

# Simple and efficient software development with the SimpleLink™ MCU platform



Henry Wiechman  
*Embedded Processor Software Marketing and Strategy*  
Texas Instruments

Adrian Fernandez  
*Microcontroller Development Experience Manager*  
Texas Instruments

# Overview

---

The software challenges associated with embedded systems development have grown significantly in recent years. These systems continue to add advanced functionality with higher speed interfaces and multiple connectivity options, driving the need for more sophisticated scheduling algorithms and increased power consumption and code size. End-product needs are also changing rapidly as companies strive to serve multiple markets with different adaptations of their base product. As time-to-market windows shrink, developers need a robust software foundation with intuitive levels of abstraction and operating system support to enable faster creation of applications. This foundation should support a broad portfolio of devices that can easily reuse application code so that initial investments can be ported to a variety of products with varying system requirements.

The SimpleLink™ microcontroller (MCU) software development kit (SDK) is a complete set of validated, fully documented drivers, stacks and code examples that enable engineers to develop innovative and differentiated applications with the [SimpleLink connected-MCU family](#) from Texas Instruments (TI). This powerful SDK provides a cohesive and consistent experience by packaging essential software components and easy-to-use examples in one comprehensive package. Everything a customer needs to quickly and efficiently develop new applications using an ARM® Cortex®-M-based MCU from TI is included in the SDK, from the drivers and communication stacks to an OS kernel. The SDK is well integrated into the development environment, leveraging the graphical configuration capabilities of SysConfig, a unified configuration software tool, to speed initial development while making advanced debug capabilities easily accessible.

This cohesive experience across SimpleLink SDKs allows developers to spend their time creating applications that differentiate products from competitors. TI's SimpleLink portfolio is a fully tested and integrated, out-of-the-box robust solution. And through 100 percent code portability, the common foundation of the SDK maximizes reuse. TI's ongoing commitment to the platform guarantees compatible APIs. In addition, the common components of the SDK are available under industry-standard, open-source licensing models that enable customer use directly in applications.

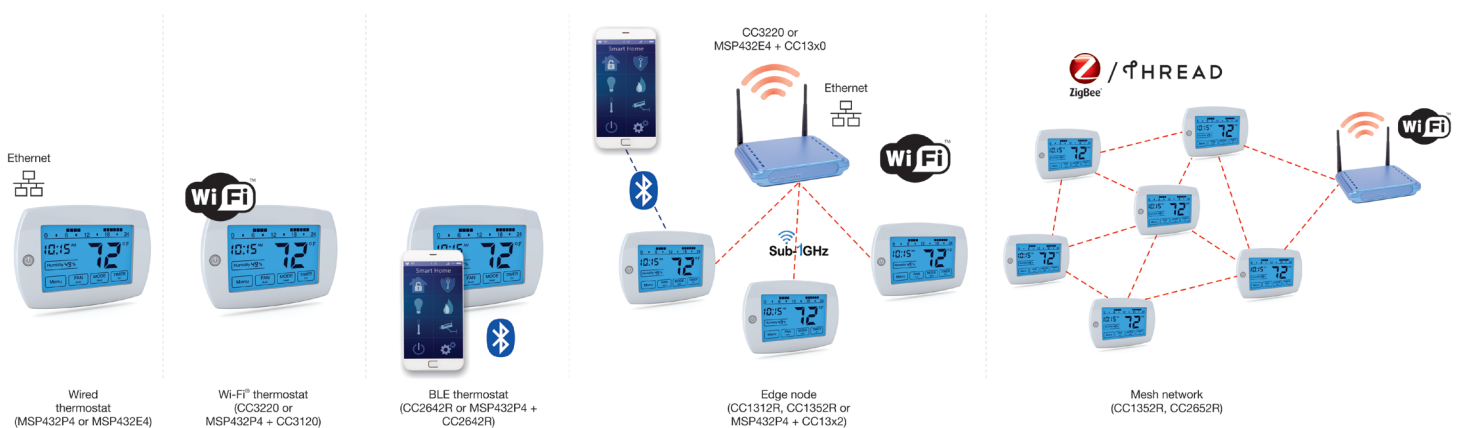
## How the SimpleLink SDK improves the development cycle

Designing new products and applications is not as easy as it may seem. Developing new products from the ground up can require a large investment in tools, software and time to learn the new design environment. How easy are the new tools to use? Are they well documented? Does training exist to help designers ramp-up faster? What kind of APIs does the software use? Are they industry-standard or proprietary and changing? How extensive is the underlying silicon portfolio that is supported? How will the software and tools be supported? When these ramp-up challenges are considered, along with the need to be nimble and able to adapt quickly to ever-changing customer needs and wants, developers can quickly become overwhelmed.

TI's SimpleLink MCU SDK enables rapid development and simplified porting across the industry's broadest portfolio of connected MCUs, providing flexibility within the portfolio as customer needs and connectivity standards requirements evolve.

The SDK supports all devices in the SimpleLink MCU portfolio, including:

- **MSP432™ MCU:** Scalable ARM Cortex-M4F-based MCUs with up to 2 MB of integrated Flash
  - **MSP432P4 MCU:** Ultra-low power and high-precision analog-to-digital converter (ADC)
  - **MSP432E4 MCU:** High-performance with integrated Ethernet MAC and PHY, USB and CAN
- **CC32x0 devices:** ARM Cortex-M4-based MCUs with integrated Wi-Fi® and advanced security features.
- **CC2640R2F and CC1310/CC1350 wireless MCUs:** Low-power wireless ARM Cortex-M3-based MCUs for high-performance RF applications. These devices support a range of wireless standards, including *Bluetooth*® Low Energy and Sub-1 GHz-based standards such as Wireless M-Bus, SigFox and 6LoWPAN, and dual-band capability.
- **CC2642R/CC2652R and CC1312R/CC1352R/CC1352P wireless MCUs:** Low-power wireless ARM Cortex-M4F-based MCUs for high-performance RF applications. These devices support a range of wireless standards, including Zigbee®, Thread, Bluetooth Low Energy, Sub-1 GHz, and multi-standard/dual-band capability.



100% application code portability

Figure 1. Application scalability examples

The SimpleLink SDK maximizes developers' return on investment. As displayed in **Figure 1** below, the SDK allows designers to develop an application on one SimpleLink MCU, such as the [MSP432P4111](#) device, and then reuse the application on a different SimpleLink MCU, like the CC3220 wireless MCU, to add new functionality without starting from scratch. A comprehensive suite of TI Drivers combined with a POSIX-compliant API and integrated RTOS support streamlines integration and ensures compatibility with numerous third-party software components.

In a typical development scenario, a customer might initially introduce an ultra-low-power thermostat using the SimpleLink MSP432P4 MCU or an Ethernet-enabled thermostat using the MSP432E4 MCU. As their business grows and customer demands change, the customer sees the need for a Wi-Fi-enabled thermostat to allow end users to leverage Wi-Fi connectivity for home automation. They can then design a new application based on the SimpleLink CC3220 wireless MCU or host MSP432 MCU and CC3120 wireless network processor. All thermostat control software, user interfaces and applications that were developed for the MSP432 MCU-based thermostat are 100 percent reusable for the new Wi-Fi-based thermostat. This allows the customer to easily add Wi-Fi to their existing product without having to start over.

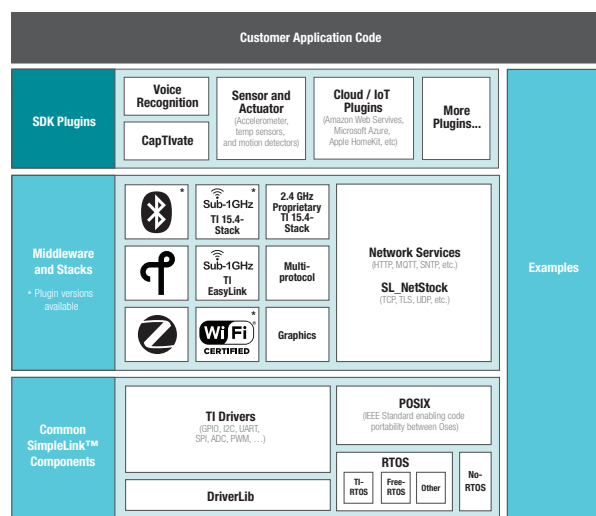
Down the road, if the customer releases a thermostat for the home that requires Bluetooth Low Energy for phone pairing or connection to an existing Thread/Zigbee network, the software is again portable to the multi-standard CC2652R wireless MCU. Even when moving to a gateway sensor network in an industrial setting, the same software developed for the standalone thermostat can be used in the Sub-1 GHz sensor network application using the CC1352R device which also incorporates dual-band functionality for a concurrent Bluetooth Low Energy connection. If

more range is required, the software can also scale to the CC1352P device that has an ultra-low power integrated PA.

TI recognizes that customers need software they can count on and that they know will be usable now and in the future. TI's SimpleLink SDK will regularly release software updates so designers have a predictable upgrade path with upward API code compatibility. All SDK components are fully integrated and tested together to provide a robust foundation for application development. And the SimpleLink SDK is developed, maintained and directly supported by TI through the 24/7 [E2E™](#) [community](#) as well as hands-on [SimpleLink Academy](#) training to make sure you are supported along every step of your design.

## SimpleLink MCU SDK: The breakdown

TI's [SimpleLink MCU SDK](#) (**Figure 2**) is a complete, out-of-the-box solution for all your development needs. The SDK allows you to spend your time building differentiated applications without worrying about creating pieces like communication stacks or developing drivers from the ground up.



**Figure 2.** SimpleLink SDK components

## TI drivers

The TI drivers API exposes the functionality of the hardware-specific drivers in the same way across all TI SimpleLink devices, giving developers portable, feature-rich access to a variety of peripherals. TI drivers are open source [Berkeley Software Distribution license (BSD)] and built on the hardware abstraction layer, offering full access to the device's complete capability. For example, though the hardware implementation of the UART may be different across each devices, the TI drivers API used to access its common functionality is the same.

This device-agnostic approach provides easy portability of the application code across SimpleLink devices now and into the future. Other peripherals exposed through these intuitive and consistent TI driver APIs include analog-to-digital converter (ADC), SPI, PWM and GPIO, among many others. With full access to the hardware abstraction layer through the TI drivers, developers can further optimize their applications by leveraging the Hardware Abstraction Layer (HAL) or directly accessing the microcontroller with register-level programming. TI drivers allow customers to make the application exactly what it needs to be.

## OS/kernel

The kernel provides real-time, multitasking services such as timing and scheduling of tasks. All TI SimpleLink SDKs come with the TI-RTOS kernel pre-installed and are POSIX-compliant. TI-RTOS is a robust solution you can trust, already deployed in thousands of applications across various TI embedded solutions. The kernel is open source (BSD license) and was developed in lock-step with TI's silicon portfolio to enable very low latency in an efficient code footprint. Developers can optimize applications for power consumption, performance and code size to meet their needs. SimpleLink SDKs also support alternative RTOS kernels, such as **FreeRTOS**. The RTOS kernel runs the hardware abstraction layer as well as a suite of functional drivers for all on-chip peripherals.

The **POSIX** layer abstracts the RTOS kernel functionality used by applications. POSIX is an IEEE industry standard for compatibility between operating systems. Requiring less than 2 KB of code in typical applications, the POSIX layer allows examples and user applications to be easily re-used and ported to a different kernel. Using this layer is optional but allows customers to use whatever OS they are currently familiar with or want to move to in the future. POSIX-compatibility also allows TI third-party partners to interface with SimpleLink SDK devices to add support for their kernel. This provides complete freedom to design with any OS, including FreeRTOS.

## Middleware

TI's middleware adds functionality on top of drivers. Communication stacks and graphics libraries are just two examples of middleware. TI has completed all testing and integration of this middleware for SimpleLink devices, making it fast and easy to integrate new technologies like Wi-Fi or Bluetooth Low Energy into an application. Some examples of middleware included for each of the SimpleLink SDK devices are shown in **Figure 3** on the following page.

## SDK plug-ins

The SimpleLink SDK was built to provide a consistent platform for developers to build their applications. This consistent platform makes it easy for TI, third parties or even community members to develop plug-ins that depend on the TI SimpleLink SDK to support external components. Initially, the SimpleLink MSP432 SDK will be complemented by the Bluetooth Low Energy and Wi-Fi plug-ins for connectivity. TI's plug-in offerings will continue to expand to include sensors, actuators, displays and more, enabling developers to build complex systems quickly with this modular software approach.

Device	CC3220	CC2642R/ CC2640R2F	CC1352P/ CC1352R/ CC1350	CC1312R/ CC1310	CC2652R/ CC2652P	MSP432P4	MSP432E4
Included Middleware Stacks and Libraries	<ul style="list-style-type: none"> <li>• Wi-Fi</li> </ul>	<ul style="list-style-type: none"> <li>• Bluetooth Low Energy</li> </ul>	<ul style="list-style-type: none"> <li>• EasyLink Sub-1 GHz RF</li> <li>• TI 15.4-Stack Sub-1 GHz RF</li> <li>• Bluetooth Low Energy</li> </ul>	<ul style="list-style-type: none"> <li>• EasyLink Sub-1 GHz RF</li> <li>• TI 15.4-Stack Sub-1 GHz RF</li> </ul>	<ul style="list-style-type: none"> <li>• Bluetooth Low Energy</li> <li>• Thread</li> <li>• Zigbee</li> <li>• Proprietary TI 15.4-Stack 2.4 GHz RF</li> </ul>	<ul style="list-style-type: none"> <li>• IQ math</li> <li>• Graphics library</li> </ul>	<ul style="list-style-type: none"> <li>• Network stack</li> <li>• USB</li> <li>• CAN</li> </ul>

**Figure 3.** Middleware to support each unique SimpleLink device

## Hardware abstraction layer (HAL)

The hardware abstraction layer (HAL) consists of C functions that abstract writes to hardware registers. The TI drivers and OS kernel are built using the HAL to access hardware features. The HAL, built on top of device-specific header files, follows the ARM CMSIS standard, simplifying access to device modules beyond the register level. With the DriverLib HAL access, developers can peel back the layers of the TI drivers for greater control of their applications or to enhance the software for peripheral and device-specific optimization.

## Examples and training

The SDK provides a wide range of free examples. Using these examples, customers can quickly and easily start writing applications straight out of the box. Each example comes with its own documentation and project files, giving you everything you need to get started. Examples are provided using the supported RTOS kernels. For certain SDKs, examples that do not use an RTOS are also provided.

Freely accessible to all customers, the SimpleLink Academy delivers dozens of exercises and workshops to help get you designing applications quickly. SimpleLink Academy curated trainings support developers of all experience levels to help them get the most out of these devices.

## Conclusion

Developers today face a myriad of challenges—rapidly expanding functionality and connectivity requirements coupled with continued pressure on power, code size and time-to-market. A robust and intuitive software foundation is necessary to expedite initial development. This foundation should support a portfolio of devices that can enable a range of different system configurations while leveraging the initial investment. By enabling 100 percent code portability and a common foundation, the SimpleLink MCU SDKs maximize reuse and portability across devices today and into roadmap devices of the future. This ultimately saves time and money, allowing designers to concentrate on developing differentiated applications. The common software platform also allows designers to easily expand and add functionality that fits new markets.

TI's SimpleLink SDK plug-ins further extend the functionality of products with external components and software, including connectivity stacks, sensors, actuators and more. These plug-ins are built on top of the TI drivers included in the SDK and are available directly from TI as well as third-party developers.

To stay up-to-date on the latest SimpleLink SDK devices, plug-ins, etc. be sure to visit

[www.ti.com/simplelink](http://www.ti.com/simplelink).

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

The platform bar, E2E, MSP432 and SimpleLink are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.



## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale ([www.ti.com/legal/termsofsale.html](http://www.ti.com/legal/termsofsale.html)) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2019, Texas Instruments Incorporated