Proprietary Flex SDK 3.2.7.0 GA
Gecko SDK Suite 3.2
July 13, 2023

The Proprietary Flex SDK is a complete software development suite for proprietary wireless applications.Per its 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):

3.2.7.0 released July 13, 2023 (RAIL: support for EFR32xG21, Revision C and later)
3.2.5.0 released March 29, 2023 (RAIL: early access part support)
3.2.4.0 released January 11, 2023 (underlying platform changes only)
3.2.3.0 released October 13, 2021
3.2.2.0 released September 8, 2021
3.2.1.0 released July 21, 2021
3.2.0.0 released June 16, 2021

((•)) Proprietary

CONNECT APPS AND STACK KEY FEATURES

- EFR32xG23 GA support
- EFR32xG21 and MGMx CONNECT GA support
- FreeRTOS Support for CONNECT
- Long Range PHY Support for CONNECT
- New example: ECDH key exchange
- Example ported from Flex 2.7: DMP Light/Switch
- Support for the RAIL-embedded IEEE 802.15.4 PHY

RAIL APPS AND LIBRARY KEY FEATURES

- Wi-SUN Profile and Certified PHYs in Radio Configurator
- New example: SimpleTRX Standards for demonstrating 15.4/BLE usage

---

Compatibility and Use Notices

For information aboutsecurity updates and notices, see the Security chapter of the Gecko Platform Release notes installed with this SDK or on the Silicon Labs Release Notes page. 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 8.50.91

- Usingwine 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 10.2.0, provided with Simplicity Studio.

Contents

1 Connect Applications

1.1 New Items

Added in release 3.2.2.0

EFR32xG23 support

1.2 Improvements

Changed in release 3.2.0.0

BRD4183C support (EFR32MG22)

1.3 Fixed Issues

None

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 https://www.si-labs.com/products/software.

ID #	Description	Workaround
652925	EFR32XG21 is not supported for "Flex (Connect) - SoC Light Example DMP" and "Flex (Connect) - SoC Switch Example"	

1.5 Deprecated Items

Connect NCP-Host applications are not supported.

1.6 Removed Items

None

2 Connect Stack

2.1 New Items

Added in release 3.2.2.0

EFR32xG23 support

2.2 Improvements

None

2.3 Fixed Issues

ID #	Description
708489	Fixed an issue causing EFR32MG1 to fail to join any network.

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 https://www.si-labs.com/products/software.

ID #	Description	Workaround
501561	In the Legacy HAL component, the PA configuration is hard-coded regardless of the user or board settings.	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 to reflect the desired PA mode, voltage, and ramp time.
711804	Connecting multiple devices simultaneously may fail with a timeout error.	

2.5 Deprecated Items

None

2.6 Removed Items

None

3 RAIL Applications

3.1 New Items

Added in release 3.2.0.0

- New Applications
- Os support (FreeRTOS / Micrium) for each application except:
    - Flex (RAIL) – RAILtest
    - Flex (RAIL) – Light
    - Flex (RAIL) – Switch
    - Flex (RAIL) – Light Standards

3.2 Improvements

Changed in release 3.2.0.0

- BRD4183C support (EFR32MG22)

3.3 Fixed Issues

None

3.4 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
732659	**On EFR32xG23:** • **Wi-SUN FSK mode 1a exhibits a PER floor with frequency offsets around ± 8 to 10 KHz** • **Wi-SUN FSK mode 1b exhibits a PER floor with frequency offsets around ± 18 to 20 KHz**	

3.5 Deprecated Items

None

3.6 Removed Items

None

4 RAIL Library

4.1 New Items

Added in release 3.2.2.0

- Added support for the EFR32xG23 series of chips.

- Added a new RAIL_ConfigDirectMode() API for configuring direct mode settings on all chips. This can set whether the data stream is synchronous or asynchronous and which GPIOs are used for the feature.

- Added PA curves for EFR32XG23 for HP, MP, LP and LLP modes for both 14dBm and 20dBm variants.

- Added a new RAIL_RxDataSource_t to capture direct mode data on supported devices.

Added in release 3.2.1.0

- Added the new RAIL_STREAM_CARRIER_WAVE_PHASENOISE RAIL_StreamMode_t for phase noise measurement.

- Added RAIL_PA_BAND_COUNT to count RAIL_PaBand_t.

Added in release 3.2.0.0

- Added RAIL_StartScheduledCcaCsmaTx and RAIL_StartScheduledCcaLbtTx APIs to allow applications to schedule a CSMA/LBT transmit.

- Added support for a new RAIL_EVENT_ZWAVE_LR_ACK_REQUEST_COMMAND, triggered on the reception of a Z-Wave Long range packet with acknowledgement request bit set, following which the application must call RAIL_ZWAVE_SetLrAckData to populate the fields of the Z-Wave Long range acknowledgement packet.

- Added RAIL_TX_OPTION_RESEND to allow re-transmitting the packet most recently loaded into the Transmit FIFO. This can be used in combination with RAIL_SetNextTxRepeat() on supported platforms to repeatedly transmit the same packet.

- Added new RAIL_SetNextTxRepeat() API and RAIL_TxRepeatConfig_t to allow configuration of repeated transmits triggered by an initial transmit.

- Added ability to configure TxToTx state transition time via RAIL_StateTiming_t::txToTx and RAIL_SetStateTiming(). This time is generally used between an autoACK transmit and a user transmit that was pending. It is also used during repeated transmits by default, but can be overridden by the RAIL_TxRepeatConfig_t::delayOrHop or RAIL_BLE_TxRepeatConfig_t::delayOrHop configuration.

- Added ability to hop channels during repeated transmits in RAIL_TxRepeatConfig_t using RAIL_TX_REPEAT_OPTION_HOP.

- Added BLE-specific hooks for repeated transmits with channel hopping in RAIL_BLE_SetNextTxRepeat() and RAIL_BLE_TxRepeatConfig_t.

- The "RAIL Utility, Callbacks" component can now be configured to not provide the RAILCb_AssertFailed() function for situations where the application wants to provide its own implementation.

4.2 Improvements

Changed in release 3.2.2.0

- Now when using a RAIL_RxDataSource_t other than RX_PACKET_DATA the receiver will be disabled and must be manually restarted any time a RAIL_EVENT_RX_FIFO_OVERFLOW occurs. If the RAIL_EVENT_RX_FIFO_OVERFLOW event is not enabled, then the receiver will continue to run and lose some amount of data until the buffer is processed that matches the old behavior.

Changed in release 3.2.0.0

- Added support for the RAIL_TX_POWER_MODE_*_HIGHEST options in the RAIL_GetTxPowerCurve() function.

- Changed when RAIL_EVENT_IEEE802154_DATA_REQUEST_COMMAND event is issued to better facilitate support for 802.15.4E-2012 Enhanced ACKing and reduce time spent in the event handler. The event is now issued *after* receiving the Auxiliary Security Header (if present) in the MAC Header of the incoming frame, and for MAC Command frames, after receiving the MAC Command byte (which may be encrypted). This change is *not* backwards-compatible with prior releases for Enhanced ACK requesting frames, but is for Immediate ACK requesting frames. Use of RAIL_IEEE802154_EnableEarlyFramePending() is no longer required to support Enhanced ACKing; the notion of early frame pending notification has also shifted to after the Auxiliary Security Header for Enhanced ACK requesting frames.

- Changed RAIL Timer Synchronization over sleep on the EFR32xG21 to use the RTCC instead of the PRORTC to reduce current consumption in EM2.

- The posting of RAIL_EVENT_TX_ABORTED now occurs before the PA ramps down, at the same time other transmit completion events get posted.
- Allow RAIL_ZWAVE_ReceiveBeam to be run on US LR regions instead of always returning an error.

4.3   Fixed Issues

Fixed in release 3.2.3.0

ID #	Description
737292	Fixed an issue on the EFR32xG21 that was causing some Bluetooth LE coded PHY packets to be sent with a corrupted sync word and payload, breaking the connection.
738158	Fixed the RX_DIRECT_MODE_DATA RAIL_RxDataSource_t data source on the EFR32xG23.
741120	Fixed an issue where the IQ data captured during Bluetooth LE's CTE could be all zeros on the 2Mbps PHY.
743258	Fixed an issue that could cause the antenna switching pattern in Bluetooth LE AoX applications to sometimes repeat a previous sequence instead of following the expected antenna sequence.

Fixed in release 3.2.2.0

ID #	Description
671651	Fixed timing problems with certain State_Transitions or Rx_Channel_Hopping delay values on the EFR32xG22 and newer parts.
714271	Fixed an issue where RAIL_IEEE802154_Config2p4GHzRadio*() and RAIL_IEEE802154_ConfigGB*Radio() functions were improperly clearing or were setting certain RAIL_IEEE802154_EOptions_t. Also documented that these functions still implicitly clear or set certain RAIL_IEEE802154_GOptions_t suitable for that configuration.
716369	Fixed an issue where incorrect radio transition times were being applied at higher temperatures when using the high power PA on EFR32xG22 parts.

Fixed in release 3.2.1.0

ID #	Description
646980	An attempt to use an unsupported built-in radio channel configuration, e.g. on a module that does not support that protocol or configuration, will now trip RAIL_ASSERT_FAILED_INVALID_CHANNEL_CONFIG rather than returning success and ignoring the configuration.
675252	Fixed an antenna diversity regression where a transmitted auto-ACK would incorrectly go out the currently configured TX antenna rather than the antenna used to receive the packet being acknowledged.
676896	Fixed an issue in OOK PHYs where dynamic adjustments made to receive a packet with a high RSSI could persist after the packet and decrease the ability to receive packets with a lower RSSI.
696198	Fixed an issue on PHYs that do not support dual sync words. If RAIL_RX_OPTION_ENABLE_DUALSYNC is called when one of these PHYs is loaded the RAIL_ConfigRxOptions function will now return RAIL_STATUS_INVALID_PARAMETER.
697097	Fixed a rare situation where a premature RAIL_EVENT_TX_BLOCKED event might occur when auto-ACK is enabled and a scheduled transmit using RAIL_SCHEDULED_TX_DURING_RX_ABORT_TX is pending when an erroneous packet is received.
700439	Fixed an issue where configuring a Selective RF Sense Wakeup packet by calling RAIL_ConfigRfSenseSelectiveOokWakeupPhy followed by RAIL_SetRfSenseSelectiveOokWakeupPayload would put RAIL in fixed length mode and leave it there even after a new PHY was loaded. We will now revert the fixed length settings to their default when loading a new PHY after the Selective RF Sense Wakeup PHY was loaded so that the incoming PHY's settings are used.
710273	Fixed an issue using RAIL_TX_REPEAT_OPTION_HOP with RAIL_SetNextTxRepeat() or RAIL_BLE_SetNextTxRepeat() when they return an error found in the channel-hopping configuration, yet would still attempt to repeat the next transmit.

Fixed in release 3.2.0.0

ID #	Description
456701	Fixed an issue on EFR32xG1x parts where calling RAIL_Init() with the MSC->CTRL.CLKDISFAULTEN bit set would cause a bus fault.
654600	On EFR32XG21, a watchdog has been added to terminate an RSSI averaging operation in case the RAIL_EVENT_RSSI_AVERAGE_DONE event does not occur.
655541	Fixed an issue on EFR32XG22 and later where packet filtering might be incomplete in FEC-enabled radio configurations causing good packets to be improperly dropped. Note that if packet filtering fails close to the end of an otherwise successfully received packet, the packet may be flagged RAIL_RX_PACKET_READY_CRC_ERROR rather than RAIL_RX_PACKET_ABORT_FILTERED. Note this issue is still present on earlier chips.
665705	Fixed an issue where a transmit with RAIL_TX_OPTION_SYNC_WORD_ID 1 to use SYNC2 would improperly indicate SYNC1 was used in the packet trace appended information.
666275	Fixed potential delays when using RAIL's channel hopping or duty cycling features in EM1P mode on the EFR32xG22 and newer parts.
667103	Fixed RAIL_ReadRxFifo() to behave as documented when passed NULL dataPtr: the data is thrown away rather than copied out.
671817	Fixed an issue when switching between certain radio configurations (for example, Z-Wave) where use of RAIL_TX_OPTION_REMOVE_CRC can become permanently stuck.
673333	Fixed an issue with RAIL_TX_OPTION_WAIT_FOR_ACK transmits where an RX overflow during the ACK wait period would silently abort the ACK timer resulting in no RAIL_EVENT_RX_ACK_TIMEOUT being generated.
682032	Fixed an issue where setting RAIL_SCHEDULED_TX_DURING_RX_ABORT_TX for a scheduled transmit could cause subsequent non-scheduled transmits to be blocked.
696665	Fixed an issue in RAIL_ConvertRawToDbm for PAs that use piecewise-linear line segment fit, where the minimum raw power level was incorrectly compared to the minimum deci-dBm value.
697097	Fixed a rare situation where a premature RAIL_EVENT_TX_BLOCKED event might occur when auto-ACK is enabled and a scheduled transmit using RAIL_SCHEDULED_TX_DURING_RX_ABORT_TX is pending when an erroneous packet is received.
699890	Fixed missing C++ compatibility in the "RAIL Utility, Init" component's generated header file.
701604	Fixed an issue where using Rx_Channel_Hopping with channels in different frequency bands would cause RAIL to assert.
703788	Fixed an issue on EFR32xG2x devices where RAIL would not allow you to initialize the radio with voltage scaling enabled even though this is supported on these devices.
705595	Fixed an issue where custom PA curves provided through the PA Module were not respected when building Silicon Labs Zigbee applications.
708511	Fixed possible RAIL_ASSERT_FAILED_UNEXPECTED_STATE_TX_FIFO when RAIL_SetTxFifo() is called after the transmit FIFO had been filled enough to cause its write offset to wrap.

4.4 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
	Using direct mode (or IQ) functionality on EFR32xG23 requires a specifically set radio configuration that is not yet supported by the radio 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 fixed length is set to 0 are not working correctly on the EFR32xG23 series chips.	

4.5   Deprecated Items

None

4.6   Removed Items

None

5   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

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

Use the Flex SDK v3.x with the Silicon Labs Simplicity Studio 5 development platform. Simplicity Studio ensures that most software and tool compatibilities are managed correctly. Install software and board firmware updates promptly when you are notified.

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

This version of the stack does not integrate Secure Vault Key Management.

Security Advisories

To subscribe to Security Advisories, log in to the Silicon Labs customer portal, then select **Account Home**. Click **HOME** to go to the portal home page and then click the **Manage Notifications** tile. Make sure that 'Software/Security Advisory Notices & Product Change Notices (PCNs)' is checked, and that you are subscribed at minimum for your platform and protocol. Click **Save** to save any changes.

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

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

www.silabs.com