major components of their base software such as the RTOS, communications stacks, software frameworks, and device drivers. Developers should be using software components that are part of a larger, integrated MCU-based hardware/software package because doing so offers many advantages such as:

- Avoiding grappling with continuously changing roadmaps of the different software components in the base software platform
- Removing integration headaches that result from using components that were never designed to work together
- Allowing the team to focus on their key differentiators rather than common, low-level base code

From the perspective of the Renesas Synergy™ development team, one thing had become vividly clear -- accelerating time-to-market has become increasingly crucial to product success in the IoT market. In an environment where product lifecycles are measured in months instead of years, developers who are late to market risk losing market share.

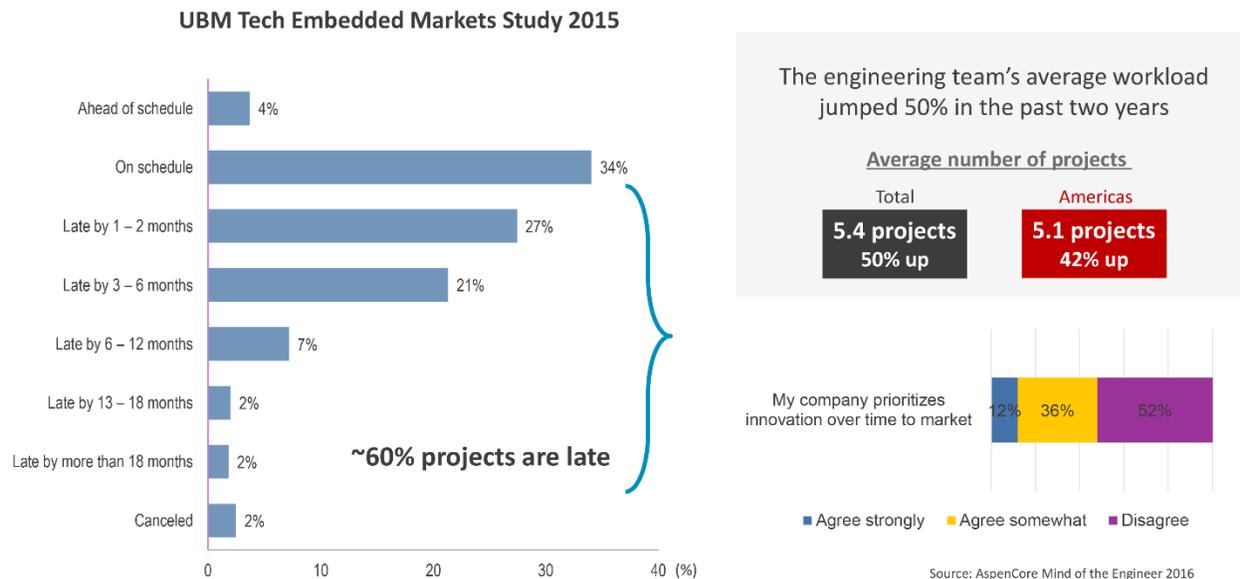


Figure 1: *The increasing complexity of software development in MCU projects is driving up development costs and extending development schedules.*

At the same time product development needs have changed, the days when each developer created his or her own solution by mixing and matching an MCU and software components are quickly fading. Today’s Internet-connected, MCU-based designs have become too complex.

The Synergy Platform design team looked at the traditional development cycle and asked why should developers devote so much of their development cycle to non-differentiated code – designing software drivers, middleware, integrating with the RTOS and connecting to the cloud? That effort doesn’t offer developers much in terms of differentiating features in the end-product. Instead, developers should be devoting more of their time to the truly innovative portion of their design - creating application code or adding new features to their product.

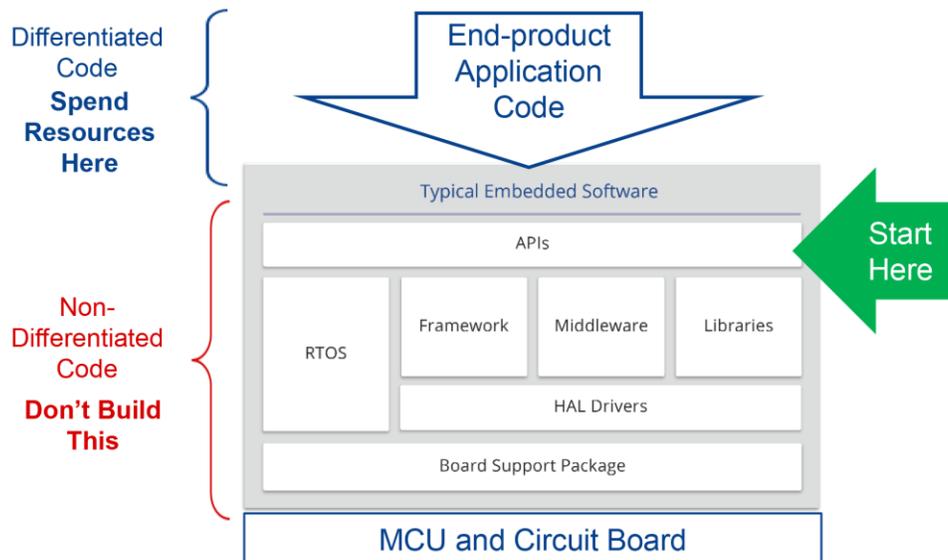


Figure 2: *Developers should focus on differentiated code in their end-product*

To minimize cost of ownership and allow developers to more quickly begin developing code for their particular application, Renesas took a radically new approach to product development. The Synergy Platform would treat hardware and software as a unified product. With this new platform, Renesas would assume responsibility for the development and integration of the non-differentiated components which are the building blocks of the platform software. That would allow developers using the platform to spend less time on those basic functions and more time on the implementation of differentiating software features in their product.

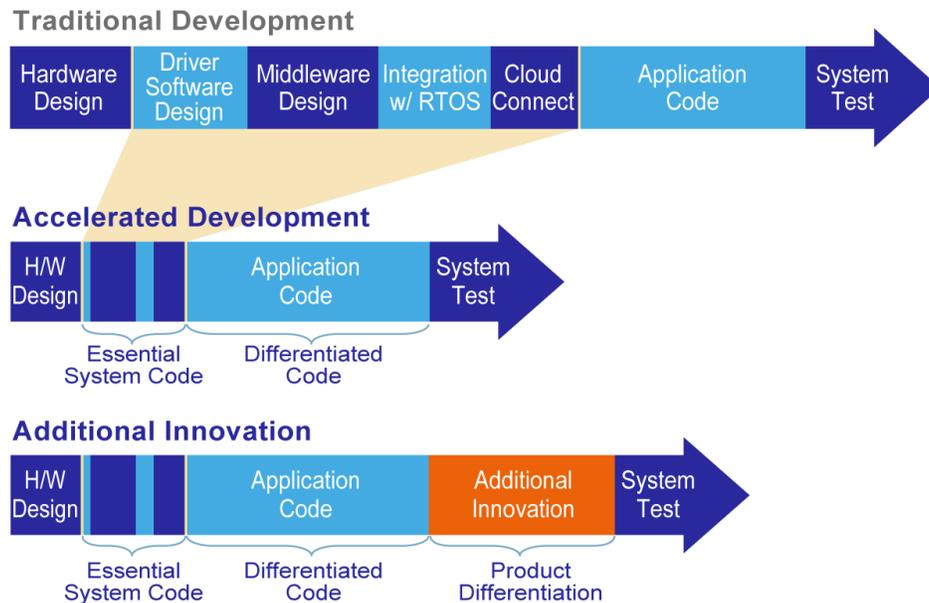


Figure 3: Developers can free themselves of spending time on low-level tasks, but instead focus on getting to market before their competitors do.

Key Functions

Next, the Synergy team asked what characteristics were essential to successful embedded IoT applications. If the Renesas SSP was going to serve as the optimal platform for IoT applications, what kind of challenges would it have to address?

Clearly any prospective IoT solution would have to offer a wide range of communications options. It would also have to offer security features to address the many threats to any networked product. For developers to confidently rely on this integrated platform, it would have to offer high quality, commercial-grade software. Moreover, that software would have to be tested and qualified on high quality hardware reference designs. Given the large number of complex software components in the platform and the massive quantity of the documentation associated with the entire package, developers would need a new, much more efficient method to quickly find the most relevant information when they needed it.

Security was a major challenge. Threats lie everywhere. How could the Synergy designers prohibit unauthorized code from being programmed and executed, protect the firmware and data on the MCU, identify the right source of communication and protect communication data from interception and tampering?

To accomplish this task, the Synergy team integrated an extensive portfolio of security features into the platform and took advantage of the significant processing capabilities of the Arm® Cortex®-M cores coupled with a hardware-based security accelerator to speed up those functions in silicon. Most of the Synergy MCUs feature accelerators for

symmetric cryptography, asymmetric cryptography, HASH, and asymmetric key generation. They also feature a true random number generator, key secure storage, limited JTAG access, and a unique ID assigned to each MCU. On the software side, designers implemented a TSL library, a cryptographic library and secure vault.

To help build the SSP, Renesas engineers drew some key pieces from Express Logic's X-Ware™, an integrated set of software components optimized for IoT applications. The SSP uses ThreadX®, a popular multitasking RTOS, with many of middleware components including NETX™, NETX DUO™, USBX™, FILEX® and GUIX™. With over six billion installations, ThreadX is an industry-proven RTOS. This priority-based, fully preemptive, deterministic RTOS offers basic system services such as pre-emptive and round-robin scheduling, semaphores, message queues, timers, interrupts and memory management. Advanced features such as preemption-threshold scheduling reduce context switches and the RTOS' deterministic performance remains consistent regardless of the number of threads or objects in an application. ThreadX also includes numerous safety certifications and features integrated event trace capability through TraceX® and the third party Percepio Tracealyzer. Developers can trace their application code which allows them to see what the application is doing rather than just guessing. Trace capability coupled with run-time stack analysis can dramatically help developers decrease the time they spent debugging their software and decrease the number of bugs in their production code.

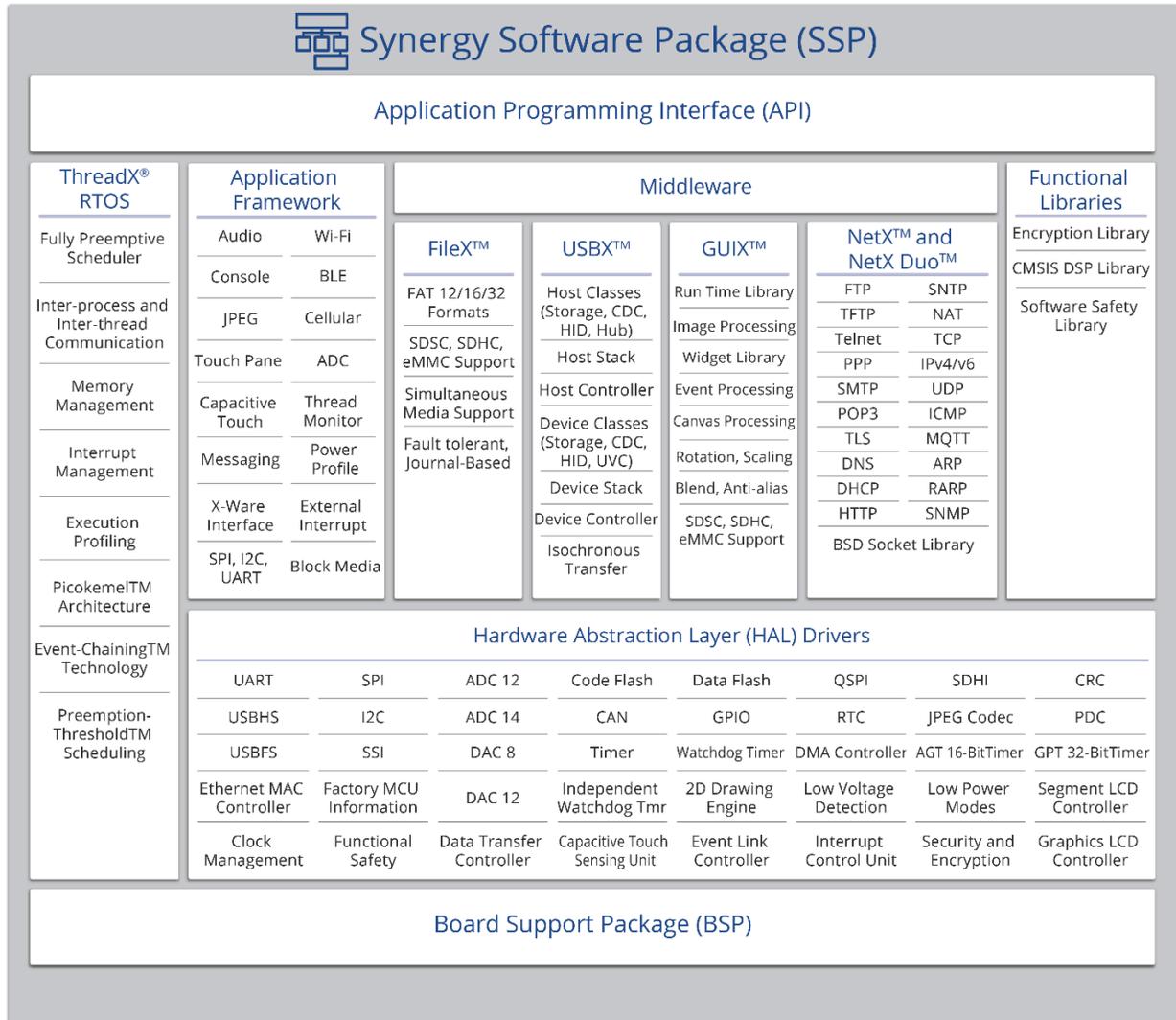


Figure 4: Synergy Software Package (SSP)

NETX and NETX Duo provide high performance TCP/IP stacks in a small footprint. NETX is IPv4 capable while NETX Duo supplies both IPv4 and IPv6 capabilities in a dual stack product. A Zero-Copy API helps optimize buffer management. UDP fast path technology allows packets to pass through NETX without copying or any system context switches.

The NETX core stack supports TCP, IP, UDP, ARP, RARP, ICMP and IGMP. The NETX applications bundle includes PPP, DHCP, DNS, FTP and other functions.

USBX, the USB protocol stack for embedded software includes host, device and On-the-Go (OTG) support. The stack supports low speed (1.5 Mbps), full speed (12 Mbps) and high speed (480 Mbps). USBX is very easy to use and supports a wide range of device classes including CDC, HID, UVC, Mass Storage.

FILEX is the SSP's MS-DOS compatible file system. Tightly integrated with ThreadX, it boosts performance by minimizing function call layering, using contiguous cluster allocation, and consecutive cluster reading and writing. Advanced features include FAT 12, 16, 32-bit support, real-time performance, internal FAT entry cache, and internal logical sector cache. It also supports RAM, flash and a variety of physical media in addition to long filename and unlimited FILEX objects. Integrated functions like error detection and recovery, fault tolerant options and built-in performance statistics boost reliability.

GUIX offers a graphical user interface framework. It includes a full runtime UI library and a matching desktop design application called GUIX studio. The high performance graphical library is optimized specifically for the Synergy MCU graphic engine to support hardware accelerated 2D graphics operations, complete windowing support with multiple displays and different resolutions, and the use of multiple languages. The core library includes a wide array of buttons, windows, scroll bars, test display and text editing controls. The supporting framework features event queues and signals, windowing and viewport management, and other functions.

When it comes to peripheral drivers, the SSP gives developers a choice. Framework peripheral functions offer easy-to-use, feature-oriented functions for application programs. The framework automatically takes care of the details of the RTOS integration. Since the drivers abstract hardware registers by logically defined values, the API and parameters are consistent across different Synergy MCUs. This approach allows the developer to build solutions without spending time learning Synergy MCU hardware specifications or ThreadX specifics. Instead developers can concentrate on building their own application. The application framework also offers commonly used system services such as an elaborate message-passing infrastructure for inter-process communications, an audio framework for simple development of audio playback capability, an infrastructure for easy management of different power profiles of an application and other functions.

A complete set of low-level peripheral driver modules are available for a wide array of functions including memory, connectivity, analog, timing, system and power management, security and encryption, safety and human machine interface. Embedded developers who want access to individual peripheral drivers directly, outside the framework, may do so with direct calls from the application to meet specific requirements or to operate within time-critical bounds.

Additionally, the SSP also includes many wireless frameworks for Wi-Fi, Bluetooth® and LTE IoT cellular connectivity. These frameworks abstract the hardware drivers of many popular radio chipsets and provide a quick and easy way to add wireless connectivity to your applications.

To extend the functionality of the SSP, Software Add-ons are available to enable a wide range of specialized functions, including communication protocols, extended security

functions, and cloud services. Some add-on components are licensed and serviced by Renesas, while others are developed by third parties and verified by Renesas to be SSP-compatible. Software add-ons “verified” to work with the Renesas Synergy Platform and are named VSA, or Verified Software Add-on components. These VSA components have been tested for functionality and compatibility with the SSP and approved by Renesas. Examples of VSA components are communications stacks for CANopen, DALI lighting, or BACnet.

Productizing software

The SSP was developed using industry-accepted best practices and following a robust software development life cycle, the SSP is a qualified software product with an accompanying datasheet. A software datasheet for MCUs is an industry first. The SSP meets widely accepted standards, tests and benchmarks, such as MISRA-C, as well as published SQA metrics, documented processes and product lifecycle practices such as ISO/IEC/IEEE 12207. Moreover, Renesas provides full product-level support for the SSP including maintenance with scheduled update releases, errata publishing and management, issue tracking and bug fixes.

Software Access and Licensing

Obtaining and licensing the SSP and the VSA components is easy using the Synergy website.

Developers can create or log in to their MyRenesas account to download Synergy Software. With those credentials, a user can download the SSP that includes an evaluation license providing access to the entire SSP during evaluation. When it's time for production, the developer can easily obtain a production license from the Synergy website by registering their company with a simple click-through process. With the production license, the customer's company gains the right to use the SSP for product of end-products. There is never an additional fee or royalty payment required for licensing and using the SSP – everything is included in the price of any Synergy MCU. There is no limitation to the number of end-products that use the SSP, and SSP maintenance is included while the Synergy MCU is commercially available from Renesas.

The source code of the entire SSP is visible during development and debugging but some protected portions are for read-only. For example, within the Synergy development tool suite, developers can view the source C code of all SSP components while single-stepping through the RTOS and communications stacks. However, some portions of SSP code are protected, meaning that those portions cannot be changed or saved outside the Synergy tool environment. However, the protected portions of SSP include the RTOS, network stacks, and other middleware that typically never need to be

altered. The vast majority of Synergy customers do not need to alter protected code for their applications.

If developers wish to obtain or modify the source code of protected SSP components, they can purchase a customized mix of these components. They will be given a quotation document that can be exercised with their local Renesas sales representative to complete the purchase of a source code license for the selected components. Once the source code license is entered into the tool suite, the purchased software components become unprotected and developers are free to save the source code of those components to a file, modify the source files, and print the source files for the term of their maintenance. Many of the remaining components of the SSP are not protected and are distributed as clear-text C code files within the SSP distribution. These include the low-level Synergy MCU peripheral drivers, board support packages (BSPs), various libraries, and other SSP components.

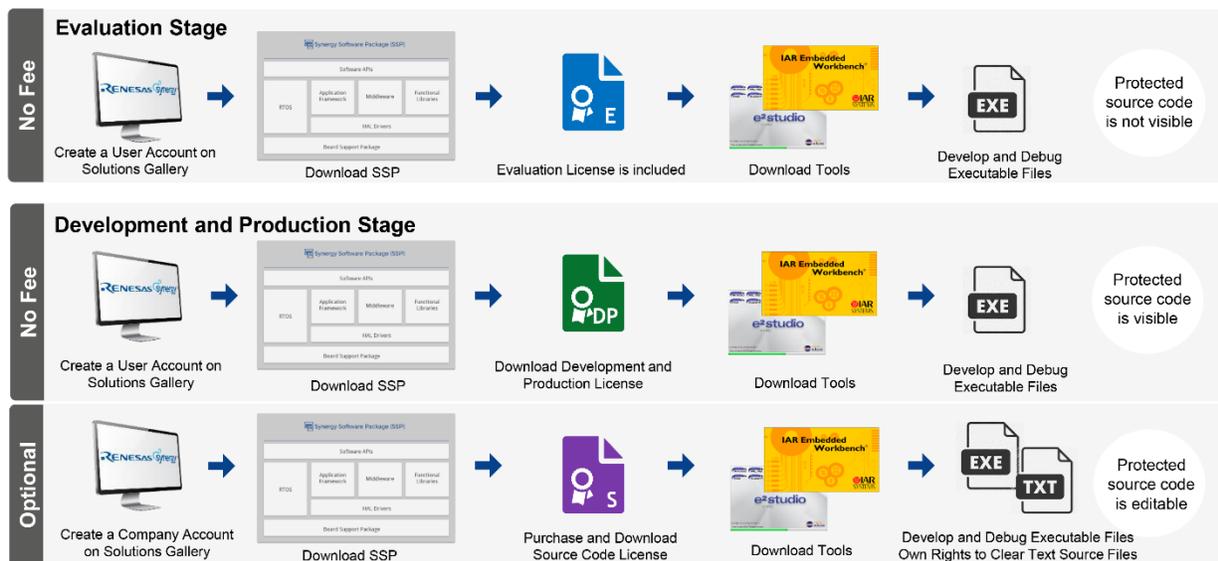


Figure 6: The SSP is available free-of-charge with any Synergy MCU, with simple click-through licensing.

Reflecting this new integrated hardware/software philosophy, Renesas has restructured its support practices to offer full product-level support for silicon and software together. Developers seeking support can resolve all hardware and software issues via a single point of contact.

Conclusion

For the MCU-based embedded market, the age of the “platform” has arrived. Today developers must build higher performance, increasingly complex solutions in a fraction of the time. To accomplish that task and meet shrinking development timetables, they need integrated platforms that allow them to virtualize the hardware and software and differentiate their end products using APIs. As an innovative hardware/software solution for embedded applications, the Renesas Synergy Platform offers developers an exciting, new opportunity to focus on the most crucial aspects of their design.

(Remarks) Contents of this article are subject to change.

© 2018 Renesas Electronics America Inc. (REA). All rights reserved. All trademarks are the property of their respective owners.