Renesas Synergy™ Platform Charts Revolutionary Path to Faster Microcontroller System Development for IoT and Embedded Products

Integrated Hardware/Software Solution Shrinks Time Spent on Non-Differentiated Functions, Frees Developers to Focus on Application Code

When Renesas engineers first began to consider how they would package microcontroller (MCU)-based solutions for the Internet-of-Things (IoT), they took a close look at the unique challenges the IoT presented. With applications running the gamut from factory automation, home appliances to wearable computing, the opportunities appeared to be endless. Clearly developers who could deliver their products to market first would gain a significant advantage. Solutions that could shorten the developer's time in development would offer a powerful advantage.

But the Renesas team noted something else as well. As they explored the IoT across multiple industries, they noticed the traditional MCU customers profile was changing. As a leading MCU supplier for decades, Renesas engineers typically worked with a customer's hardware engineers. Often, the customer's hardware team came to Renesas with a specific specification they were looking to implement. Once the hardware was defined, the customer's hardware engineers would pass the solution onto its software team to develop solutions. For decades hardware engineers drove the MCU selection process.

As the IoT market began to emerge, roles were changing. The more Renesas interacted with its customers, the more it became increasingly clear that the lead negotiator was no longer the hardware engineer. Software developers were now defining the end solution and, with it, which MCU would best fit their systems software requirements.

The Renesas team grappled with how to address this new reality where software has become the major driver in embedded system design rather than the hardware. This resulted in several new questions being posed:

- What were the key values driving customers?
- How would those priorities impact MCU selection?

Clearly any solution targeted at the huge IoT market would have to take into account the increasing importance of the software engineer's point of view.

Increased Embedded Design Complexity

Time-to-market has always played an important role in embedded system design – but with the IoT and connected devices, now it was critical. Embedded systems have steadily become more complex as end users request more and more features in their devices. As devices become connected, developers are now faced with far more

complex applications with time spent developing software consuming ever larger portions of the development cycle. A recent study by UBM found that software development consumed almost twice as much time, budget and manpower as hardware development and as projects have taken longer to develop, more projects have been delivered late.

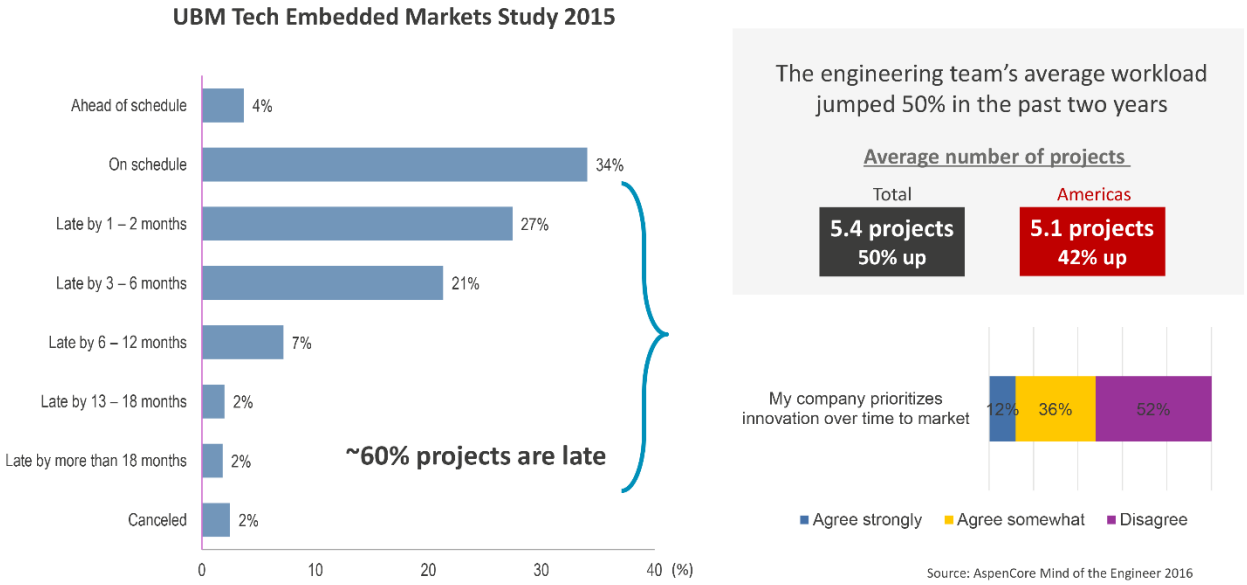

Figure 1: *Recent research indicates that software development is driving up the cost and time needed to finish embedded designs.*

What's responsible for these delays? According to the UBM survey, the problem was attributed to increasing code complexity, integration of new technologies, and increasing amounts of time spent with software development tools and operating systems. In traditional development cycles, designers devote large blocks of time to hardware design, driver design, middleware development, RTOS integration, and connectivity options. These common software development tasks take time away from the actual application code. Yet all those software design tasks mentioned above represent non-differentiated functions. Often at the end of their development cycle when designers are typically under severe time constraints to get to market they are forced to devote substantial amounts of time to the truly differentiating aspects of their design – the application code.

At the same time, other trends are escalating the pressure on design teams. US and European electronics developers are increasing outsourcing except for the core engineering functions required to differentiate their end-products. As engineering resources shrink, a growing number of firms are finding they can no longer afford to develop new base technologies, or new technology building blocks in-house as they have in the past. As a result, developers are increasingly willing to move away from the

traditional in-house development of these core technologies and use external resources and off-the-shelf components. However, it's still a time-consuming and costly approach to use a standard MCU and then research, license, integrate, test, and maintain a RTOS, connectivity stacks, middleware, and libraries from multiple software vendors. What if they could get everything from one source - their MCU vendor?

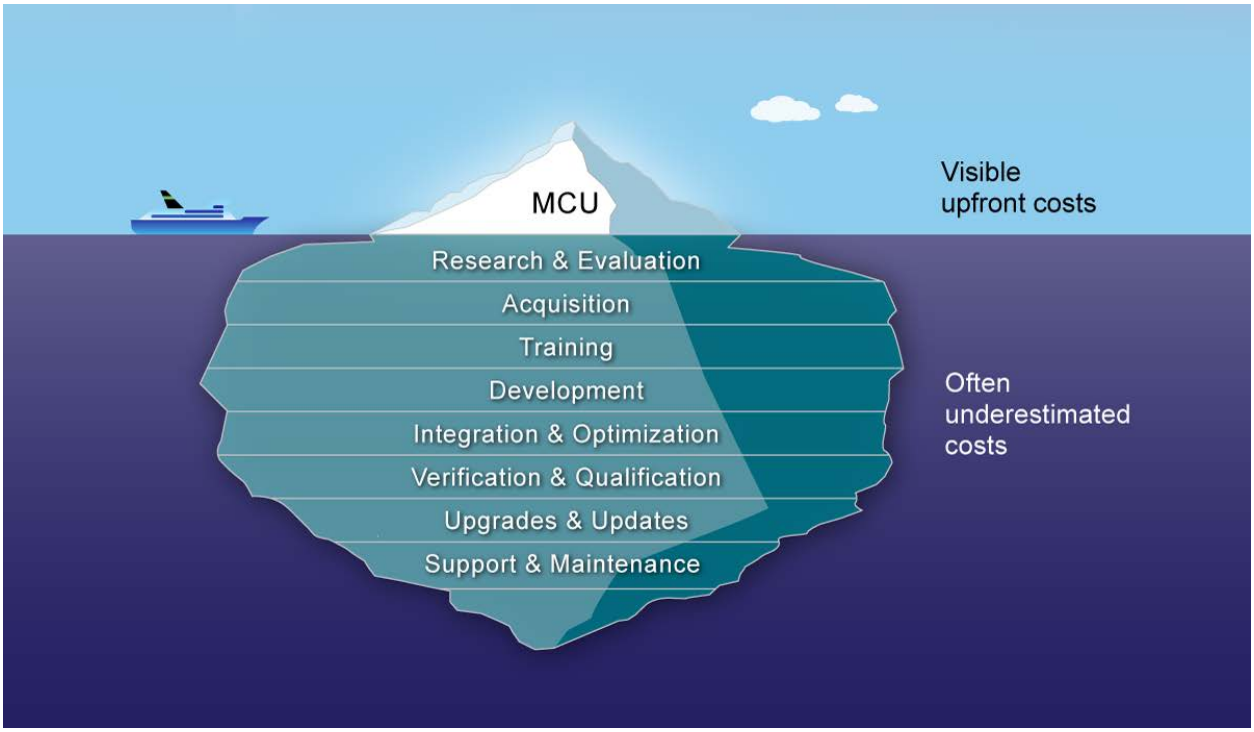

Figure 2: *By simplifying the creation and management of key software components and tools, the Synergy Platform reduces the total cost of ownership.*

Traditionally, MCU vendors have provided example code that functioned but was far from being usable in a production intent system. To source their software from their MCU vendor, developers would need assurances such as:

- How long will the silicon vendor supply and support the software?
- Will the vendor make a long-term commitment to maintain, upgrade and support the software?
- Developers generally equate commercial quality with longevity. If the silicon vendor was willing to sell its software as a high quality commercial product, would that assure embedded system developers?

From the developer's standpoint, sourcing the software from an MCU supplier offers numerous advantages. Large, established silicon vendors generally have the resources to support the development and maintenance of software over long periods of time. And

since the silicon manufacturer usually has a large customer base, the software will be tested and proven over a very large number of customers and applications. Moreover, sourcing the silicon and software from the same company links supplier and vendor interests. Since the silicon vendor only makes money when products go into production, it has the same interest as the customer in accelerating the development cycle. Finally, the silicon manufacturer has a vested interest in maintaining software quality because the success of its software is directly tied to its ability to continue selling MCUs.

The Renesas team wondered what if designers could re-invent the development cycle so that more of their time is spent focusing on the truly innovative aspects of their solution? What if MCU suppliers like Renesas could reduce the amount of time designers spent on those non-differentiating software development functions and, instead, allow developers to increase the amount of time they could spend on their application code?

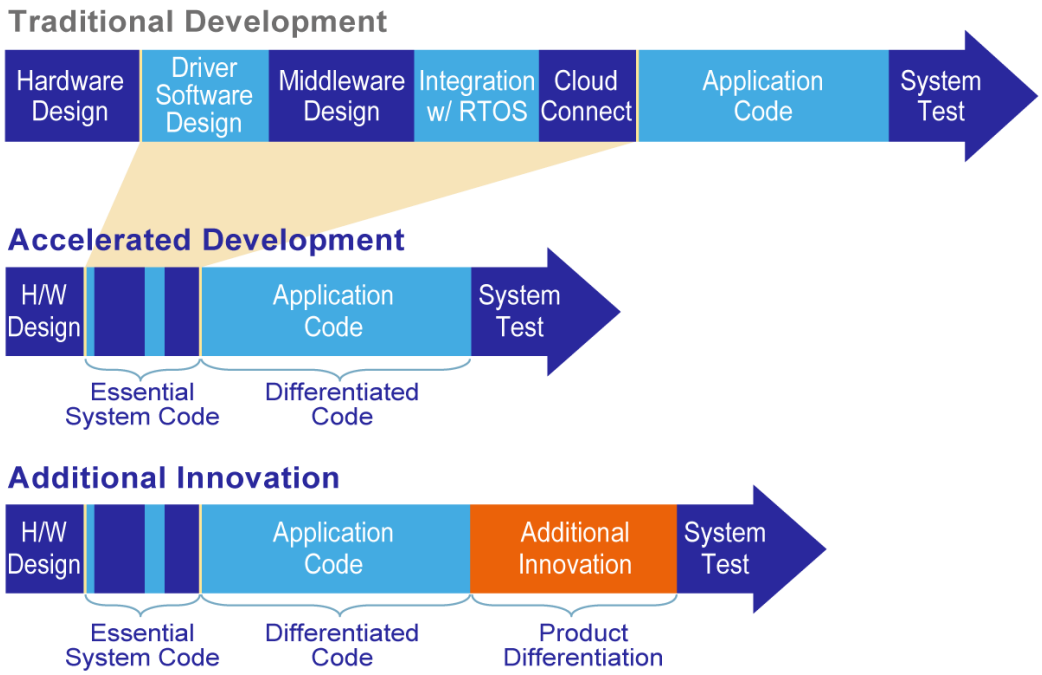

Figure 3: *Developers can free themselves of spending time on low-level tasks, but instead focus on getting to market before their competitors do.*

What would this new product development cycle look like? In the traditional development cycle designers must grapple with continual updates and constantly changing roadmaps for their RTOS, stacks and tools. What if the MCU supplier offered a synergistic platform that took care of the integration of new updates so developers could shrink that portion of their development cycle and spent more time innovating? And what if the MCU supplier offered and supported that software as a high quality

commercial product free-of-charge? What if the software would be complete with APIs and abstraction layers that would allow the IoT developer to simply build his or her own applications on top of that software base with minimal knowledge of the hardware? Ultimately, those questions led to the development of the Renesas Synergy™ Platform, an integrated hardware and software solution optimized for embedded and IoT applications.

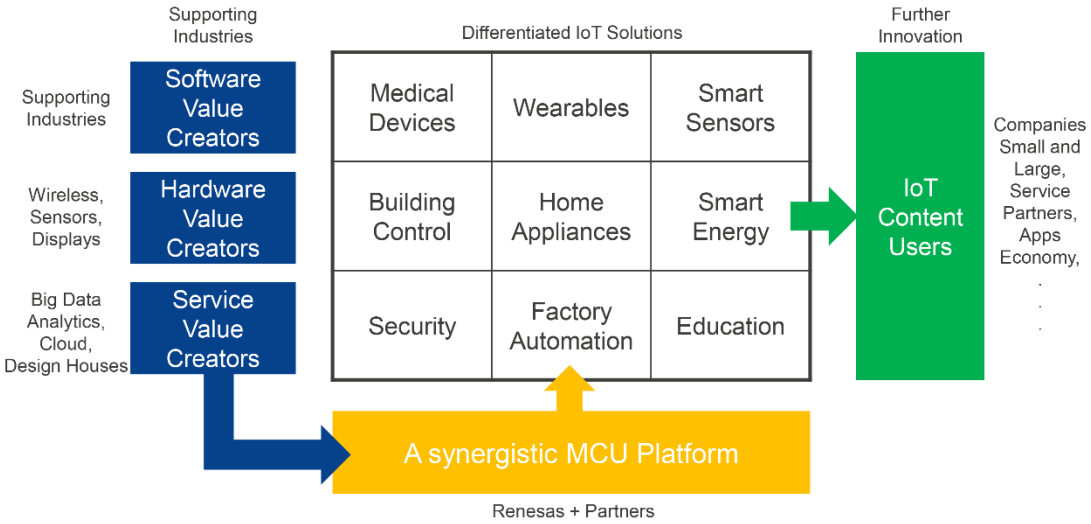

Figure 4: *The Synergy Platform offers a model for the future of embedded computing.*

Synergy Platform Elements

The Synergy Platform consists of three main elements:

1. Synergy Software (renesassynergy.com/software)
2. Synergy Hardware (renesassynergy.com/hardware)
3. Synergy Solutions Gallery (renesassynergy.com/solutionsgallery)

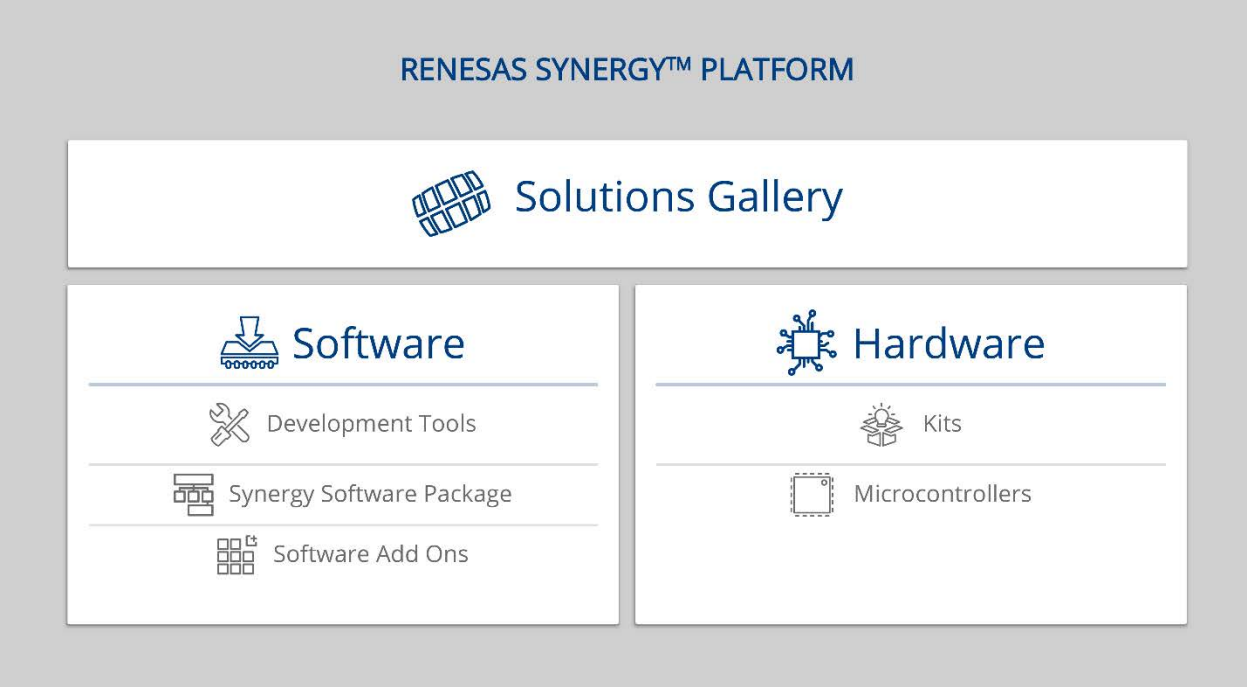

Figure 5: *The core elements of the Synergy Platform*

1. Synergy Software

Synergy Software consists of Synergy Software Package, software add-ons and development tools. The leading element of the Synergy Platform is the qualified and maintained embedded software which is tested to commercial standards and has ensured compatibility with the Synergy MCUs, all supported by Renesas.

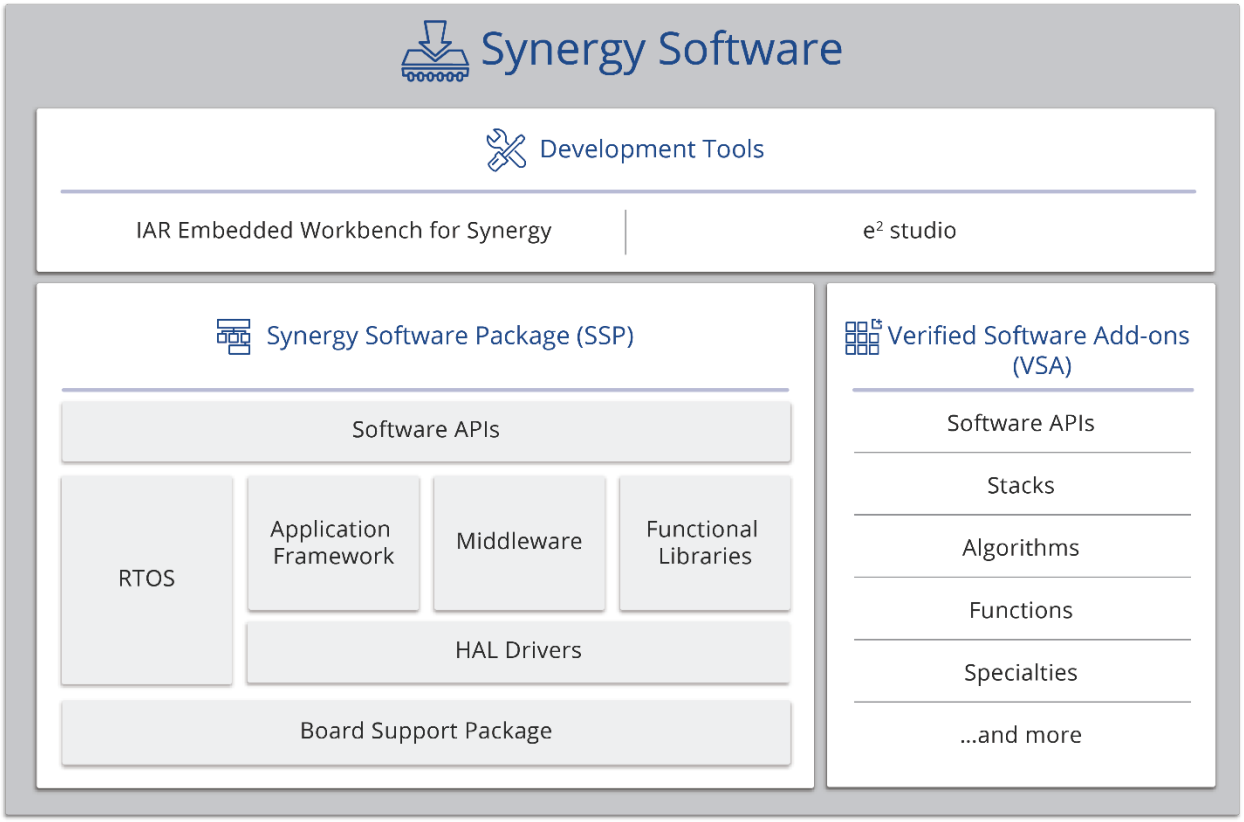

Figure 6: *Synergy Software*

.

The qualified Synergy Software components are not only optimized and integrated to work seamlessly with the MCUs, they are also rigorously tested for both performance and reliability. Sold, maintained, and directly supported by Renesas as a product, these qualified components undergo multi-level testing, and are documented with a software datasheet and a test report - a first in the MCU industry.

1.a Synergy Software Package (SSP)

The SSP supplies the key software components needed for functions essential to most of embedded system and IoT applications. Renesas engineers selected premium, quality components to form the basis for the SSP. These components include Express Logic's ThreadX® RTOS, and middleware components from Express Logic's X-Ware™. These components are integrated with MCU device-specific Renesas software components including device drivers, middleware, libraries, and a flexible application framework with an API.

ThreadX is a popular industry-proven, priority-based and deterministic multitasking RTOS offering basic system services such as pre-emptive and round-robin scheduling, semaphores, message queues, timers, interrupts and memory management with

advanced features such as preemption-threshold scheduling to reduce context switches and an integrated event trace capability and run-time stack analysis. ThreadX RTOS has been deployed in more than six billion electronic products spanning a variety of markets since 1996.

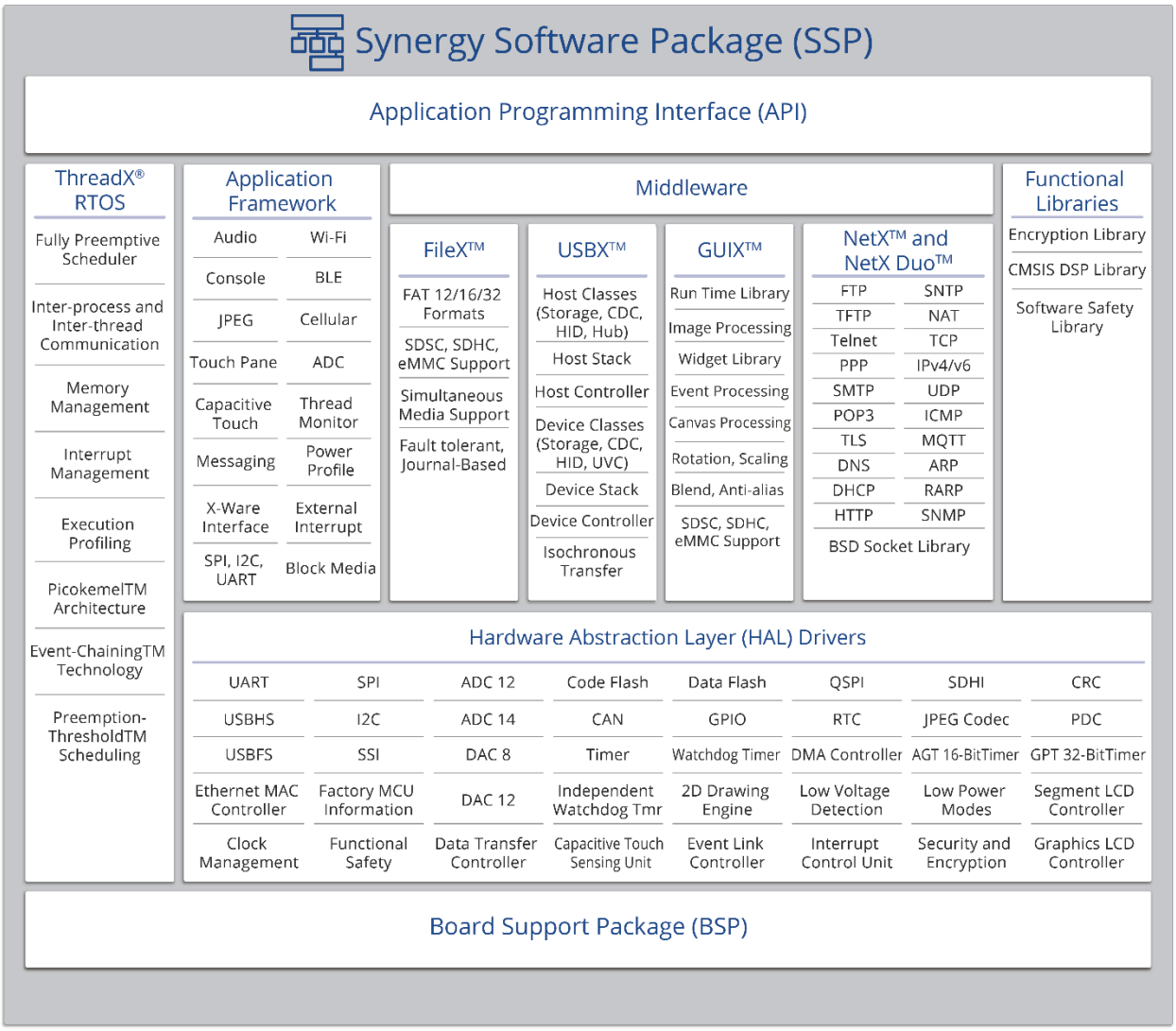

Figure 7: *Synergy Software Package*

For TCP/IP stacks, Renesas engineers chose NetX™ and NetX Duo™. These two X-Ware™ components offer IPv4 and IPv6 capability in a small and efficient footprint that was perfect for embedded applications. USBX™ supplies the USB protocols stack with host, device and On-The-Go support. FileX®, another X-Ware component, gives the SSP a MS-DOS compatible file system for interfacing to memory devices such as an SD card. Developers interested in developing a Graphical User Interface also have access to GUIX™, an X-Ware component that Renesas and Express Logic engineers

specifically optimized for the Synergy MCU graphics engine. GUIX Studio is a desktop design application provided to the platform developers at no charge, enabling them to easily create a GUI layout before realizing it at the silicon and embedded software level.

The SSP blends the RTOS, middleware and libraries with the low level peripheral functions through an API and hardware abstraction layer (HAL). This allows the application to access peripherals as easy-to-use, feature-oriented functions. The framework automatically takes care of the details of the RTOS integration. Since the driver frameworks abstract hardware registers by using logically defined values, the API and parameters are consistent across the different microcontroller series within the Synergy MCU Family. This approach allows the developer to build solutions without spending time learning detailed MCU hardware specifications, specific register definitions, or ThreadX specifics.

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 application-specific requirements or to operate within time-critical bounds.

1.b Synergy Software Add-ons

Software Add-ons extend the functionality of the SSP 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.

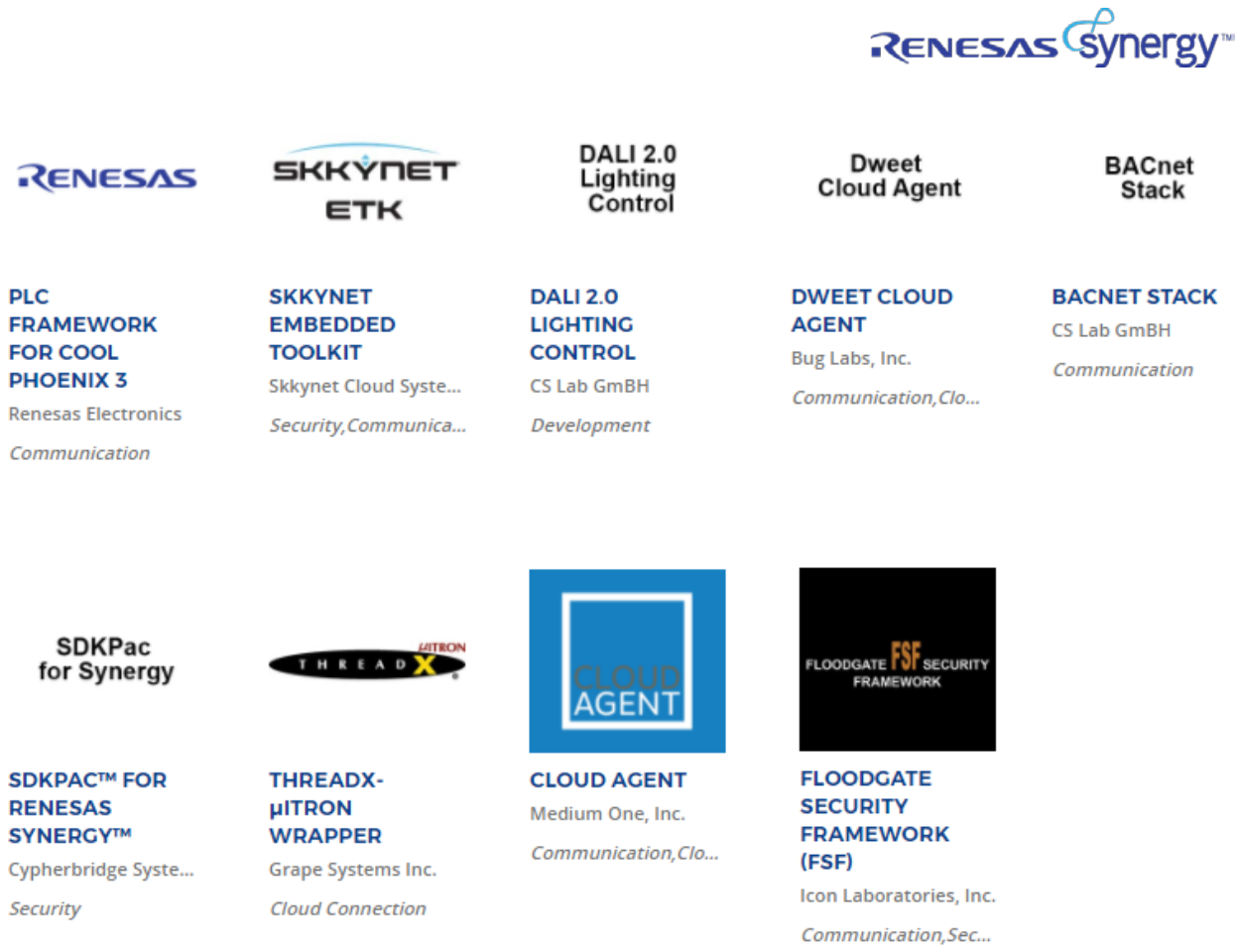

Figure 8: *Example of Synergy Verified Software Add-Ons (VSAs) on the Synergy website,* renesassynergy.com/software

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.

VSA components are available on the Synergy website for download as evaluation files which are in binary form or time-limited versions that have been verified to be compatible with the Synergy Platform by Reneas. Customers who want to purchase these VSA components can access the third-party VSA vendor to purchase a license, source files, maintenance, and support on the VSA vendor’s terms.

1.c Development Tools

Designed to accelerate time to market through rapid code development, Synergy Development Tools facilitate file management, software and MCU configuration, code generation, compilation, debugging, and intuitive graphic interface design. All Synergy Tools, support, development seats, and maintenance are included with the Synergy Platform. Developers can use e^2 studio, the Eclipse-based Integrated Development Environment (IDE) or IAR Embedded Workbench® for Renesas Synergy™ IDE for creating, compiling, and debugging the embedded applications. The development tools are available at no additional charge.

Eclipse is the de-facto standard when it comes to embedded IDEs and by adding new, solution-oriented components, Renesas engineers transformed the environment and e^2 studio into a true ISDE. This platform provides easy and innovative ways to develop applications on the Synergy Platform. From mapping I/O pins to setting up a clock tree or easily adding and configuring software modules, it can all be done graphically with

initialization C source code being generated in the background. Even adding and configuring ThreadX RTOS threads becomes an easy task. The e^2 studio IDE provides the industry-standard GNU GCC Arm Cortex®-M compiler or the IAR C compiler. The popular J-Link® from SEGGER was selected as the debug probe for the Synergy Platform, with code analysis functionality provided via the Codan plug-in for Eclipse.

In addition to having e^2 studio available for use as an IDE, developers can also use IAR Embedded Workbench for Synergy. Included in the IAR IDE is its full suite of development tools such as the C-Stat static code analyzer. IAR integrates seamlessly into the Synergy design process using the Synergy Standalone Configurator (SCC) tool that allows developers to easily configure their software stacks and drivers along with the RTOS.

To match traditional hardware tracing capabilities on this integrated hardware/software platform, the tool environment adds a high degree of RTOS awareness. Developers need to view correlation between different operations to trace RTOS functionality over time. To accomplish this task, the platform adds Express Logic's TraceX, a host-based analysis tool that provides a graphical view of real-time system events providing a holistic view of code execution and timing.

Developers using TraceX can track the occurrence of system events such as interrupts and context switches, identify the timing of events in the context of overall system operation, and more easily resolve programming problems. TraceX works with ThreadX which constructs a database of system and application events on the target system during runtime. Events are logged with time-stamps and active threads are identified so they can be displayed later with a proper time sequence and associated with the appropriate thread. TraceX displays events graphically on a horizontal axis representing time with the various application threads and system routines to which events are related listed along a vertical axis. A summary display helps a developer analyze systems with many threads by showing all system events on a single horizontal line.

Developers can also trace their applications using third party tools such as the Percepio Tracealyzer. Tracealyzer provides functionality similar to TraceX but provides advanced capabilities for analyzing the software trace to determine whether the application is behaving as expected and for discovering those difficult to find bugs.

Taken together with the environment's traditional hardware debug capabilities, these capabilities give developers using the Synergy Platform tracing capability on both the hardware and the RTOS level. Visit renesassynergy.com/devtools to learn more.

2 Synergy Hardware

Synergy Hardware consists of Synergy MCUs and Kits for developers to design a wide range of embedded systems and IoT devices, from efficient battery-powered applications to high performance connected products.

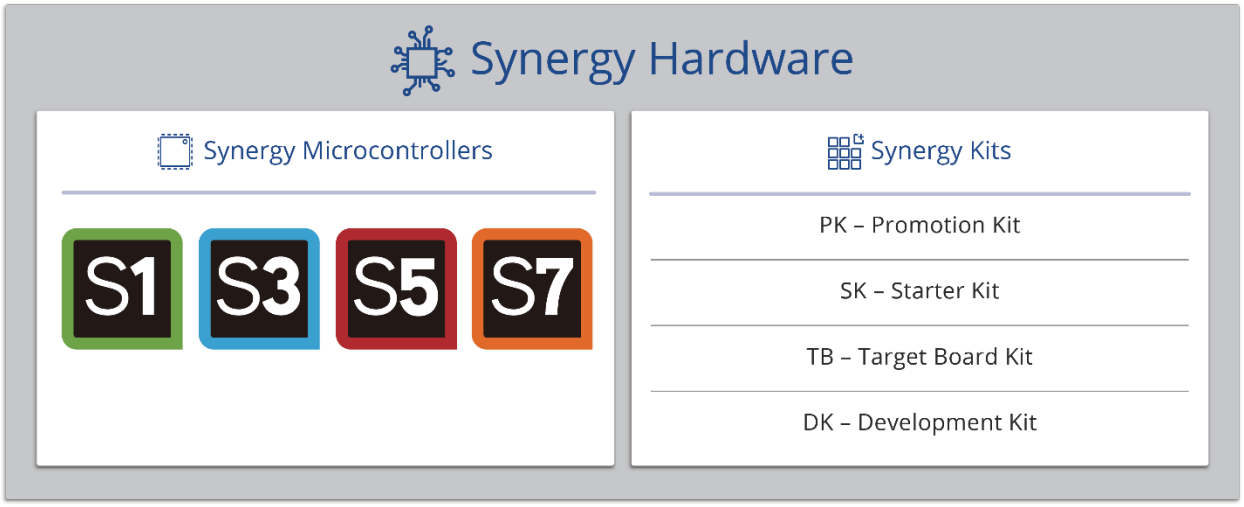

Figure 9: *Example of Synergy Verified Software*

2.a Synergy MCUs

The device foundation for the Synergy Platform is a family of compatible and scalable 32-bit MCUs based on Arm's Cortex-M CPU cores. Since the Synergy MCUs were designed from a clean slate with no legacy requirements, compatibility and scalability were factored in from the beginning. Compatibility means that across the board, all members of the Synergy MCUs have the same or similar peripherals to minimize the learning curve and maximize re-use of software. Also, the pin definition is the same or similar for all like-packages across the entire family for easy migration to higher or lower functionality. Scalability means that peripheral capabilities scale from lower to higher and higher to lower while keeping the same register footprint. For example, a simple 16-bit timer peripheral and a complex 32-bit version of the same timer have the same basic control registers. The 32-bit version adds registers to match the added functions orthogonally which have no effect on the 16-bit version. Also, the register address offsets are designed to simplify the software; if a timer function does not exist, the register also does not exist, but it does not change the overall register address offset scheme.

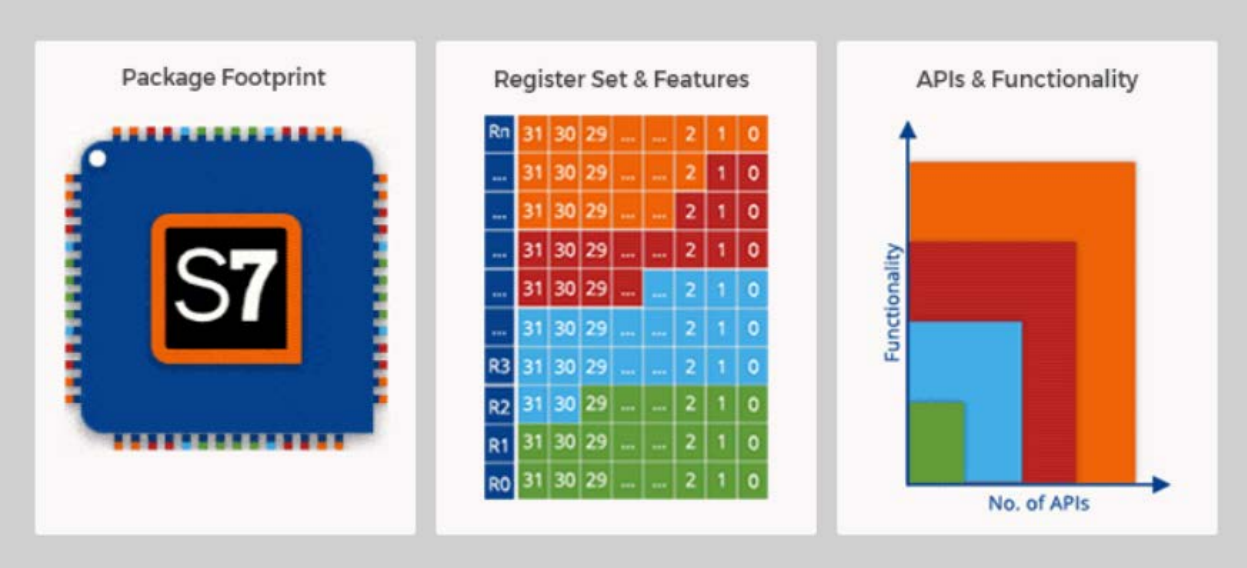

Figure 10: *Synergy MCUs are designed to enable high software re-use when migrating from one MCU to another, reduce PCB layout efforts, and increase manufacturing efficiency.*

The Synergy MCUs begins with the S1 Series, an ultra-low-power MCU based on a 32 MHz Cortex-M0+ core. Three additional series of the Synergy MCUs are, the S3, S5 and S7 series that use Cortex-M4 cores to support operating frequencies that range up to 240 MHz. Designed with industrial automation, Human Machine Interface, communication gateways, and similar embedded applications in mind, the Cortex-M4 features extended single cycle multiply accumulate (MAC) instructions, optimized SIMD arithmetic, saturating arithmetic instructions and a single precision Floating Point Unit (FPU). These architectural attributes along with integrated sleep modes and state retention capability allow the Cortex-M4 to deliver excellent performance at very low power levels.

S3 Series MCUs are based on a 48 MHz Cortex-M4 core and serve applications that demand higher levels of integration than the Cortex M0+-based S1 series. The S5 Series MCUs target more complex IoT applications using a Cortex-M4 core running at 120 MHz. At the top end of the performance spectrum, Renesas' S7 Series combines a 240 MHz Cortex-M4 core with a wide array of high-speed peripherals. The Synergy MCUs offer significant amounts of memory on-chip all the way up to an industry-leading 4 MB of code flash and 640 KB of SRAM on the S7 Series. Shown below is the MCU portfolio. Visit renesassynergy.com/mcus to learn more.

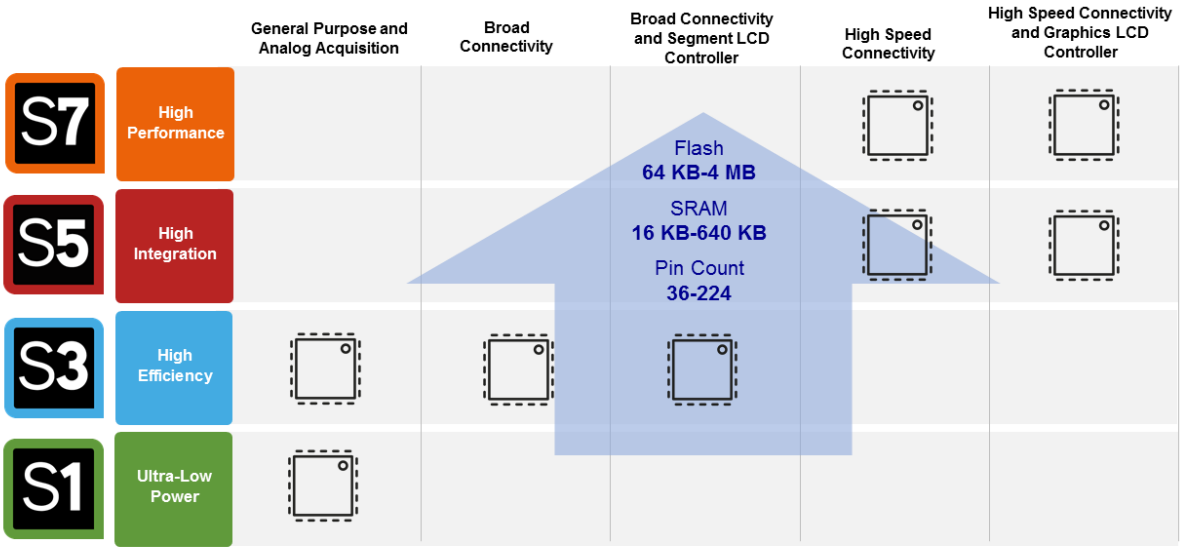

Figure 11: *The Synergy MCU Family covers a broad application range.*

2.b Synergy Kits

Developers who want to accelerate their development cycle and take the guesswork out of using a wide variety of technologies in their own IoT end-products can do so using the Synergy Platform's wide array of kits and design examples. Synergy Kits are designed to help developers evaluate, prototype, and start developing using Synergy Platform quickly. Developers can choose from Target Boards Kits (TB), Starter Kit (SK), Development Kit (DK), Promotion Kit (PK), Application Examples Kit (AE) and Product Example Kit (PE). The Synergy Kits feature many components that make it easy to utilize the peripherals of the Synergy MCUs and to connect a variety of sensors, actuators, etc. so you can prototype your end-applications faster.

Development Kits offer a complete MCU hardware development platform with full access to virtually all the capabilities of the MCU. Developers may access all the MCU features and pins to evaluate device performance and power consumption, to build a core software application until their own specific hardware development platform is available, and even to expand capabilities by plugging in specialized circuit boards into DK expansion connectors and industry standard Pmod™ connectors. DKs feature on-board J-Link JTAG debug access, as well as a Bluetooth low energy (BLE) radio for wireless connection to a mobile device.

Figure 12: *Synergy Development DK-S7G2*

As a lower cost alternative to the Synergy DKs, Renesas offers Synergy Starter Kits that introduce the entire Synergy Platform and outline steps to begin development. The Starter Kit also provides expansion via sets of connectors based on the Pmod standard and on the standard Arduino™ format for Arduino Shield plug-in boards.

Figure 13: *Synergy Starter Kit SK-S7G2*

The Target Board Kits offer the lowest cost means to access to all pins of each of the Synergy MCU Group. It features on-board debugging interface, Pmod connectors for expansion, USB device, capacitive touch sensing, means to measure MCU current consumption, and debug with the J-Link OB on-board debugger, or bypass key functionality and access pins directly.

Figure 14: *Synergy Target Board TB-S3A6*

Developers who want guidance on how to implement a specific end-product or on how to use multiple technologies can refer to multiple design examples in the form of Product Examples (PEs) and Application Examples (AEs). Products Examples represent an actual end-product for reference, and Application Examples showcase specific technologies. Visit renesassynergy.com/kits to learn more.

3.Synergy Solutions Gallery

The Synergy Solutions Gallery makes it easy to find all Synergy Platform-compatible components, software, hardware, tools, kits, and services.

In the Solutions Gallery, users can find items in these sections:

- Software
- Software Tools

- Hardware Components
- Kits
- Partner Projects
- Hardware Tools
- Production Tools
- Application Projects
- Security
- Cloud Connectivity
- Human Machine Interface
- Design & Testing Services

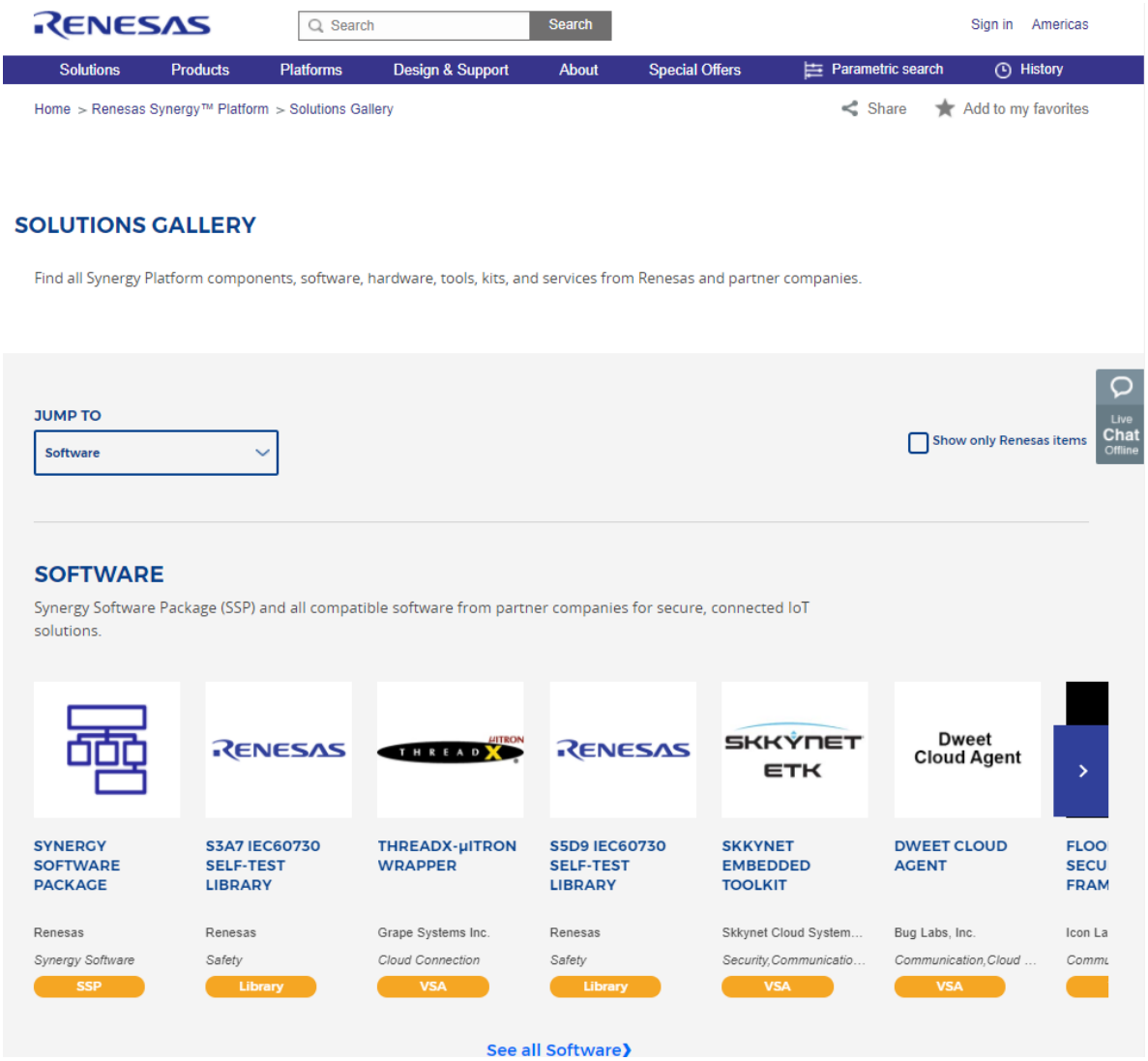

Figure 15: *Synergy Solutions Gallery, example of Software section*

Security and Safety

To address the design challenges inherent in the majority of IoT applications, the Synergy Platform offers developers a wide array of options to boost security, safety and communications capabilities. The security and encryption blocks available on Synergy MCUs enable developers to protect data inflight and at rest on the MCU, ensure authenticity of data and design IP, and achieve secure product lifetime management. Functional security blocks and software libraries make use of primitives for hashing algorithms and symmetric/asymmetric cryptography, as well as secure key generation and storage to provide a full set of secure services including secure boot and over-the-air firmware updates. And each MCU adds a long list of safety functions including ECC in RAM, ADC diagnosis, CRC Calculator, Flash Area Protection, SRAM Parity Error Check and Data Operation Circuit to name just a few.

Threat	S7	S5	S3	S1
Product cloning	Best	Best	Better	Good
Product disruption with malware injection during update	Best	Best	Better	Good
Eaves-dropping during update	Best	Best	Better	Good
Privacy threat by firmware/date exposure	Best	Best	Best	Good
Add-on program to damage or steal	Best	Best	Best	Limited

Figure 16: *Synergy MCUs security features*

Connectivity and Signal Acquisition

Interconnectivity is obviously a high priority in the IoT market and the Synergy Platform addresses this need with an extensive lineup of capabilities. The S7 series MCUs, for instance, offer dual Ethernet with IEEE-1588 synchronization, high-speed USB, plus many serial interfaces on chip including UART, I^2C, SPI, IrDA, QSPI, I^2S, SDHC/MMC and CAN. For applications closer to the edge, virtually every Synergy MCU also features a full array of analog interfaces including analog-to-digital, digital-to-analog converters, analog comparators, as well as temperature sensors. They also add a suite of timing functions for motor and industrial control applications. For cloud-related connectivity, the Synergy Platform offers secure TLS-based communications and connection to major enterprise cloud service providers.

Best in Class Help and Support

Consistent with the unified hardware/software platform concept, Renesas provides product-level support for both silicon and software. Online self-help and personalized support is provided in the form of extensive documentation, knowledgebase, forums, training, videos, chat and web tickets at renesassynergy.com/support.

Conclusion

The fast-moving IoT market is forcing developers to re-evaluate their traditional approach to product development. Today's embedded designs are simply too complex to develop in a step-by-step, serial fashion where designers take a hardware design and painstakingly build their software infrastructure around it. The market demands a faster response and shorter time-to-market.

The Synergy Platform promises to meet that need. By offering a compatible, scalable portfolio of solutions, the platform allows developers to find the best fit for their application. And by adding pre-integrated, tested, commercial-grade software, Synergy Platform assures developers of predictable behavior while reducing the time and energy spent developing non-differentiating functions and ensuring maintenance and support. Most importantly, this new approach allows developers to devote more time to the truly innovative aspects of their design and deliver better products to market faster.

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