How the robust, compatible and consistent expansion of SimpleLink™ software leads to accelerated development

Henry Wiechman
SimpleLink MCU software product management
Texas Instruments

The complexity of microcontrollers (MCUs) is increasing exponentially, providing a myriad of challenges for software developers. These challenges include rapidly expanding functionality and connectivity requirements, coupled with continued pressure on power, code size and time to market.

Much of the foundational software used in embedded applications is common including drivers, operating systems, middleware and stacks. Many silicon manufacturers are starting to provide common, foundational software for their devices, including a number of examples that illustrate optimal usage for their devices. A robust and intuitive software foundation can reduce development time, enabling developers to focus on their application code and thus accelerating the ramp to volume production and revenue.

But how much does the silicon manufacturer know about producing high-quality software? Can the software they provide be easily integrated into an end application? Is the code robust? Will it be supported through the life cycle of the end product?

This white paper describes an approach that we've used at Texas Instruments (TI) to create and support SimpleLink™ MCU software development kits (SDKs). TI releases SimpleLink SDKs quarterly, expanding the functionality with new application examples, driver updates and operating system (OS) kernel features, as well as fixing any issues from previous releases. TI preserves backward compatibility as much as possible and documents any breaks when necessary.

SimpleLink SDK releases support multiple integrated development environments (IDEs), including TI's Code Composer Studio™ IDE; IAR Systems Embedded Workbench for Arm™ (EWARM); GNU Compiler Collection; multiple real-time OS (RTOS) kernels including TI-RTOS and FreeRTOS; and a variety of desktop environments (Windows, Mac and Linux®). All SimpleLink SDK releases include documentation and online training through TI's SimpleLink Academy to explain how to integrate and use the software in creating differentiated applications.

Each SimpleLink SDK release follows a four-stage software development process, as outlined in **Figure 1**. To advance from one phase to another, a team that includes marketing and systems engineers, the project manager, and an independent software quality assurance (SQA) engineer who does not report to the SDK development team reviews and approves project artifacts including requirement documents, schedules, resource plans, risk mitigation plans, and documentation and training plans.

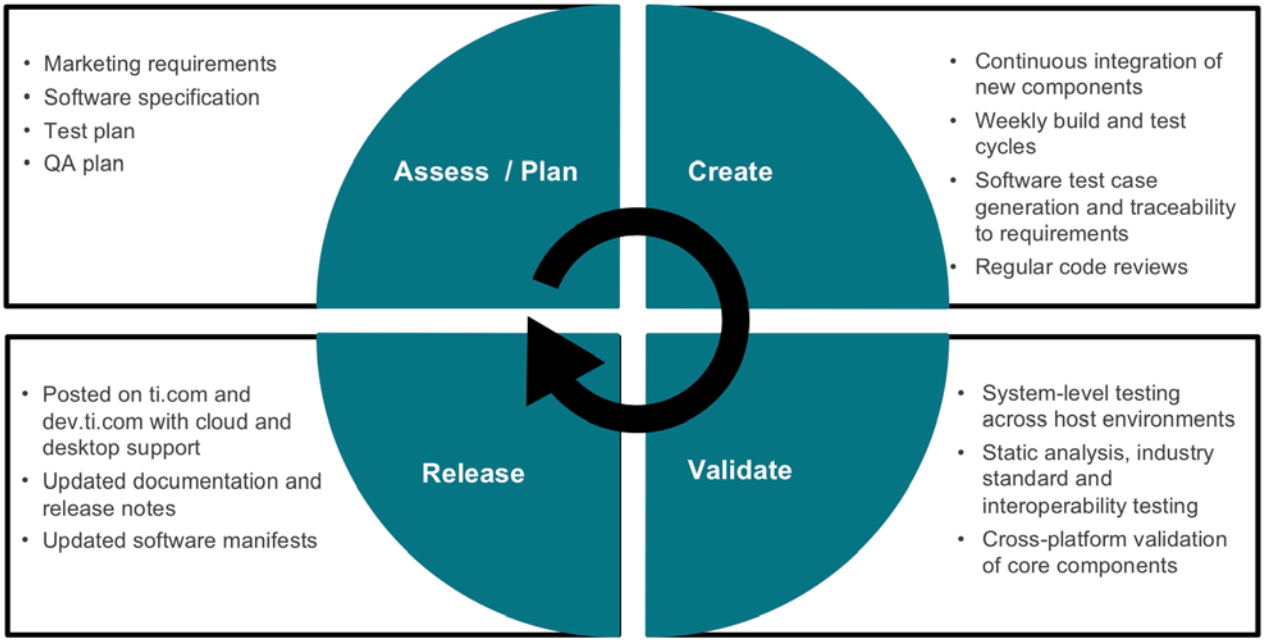

Figure 1. *The SimpleLink SDK quarterly release cycle.*

Upfront planning

During the initial assess/planning phase, the team scopes the project and creates high-level marketing requirements based on identified customer development needs. The marketing requirements are broken down, or decomposed, into low-level functional requirements. This decomposition is critical in order to understand and plan for the engineering effort. The team establishes test and SQA strategies for the release, along with training and documentation needs, and carefully articulates dependencies and risks for all deliverables. A traceability matrix between high- and low-level requirements helps ensure that the team meets all of the objectives.

At the end of the planning phase, the team reviews the project artifacts. If in satisfactory condition, they establish a project execution baseline and the project continues to the create phase.

Continuous integration of new components

The create phase includes an architecture of new features and the generation of code. SimpleLink SDK teams use an agile software development methodology, with continuous integration including daily/weekly integration test runs to identify issues as early as possible. There is a heavy reliance on an automated build and test environment for efficient execution, as well as peer code reviews.

A fresh set of eyes can often identify whether any sections of code need further optimization to improve performance, minimize code size, reduce power consumption or improve reliability. The development methodology also includes running static code analysis and the proper dispositioning of all identified issues.

Along with code construction, the team creates new test cases and automates them as much as possible. Test automation helps ensure that there are no regressions from one release to the next in terms of existing functionality. Linking all test cases to the functional requirements ensures that they cover all functionality.

At the end of the create phase, the team reviews the current project artifacts (code completion, unit test results, static analysis results, test plans and the test case traceability matrix). Again, if satisfactory, they update the project baseline and the project proceeds to the validate phase.

Comprehensive system-level testing

A release candidate SDK is generated at the completion of the create phase. This release candidate will now be subjected to extensive system-level testing that can take several days to complete across all host OS and IDE variants. The team logs the test results and records any issues in a defect tracking system. Weekly project/change control board meetings include the dispositioning of defects.

Interoperability testing is extremely important for connected devices, and TI has invested significantly in test labs for wireless protocols. For example, we test our Wi-Fi® stacks with over 200 different routers from over 50 vendors. We test our *Bluetooth*® low energy stacks with over 30 phones and operating systems and our 15.4 stack over Sub 1-GHz with more than 150 nodes, as shown in **Figure 2**.

We perform precertification testing against regulatory standards in-house, followed by full certification by the respective regulatory agency or certification body. We also conduct a variety of performance characterization tests to assess power consumption, throughput and related parameters. After ensuring cross-platform validation for all common components, the QA and development managers must approve a validation review before release.

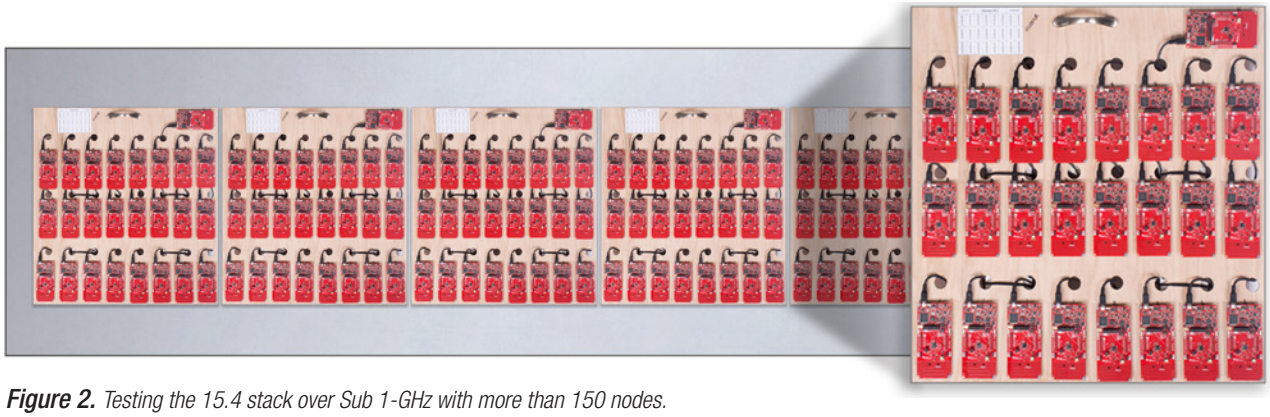

Figure 2. *Testing the 15.4 stack over Sub 1-GHz with more than 150 nodes.*

Test Category	Description
Build Test	All projects compile without any “undispositioned” warning/error
Functional tests	Tests all functionality under normal conditions Tests functionality under corner conditions (interference, fault injection, different sequence of exec etc.)
Platform specific tests	Tests that are specific to a platform (special RF front end, LCD etc..)
Stress tests	Ability to handle increased load for shorter periods of time (back to back packets, repeated join-disconnect etc...)
Reliability tests	Verify operational stability over long hours under normal load conditions
Interoperability tests	Interoperability with other platforms/vendor solutions wherever applicable
Interoperability across SDKs	SDK Compatibility: Earlier released example should be compiled using newer libraries/drivers Communication Interoperability: Interoperability of current SDK with previously released SDK for features that were common between the two releases
Compliance tests (regional compliance/ standard compliance)	Verified in house by testing for timing requirements, frame formats etc. Verified through test labs
Performance tests	Performance characterization (power consumption, throughput …)
User experience tests	Verify overall usability and consistency of the experience

Table 1. *System test categories for the SimpleLink SDK*

Quarterly online releases with complete documentation and training

The final phase includes posting the release online. SimpleLink SDKs can be downloaded directly from www.ti.com/simplelink or evaluated through the Code Composer Studio IDE cloud at www.dev.ti.com. Marketing, systems, applications, software development and QA must all sign off on any release. In order to receive approval, each release must include updated documentation and release notes, as well as revised software manifests documenting the licenses used for all included components. You can view the documentation for each release online before downloading and find out what's new in the release notes. Most choose to update to the newest release of the SDK until they have frozen their code for final validation.

We test SimpleLink Academy modules with each release to ensure that all labs and examples continue to work as expected. We often create modules to highlight new SDK functionality. You can sign up for email notifications when new releases are posted by clicking the Alert Me button from the download page, as shown in **Figure 4**.

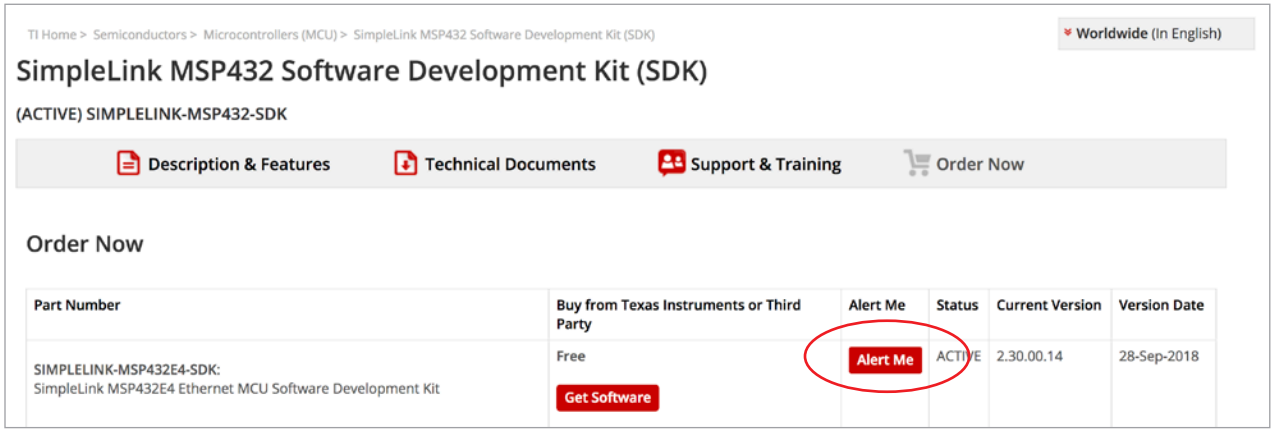

Figure 4. *Alert Me button to register for quarterly SDK release notifications.*

All SimpleLink SDKs are supported 24/7 by TI engineers via the TI E2E™ Community forum; feedback from our community helps drive requirements for future releases .

With a comprehensive development process and a commitment to quarterly releases, SimpleLink SDKs offer a firm software foundation for our portfolio of connected MCUs. Featuring 100% application code portability across the SimpleLink MCU portfolio of 10 different wired and wireless communication technologies, it's possible to easily enhance initial designs and create complementary products with full confidence in the SimpleLink SDK software foundation and a consistent release schedule. Learn more at www.ti.com/simplelinksdk.

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 is a trademarks of Texas Instruments. All other trademarks are the property of their respective owners.

© 2018 Texas Instruments Incorporated

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) or other applicable terms available either on 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