Z-Wave 700 SDK 7.13.10.0 GA
Gecko SDK Suite 2.7
August 18, 2021

Z-Wave 700 is designed to meet the demands of the future smart home, where increasing needs for more sensors and battery-operated devices require both long range and low power. Context-aware environments are the next evolution in the smart home market, and they require technologies that have been optimized specifically for these applications.

100% Interoperable: Every product in the Z-Wave ecosystem works with every other product, regardless of type, brand, manufacturer or version. No other smart home/IoT protocol can make this claim.

Best-In-Class Security: Z-Wave's Security 2 (S2) framework provides end-to-end encryption and the most advanced security for smart home devices and controllers. Homes with S2 Z-Wave devices are virtually un-hackable.

SmartStart Easy Installation: SmartStart radically simplifies the installation of smart devices by using QR code scans for uniform, trouble-free setup. Devices and systems can be pre-configured dramatically easing deployments.

Backwards-Compatible: Z-Wave certification mandates backward-compatibility. The very first Z-Wave devices on the market, more than ten years old still perform as intended in networks with the latest Z-Wave technologies.

The Z-Wave 700 SDK v7.13.10 GA release is intended for development of Z-Wave-certifiable, 700-based products entering volume production. This release contains underlying platform changes only. Notice: The Z-Wave 700 SDK v7.13.0 was a beta release and therefore could not be used for Z-Wave certification, see section 6 - Product Life Cycle and Certification.

These release notes cover SDK version(s):

7.13.10.0 released August 18, 2021 (underlying platform changes only)
7.13.9.0 released March 3, 2021
7.13.8.0 released October 28, 2020
7.13.7.0 released August 12, 2020
7.13.6.0 released May 27, 2020
7.13.5.0 released April 29, 2020
7.13.4.0 released April 15, 2020
7.13.3.1 released March 27, 2020
7.13.3.0 released March 20, 2020
7.13.2.0 released February 21, 2020
7.13.1.0 released January 24, 2020
7.13.0.0 released December 13, 2019

KEY FEATURES

- Z-Wave certified apps; Door Lock Key Pad, Power Strip, Sensor PIR, Switch On/Off & Wall Controller.
- Support for delayed activation in Firmware Update Meta Data Command Class.
- Support of ZGM130S version N2 SIP (Ordering code ZGM130S037HGN2).
- FLiRS power consumption improved for Korea and Japan

Compatibility and Use Notices

If you are new to the Z-Wave 700 SDK, see Using This Release.

Contents

1 Z-Wave Protocol

1.1 New Items

Added in release 7.13.0.0 Beta

Code execution before entering sleep mode: Several callbacks from the protocol to the framework supported allowing code execution last before entering sleep mode. Up to three callbacks are available. For details refer to function ZAF_PM_SetPowerDownCallback() in ZAF/ApplicationUtilities/PowerManagement/ZAF_PM_Wrapper.h.

SmartStart (6.81.0x+)

SmartStart introduces a number of new APIs for using learn mode and adding nodes to the network. For end nodes, the `ZW_NetworkLearnModeStart()` is now used to control learn mode. For controller nodes, ZW_AddNodeToNetwork() is used.

SmartStart and S2 QR Code Generation

Z Wave devices supporting the Security 2 (S2) Command Class or SmartStart provisioning must provide a QR code physically on the device as well as on packaging. The actual marking and layout requirements are documented in *SDS11847: Z Wave Plus Device Types Specification*.while the data string encoded in the QR code is specified in *SDS13937: Node Provisioning QR Code Format*.

Both the QR code and S2 DSK are generated in the SmartStart device itself and Simplicity Commander facilitates readout of the QR code for printing.

The current SDK release contains two software utilities described in *INS13975: Smart Start Production Control* to assist developers in creating and verifying the contents of a QR code:

- QrCodeEncoder.xlsm
  - Encoding of QR code fields
  - Single-sample generation of QR codes for prototyping
  - Generation of dynamic string for the Production Control File with fields to be replaced during production
- QrCodeDecoder.xlsm
  - Decodes the string contained in a QR code using an arbitrary smart phone QR code scanner application

These utilities are implemented using Excel sheets, incorporating several macro functions for SHA-1 checksum calculation, QR code rendering, and control file generation. The utilities must therefore be stored in a folder that is not write protected. All gray fields in the spreadsheets should be left untouched.

1.2 Improvements

Changed in release 7.13.0.0 Beta

FLiRS enhancement: FLiRS power consumption is reduced to 30 µA in average for Korea and Japan.

File system enhancement: Improve access time to filesystem NVM3 by decreasing the number of files and code optimizations. For example, it is six times faster to add a node to the network than in the previous SDK release.

1.3 Fixed Issues

Fixed in release 7.13.4.0 GA

ID #	Description
448052	During inclusion of a secondary controller in a network with existing FLiRS nodes the neighbor discovery of FLiRS nodes from the secondary controller will time out too fast and prevent the secondary controller from obtaining correct routing information for FLiRS nodes. This timeout will prevent the secondary controller from routing to FLiRS nodes that existed in the network when it was included.
456447	Slave device does not ack incoming frames when not included. Used for test on production line.

Fixed in release 7.13.3.0 GA

ID #	Description
379589	The RSSI values in received frames and in transmit complete events are offset with 10-20dBm compared to the signal strength coming into the radio. The RSSI is offset by ~10dBm at high signal strength and ~20dBm at low signal strength.
463149	The workaround for the chip issue RTCC_E205 in the ZGM130s was not 100% safe and could result in end devices missing timer interrupts and lock up.
466398	Re-transmission timeout on routed Ack was too long because the LBT part of the timeout is included in the random part.
469941	The protocol can enter a situation where an old failing return route attempt is performed before the correct response route attempt. Potential communication latency and battery consumption increased in case the battery-operated device or potential repeater devices are moved substantially. Issue is only present for devices that go into EM4 as the response route step in the overall routing algorithm was not correctly stored in retention registers.

Fixed in release 7.13.2.0 GA

ID #	Description
457708	The retransmit timeout is too short when routing to a FLiRS node that is difficult to communicate with. In this scenario the source node make a new attempt to early colliding with the ongoing attempt. This applies only for 3-ch frequencies (JP and KR).
462766	Improved routed ack handling to reduce retransmissions during S2 communication.

Fixed in release 7.13.1.0 GA

ID #	Description
420261	Request node neighbor update from FLiRS never returns Done.
449613	The controller serialAPI function FUNC_ID_ZW_REQUEST_NODE_NEIGHBOR_UPDATE will not always return a callback when there is a FLiRS node in the network.
450440	Inclusion controller S2 bootstrapping when including FLiRS nodes can sometimes fail.
451884	Serial API Bridge Controller can unintentionally hang when trying to add a node.
452266	Door Lock Key Pad (FLiRS) for region KR and JP (3-channel) do not send battery reports when it wakes up from sleep initiated by timer. It works for EU, US etc. (2-channel).
452873	Serial API Bridge Controller can unintentionally stop sending frames.
453179	UZB-7 NVM is not written correctly during backup recover in case the first 64 bytes are unchanged.

Fixed in release 7.13.0.0 Beta

ID #	Description
363439	Re-transmission rate when sending data to a FLiRS node is too high (around 10%) on 2 channel solutions such as EU, US, etc.
407674	SensorPIR early wakeup phase is much longer than expected.
435611	Add Door Lock Key Pad with S0 as node ID 232 failed.
436168	OTA firmware update fails on 3-channel frequencies for S2 and non-secure. Fragment size used in apps are too big causing buffer overflow and app failure.
449871	SRAM .bss section not initialized to zero.

1.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
355095	In small networks Assign Return Routes will only generate direct range or one hop routes even though multi hop routes are possible.	None
361273	Transport Service is used when it is necessary to split a frame in two parts due to size. However, Transport Service does not forward RSSI information from the lower layers but only routing information. The RSSI value is the difference between LWR RSSI and background RSSI. As a consequence it is not possible to use RSSI for large frames handled by Transport Service in a network health calculation.	None
433582	The supply voltage of the EFR32ZG14 SoC for gateways must be 2.5V or higher. This will ensure stable operation since low noise DCDC conversion is enabled on the SoC instead of DCDC bypass.	None
436188	Priority routes are written to a cache in RAM and not flushed to file system NVM3 when a soft reset is issued.	Host application must always restore priority routes in controller at startup.
436380	Serial API-based controller can seldom reset during SmartStart inclusion in large networks. Seen rarely in networks larger than 40+ nodes.	Important to enable watch dog to recover from SmartStart failure. Host application must set controller in SmartStart mode again to proceed.
448729	Virtual nodes on a bridge controller will ack frames from a foreign homeID when the bridge controller is in the process of adding a node to the network	None

1.5 Deprecated Items

None

1.6 Removed Items

None

2 Z-Wave Plus V2 Application Framework

2.1 New Items

None

2.2 Improvements

Added in release 10.13.1.0 GA

Improved Z-Wave Plus V2 Framework: Added delayed activation functionality in the Firmware Update Meta Data Command Class fulfilling requirements according to version 5. The delayed activation functionality enables programming of a device using a previously transferred firmware image.

Added in release 10.13.0.0 Beta

Improved Z-Wave Plus V2 Framework: The Z Wave Plus V2 Framework is an extension of the well-known Z-Wave Plus certified solutions. It features a selected set of extended features and capabilities that enhance the end user experience and make Z-Wave installations even faster and easier to install and set up.

The Z-Wave Plus V2 requirements are as follows:

- SmartStart is mandatory.
- OTA Firmware update is mandatory.
- Extended CC support for root devices and Multi-Channel End Points. All actuator Device Types must support Basic CC.
- Indicator to identify device such as a visible LED.
- Dynamic capabilities and node discovery. Capabilities may change due to user interaction.
- New controller requirements to strengthen interoperability; for instance, blocking or forced exclusion of non-preferred devices is no longer allowed.
- Minimum CC to be controlled by a controller extended. This applies also for bridging devices interfacing to another technology.
- Detection of Z-Wave Plus V2-compliant nodes using Z-Wave Plus Info CC.

For a detailed description of application development using the Z-Wave Plus V2 Framework, refer to *INS14259: Z-Wave Plus V2 Application Framework SDK7.*

2.3 Fixed Issues

Fixed in release 10.13.9.0 GA

ID #	Description
495621	Fixed a bug regarding Association initialization when doing OTA migration from 7.13.1 to 7.13.8. In version 7.13.4 the format for storing unassigned associations in the ZAF_FILE_ID_ASSOCIATIONINFO file was changed. Unfortunately the necessary code was not added for migrating old files to the new format at that release. That code is necessary to keep track of unused associations when doing OTA firmware update from any earlier version to v 7.13.4 or later. The bug was discovered in June 2020 and a fix was subsequently pushed to v7.14.2 and to v7.15.x. Unfortunately the fix was not pushed to later patches of v7.13.x like v7.13.8.

Fixed in release 10.13.4.0 GA

ID #	Description
471325	Multichannel/Multicast with single cast follow up does not work for all association combinations.
472305	Association Set Command does not set the correct associations. Associations Report Command returns an incorrect result. First and last association is correct but associations in between are all set to the second association. Device fails certification!

ID #	Description
472886	Sample apps build correctly but fail to build GBL files used to make OTA firmware update.

Fixed in release 10.13.1.0 GA

ID #	Description
402207	Wakeup Notification Command Class – Callback function ZCB_WakeUpNotificationCallback() must have the same arguments as pCallback.
452390	The Z-Wave Product Type value remains 4 in the QR code despite changing APP_PRODUCT_TYPE_ID in config_app.h file.
453561	SessionID in Supervision Command encapsulated frames is not incremented.

Fixed in release 10.13.0.0 Beta

ID #	Description
396608	UART1 Tx/Rx PORT/PIN defined incorrectly.
436136	S2 Commands Supported Get does not trigger a correct response depending on the security type of the inclusion.

2.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
369430	All S2 multicast frames are sent using verified delivery S2_TXOPTION_VERIFY_DELIVERY whether or not a response is expected.	Change source code depending on frame sent.
412848	Multichannel association groups works incorrectly when having multiple associations to the same device.	Change source code according to specification.

429745	In CC_Supervision.c the session_id gets increased before Supervision GET is sent. So condition
	if ((supervision_session_id - 1) == pCmd->ZW_SupervisionReportFrame.properties1)
	is never true.
		Replace
		if ((supervision_session_id - 1) == pCmd->ZW_SupervisionReportFrame.properties1)
		with
		if (supervision_session_id == pCmd->ZW_SupervisionReportFrame.properties1)

2.5 Deprecated Items

None

2.6 Removed Items

None

3 Certified Applications

3.1 Door Lock Key Pad

3.1.1 New Items

Added in release 10.13.0.0 Beta

Current Consumption of Door Lock Key Pad in Sleep Mode: The current consumption of Door Lock Key Pad FLiRS device is typical 19 µA on average. The configurations of the FLiRS device is configured as follows:

- 2-channel frequency, such as EU, US, etc.
- Wakeup interval of 1000 ms
- No communication

3.1.2 Improvements

None

3.1.3 Fixed Issues

Fixed in release 10.13.1.0 GA

ID #	Description
449819	The Door Lock Key Pad application disables the ADC which in turn makes the protocol layer set the DCDC converter to pass mode and thereby increase power consumption.
450070	Door Lock Configuration Set Command with Supervision should return FAIL when setting unsupported components in the command parameters.
450243	The Door Lock Operation Set command handler incorrectly modifies the Outside- and Inside Door Handles Mode values. Those values should only be set by the Door Lock Configuration Set command.
451858	The Door Lock Key Pad responds with a Door Lock Operation Report Command after a locally initiated lock/unlock operation is incomplete. The last two bytes of the fields "Target Door Lock Mode" and "Duration" are missing.
456219	When Door Lock Key Pad receives a DoorLock Operation Get Command then it must report 0 for the Door Handles Mode if the Door Lock Mode is secured (0xFF).

Fixed in release 10.13.0.0 Beta

ID #	Description
448027	The Door Lock Key Pad application does not support inside door handles. Hence, the Door Lock Operation Set command should not alter these values.

3.1.4 Known Issues in the Current Release

None

3.1.5 Deprecated Items

None

3.1.6 Removed Items

None

3.2 Power Strip

3.2.1 New Items

None

3.2.2 Improvements

None

3.2.3 Fixed Issues

Fixed in release 10.13.1.0 GA

ID #	Description
455109	PowerStrip doesn't send the Alarm Event Notification Reports to nodes added in the EP 1 and EP 2 association groups.

Fixed in release 10.13.0.0 Beta

ID #	Description
434820	DUT must allow a command to go through with a higher security level. Both a S2-Access Msg Encap [S0 Security Commands Supported Get] and a S0 Msg Encap [S0 Security Commands Supported Get] must return a S2-Access Msg Encap [S0 Security Commands Supported Report (empty list)].
436136	When the DUT is included non-securely, the endpoint capability report does not advertise the S0 Security CC but it is still listed in the NIF for the root device.

3.2.4 Known Issues in the Current Release

None

3.2.5 Deprecated Items

None

3.2.6 Removed Items

None

3.3 Sensor PIR

3.3.1 New Items

None

3.3.2 Improvements

None

3.3.3 Fixed Issues

Fixed in release 10.13.1.0 GA

ID #	Description
450403	Sensor PIR doesn't extend the stay awake period by 10 seconds when receiving a Request Node Info frame.
452674	The macro DISABLE_UART0 is missing in the Simplicity Studio Sensor PIR project causing power consumptions up to 50 µA in EM2 sleep.

Fixed in release 10.13.0.0 Beta

ID #	Description
407674	Sensor PIR wakeup phase is much longer than expected. Measurements show that the initial wakeup phase is prolonged by ≈300 ms. It should not take more than ≈10 ms.

3.3.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
386208	Sensor PIR does not always generate SHORT_PRESS events for short button presses in EM4. If the device has been awakened by a button press, the button handling logic starts by looking at the current state of the button. If the button is UP, a SHORT_PRESS event is immediately sent to the application. If the button is DOWN, then the de-bouncing logic is triggered to properly generate HOLD or LONG_PRESS events as needed. However, this leaves a tiny timing window where the button could be DOWN when initially tested, but is released before the DOWN time required for the de-bouncing logic to generate even a SHORT_PRESS event.	Prolong button press to allow detection of SHORT_PRESS by the de-bouncing logic.

3.3.5 Deprecated Items

None

3.3.6 Removed Items

None

3.4 Switch On/Off

3.4.1 New Items

None

3.4.2 Improvements

None

3.4.3 Fixed Issues

Fixed in release 10.13.1.0 GA

ID #	Description
450496	Long press on BTN0 on the BRD8029A Button Board triggers watchdog reset.

3.4.4 Known Issues in the Current Release

None

3.4.5 Deprecated Items

None

3.4.6 Removed Items

None

3.5 Wall Controller

3.5.1 New Items

None

3.5.2 Improvements

None

3.5.3 Fixed Issues

None

3.5.4 Known Issues in the Current Release

None

3.5.5 Deprecated Items

None

3.5.6 Removed Items

None

4 Serial API Bridge Controller

Unchanged serial interface version 8.

4.1 New Items

None

4.2 Improvements

None

4.3 Fixed Issues

None

4.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
387655	Pre-built SerialAPI delivered in Simplicity Studio will not work if the ZG14 bootloader is also flashed to the radio board.	Use serialAPI withput bootloader or, if OTW support is needed, contact the Z-Wave Apps team for workaround.

4.5 Deprecated Items

None

4.6 Removed Items

None

5 Using This Release

This release contains the following

- Z Wave Plus V2 Application Framework
- Z-Wave Certified Applications for a broad range of smart home applications
- Z-Wave Protocol and Serial API Applications

If you are a first-time user, Z-Wave documentation is installed with the SDK. See INS14280: Z-Wave 700 Getting Started for End Devices, INS14278: How to Use Certified Apps in Z-Wave 700, and INS14281: Z-Wave 700 Getting Started for Controller Devices for instructions.

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.

5.1 Installation and Use

Order a Z-Wave 700 Wireless Starter kit. The kit offers the easiest and fastest way to start evaluation and development of your own Z-Wave 700 mesh application. It provides a single world-wide development kit for both end devices and gateways with multiple radio boards, to enable developers to create a mesh network and evaluate the Z-Wave 700 module.

Download and install Simplicity Studio from https://www.silabs.com/support/getting-started/mesh-networking/z-wave/z-wave-700. Simplicity Studio ensures that most software and tool compatibilities are managed correctly. Install software and board firmware updates promptly when you are notified.

After Simplicity Studio installs, select **Install By Product Group**, check **Z-Wave**, and follow the steps to install the SDK.

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

To implement a specific application, Silicon Labs recommends starting with one of the existing pre-certified apps with the desired Role Type.

5.2 Support

Development Kit customers are eligible for training and technical support.

See support resources and contact Silicon Laboratories support at http://www.silabs.com/support.

6 Product Life Cycle and Certification

Silicon Labs will add new features based on market requirements and continuously improve the Z-Wave Protocol to position the Z-Wave Ecosystem. The Z-Wave Protocol Life Cycle is a process to provide rapid innovation, new features and robust matured protocol release to Z-Wave Partners. The Z-Wave Protocol Life Cycle defines the maturation process of Z-Wave Protocol generations and consist of three phases divided in five Life Cycle stages.

Ascent Phase (BETA)

Silicon Labs releases new Z-Wave protocol generations (branches), i.e. initial BETA release of a Z-Wave Protocol generation that will introduce major new features/functions or support for a new Z-Wave Single Chip generation. This release is not certified and not eligible for certification.

Maturity Phase (ACTIVE/MAINTAINED)

Each new generation will generate follow on matured releases to resolve protocol issues prioritized by Silicon Labs and based on input from Z-Wave Alliance Partners.

Decline Phase (MONITORED/OBSOLETE)

After a period of 17-24 months in the maturity phase a branch/release is discontinued and for an additional period (up to 24 months) a discontinued branch/release will be monitored since products based on this branch may still be shipping or under warranty in the field.

Table 6-1. Z-Wave SDK Life Cycle Status

Series	Branch	SDK Version	Release Date [DD/MM/YYYY]	Life Cycle Status
700	7.1x.x	7.13.10 GA	18-AUG-2021	Active
		7.13.9 GA	03-MAR-2021	Maintained
		7.13.8 GA	28-OCT-2020	Maintained
		7.13.7 GA	12-AUG-2020	Maintained
		7.13.6 GA	27-MAY-2020	Maintained
		7.13.5 GA	29-APR-2020	Maintained
		7.13.4 GA	15-APR-2020	Maintained
		7.13.3 GA	20-MAR-2020	Maintained
		7.13.2 GA	21-FEB-2020	Maintained
		7.13.1 GA	24-JAN-2020	Maintained
		7.13.0 Beta	13-DEC-2019	Obsolete
		7.12.2 GA	26-NOV-2019	Maintained
		7.12.1 GA	20-SEP-2019	Obsolete
		7.11.1 GA	12-JUL-2019	Maintained
		7.11.0 GA	29-MAR-2019	Monitored

A change in the Z-Wave SDK utilized for a specific device does require recertification; however, the type of certification required, the amount of testing needed, and the associated fees depend on the scope of the change.

Table 6-2. Z-Wave Certification in case of a SDK upgrade.

SDK Version	Upgrade to SDK Version	Type of Certification
7.13.10 GA	NA	-
7.13.9 GA	7.13.10 GA	Re-certification
7.13.8 GA	7.13.9 GA	Re-certification
7.13.7 GA	7.13.8 GA	Re-certification
7.13.6 GA	7.13.7 GA	Re-certification
7.13.5 GA	7.13.6 GA	Re-certification
7.13.4 GA	7.13.5 GA	Re-certification
7.13.3 GA	7.13.4 GA	Re-certification
7.13.2 GA	7.13.3 GA	Re-certification
7.13.1 GA	7.13.2 GA	Re-certification
7.13.0 Beta	7.13.1 GA	Full certification
7.12.2 GA	7.13.1 GA	Re-certification
7.12.1 GA	7.13.1 GA 7.13.0 Beta 7.12.2 GA	Re-certification NA Re-certification
7.11.1 GA	7.13.1 GA 7.13.0 Beta 7.12.2 GA 7.12.1 GA	Re-certification NA Re-certification NA
7.11.0 GA	7.13.1 GA 7.13.0 Beta 7.12.2 GA 7.12.1 GA 7.11.1 GA	Re-certification NA Re-certification NA Re-certification

7 Legal

1. Disclaimer

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

2. Trademark Information

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