Silicon Labs OpenThread SDK 1.1.1.0 GA

Gecko SDK Suite 3.1

January 27, 2021

Thread is a secure, reliable, scalable, and upgradeable wireless IPv6 mesh networking protocol. It provides low-cost bridging to other IP networks while optimized for low-power / battery-backed operation. The Thread stack is designed specifically for Connected Home applications where IP-based networking is desired and a variety of application layers may be required.

OpenThread released by Google is an open-source implementation of Thread. Google has released OpenThread in order to accelerate the development of products for the connected home and commercial buildings. With a narrow platform abstraction layer and a small memory footprint, OpenThread is highly portable. It supports system-on-chip (SoC), network co-processor (NCP), and radio co-processor (RCP) designs.

Silicon Labs has developed an OpenThread-based SDK tailored to work with Silicon Labs hardware. The Silicon Labs OpenThread SDK is a fully tested snapshot of the GitHub source. It supports a broader range of hardware than does the GitHub version, and includes documentation and example applications not available on GitHub.

These release notes cover SDK version(s):

1.1.1.0 released on January 27, 2021
1.1.0.0 released on December 9, 2020

Thread

KEY FEATURES

- Alpha support for IEEE 802.15.4 managed coexistence with Wi-Fi
- Antenna Diversity
- Border router sample application
- FreeRTOS integration with sample applications
- MGM210L module support

Compatibility and Use Notices

If you are new to the Silicon Labs OpenThread SDK, see Using This Release.

Compatible Compilers:

Note: The supported compiler will be upgraded to ARM GCC-10-2020-q4-update in the next major release.

GCC (The GNU Compiler Collection) version 7.2.x, provided with Simplicity Studio.

Contents

1 New Items

1.1 New Features from OpenThread GitHub Repo

Added in release 1.1.0.0

The Silicon Labs OpenThread SDK includes all changes from the OpenThread GitHub repo (https://github.com/openthread/openthread) up to and including commit **5c2ad91cf**. Any features introduced between commits **f411a412bee** and **5c2ad91cf** can be considered new items with the Silicon Labs OpenThread 1.1 SDK. A copy of the OpenThread repo can be found in the following Simplicity Studio 5 location:

Simplicity Studio\developer\sdks\gecko_sdk_suite\<version>\util\third_party\openthread

1.2 New Features from OpenThread Border Router GitHub Repo

Added in release 1.1.0.0

The Silicon Labs OpenThread SDK includes all changes from the OpenThread border router GitHub repo (https://github.com/openthread/ot-br-posix) up to and including commit **a37e299ff**. Any features introduced between commits **f93719a08c9** and **a37e299ff** can be considered new items with the Silicon Labs OpenThread 1.1 SDK. A copy of the OpenThread border router repo can be found in the following Simplicity Studio 5 location:

Simplicity Studio\developer\sdks\gecko_sdk_suite\<version>\util\third_party\ot-br-posix

Refer to section 7.3 for information on using the border router.

1.3 New Components

Added in release 1.1.0.0

ot_debug_channel

This component provides a set of APIs for printing to the debug channel. Information printed to the debug channel can be viewed using Simplicity Studio's Network Analyzer.

1.4 New Features

Added in release 1.1.0.0

Antenna Diversity

Antenna Diversity support has been added. Refer to *AN1294: Configuring Antenna Diversity for OpenThread* for details on configuration and use.

Wi-Fi Coexistence

Wi-Fi Coexistence support has been added. Refer to *UG103.17: Wi-Fi Coexistence Fundamentals* and *AN1017: Zigbee and Thread Coexistence with Wi-Fi* for details on configuration and use.

OpenThread Border Router

The OpenThread border router has been integrated into the Silicon Labs OpenThread SDK. Refer to AN1256: Using the Silicon Labs RCP with the OpenThread Border Router for details on configuration and use

Thread 1.2

Alpha support for Thread 1.2 has been added. Refer to *AN1295: Developing with Thread 1.2* for details on configuration and use.

FreeRTOS integration with sample applications

FreeRTOS support has been added to all OpenThread sample applications. Refer to *AN1264: Using OpenThread with Free RTOS* for configuration and use.

2 Improvements

2.1 Component Changes

Changed in release 1.1.0.0

ot_cli

The ot_cli component has been enhanced to provide CLI support for Wi-Fi coexistence and antenna diversity.

ot_platform_abstraction

The ot_platform_abstraction component has been enhanced to support CMSIS-RTOS2 abstraction layer, Wi-Fi coexistence, and antenna diversity.

ot_thirdparty

The ot_thirdparty component has been updated to use mbedTLS 2.24.0.

3 Fixed Issues

Fixed in release 1.1.1.0

ID #	Description
605107	We no longer allow OpenThread samples to be selected in Simplicity Studio with boards where the app would fail to build due to limited flash/RAM.
649169	In the platform abstraction layer we now populate the RSSI, LQI and timestamp values (aka RX info) for the received ACK frames before passing to the OpenThread core. Previously, this info was only populated for non-ACK packets. The problem was that when a SED polled its parent, the ACK frame was passed up to the OpenThread core, but the RX info was incorrect (it was actually the RX info for the last received non-ack packet).
651053	Fixed an issue with the ot-ble-dmp sample app where, if you open a BLE connection and then send a lot of OpenThread traffic, it was possible to run out of OpenThread buffers.
651489	The sleepy-demo-ftd and sleepy-demo-mtd sample apps were only initializing 12 of the 16 bytes of the predefined master key. Now all 16 bytes are initialized. The previous code did not cause a problem with the execution of the sample apps but did cause confusion to readers of the code.
653069	Fixed an issue with the Coexistence Component where "TX Priority Escalation when CCA Fail" was not working due to a header misconfiguration.
653385	Fixed issues with the coexistence get-pta-options and set-pta-options CLI commands. The output displayed for the get-pta-options command missed line spaces and used incorrect format specifiers. The set-pta-options command read the input in decimal format instead of hexadecimal.
653466	Fixed issues with logic in the platform abstraction layer responsible for passing mAckedWithFramePending to the OpenThread core.
654789	Fixed an issue where GPIO initialization was not occurring, preventing antenna diversity from working correctly.
655515	The txpower CLI command now displays the correct dBm value.

Fixed in release 1.1.0.0

ID #	Description
630624	In both the otPlatRadioSetTransmitPower and otPlatRadioGetTransmitPower implementations a variable sTxPowerDbm is used to keep track of the radio power level. But this variable can easily get out of sync because the call RAIL_SetTxPowerDbm silently truncates any power level to the nearest power level available for the radio. Fix is to remove this extra state variable and use RAIL_GetTxPowerDbm (divided by 10) to report the actual, set power level for the radio.
635190	Fixes for the following energy scan issues: • When performing an Energy scan on a node running the ot-cli-ftd app, with large scan duration, the energy scan expectedly fails. However, every subsequent energy scan returns a channel busy message. • Similar to the previous case, when performing an Energy scan on a node running the ot-ble-dmp app, with large scan durations, the energy scan expectedly fails. However, in this case, the node becomes unresponsive on the network and every subsequent energy scan returns a channel busy message.
643623	Removed an extra call to sl_uartdrv_init_instances that was causing excess DMA channels to be allocated.
642953	Fixed an issue where radio.c can assert with very heavy traffic and very heavy MCU load.
629088	otPlatEntropyGet should directly call mbedtls_hardware_poll and avoid dropping 3 bytes of entropy by using all 4 bytes of gathered entropy.
642968	In otPlatGetResetReason, the return value previously implied that the last reset was caused by an assert, when actually it was a brownout or an EM4 reset. It now reports OT_PLAT_RESET_REASON_OTHER.
642965	Ensure that OPENTHREAD_CONFIG_MAC_SOFTWARE_CSMA_BACKOFF_ENABLE and OPENTHREAD_CONFIG_MAC_SOFTWARE_ENERGY_SCAN_ENABLE are defined to 0 to avoid CMSA being performed twice.

4 Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on https://www.si-labs.com/products/software.

ID #	Description	Workaround
482915 495241	A known limitation with the UART driver can cause characters to be lost on CLI input or output. This can happen during particularly long critical sections that may disable interrupts, so it can be alleviated by repeating the CLI or waiting long enough for state changes.	No known workaround
653058	Coexistence Component: "TX fails to abort on Grant abort when set to do so."	No known workaround
653068	**Coexistence Component: "Sync MAC to Grant" is not working. Grant doesn't abort even when the tx abort grant abort is enabled in component and pta master is configured.**	**No known workaround**
653070	**Coexistence Component: "PWM on Request" not currently working. Request, Grant, and Priority are held high.**	**No known workaround**
653076	**Coexistence Component: "Shared Mode Request back off not working" is currently not working. Simulating regular request and not asserting priority.**	**No known workaround**

5 Deprecated Items

None

6 Removed Items

None

7 Using This Release

This release contains the following

- Silicon Labs OpenThread stack
- Silicon Labs OpenThread sample applications
- Silicon Labs OpenThread border router

For more information about the OpenThread SDK see QSG170: Silicon Labs OpenThread QuickStart Guide. If you are new to Thread see UG103.11: Thread Fundamentals.

7.1 Installation and Use

A registered account at Silicon Labs is required in order to download the Silicon Labs OpenThread SDK. You can register at https://silabs.force.com/apex/SL_CommunitiesSelfReg?form=short.

Stack installation instruction are covered in the Simplicity Studio 5 online User's Guide.

Use the OpenThread SDK v1.x with the Silicon Labs Simplicity Studio 5 development platform only. The SDK is not compatible with Simplicity Studio 4.

Documentation specific to the SDK version is installed with the SDK. API references and other information about this release are available on https://docs.silabs.com/openthread/1.1/.

7.2 Enabling Thread 1.2

Thread 1.2 support is included in the Silicon Labs OpenThread SDK but is not enabled by default. It can be enabled by following the instructions in *AN1295: Developing with Thread 1.2*. Thread 1.2 support is alpha quality and is not fully tested as the specification is yet to be certified and some features are unstable. We have included a subset of features: Domain Unique Address (DUA), Multicast Listener Registration (MLR) and Backbone Router support. Low power features such as Enhanced Frame Pending are currently disabled by default, and CSL support is not included.

7.3 Using the Border Router

For ease of use, Silicon Labs recommends the use of a Docker container for your OpenThread border router. Please refer to AN1256: Using the Silicon Labs RCP with the OpenThread Border Router for details on how to set up the correct version of OpenThread border router Docker container. It is available at https://hub.docker.com/r/siliconlabsinc/openthread-border-router.

The Silicon Labs OpenThread RCP is meant for use with the OpenThread border router built using the ot-br-posix repository (git commit a37e299ff). Our copy of the OpenThread stack (git commit 5c2ad91cf) should be symbolically linked under "third_party\openthread\repo" in the ot-br-posix repository.

For your convenience we have included copies of these repositories in the GSDK, located here:

Simplicity Studio\developer\sdks\gecko_sdk_suite\<version>\util\third_party\ot-br-posix

Simplicity Studio\developer\sdks\gecko_sdk_suite\<version>\util\third_party\openthread

If you are manually installing a border router, using the copies provided above is recommended. Refer to AN1256: Using the Silicon Labs RCP with the OpenThread Border Router for more details.

Although updating the border router environment to a later GitHub version is supported on the OpenThread website, it may make the border router incompatible with the OpenThread RCP stack in the SDK.

7.4 NCP/RCP Support

The OpenThread NCP support is included with OpenThread SDK but any use of this support should be considered experimental. The OpenThread RCP is fully implemented and supported.

7.5 Support

Development Kit customers are eligible for training and technical support. Use the Silicon Laboratories Thread web page to obtain information about all Silicon Labs OpenThread products and services, and to sign up for product support.

You can contact Silicon Laboratories support at http://www.silabs.com/support.

8 Legal

8.1 Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required, or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

8.2 Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, ClockBuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.