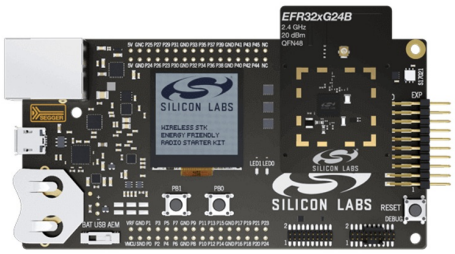


SILICON LABS 3.6.3.0 GA Proprietary Flex SDK User Guide

Proprietary Flex SDK 3.6.3.0 GA
Gecko SDK Suite 4.3
March 13, 2024

Contents

3.6.3.0 GA Proprietary Flex SDK

The Proprietary Flex SDK is a complete software development suite for proprietary wireless applications.
Perits namesake, Flex offers two implementation options.
The first uses Silicon Labs RAIL (Radio Abstraction Interface Layer), an intuitive and easily-customizable radio interface layer designed to support both proprietary and standards based wireless protocols.
The second uses Silicon Labs Connect, an IEEE 802.15.4-based networking stack designed for customizable broad-based proprietary wireless networking solutions that require low power consumption and operates in either the sub-GHz
or 2.4 GHz frequency bands.
The solution is targeted towards simple network topologies.
The Flex SDK is supplied with extensive documentation and sample applications. All examples are provided in source code within the Flex SDK sample applications.
These release notes cover SDK version(s):
3.6.3.0 GA released March 13, 2024

3.6.2.0 GA released October 9, 2023
3.6.1.0 GA released July 26, 2023
3.6.0.0 GA released June 7, 2023

RAIL APPS AND LIBRARY KEY FEATURES

- FG23 Direct Mode settings in Radio Configurator
- WM-BUS T+C PHY support
- FGM230S WM-BUS PHYs and Application support
- RAIL NCP Sample Applications
- PSM support for DSSS-OQPSK Long Range PHYs

CONNECT APPS AND STACK KEY FEATURES

- Connect NCP support
- FGM230S Connect support

Compatibility and Use Notices

For information about security updates and notices, see the Security chapter of the Gecko Platform Release notes installed with this SDK or on the TECH DOCS tab on **https://www.silabs.com/developers/flex-sdk-connect-networking-stack**. Silicon Labs also strongly recommends that you subscribe to Security Advisories for up-to-date information. For instructions, or if you are new to the Silicon Labs Flex SDK, see Using This Release.
Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 9.20.4

- Using wine to build with the IarBuild.exe command line utility or IAR Embedded Workbench GUI on macOS or Linux could result in incorrect files being used due to collisions in wine's hashing algorithm for generating short file names.
- Customers on macOS or Linux are advised not to build with IAR outside of Simplicity Studio. Customers who do should carefully verify that the correct files are being used.

GCC (The GNU Compiler Collection) version 10.3-2021.10, provided with Simplicity Studio.

Connect Applications

1.1 New Items

Added in release 3.6.2.0

- NCP – Host solution
- Added a new API to read the NCP GSDK, stack and bootloader versions.
- Added a new API emberGetKeyId to read the PSA Crypto KeyId currently used in the NCP.

- Added a new API emberSetNcpSecurityKeyPersistent that stores the key in NVM. emberSetPsaSecurityKey could only store the key in RAM.
- Implemented sl_cpc_secondary_app_version to return the GSDK version.

Added in release 3.6.0.0

- XG27 Support
- NCP – Host solution
- Host: Connect Host Sink CLI Application
- NCP: Connect – NCP Application
- Security and OTA Bootloader support

1.2 Improvements
None
1.3 Fixed Issues

1.4 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 the TECH DOCS tab on **https://www.silabs.com/developers/flex-sdk-connect-networking-stack**.

ID #	Description	Workaround
652925	EFR32XG21 is not supported for “Flex (Connect) – SoC Light Example DMP” and “Flex (Connect) – So C Switch Example”	
1076409	OTA Bootloader is not working on Series2	
1139850	DMP instabilities with XG27	

1.5 Deprecated Items
None
1.6 Removed Items
None

Connect Stack

2.1 New Items
None
2.2 Improvements
None
2.3 Fixed Issues
Fixed in release 3.6.2.0

ID #	Description
1173046	Fixed an interoperability issue between xG28 and xG1x. Forced RX to TX and TX to RX transition duration to greater than a predefined value.
1187053	Fixed version macros.

2.4 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 the TECH DOCS tab on **https://www.silabs.com/developers/gecko-software-development-kit**.

ID #	Description	Workaround
	When running the RAIL Multiprotocol Library (used for example when running DMP Connect+BLE), IR Calibration is not performed because of a known is sue in the RAIL Multiprotocol Library. As result, the re is an RX sensitivity loss in the order of 3 or 4 dB m.	
501561	In the Legacy HAL component, the PA configuratio n is hard- coded regardless of the user or board set tings.	Until this is changed to properly pull from the configuration header, the file ember-phy.c in the user's project will need to be modified by hand t o reflect the desired PA mode, voltage, and ra mp time.
711804	Connecting multiple devices simultaneously may fa il with a timeout error.	

2.5 Deprecated Items

None

2.6 Removed Items

None

RAIL Applications

3.1 New Items

Added in release 3.6.0.0

- XG27 Support
- Host – NCP support
- Host: RAIL Host Simple CPC to Serial
- NCP
- RAIL – NCP Simple TRX with CPC Support (VCOM)
- RAIL – NCP Simple TRX with CPC Support (SPI)

3.2 Improvements

Changed in release 3.6.0.0

- Amazon Sidewalk PHYs are added to the RAIL – SoC Range Test applications

3.3 Fixed Issues

Fixed in release 3.6.2.0

ID #	Description
1151826	On XG23, XG25 and XG28 Sidewalk profile was greyed out in radio configurator

3.4 Known Issues in the Current Release
None
3.5 Deprecated Items
None
3.6 Removed Items
None

RAIL Library

4.1 New Items
Added in release 3.6.2.0

- Added support for a new assert, which will be thrown if the loaded PHY is not supported by the software-defined modem on EFR32xG25.
- Added the RAIL_TX_REPEAT_OPTION_START_TO_START option to measure the delay between repeated transmits from the start of TX to start of TX instead of the default from end of TX to start of TX.

Added in release 3.6.1.0

- Added new RAIL_GetAutoAckFifo() API and allow NULL for RAIL_WriteAutoAckFifo() or RAIL_IEEE802154_WriteEnhAck() ackData parameter, which gives applications direct access to the AutoAck FIFO to construct Ack packets in pieces.
- Added support for antenna selection through the applicable RAIL_ RxOptions_ t and RAIL_TxOptions_tvalues when using OFDM on the EFR32xG25.
- Added a new “RAIL Utility, SFM Sequencer Image Selection” component to allow selection of modulations supported by EFR32XG25 software modem (SFM). These changes can save considerable flash space by reducing the set of modulations to just those that are needed.
- Added support for Sidewalk PHYs for EFR32xG23 and EFR32xG28 chips.
- Added a new component to switch between Coexistence, Antenna Diversity and FEM Utilities for EFR32xG21 and EFR32xG24 chips supporting 15.4 Fast Channel Switching feature.
  Added in release 3.6.0.0
- Added support for a new RX_DIRECT_SYNCHRONOUS_MODE_DATA RAIL RX data source to capture direct mode data in sync with the configured bit rate for the PHY. This requires a PHY that supports this mode of capture from the Radio Calculator and is only currently supported on the EFR32xG23.
- Added new RAIL_ Enable Cache Synth Cal function to enable the radio sequencer to cache calibration values instead of recalculating them on every RX and TX event. This allows you to lower the minimum transition time for most RAIL_ State Timing_ ttransitions in typical cases.
- Added a new RAIL_RX_OPTION_FAST_RX2RX which will force the radio sequencer to immediately transition to RXSEARCH to get ready to receive the next packet while still processing the previous one. This will minimize the RX to RX state transition time.
  This is only supported on chips that have RAIL_SUPPORTS_FAST_RX2RX set to true.
- Added RAIL_ Packet Time Stamp_t::packet Duration Us field which is currently set only on EFR32xG25 for received OFDM packets.
- Added RAIL support for the MGM240L lighting modules.
- Added the new RAIL_WMBUS_ Config API to allow configuring WMBUS and simultaneous M2O RX of T and C mode packets.

- Added a new API RAIL_ SetTxFifoAlt() which provides a new start offset parameter to specify where the data begins in the TX FIFO.
- Added support for RAIL_IEEE802154_SupportsRxChannelSwitching on the EFR32xG21. This is also still supported at an alpha quality level on the EFR32xG24.
- Added RAIL_IEEE802154_SetRxToEnhAckTx() to allow IEEE 802.15.4 stacks to specify a different rxToTx state transition turnaround time for Enhanced ACKs, which generally need more time to construct and secure. Immediate ACKs will continue to use the rxToTx time specified in RAIL_IEEE802154_Config_t::timings.
- Added Tx packet duration information for EFR32xG22 and newer chips.

4.2 Improvements

Changed in release 3.6.2.0

- Reduced the delay between TX completion and the start of PA ramp down on the EFR32xG24.

Changed in release 3.6.1.0

- Updated the RAIL_PacketTimeStamp_t::packetDurationUs field on the EFR32xG25 to be populated for non-OFDM packets.

Changed in release 3.6.0.0

- Added support for PHY-specific RSSI offsets on the EFR32xG27 and EFR32xG28 platforms.
- Updated Packet Trace on the EFR32xG25 and EFR32xG28 when using the Wi-SUN protocol to have a more informative PHY
  identifier and to support the whole channel number range.
- Fixed the RAIL_PA_CURVES_2P4_LP power curves on the EFR32xG24 to better match characterization data.
- Corrected the sign of the frequency error reported by RAIL_GetRxFreqOffset() when using OFDM on the EFR32xG25 to match how
  this was handled for other modulations (e.g., Freq_error=current_freq-expected_freq).
- Added new RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE to trigger RAIL_EVENT_ZWAVE_BEAM on all beam frames.
- Added RAIL_ZWAVE_GetBeamHomeIdHash() to retrieve the beam frame's HomeIdHash when handling that event and made sure
  that the HomeIdHash byte is now present on PTI for Z-Wave beam frames even when NodeId does not match.
- Fixed an issue on EFR32ZG23 where multiple beam frames were lumped together on PTI as one large beam chain.
- Adjusted channel power restrictions for the 802.15.4 PHYs on new xGM210 modules.
- Added separate curves when the 20 dBm PA is used at 3.3 V and 1.8 V for EFR32xG24.
- Increased EFR32ZG13 and EFR32ZG14 Z-Wave long-range beam detect time to improve FLiRS performance.

4.3 Fixed Issues

Fixed in release 3.6.2.0

ID #	Description
1191666	Fixed an issue where high reference spurs may appear during radio communication on EFR32xG22 and newer platforms.

Fixed in release 3.6.1.0

ID #	Description
707731	Fixed an issue when using RAIL_BLE_SetNextTxRepeat() that would cause an incorrect Protocol Config ID to appear in the packet trace for repeated transmits.
1079816	Fixed a race condition on EFR32xG22 and later during RX channel hopping or duty-cycling where frame detection occurring close to when a hop should happen could leave the radio stuck in reception but not receiving anything, with the only remedy being to idle the radio.
1088439	Fixed an issue which would cause the incorrect antenna to be reported for a received packet on the EFR32xG25 when using OFDM and antenna diversity.
1141539	Fixed an issue to improve CCA behavior for regional certification test suites for EFR32xG25 chips.
1153679	Fixed an issue in “RAIL Utility, Coexistence” component where a GRANT signal pulse less than 100 µs might result in the radio not being properly placed in hold off after GRANT is deasserted.
1156980	Fixed an issue with channel hopping on EFR32xG22 and later where use of RAIL_RX_CHANNEL_HOPPING_OPTION_RSSI_THRESHOLD can prevent the timed RX channel hopping modes including RX duty-cycling from timing out properly.
1167235	Fixed an issue with RAIL_SupportsFastRx2Rx() where it used to return an incorrect value on supported platforms.

Fixed in release 3.6.0.0

ID #	Description
824355	Fixed an issue in IEEE802.15.4 MAC address filtering when receiving small OFDM packets.
832743	Clarified use of RAIL_SetNextTxRepeat() must be prior to initiating a transmit operation via API call and fixed an issue where it did not properly return an error when called while a transmit operation was in progress.
1055824	Fixed an issue with low-side synth injection (negative IF) on proprietary 2.4GHz PHYs when using EFR32xG22 and newer chips. This fix requires regenerating the PHY with the latest version of the Radio Configurator to work.
1058480	Fixed an RX FIFO corruption on EFR32xG25 that occurred when receiving/sending certain OFDM packets using FIFO mode.
1082274	Fixed an issue on the EFR32xG22, EFR32xG23, EFR32xG24, and EFR32xG25 chips that could cause the chip to lock up if the application attempts to re-enter EM2 within ~10 µs after wake-up and hits a <0.5 µs timing window. If hit, this lockup requires a power-on reset to restore normal operation to the chip.
1083615	Fixed an issue for certain ramp time and power level combinations on the EFR32xG21 where the PA ramp would stop one power level short of the desired output level.

1090336	Fixed an issue in the “RAIL Utility, Protocol” component where BLE would be required to select a Zigbee PHY.
1090512	Fixed an issue in the “RAIL Utility, PA” component where certain functions would attempt to use the RAIL_TX_POWER_MODE_2P4GIG_HIGHEST macro even though they didn’t support it. This would result in undefined behavior previously, but will now correctly error.
1090728	Fixed a possible RAIL_ASSERT_FAILED_UNEXPECTED_STATE_RX_FIFO issue on EFR32xG12 with RAIL_IEEE802154_G_OPTION_GB868 enabled for a FEC-capable PHY which can happen when aborting a packet at frame detection, for instance by idling the radio.
1092769	Fixed an issue when using Dynamic Multiprotocol and BLE Coded PHYs where a transmit could underflow depending on what protocol was active when the PHY and sync word were loaded.
1096663	Fixed a compilation error in “RAIL Utility, Coexistence” component when the Coexistence WiFi TX GPIO is enabled.
1096665	Fixed a compilation issue in “RAIL Utility, Coexistence” component when the SL_RAIL_UTIL_COEX_WIFI_TX_PORT is defined.
1103966	Fixed an unexpected Rx packet abort on the EFR32xG25 when using the Wi-SUN OFDM option4 MCS0 PHY.
1104033	Fixed an issue in the RAIL_ZWAVE_Receive Beam function so that it idles the radio regardless of whether a beam is detected on the EFR32ZG23.
1104441	Fixed an issue with the “RAIL Utility, Coexistence” component counters for Zigbee that could prevent them from ticking as expected depending on how things are linked.
1105134	Fixed an issue when switching between certain PHYs that could cause the first received packet to be reported as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS. This issue could potentially impact EFR32xG22 and newer chips.
1105529	Fixed an issue on EFR32xG22 and later platforms when using a FrameType decoding PHY where a bad frame type packet was mis-reported as RAIL_RX_PACKET_ABORT_ABORTED instead of the proper RAIL_RX_PACKET_ABORT_FORMAT.
1109574	Fixed an issue on EFR32xG22 and newer chips where a radio sequencer assert could cause the application to hang in an ISR rather than report the assert via RAILCb_AssertFailed().
1118063	Fixed issue with recent RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE on EFR32xG13 and EFR32xG14 where the NodeId of the promiscuous beam was not properly recorded for RAIL_ZWAVE_GetBeamNodeId(), causing it to report 0xFF.
1126343	Fixed an issue on EFR32xG24 when using the IEEE 802.15.4 PHY where the radio could become stuck when doing an LBT transmit if a frame is received during the CCA check window.
1134223	Fixed an issue when using “RAIL Utility, Coexistence” component where the request line is left asserted after TX is aborted following a sync detect.
1135418	Fixed incorrect RAIL_RxPacketInfo_t::filterMask on received 802.15.4 Beacon frames, which now reflects which PanId and address the Beacon’s Source PanId and Source Address match, if any. Note that RAIL generally accepts all Beacons so the filterMask can be 0x00.
1138522	Fixed an issue on the EFR32xG25 for SUN FSK PHYs where receiving a packet after calling RAIL_IEEE802154_Init() but before configuring RAIL_IEEE802154_ConfigGOptions could break reception.
1140569	Fixed a rare timing issue on EFR32xG24 where an ACK timeout might cause the next packet to be received as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS.

1150779	Fixed the 15.4 channel configurations on the MGM240PA32 and MGM240PB32 modules to use th e correct frequency for channel 26.

4.4 Known Issues in the Current Release
Issues in bold were added since the previous release.

ID #	Description	Workaround
	Using direct mode (or IQ) functionality on EFR3 2xG23 requires a specifically set radio configuration that is not yet supported by the ra dio configurator. For these requirements, reach out to technical support who could provide that configuration based on your specification	
641705	Infinite receive operations where the frame's fix ed length is set to 0 are not working correctly on the EFR32xG23 series chips.	
732659	On EFR32xG23: · Wi-SUN FSK mode 1a exhibits a PER floor wit h fre- quency offsets around ± 8 to 10 KHz · Wi-SUN FSK mode 1b exhibits a PER floor wit h fre- quency offsets around ± 18 to 20 KHz	

4.5 Deprecated Items
None
4.6 Removed Items
None

Using This Release

This release contains the following

- Radio Abstraction Interface Layer (RAIL) stack library
- Connect Stack Library
- RAIL and Connect Sample Applications
- RAIL and Connect Components and Application Framework

This SDK depends on Gecko Platform. The Gecko Platform code provides functionality that supports protocol plugins and APIs in the form of drivers and other lower layer features that interact directly with Silicon Labs chips and modules. Gecko Platform components include EMLIB, EMDRV, RAIL Library, NVM3, and mbedTLS. Gecko Platform release notes are available through Simplicity Studio's Documentation tab.
For more information about the Flex SDK v3.x see **UG103.13: RAIL Fundamentals** and **UG103.12: Silicon Labs Connect Fundamentals.**
If you are a first time user, see **QSG168: Proprietary Flex SDK v3.x Quick Start Guide** .

5.1 Installation and Use
The Proprietary Flex SDK is provided as part of the Gecko SDK (GSDK), the suite of Silicon Labs SDKs. To quickly get started with the GSDK, install Simplicity Studio 5, which will set up your development environment and walk you through GSDK installation. **Simplicity Studio 5** includes everything needed for IoT product development with Silicon Labs devices, including a resource and project launcher, software configuration tools, full IDE with GNU toolchain, and analysis tools. Installation instructions are provided in the online **Simplicity Studio 5 User's**

Guide.
Alternatively, Gecko SDK may be installed manually by downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/gecko_sdk for more information.
Simplicity Studio installs the GSDK by default in:

• (Windows): C:\Users\<NAME>\SimplicityStudio\SDKs\gecko_sdk
• (MacOS): /Users/<NAME>/SimplicityStudio/SDKs/gecko_sdk
Documentation specific to the SDK version is installed with the SDK. Additional information can often be found in the knowledge base articles (KBAs). API references and other information about this and earlier releases is available on https://docs.silabs.com/.

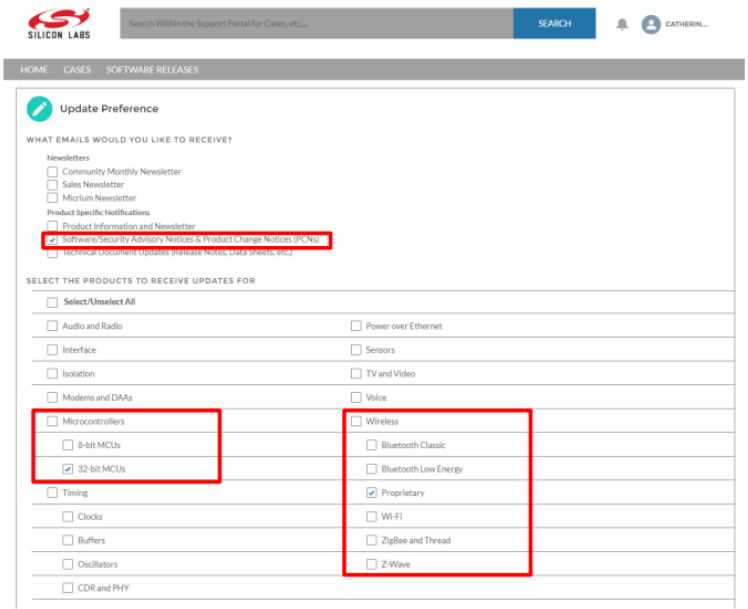
5.2 Security Information

Secure Vault Integration

When deployed to Secure Vault High devices, sensitive keys are protected using the Secure Vault Key Management functionality. The following table shows the protected keys and their storage protection characteristics.

Wrapped Key	Exportable / Non-Exportable	Notes
Thread Master Key	Exportable	Must be exportable to form the TLVs
PSKc	Exportable	Must be exportable to form the TLVs
Key Encryption Key	Exportable	Must be exportable to form the TLVs
MLE Key	Non-Exportable	
Temporary MLE Key	Non-Exportable	
MAC Previous Key	Non-Exportable	
MAC Current Key	Non-Exportable	
MAC Next Key	Non-Exportable	

Wrapped keys that are marked as “Non-Exportable” can be used but cannot be viewed or shared at runtime.
Wrapped keys that are marked as “Exportable” can be used or shared at runtime but remain encrypted while stored in flash.
For more information on Secure Vault Key Management functionality, see AN1271: Secure Key Storage.



5.3 Support

Development Kit customers are eligible for training and technical support. Use the **Silicon Labs Flex web page** to obtain information about all Silicon Labs Thread products and services, and to sign up for product support.
You can contact Silicon Laboratories support at **http://www.silabs.com/support**.

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio **www.silabs.com/IoT**	SW/HW **www.silabs.com/simplicity**	Quality **www.silabs.com/quality**	Support & Community **www.silabs.com/community**

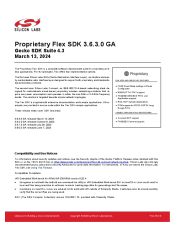
weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA
www.silabs.com

Documents / Resources

	[SILICON LABS 3.6.3.0 GA Proprietary Flex SDK] [pdf] User Guide 3.6.3.0 GA Proprietary Flex SDK, 3.6.3.0 GA, Proprietary Flex SDK, Flex SDK, SDK

References

- Silicon Labs
- Silicon Labs
- About Us - Silicon Labs
- Silicon Labs Community
- Internet of Things (IoT) - Silicon Labs
- Quality - Silicon Labs
- Simplicity Studio - Silicon Labs
- Technical Support - Silicon Labs
- Software Developer Docs - Silicon Labs

- Overview - latest - Simplicity Studio 5 Users Guide Silicon Labs
- GitHub - SiliconLabs/gecko_sdk: The Gecko SDK (GSDK) combines all Silicon Labs 32-bit IoT product software development kits (SDKs) based on Gecko Platform into a single, integrated SDK.
- Silicon Labs Community
- Flex SDK - Connect Networking Stack - Silicon Labs
- Gecko Software Development Kit (GSDK) - Silicon Labs
- Proprietary Wireless Protocols for Sub-GHz and 2.4 GHz - Silicon Labs
- User Manual