Zigbee EmberZNet SDK 6.7.5.0 GA

Gecko SDK Suite 2.7

May 1, 2020

Silicon Labs is the vendor of choice for OEMs developing Zigbee networking into their products. The Silicon Labs Zigbee platform is the most integrated, complete, and feature-rich Zigbee solution available.

Silicon Labs EmberZNet SDK contains Silicon Labs' implementation of the Zigbee stack specification.

These release notes cover SDK version(s):

6.7.5.0 released May 1, 2020
6.7.4.0 released April 22, 2020 (This release will not be certified)
6.7.3.0 released March 20, 2020
6.7.2.0 released February 21, 2020
6.7.1.0 released January 24, 2020
6.7.0.0 released December 13, 2019.

Zigbee

KEY FEATURES

- Support for EFR32MG22 parts
- Adds sample application for Zigbee Alliance TIS/TRP RF testing
- New optimized network steering algorithm
- Limited token set option for Smart Energy CommsHub
- Expanded EZSP Frame Header to support additional frame IDs

Compatibility and Use Notices

If you are new to the EmberZNet SDK, see Using This Release.

Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 8.30.1.

- 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 7.2.1, provided with Simplicity Studio.

- GCC is not supported with the EM35x, EM358x, and EM359x device families.

Contents

1 New Items

Note on Stack Protection

Stack protection is added by the compiler and has a fixed cost for the Zigbee stack of approximately 10 kB of flash and additional RAM for call stack overhead. If customers add to the application code, it will take additional flash. Silicon Labs cannot change the impact of using the stack protection feature. However, customers can choose not to use stack protection in their application configuration to save space.

Silicon Labs does not recommend stack protection for highly flash-limited devices. Stack protection has limited exposure in our installed customer base. While it passes all of our internal tests, it does not come with the many years of experience across millions of devices as do our normal (that is, non-stack-protected) libraries. Customers putting stack protection into production should be responsible for system testing.

1.1 New Plugins

Added in release 6.7.3.0

Green Power Translation Table

Implements the translation table functionality of the Green Power(GP) cluster, originally part of the Green Power Server plugin.

1.2 New APIs

For additional documentation please refer to our online Zigbee reference material at https://docs.silabs.com/zigbee/latest/

Added in release 6.7.4.0

emberAfGreenPowerClusterGpSinkCliCommissioningWindowExtend - Introduced to extend the commissioning window.

bootloader_secureBootEnforced - Added to check if signature verification on the application is enforced before every boot.

Added in release 6.7.3.0

emberGpdAfPluginGetGpdIeeeCallback - Called by the framework to get the IEEE (EUI64) address of the Green Power Device (GPD). Typically GPD application may have the IEEE address for each of the GPD programmed at a different location or storage, so the framework obtains the address through this callback.

Added in release 6.7.2.0

emberIncomingNetworkStatusHandler - This API is a superset of emberIncomingRouteErrorHandler. It is invoked when a network status message is received, informing the application of the over-the-air error codes for the specific destination.

Added in release 6.7.1.0

New APIs are added to expose parts of the source routing functionality to SoC apps.

- **emberOverrideAppendSourceRouteHandler**
- **emberOverrideIncomingRouteRecordHandler**

Added in release 6.7.0.0

emberAfNcpGpSentHandler – allows xNCP applications to handle DGP Sent event processing.

emberGetNeighborFrameCounter – fetch the last received frame counter as found in the Network Auxiliary header for the specified neighbor or child. This API returns an EmberStatus depending on whether the Frame Counter is found in the neighbor or child table.

MAC Timestamps

MAC timestamps for received packets are now exposed on the Zigbee stack. We provide an API **emberGetLastHopTimestamp(uint32_t *lastHopTimestamp)** to grab the value of the MAC timestamp for the latest received packet. The value returned is the MAC timer (ticks since up) at the moment when the Start Frame Delimiter (SFD) was received for the latest message.

This function may only be called from within:

- emberIncomingMessageHandler()
- emberNetworkFoundHandler()
- emIncomingRouteRecord()
- emberMacPassthroughMessageHandler()
- emberIncomingBootloadMessageHandler()

Please see the documentation for **emberGetLastHopTimestamp** for more details, including platform-specific information.

1.3 New Sample Applications

Added in release 6.7.3.0

352kB Internal Flash Bootloader

A new internal storage bootloader sample application bootloader-storage-internal-single-352k for devices with 352kB internal flash.

Added in release 6.7.2.0

Greenpower-sensor

A sensor sample application is added to the GPD framework.

Added in release 6.7.1.0

StandardizedRfTesting

Sample application demonstrating RF testing through TIS (Total Isotropic Sensivity) / TRP (Total Radiated Power) interfaces. The current version implements the latest test specification version (docs-19-01701-00). The version released with SDK 6.7.0.0 implemented a pre-standardized version of the specification.

2 Improvements

2.1 Plugin Changes

Changed in release 6.7.3.0

Green Power Server

- The translation table is now an optional configuration.
- To reduce NVM wear out of sink table token, the security frame counter in sink table is made as a separate token and the writing interval to NVM is made available as application configuration.
- More than one GPD may now perform interleaved bidirectional commissioning transactions in a commissioning session.
- During bidirectional commissioning if success frame is lost, the sink can now consider a correctly protected GPDF operational frame to announce pairing.
- The function for outgoing GPDF encryption considers all valid security levels.

ZCL Scenes CLI

The 'Transition Time' argument of the 'Recall Scene' CLI command for the Scenes cluster, "zcl scenes recall [groupId:2] [sceneId:1] [transitionTime:2]", is now optional, to bring it into line with the Zigbee Cluster Library specification, Version 7, 3.7.2.4.7.1

Zll Commissioning

Added plugin option and CLI command to allow a device on a centralized network to reject an interpan reset request. The new CLI command is: `plugin zll-commissioning noreset-nfn`

Changed in release 6.7.2.0

Zigbee PRO Stack Leaf Library

Added option to set broadcast table size to a value in the range 15-254. The default is 15.

Changed in release 6.7.1.0

Mbed TLS

Updated mbed TLS library to version 2.7.12.

Changed in release 6.7.0.0

Network Steering

Two new options have been added to the **Network Steering** plugin to improve performance.

- The Optimize Scans option (enabled by default) collects channel scans together and then joins networks that have been saved in a cache, resulting in fewer scans issued by the device when joining a network. For example, all channels will be scanned on the current mask, then networks are joined one by one without issuing another scan.
- The Try All Keys At Once option (requires the Optimize Scans option to be enabled) will use the install code derived key, centralized key, and distributed key when joining each candidate network. This results in faster joining times but may suffer from joining promiscuous networks allowing joining using the centralized key (e.g. if a device wishes to join with an install code, it may end up joining a Home Automation network before it finds its desired network using its install code derived key). Users that wish to prioritize joining using install code keys should not check this option.

The default for bdbSecondaryChannelSet has been changed from 0x07FFF800 to 0x4E73000, resulting in no overlap between the primary and secondary channel set. This is reflected in the **Network Steering** plugin with Optimize Scans enabled.

The Scan Duration Component option in the **Network Steering** plugin default has been changed from 5 to 4. This change results in spending around 260 milliseconds when scanning a channel, down from around 500 milliseconds per channel with value 5. Users should ensure that beaconing devices do not jitter for longer periods than this value when sending a beacon.

OTA Bootload Cluster Storage

The OTA Cluster Platform bootloader plugin has been updated with the ability to flash all images embedded within the .ota image. Each image is consecutively flashed onto the device with a device reset in-between flashes.

SE Token Libraries

Created an optional optimized token map for ZigBee Smart Energy Commshub devices (em356, em357, efr32xg1x). Removes unrelated tokens from stack libraries (including Green Power, Multi-Network, and ZLL). This feature can be enabled on select devices by choosing the appropriate option in the "Other Options" tab of AppBuilder in studio.

Zigbee PRO Stack Library

All ZigBee applications will now be built with the default security features enabled. The **Security Library Core** plugin has been merged into the main stack libraries (**ZigBee PRO Stack Library**, **ZigBee PRO Leaf Library**, etc) and the **Security Library Core Stub** has been removed.

The emberSetLogicalAndRadioChannel() API has moved from **ZigBee Light Link Library** to **ZigBee PRO Stack Library.**

HAL Library

Most applications running on EFR32 devices now take advantage of the new Simple MPU component, which disallows running from all RAM except the area specifically set aside for RAM functions. Due to a known incompatibility, the Simple MPU component is not used to initialize the MPU when running DMP or other Micrium-based applications.

Micrium RTOS

Stack size of the statistic task of the **Micrium RTOS** plugin was increased from 128 bytes to 256 bytes.

2.2 API Changes

Changed in release 6.7.4.0

The emberAfExternalAttributeReadCallback() function's signature has changed

The emberAfExternalAttributeReadCallback() function's signature has been changed to add the maxReadLength argument. This is a deliberate change to the API to improve defensive coding and to compel existing implementations to be modified. An implementation of the callback function must ensure that data copied to the provided buffer does not exceed the buffer length, indicated by maxReadLength. If the size of the external attribute value exceeds the indicated buffer size, no copy should be performed; instead, an appropriate error such as EMBER_ZCL_STATUS_INSUFFICIENT_SPACE should be returned.

Changed in release 6.7.2.0

Increased range of MacPollFailureWaitTime

The “waitBeforeRetryInterval” parameter of the `emberSetMacPollFailureWaitTime()` function has been changed from seconds to milliseconds with the type changed from uin8_t to uint32_t. This mac poll feature can be disabled by setting the “waitBeforeRetryInterval" to zero.

Changed in release 6.7.0.0

EZSP Protocol Version 8 and Secure EZSP Protocol Version 2

Both EZSP and Secure EZSP have adopted a new frame format with the following changes: (1) the fields of "Frame Control" and "Frame ID" are now two bytes; (2) no longer use "Legacy Frame ID"; (3) consume two bits of "Frame Control" to indicate the frame format version which is version 1 now.

The new frame format of EZSP protocol version 8 is described in the "UG100: EZSP Reference Guide" and the new frame format of Secure EZSP protocol version 2 is described in the "AN1125: Creating and Using a Secure EZSP Host-to-NCP Interface". More specific EZSP examples are shown in the "AN711: SPI Host Interfacing Guide for Zigbee".

EZSP_SET_GPIO_RADIO_POWER_MASK has been moved into the "EzspValueId" section and renamed to EZSP_VALUE_GPIO_RADIO_POWER_MASK. Use the EZSP_SET_VALUE command to achieve the equivalent function.

EZSP_SET_CTUNE and EZSP_GET_CTUNE have been moved into the "EzspConfigId" section and renamed to EZSP_CONFIG_CTUNE_VALUE. Use the EZSP_SET_CONFIGURATION_VALUE and EZSP_GET_CONFIGURATION_VALUE commands to achieve the equivalent functions.

Beacon Caching

The beacon caching code now records the most prioritized beacon per network. See emberGetFirstBeacon and emberGetNextBeacon for APIs on how to traverse the beacon cache. The most prioritized beacon is subject to parent classification, RSSI, LQI, and other factors.

2.3 Sample App Changes

Note: Sample applications in releases beginning with June 2020 will no longer target 256 kB parts, including EM35x, EFR32MG1, and EFR32MG14. This is due to new developments in the Zigbee specification that have led to increased flash requirements beyond the 256 kB provided by these parts.

Changed in release 6.7.4.0

The Green Power Device sample application has been renamed to Green Power On/Off Switch.

The Green Power Sensor sample application has been renamed to Green Power Sensor Device.

Changed in release 6.7.3.0

The Z3 Light Gp Combo sample application is updated with the description and LED indication during commissioning.

Changed in release 6.7.2.0

Green Power device (GPD) applications now have a command line interface option.

Changed in release 6.7.1.0

Green Power device (GPD) has been updated as follows:

- Low power timing for rxoffset, application idle loop and rx window timing are implemented using sleep timer driver module
- GPD Sample application button action have changed - please follow the "Description" in .isc file in the the generated sample application project.
- gpd-startup plugin that was using device start up files are removed as the framework uses the hal libray plugin.
- added LED and debug print in the sample application to help development , these can be turned off by the user.

2.4 Documentation Changes

None

3 Fixed Issues

Fixed in release 6.7.5.0

ID #	Description
471327	Fixing early rollover of halCommonGetInt32uMillisecondTick (482349) has resulted to fix the issue with event scheduling, where a device may stop sending periodically scheduled messages (such as ZCL reports and Link Status messages) after around 2 days. This affected EmberZNet 6.7.0- 6.7.4.
482349	Fixed early rollover of halCommonGetInt32uMillisecondTick on EFR32 devices. It now once again rolls over at 2^32 milliseconds. Also fixed a bug in halCommonGetInt64uMillisecondTick that was limiting the return value to 32-bits. Due to time unit conversion, it still does not quite use the full 64-bit range. The maximum value, however, represents an uptime of millions of years.

Fixed in release 6.7.4.0

ID #	Description
401203	AppBuilder was not properly substituting Stack Protection variants of libraries into generated EM35x NCP projects (neither for Studio's .cproject file nor IAR's .ewp file) when the Stack Protection option from Other Options tab was selected. Also, the "--stack_protection" compiler flag was not being added to the generated IAR .ewp file. Now both issues are fixed beginning with Simplicity Studio release SV4.1.13.5+.
452745	Resolved the issue where an end device may unexpectedly relay a Route Record message to its parent, which can lead to an address conflict being reported for the end device ID.
459215	Prior to this change, if an APS encrypted message needed to be sent to a destination whose EUI64 was not known, the sender would get an error code EMBER_APS_ENCRYPTION_ERROR. This fix enables the stack to initiate an IEEE address discovery on behalf of the application and save the received EUI64 in the Trust center cache section of the address table (size of this section is configurable in the address table plugin). We now use the error code EMBER_IEEE_ADDRESS_DISCOVERY_IN_PROGRESS. When this error message is seen, the application may retry the message to this destination after at least 2 seconds, to allow the IEEE address discovery to complete. If subsequent tries also result in the same error code, other issues on the network might need to be diagnosed, like: a) destination node is not on the network b) there are problems with the health of the network c) there may not be any space set aside in the address table for the newly discovered address - this can be rectified by reserving more entries for the trust center cache in the address table plugin
466334	Minor modifications are made to the way the short nodeId <-> EUI64 lookup is determined.
467374	Defensive coding is improved for certain buffer copies to arrays sized by the ATTRIBUTE_LARGEST macro to ensure the array is not overrun. This fix pertains only to conditional code enabled by definition of the BIGENDIAN_CPU macro.
467424	Incorrect handling of length for ZCL string has. been fixed. For instances in the code where ZCL string-type attributes are processed (CHAR, OCTET, LONG_CHAR, LONG_OCTET), string size checks are improved to ensure that a string does not exceed the size of a buffer to which it is copied; or, a malformed string contained in a received message is more defensively identified and handled (for example, a string whose length prefix specifies a size greater than the size of the message that contains the string).
467589	Updated default dynamic multiprotocol (DMP) transition timings to make them work with Zigbee and BLE DMP applications. The previously suggested workaround of adding 30 µs to the default transition time using RAIL_SetTransitionTime() is no longer required.
470511	GPD sample applications could be built properly with both Simplicity IDE and IAR IDE.
471107	The CMU_LFRCOSetPrecision() function is now only available on devices that actually have the precision LFRCO feature.
471824	Two green power callback configurations are updated, hence when exporting an existing Combo application the following callbacks need to be resolved in the .isc file before regeneration. 1. emberAfGreenPowerClusterGpSinkTableRequestCallback - because this callback is implemented in the green-power-server plugin. 2. emberAfGreenPowerClusterGpPairingSearchCallback must be excluded in the callback selection list.

ID #	Description
472395	Application Framework plugins that implement processing of received ZCL cluster commands are modified to improve handling of ZCL string types (CHAR, OCTET) conveyed in those commands. If a string is copied from a ZCL command payload to internal storage, defensive code limits the data copied to the maximum capacity of the internal storage buffer receiving it.
477808	Re-enabled MPU protected guard region to trap stack overflows that corrupt this guard region, and to correctly display a corresponding error code on EM35x NCP parts. Stack overflow may occur on EM35x NCPs when the IAR stack protection plugin is turned on, as these parts are RAM-constrained. The default call stack size for EM35x applications with Stack Protection enabled has been increased to 3200 bytes.
479024	When End Device Bind Stub Library was selected and the Smart Energy Tokens option was selected, AppBuilder would look for an invalid library path for this library, causing the build to fail. The issue is now fixed.
480442	Enabling Smart Energy (SE) token optimization on SoC did not include the correct libraries on previous releases. Starting from studio release SV4.1.13.5+ SE token optimizations will include correct non-stub libraries for Zigbee System on Chip applications.

Fixed in release 6.7.3.0

ID #	Description
400418	Fixes an issue where a touchlink initiator cannot link to a non-factory-new end-device target.
400427	Fixes an issue where a factory-new touchlink initiator is unable to restore its original radio channel.
401831	Fix to prevent a touchlink initiator acting as a touchlink target, if it doesn't have the Touchlink Commissioning server-side cluster and plugin. This is implemented using the new Touchlinking Policy bitmap mechanism.
414462	On sub-GHz devices, the per-device duty cycle now increases and decreases in a more linear way over the period of time.
414941	In earlier stack versions dating back as far as EmberZNet 6.0.0, after ECC operations such as CBKE are performed, the stack was vulnerable to a race condition wherein, if packet reception (excluding MAC ACK reception) occurs at a certain point during MAC-level RX processing in emberTick(), the PHY-to-MAC queue can enter an invalid state in which incoming messages are no longer processed at any level of the stack after the radio RX interrupt completes. This condition manifested as the device appearing to be "deaf" to incoming packets, aside from responding with MAC ACKs and showing the packet reception in the PTI capture stream, and this condition would persist until the device was rebooted. Due to a related issue, the emberCounterHandler was not being triggered for PHY To MAC Queue Limit Reached when this state was entered. Workaround for affected releases: Reboot the device after CBKE or other ECC operations are completed to ensure device is not vulnerable to this race condition.
419564	Fixed issue where a router initiator would hang after performing a network update resulting from receipt of a scan response with a higher network update id.
421702	Fixed issue where a second touchlink request may fail the client-side validation in certain circumstances, after the first request failed. (The touchlink commissioning client uses the tokenized node type, rather than the correct run-time node type.)
423335	A touchlink device should be an initiator in order to start a scan: fix an issue where a non-initiator is permitted to do so in the Zll Commissioning Client.
423471	Fix incorrect channel scanning order for a touchlink request: the Base Device Specification states that the primary channels shall be scanned first, optionally followed by the secondary channels. Currently, the ZLL Commissioning plugin sets up a combined channel bitmask, and uses that to scan.
423478	Allow a local (or remote) factory reset to override an in-progress touchlink. This fixes an old issue where an end-device touchlink initiator failed to complete a touchlink request, because it was unable to rejoin to its parent. This should not now occur, since the touchlink will automatically be aborted after a number of rejoin retries, but it will also provide a last-chance recovery mechanism should a touchlink hang for any other reason.
423821	A local (CLI) or remote (interpan) touchlink reset to factory new will now automatically clear the binding table.
424355	Fixes an issue where a non-factory-new sleepy end device touchlink target-capable initiator is not able to receive a device information response in certain circumstances
428215	Fixes the check which prevents a coordinator being stolen: it should use the run-time node type, not the ZLL node type, as in the latter, a CO will always report as a router.
453315	DynamicMultiprotocolLightSedSoc now defaults to using LFXO rather than LFRCO for EM2/3 clock source.

ID #	Description
453425	When a route error occurs, we now always send it back to the source so that a child is able to get a notification when a message could not be routed. Customers can implement the "emberIncomingNetworkStatusHandler" to handle the route error.
453570	The incoming packet mic calculation routine now takes into account the rx-after-tx bit.
457293	EM4 sleep mode is now working for Green Power Device (GPD).
459713	After power cycle the nodeId is looked up for light weight unicast mode.
462628	Series 2 80MHz selection is now done by selecting HFRCODPLL as the HFCLK Source and then selecting 80MHz under HFRCODPLL Frequency Selection. 76.8MHz has been added as an option for some boards as well.
464545	Add EMBER_COUNTER_PHY_TO_MAC_QUEUE_LIMIT_REACHED counter to log when max PHY to MAC queue limit is reached. After that limit is reached, packets will be dropped. The number of dropped packets will be recorded in this counter.
465244	The sink test cases 4.4.4.3 and 4.4.4.6 are fixed.
465334	Fixed an issue where TEMPDRV_GetTemp() could return incorrect temperature results if not called immediately following a temperature interrupt.
465887	Fixed issue where calling LETIMER_Reset() could cause BusFault when called on a disabled LETIMER peripheral.
466779	Added a routine that validates if public-key is on the curve in mbedtls_ecdh_compute_shared for Series-2 products.
468127	A DMP sleep issue was fixed, where the application would stay in EM0 state after returning from EM1 longer than required.

Fixed in release 6.7.2.0

ID #	Description
349232	ota-server and ota-client will use the app builder's default response policy in outgoing commands. Note that there is a mandatory setting for this bit in most cases and in these instances, the specification trumps the behavior.
399475	Per-device duty cycle for sub-GHz devices will be reset after a sub-GHz channel change by calling the "emberMultiPhySetRadioChannel()" function or using the "network multi-phy-start" command.
449010	Added function to check if any (W)TIMER instance supports Dead Time Insertion (DTI) and added configuration support for Dead Time Insertion (DTI) for other timers than TIMER0.
450035	HFLE clock branch handling is now correct on all Gecko families.
456751	Fixed issue where GPIO_EM4EnablePinWakeup() sometimes did not clear interrupts, causing an immediate wakeup from EM4.
458106	Fixed issue where function printDiscoverCommandsResponse() can potentially get into a watchdog timeout or an infinite loop (in case the default watchdog functionality is disabled) if a specific malformed packet is received that causes the length argument to be greater than 255.
459690	Fixed bug in EFP driver sl_se_init() function. This function could return SL_STATUS_OK even though an error was detected.
461130	Fix a timer initialization issue when radio coexistence PHY Select GPIO is defined.
462054	Add support for runtime configuration of radio coexistence priority options if directional priority is defined but not static priority.
462527	A version check has been added to guard against use of Secure Boot with anti-rollback on products with (V)SE version lower than v1.2.1.

Fixed in release 6.7.1.0

ID #	Description
235222	Virtual UART (VUART) functionality over USB on WSTK has been restored in WSTK firmware version 1.4.4

ID #	Description
401119	The Image Builder utility no longer incorrectly flags a "Bad length for tag. Size X < Min size Y" warning when creating an OTA image from a .gbl file. This fix was available in 6.7.0.0, but was erronenously documented as a known issue in the release notes.
410372	Fixed a buffer allocation bug in the NCP framework which resulted in all packet buffers being re-initialized under exceptional high load conditions.
434755	Fixed an issue of ID Conflicts being flagged when none were present in the network. Underlying issue was memory corruption triggered when a router performs a secure rejoin with another router that is not its parent. This could alter data in Neighbor Table entries, leading to false flag ID Conflicts.
451828	Applications configured for NVM3 will now be automatically reconfigured to using sim-eeprom1 when selecting an EM35x device platform. Similarly, applications configured for sim-eeprom1 or sim-eeprom2 will be automatically reconfigured to use NVM3 when selecting an EFR32xG21 device platform. There is no change to the configuration when selecting an EFR32xG1x device platform.
451867	Two issues in the Network Steering plugin have been resolved. The first involved failing to join any networks if the "Optimize scans" and "Try all keys at once" options were set and an install code was not flashed to the device. The device would exit the join process with an output of "Error: NWK Steering could not setup security: 0xB7." The device now continues the steering process using the well-known key and the distributed key, skipping the absent install code. The second issue involved failing to join networks using the install code derived key if the "Optimize scans" option was set and the network was found as a result of the secondary channel scan. The issue did not affect a network found on the primary channel mask. This issue has been resolved.
452263	Incorrect bit masks were fixed in the EFR32 sleep code.
452377	RAM functions for GCC built applications are now placed in .ram section.
452682	Dynamic Multiprotocol applications now have 80MHz as the default clock configuration for EFR32xG21.
452685	Fixed the accuracy of the value reported by emberGetLastHopTimestamp for EFR devices. We now provide the exact 24 bit value passed up from the RAIL timer.
453206	Fixed an issue in NVM3 that could cause an error in the content of an existing data object after firmware upgrade from GSDK2.4.0 to GSDK2.5.0 or higher. The problem is related to the fact that NVM3 released in GSDK2.4.0 and earlier could write data objects in a format that was off spec. Although the format was incorrect, all functions handled the format correctly. From GSDK2.5.0 this format issue was fixed, but as a side effect, repacking data written with GSDK2.4.0 or earlier could unintentionally cause data error. The fix ensures that both the pre- and the post-GSDK2.5.0 formats are handled correctly.
453254	Fixed an issue where Rx packet timestamps passed up to MAC layer on EFR32XG2 series were not at sync-detect like on EFR32XG1 series and older platforms.
453420	Fixed an issue where studio was loading old stacks instead of newer stacks for Zigbee.
455341	Image builder was updated in the 6.7.0.0 release without updating the version number. The version has now been updated to 1.5.2.
456319	Fixed a bug in the processing of optional arguments for generated ZCL CLI commands "zcl ..." whose effect is to construct a ZCL message buffer. Depending on the number and data types of optional arguments, could result in CLI error, code hang / watchdog reset, or malformed ZCL message content.

Fixed in release 6.7.0.0

ID #	Description
239480	Fixed a code flow issue causing network leave announcements to be sent before the leave response. The network leave announcement will now wait until the ZDO Leave response goes out (or until APS retries run out, if enabled).
330132	NCP applications can now be augmented with custom token headers.
331528	There was an issue with the name of the Manufacturing Library CLI event to check that sending is complete. The metadata has been fixed to correctly generate the event code and the handler function has been renamed to match. The correct event is now set active in the mfglibRxHandler.

ID #	Description
348357	As the ApsFrameCounter currently is a shared variable for multi networks, resetting it every time when the node is joining a new network may cause the TC of the joined network to drop the APS encrypted packets. For a solution, we will check whether or not the current joining network is the only active network, if no, don't reset the ApsFrameCounter.
359622	If the OTA incoming message is an image page request, then don't send a duplicate response when handling the incoming message, because the response for an image page request is also handled by the "sendBlockRequest()" function.
371180	Fixed bug where including the Battery Monitor plugin could disable other features of the chip inadvertently.
385119	Fixed "option install-code" CLI command to stop adding derived key into the link key table for Z3.
385412	After receiving a device announce, the trust center will not send out an immediate route request with the default HIGH RAM concentrator type, anymore. Instead, an internal message is passed to the source route library to schedule a route request with correct concentrator type. The new request is scheduled according the min/max timing period between MTORRs, which itself could be configured through the Concentrator Support plugin.
385812	The existing em4 plugin callbacks have been completely renamed. Now we have 4 callbacks, 2 of which needs to be implemented by the application if it wishes to use the em4 mode. These 2 callbacks are emberAfPluginEm4EnterCallback() and emberAfPluginEm4ExitCallback().
387743	Zigbee leaf stack no longer consumes any memory creating/storing a routing table.
390692	A device that receives a unicast ZCL Groups cluster AddGroupIfIdentifying command, and which is not identifying, now returns ZCL status SUCCESS in the Default Response message. Previously it returned status FAILURE, which is wrong since the device is behaving correctly (not identifying, thus does not add group) per ZCL specification. Note that this is observed only for a received unicast command; no response is sent for a command received via multicast/broadcast.
398189	The level-control plugin's implementation of the emberAfOnOffClusterLevelControlEffectCallback() function is refactored to eliminate a circumstance where an "if" conditional statement tests a variable for a value that was earlier assigned to that variable. Coverity flags this as an always-true condition.
400188	Plugins enabled/disabled from Appbuilder can now enable/disable hardware configurator modules/peripherals given that an Appbuilder plugin is associated with a hardware configurator.
403200	The application image upload window has been updated to remove all pre-built Ember (em*) binaries for HA/Se/Sensor/Sink.
408103	An issue with debug output timing on EFR32xG21 has been fixed. This results in improved speed and/or reliability in a number of areas, including sleep and NCP communications, when the Debug plugin is enabled.
424410	Fixed the issue of emberSerialReadAvailable returning values greater than the buffer size.
426464	CommsHub will now send a response when receiving an invalid enum in a EndDeviceTimeOut Request.
429083	Fixed EZSP host makefile compatibility issue with Windows 10/Cygwin environments.
430883	Fixed an issue where Dual Band Coordinators could cause the network retry queue to overflow when forwarding a broadcast fails on one interface, but succeeds on another. We now submit broadcast packet on second MAC only if it is succeeded on native MAC.
430956	Fixed an issue where the original channel's used duty cycle could be reset during a channel scan.
431858	Incorrect interrupt names were corrected in diagnostic code for EFR32xG21 and xGM21 parts.
431860	Fixed a potential out of bounds memory access in token code.
434267	Fixed bug on EFR32 where checking for USART TX being idle returned true before transmission was complete, which had resulted in some NCP bootloading issues. The affected functions were halInternalUartTxIsIdle and COM_InternalTxIsIdle.
437423	emberGetChildData was not properly masking an internal flag in the timeout before returning the data. This resulted in the stack diagnostics plugin printing incorrect timeout values. The issue has been corrected.
440341	Fixed a GCC linker issue. The linker will error out if there is a possibility of overlap of data_init section.
440898	Fixed issue where NVM3 library was not included properly in GCC projects for EFM32 in Simplicity Studio.
442654	Fixed a race condition that could result to network retry queue overflow. The race condition could occur on a router or coordinator device, causing the NWK Retry Queue to lock up and overflow, such that no more NWK or APS layer frames would be transmitted. This could occur when all of the following conditions were met: (1) An APS broadcast frame to addressed to 0xFFFF is pending on the parent for relaying to at least one sleepy child on next poll. (2) Parent queues this broadcast data as a MAC unicast for transmission to its sleepy child after receiving a Data Request and begins transmission of the data. (3) A second broadcast to 0xFFFF is queued for relaying to sleepy children before the in-flight transmission of the first broadcast completes its handoff to the end device.

ID #	Description
444155	We have observed a minor issue in password-protected-cli. In case of a successful password entry after a few wrong attempts, number of wrong password entry attempts did not reset in the tokens. Therefore, the number of wrong password attempts carried over after the next reset. Depending on the wrong password entry threshold, this could introduce addition of an unwanted lockout period in between resets. To fix this, now we reset the token value, after a successful password entry attempt.
444215	There was an issue when setting a default value for a 64-bit attribute value (with ZigBee data type: BITMAP64) in the appbuilder.
444454	An issue has been fixed where emberGetNextBeacon would fail to return a valid, next beacon. This would cause network steering to occasionally fail to join a network when the Optimize Scans option was ticked.
444531	A timeout issue with em35x SPI NCPs has been resolved by moving the seeding of the PRNG with strong rand numbers up in the initialization process before initialization of the serial connection with the host. Additionally, the default wait time in host apps has been increased from 300ms to 350ms (WAIT_SECTION_TIMEOUT_MS).
444929	Fixed a race condition that could result to network retry queue overflow. The race condition could occur on a router or coordinator device, causing the NWK Retry Queue to lock up and overflow, such that no more NWK or APS layer frames would be transmitted. This could occur when all of the following conditions were met: (1) An APS broadcast frame to addressed to 0xFFFF is pending on the parent for relaying to at least one sleepy child on next poll. (2) Parent queues this broadcast data as a MAC unicast for transmission to its sleepy child after receiving a Data Request and begins transmission of the data. (3) A second broadcast to 0xFFFF is queued for relaying to sleepy children before the in-flight transmission of the first broadcast completes its handoff to the end device.
445334	Network steering did not handle a busy channel status properly, and therefore it would have failed joining any channel, if one channel was busy. The issue is now fixed.
445625	The Device Version for endpoints in the following dynamic multiprotocol sample applications has been corrected from 2 to 1: DynamicMultiprotocolLight, DynamicMultiprotocolLightSed, DynamicMultiprotocolZigbeeZllLightBle.
445764	Fixed a problem with energy scans on sub-GHz channel pages which could result in a packet buffer overflow, corrupting other packet data used by the stack.
447448	Fixed emlib issue where the core clock frequency was used instead of the hf clock frequency when calculating the frequency of HFCLKLE which can be used in the LFB clock tree. This bug could cause wrong baudrate calculations for LEUART when using HFCLKLE in systems where the HF clock and core clock are different frequency.
447797	EMLIB IADC: The definition of `iadcNegInputGnd` has been modified to set PINNEG to 1. This prevents a polarity error when performing IADC conversions between supply pins and ground.
449148	Fixed issue where EFR32xG21 NCP apps would fail to compile due to missing GPIO definitions.
452265	Restored the use of USART0_RX_IRQHandler on EFR32xG2 devices. This will allow use of wake up from idle plugin and software flow control on EFR32xG21.

4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
60858	Sleepy broadcast payload is sometimes corrupted when relaying to child	No known workaround
60975	EZSP_VALUE_TOKEN_STACK_NODE_DATA, EZSP_VALUE_UART_SYNCH_CALLBACKS, EZSP_VALUE_MAXIMUM_INCOMING_TRANSFER_SIZE, and EZSP_VALUE_MAXIMUM_OUTGOING_TRANSFER_SIZE ValueIDs are writable but not readable.	No known workaround
70851	System-timer implementation precludes runtime adjustment to GPIO wake handling	No known workaround
82569	RAM corruption (in Packet Buffers) could occur if MAC Filter Match List Size is non-zero and a list of the maximum size is provided to ezspSetValue for EZSP_VALUE_MAC_FILTER_LIST.	No known workaround
106307	Nodetest calChannel command does not wake the radio to work properly.	No known workaround
119939	ZDO IEEE Request's APS ACK proxied by parent incorrectly includes long source address.	No known workaround
135649	Multi-networking can cause APS frame counter confusion between networks.	Use emberAfSecurityInitCallback to add EMBER_NO_FRAME_COUNTER_RESET to EmberInitialSecurityBitmask.
229938	ZLL devices are sending ZLL device information frames with the 0x0104 HA/Z3 common profile ID instead of 0xC05E ZLL profile ID.	No known workaround
251287	Lowest current is not achieved during sleep on lowest current during sleep on EFR32xG12, EFR32xG13, and EFR32xG14.	To achieve the lowest current during sleep on EFR32xG12, EFR32xG13, and EFR32xG14 parts, you must turn on voltage scaling. However, the radio will not operate with voltage scaling turned on, so to turn it on you must also make sure to disable it after each wake-up. Furthermore, some resets will not turn off voltage scaling, so please ensure that it is disabled before attempting to turn on the radio. Note that there is a ramp when turning voltage scaling on or off, so enabling this feature may increase the time it takes to go to sleep or wake up.
261670	Harden the ZLL touchlink process to mitigate malicious attacks	No known workaround
266341	Z3 Light sample app has two endpoints that support similar cluster commands, so duplicate responses may be generated for certain commands.	No known workaround
271644	A device that performs a classic join to a legacy ZLL gateway may eventually leave the network on its own initiative.	No known workaround
278063	Smart Energy Tunneling plugins have conflicting treatment/usage of address table index	No known workaround
281231	Enabling Serial 3 or USB functionality on EM358x and EM359x may cause memory management faults and other errors. As EM358x and EM359x USB support has been deprecated, please ensure that Serial 3 and USB functionality are disabled.	No known workaround
281832	Green Power Common plugin incorrectly formats groupList and groupListCount parameters of GP Pairing Configuration frame.	No known workaround

ID #	Description	Workaround
289569	Network-creator plugin power level picklist doesn't offer full range of supported values for EFR32	No known workaround
295498	UART reception sometimes drops bytes under heavy load in Zigbee+BLE DMP use case	Use hardware flow control or lower the baud rate.
301024	Currently the Dynamic Multi-Protocol Sample applications do not build with GCC, they require the IAR compiler due to dependencies on both the BLE and Micrium stacks.	No known workaround
312291	The halCommonGetIntxxMillisecondTick functions on linux hosts currently use the gettimeofday function, which is not guaranteed to be monotonic. If the system time changes, it can cause issues with stack timing. The workaround is to modify these functions to use clock_gettime with the CLOCK_MONOTONIC source instead.	No known workaround
331438	Service discovery may time out too quickly in busy networks.	Define EMBER_AF_DISCOVERY_TIMEOUT_QS to customize the timeout period.
333146	The coexistence plugin behavior has changed to be required for all applications. By default coexistence functionality should be stubbed out when not configured via board header (EM35x) or not enabled via HWCONF (EFR32). Appbuilder may automatically enable the HWCONF coexistence module without warning which can trigger an error for dynamic multiprotocol applications, which do not yet support coexistence.	No known workaround
338151	Initializing NCP with a low packet buffer count value may cause corrupt packets.	No known workaround
345167	Sleepy end devices occasionally not sending APS ACK for received APS unicasts polled from parent.	No known workaround
356937	Read/write attribute CLI commands do not support manufacturer-specific ZCL attributes. Some implementations may allow local CLI debug access to display or modify these attributes.	Access the attributes from a remote device in the network via ZCL global Read/WriteAttributes commands.
362133	The default chip RSSI offset on the EFR32 chips is incorrect. For accuracy, we recommend measuring this on your hardware and then applying it with the \ref RAIL_SetRssiOffset() API. In the future, more reasonable defaults will be provided, however, the board dependent component may still need to be measured for custom hardware.	Use RAIL_SetRssiOffset()
363162	There is a bug in emberAfAddAddressTableEntry which could allow for duplicate entries in the address table	Under Investigation
398694	Disabling endpoint 2 (Touchlink) in the Z3Light sample app causes high emberRunTask execution time.	Under Investigation
426066	OTA client could reset when it starts to apply the new image if all debug print is disabled.	No known workaround
426200	Enabling a large number of endpoints (~40) and setting a large reporting table entries (close to 255) causes emberAfPluginReportingLoadReportingConfigDefaults routine to take ~14 seconds to run, which results in a watchdog reset.	Disable or reset the watchdog in the loop.
437502	halInit() is called twice in EmberZNet SPI NCP Host Application.	No known workaround
437704	The OccupiedCoolingSetpoint attribute should be optional by default in Appbuilder.	Modify the "optional" flag from false to true in the OCCUPIED_COOLING_SETPOINT attribute in app/zcl/ha.xml

ID #	Description	Workaround
437817	After EmberZNet 6.6.0, joining and insecurely rejoining children are now subject to a new, temporary timeout EMBER_SHORT_CHILD_TIMEOUT. This defaults to 2 minutes if undefined and should be able to be overridden. However, it doesn't function as expected since defining and changing the value of EMBER_SHORT_CHILD_TIMEOUT doesn't take any effect. The short temporary timeout for the joining device will always stay as 2 minutes.	No known workaround
438670	GCC compiled XNCP applications will assert whenever attempting to resize the address table size to a larger size.	Define EMBER_ADDRESS_TABLE_SIZE to the desired value when compiling the XNCP app.
438993	Certain IRQ Pin/Port settings do not allow wake on interrupt in some EM3xx parts.	No known workaround
442664	In a dense network with many devices joining simultaneously, a packet buffer assert is sometimes seen when collecting beacons during network steering.	No known workaround
444721	There is an incompatibility between the prebuild/postbuild scripts, wine 4.0, and IAR Build. As a result, this release does not contain any prebuilt EBL or GBL files.	Please use wine 1.8.5 to trigger IAR build processes in a unix environment.
451029	The Z3Light is configured as a Router Zigbee Device Type instead of Coordinator/Router. As a result, whenever modifications to the ZCL configuration are made in an application, the resulting update will disable coordinator-specific plugins such as Network Creator.	Change the Zigbee Device Type to “Coordinator or Router”. Re-enable the Network Creator plugin if disabled.
451205	The Enable EM1/EM2 option in the Micrium RTOS plugin is currently disabled for EFR32xG21 devices due to stability issues.	No known workaround
453086	ZCL Diagnostic Cluster attribute printing has formatting errors (duplicated values).	No known workaround
456350	"emberAfPluginIasZoneClientReadAttributesResponseCallback" in "ias-zone-client.c" incorrectly sets the type of "zoneStatus" and "zoneType" attributes to uint8_t. The correct type should be uint16_t.	Modify the type of these two attributes to "uint16_t" in "emberAfPluginIasZoneClientReadAttributesResponseCallback".
458128	When changing architecture to an ISC, the generation step might fail with a duplicate API implementation error. This happens when changing the platform of a sim-eeprom2 enabled application to an EM3xx device. This automatically enables the sim-eeprom1 but does not disable sim-eeprom2.	After changing the platform, ensure the correct sim-eeprom version is enabled and disable all other token storage plugins.
465180	The Coexistence Radio Blocker Optimization item "Enable Runtime Control" may block proper Zigbee operation.	Optional 'Wi-Fi Select' Control of Blocker Optiomization should be left "Disabled".
469704	On EFR32xG2x there is a known issue with the debug interface where a pin reset, connection to a debugger, or flashing an image on device could cause a pull up on DBG_TDI pin, which cannot be reliably disabled in software. On EFR32XG22 this could cause the device to be unable to talk to the Serial flash since the FLASH_SCS and DBG_TDI are sharing the port and the pin. As a result a reset loop is possible when a bootloader is used. A pin reset is needed to get the device out of the reset loop.	Avoid a pin reset of EFR32XG2X when a debugger is attached. If a pin reset is necessary, avoid using PA3 and PA4, or use JTAG instead of SWD.
474624	**The MSC_WriteWordDma() function is only able to copy data from RAM to the flash. The function will fail if you copy data from flash to flash.**	**No known workaround**

ID #	Description	Workaround
481128	Detailed Reset Cause and crash details should be available by default via the Virtual UART (Serial 0) on NCP platforms when Diagnostics plugin and Virtual UART peripheral are enabled.	Since Serial 0 is already initialized in the NCP, customers can enable the emberAfNcpInitCallback in the Zigbee NCP Framework and call the appropriate diagnostic functions (halGetExtendedResetInfo, halGetExtendedResetString, halPrintCrashSummary, halPrintCrashDetails, and halPrintCrashData) in this callback to print this data to Serial 0 for viewing in the Network Analyzer capture log. For an example of how to use these functions, refer to the code included in af-main-soc.c's emberAfMainInit() when EXTENDED_RESET_INFO is defined.
481964	When a Router or Coordinator device receives a "Parent Announce" ZDO message, a packet buffer will fail to be allocated after the Parent Announce Response is issued. Over time, if many router/coordinator devices in the network are resetting and emitting this announcement, the affected device will begin to see Packet Buffer Allocation Failures, and incoming and outgoing messages may fail to be processed after all buffers are exhausted, until the device reboots.	Handle Parent Announce messages manually at the application laying by enabling the optional emberPacketHandoffIncoming stack callback and intercepting messages with PARENT_ANNOUNCE cluster ID when EmberZigbeePacketType is EMBER_ZIGBEE_PACKET_TYPE_ZDO. After manually handling the message and sending an appropriate Parent Announce Response, the packet should be discarded by returning the EMBER_DROP_PACKET EmberPacketAction to avoid the code path in the stack that leads to this buffer leak.

5 Deprecated Items

Advanced Notice

Zigbee support – The upcoming Zigbee R23 specification will not be supported on any devices matching the following criteria:

- All EM35x devices
- Any devices with 256 kB flash or less [All wireless MCU families]

Reason: The Zigbee R23 specification will lead to increased flash requirements and introduce new security requirements that these devices cannot adequately support. In preparation for that, the next major release (expected June 2020) will be the last major release to support these devices. Customers are advised to start any new Zigbee developments using EFR32 series devices that exceed the criteria as listed above.

Maintenance Period: Critical bug fixes and security patches for the next major release implementing Zigbee specification R22 will continue to be made available for the lifetime of EM35xx series and wireless MCU devices with less than 256 kB, as specified by the wireless longevity commitment https://www.silabs.com/wireless/longevity-commitment

Deprecated in release 6.7.0.0

Network Steering Configured Key API

The "plugin network-steering pre-configured-key-set" CLI command and emAfPluginNetworkSteeringSetConfiguredKey API will be removed in a future software release. Users that wish to use a configured key during network steering should instead add the desired key value into the transient key table before starting network steering. The EUI used for this key should be the wildcard EUI (FFFFFFFFFFFFFFFF).

6 Removed Items

Removed in release 6.7.1.0

SDK no longer contains prebuilt application binaries for EM35x and 256k flash size devices except for the following:

- NCP images(ncp-spi, ncp-uart-hw, ncp-uart-sw) for
  - EM3588
  - efr32mg1p133f256gm48 - brd4150c
  - efr32mg1p232f256gm48 - brd4151a
- Sniff image for EM3588
- xncp-commshub-uart-dual for efr32mg1p132f256gm48 – brd4155a

Applications may still be generated and built via Studio.

Removed in release 6.7.0.0

In previous releases, source routing (add/remove table) and concentrator code (scheduling many-to-one route requests, handling route error) previously existed as 2 different copies, one host side plugin version and one NCP version. The host side source routing table and MTORR scheduling have now been removed.

To enable source routing, include both the concentrator plugin (configuration part) and the source routing library (the core table management and MTORR scheduling functionality).

As a side effect most of the APIs/callbacks allowing the app to interfere with the source routing has been deprecated. Specifically, the ones that depended on the host-side source-route table, including:

- emberAppendCachedSourceRouteFromHost()
- ezspSetSourceRoute()
- emberFindSourceRoute()
- emberAfEzspSetSourceRoute()
- emberAfEzspSetSourceRouteCommandCallback()
- setConcentrator/ezspsetconcentrator()
- emberAfPluginEzspZigbeeProSetConcentratorCommandCallback()
- emberAfPluginConcentratorPrintHostSourceRouteTable() and print-host-table CLI command.

7 Using This Release

This release contains the following:

- Zigbee stack v6.7.3.0
- Zigbee Application Framework v6.7.3.0
- Zigbee Sample Applications

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 Launcher Perspective, under this SDK's **Release Notes** doc header.

For more information about Zigbee and the EmberZNet SDK see UG103.02: Zigbee Fundamentals.

If you are a first-time user, see QSG106: Getting Started with EmberZNet PRO, for instructions on installing and configuring your development environment, building and flashing a sample application, and documentation references pointing to next steps.

7.1 Installation and Use

Stack installation instruction are covered in QSG106: Getting Started with EmberZNet PRO.

Use the EmberZNet SDK with the Silicon Labs Simplicity Studio V4 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/.

7.2 Support

Development Kit customers are eligible for training and technical support. You can use the Silicon Laboratories Zigbee web page to obtain information about all Silicon Labs Zigbee 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 and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System. 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.

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®, ISOmodem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, Z-Wave and others are trademarks or registered trademarks of Silicon Labs.

ARM, CORTEX, Cortex-M0+, Cortex-M3, Cortex-M33, Cortex-M4, TrustZone, Keil and Thumb are trademarks or registered trademarks of ARM Holdings.

Zigbee® and the Zigbee logo® are registered trademarks of the Zigbee Alliance.

Bluetooth® and the Bluetooth logo® are registered trademarks of Bluetooth SIG Inc.

All other products or brand names mentioned herein are trademarks of their respective holders.