EmberZNet SDK 6.0.0.0 GA

Silicon Laboratories, Inc.

October 26, 2017

1 Release Highlights

1.1 Version 6.0.0.0 GA:

- Dynamic Multi Protocol support for Zigbee/BLE
- Dynamic Multi Protocol Sample Application
- Gecko Bootloader with Compression
- Green Power updates for Zigbee 3.0 and Green Power Proxy support
- Support for EFR32xGM13P parts
- Support for EFR32xG14 family of parts
- MISRA compliance for all Zigbee stacks and sample applications
- GA release of GB868 (Sub-GHz Zigbee) for Great Britian Smart Metering
- Integration of the Hardware Configurator.
- GCC compiler support for Zigbee and the EFR32
- Zigbee support for the Micrium OS Kernel
- Zigbee + Micrium sample application
- Support for Zigbee default reporting configurations

2 Using This Release

This release contains the following

- Zigbee stack EmberZNet Pro v6.0
- Zigbee Application Framework v6.0
- Zigbee Sample Applications

For more information about the Silicon Labs EmberZNet Pro stack see UG103.02 - Zigbee Fundamentals. If you are a first time user, see QSG106: Getting Started with EmberZNet Pro.

2.1 Support

Development Kit customers are eligible for training and technical support. You can use the Silicon Laboratories web site http://www.silabs.com 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

3 Added Items

3.1 Version 6.0.0.0 GA:

3.2 New Application Plugins:

3.3 New Stack APIs:

The following is a list of new APIs included with this release. A brief description is given for each.

- emberGetDutyCycleState
  Obtains the current duty cycle state.

- emberSetDutyCycleLimitsInStack
  Set the current duty cycle limits configuration. The Default limits set by stack if this call is not made.

- emberGetDutyCycleLimits
  Obtains the current duty cycle limits that were previously set by a call to emberSetDutyCycleLimitsInStack(), or the defaults set by the stack if no set call was made.

- emberGetCurrentDutyCycle
  Returns the duty cycle of the stack's connected children that are being monitored, up to maxDevices. It indicates the amount of overall duty cycle they have consumed (up to the suspend limit). The first entry is always the local stack's nodeId, and thus the total aggregate duty cycle for the device. The passed pointer arrayOfDeviceDutyCycles MUST have space for maxDevices.

- emberDutyCycleHandler
  Callback fires when the duty cycle state has changed. This callback is only on SoC.

- ezspDutyCycleHandler
  Callback fires when the duty cycle state has changed. This callback is only on NCP.

- emberMultiPhyStart
  This causes to initialize the desired radio interface other than native and form a new network by becoming the coordinator with same panId as native radio network.

- emberMultiPhyStop
  This causes to bring down the radio interface other than native.

- emberMultiPhySetRadioPower
  Sets the radio output power at which a node is to operate for the desired phy interface.

- emberMultiPhySetRadioChannel
  Sets the channel to use for sending and receiving messages on the desired phy interface.

- emberGetPhyInterfaceCount
  Returns number of phy interfaces present.

- emberSendLinkPowerDeltaRequest
  Sends a link power delta request to the parent.

- emberGetRadioParameters
  Copies the current radio parameters into the structure provided by the caller for desired phy interface.

3.3.1 New Sample Applications:

- Two new sample applications have been added. The DynamicMultiprotocolDemoLight is a zigbee 3.0 light using dynamic multiprotocol (BLE and zigbee). The DynamicMultiprotocolDemoSwitch is a zigbee 3.0 switch that is intended to work with the DynamicMultiprotocolDemoLight. Please read their respective descriptions for more information.

- The xNCP CommsHub Dual is a demonstration of a network coprocessor application for simultaneous dual phy device (2.4Ghz and subghz bands) that communicates to a host application over a UART interface.

4 Changed Items

4.1 Version 6.0.0.0 GA:

- Trust Center (insecure) rejoins for devices using the well-known link key are now rejected by the stack by default. Sending the network key encrypted with the well-known key is a security hole and, as such, the default behavior of the stack is to not answer those rejoin requests.

  – A new API has been created to allow Trust Center rejoins for devices using the well-known key. emberSetTcRejoinsUsingWellKnownKeyAllowed() or EZSP command ID EZSP_TC_REJOINS_USING_WELL_KNOWN_KEY_POLICY can be used to set the policy for Trust Center rejoins using the well-known key. If enabling rejoins, a timer will be set which will disable those rejoins after timing out. A user can alter the timeout by overriding emAllowTcRejoinsUsingWellKnownKeyTimeoutSec or with EZSP command ID EZSP_CONFIG_TC_REJOINS_USING_WELL_KNOWN_KEY_TIMEOUT_S. A timeout value of zero will not arm the timer and keep the rejoin policy as configured. If issuing the EZSP command ID to change the timeout, the command must be issued prior to forming a network.

- Changed the API for emberCounterHandler and emberAfCounterHandler. The new argument ïnfoc̈ould be mapped to:

  1. data: For transmission events, the number of retries used. For most other events, this parameter is unused and is set to zero. This parameter is always available as ïnfo.data.̈
  2. phyIndex: used for mac specific counters identifying if they belong to SubGhz or 2.4. Depending on the type of the counter (emberCounterRequiresPhyIndex(type) and/or emberCounterRequiresDestinationNodeId(type)) this field could be accessed via one of the following:

     (a)

     $$((EmberExtraCounterInfo)(info.otherFields)) \rightarrow phy\_index$$

     (b)

     $$((uint8\_t)info.otherFields)$$

  3. destinationNodeId: identifying which connection/destinationId a specific counter refers to. Depending on the type of the counter (emberCounterRequiresPhyIndex(type) and/or emberCounterRequiresDestinationNodeId(type)) this field could be accessed via one of the following:

     (a)

     $$((EmberExtraCounterInfo)(info.otherFields)) \rightarrow destinationNodeId$$

     (b)

     $$((uint8\_t)info.otherFields)$$

  4. Any combination of parameters above.

- Changed the API for emberGetChildData to return new EmberChildData struct
  - New Signature: emberGetChildData(uint8 index, EmberChildData *data)
  - EmberChildData fields:
    - EmberEUI64 eui64; – The eui64 address of the child
    - EmberNodeType type; – The node type of the child
    - EmberNodeId id; – The short address of the child
    - uint8_t phy; – The phy the the child is operating on
    - uint8_t power; – The power of the network the child is on
    - uint8_t timeout; – The timeout duration of the child in seconds
- The networkInit API has changed in order to comply with Z3 standards
  - 'emberNetworkInitExtended(EmberNetworkInitStruct* emberNetworkInitStruct)' has been deprecated
  - 'emberNetworkInit()' is now 'emberNetworkInit(EmberNetworkInitStruct* emberNetworkInitStruct)'
  - EmberNetworkInitStruct contains one field named EmberNetworkInitStructBitmask. This bitmask now contains an additional bit named EMBER_NETWORK_INIT_END_DEVICE_REJOIN_ON_REBOOT. This can be set manually – by default this is set when Z3 security is enabled.

5 Deprecated Items

5.1 Version 6.0.0.0 GA:

- emberNetworkInitExtended(EmberNetworkInitStruct* emberNetworkInitStruct)
  - The functionality was moved to the function 'emberNetworkInit'

6 Removed Items

6.1 Version 6.0.0.0 GA:

- Support for the EM341, EM342, and EM346 has been removed
- Support for RF4CE has been removed

7 Fixed Issues

7.1 Version 6.0.0.0 GA:

- 113213 The Request Block Delay value on an OTA client is now read correctly if the source EUI is present in the Image Block Response.
- 243463 Serial bootloading via XModem may be unreliable when performed through WSTK USB connection.
- 244182 Update ZCL metadata for device types and required clusters so they match the latest ZCL revision
- 260134 ZLL stack should reject broadcast Interpan commands besides Scan Request
- 260995 Building zigbee applications with PS Store plugin in IAR IDE results in build error about missing psstore-cli.c file

- 261230 ”Trust Center (insecure) rejoins for devices using the well-known link key are now rejected by the stack by default. Sending the network key encrypted with the well-known key is a security hole and, as such, the default behavior of the stack is to not answer those rejoin requests.
  A new API has been created to allow Trust Center rejoins for devices using the well-known key. emberSetTcRejoinsUsingWellKnownKeyAllowed() or EZSP command ID EZSP_TC_REJOINS_USING_WELL_KNOWN_KEY_POLICY can be used to set the policy for Trust Center rejoins using the well-known key. If enabling rejoins, a timer will be set which will disable those rejoins after timing out. A user can alter the timeout by overriding emAllowTcRejoinsUsingWellKnownKeyTimeoutSec or with EZSP command ID EZSP_CONFIG_TC_REJOINS_USING_WELL_KNOWN_KEY_TIMEOUT_S. A timeout value of zero will not arm the timer and keep the rejoin policy as configured. If issuing the EZSP command ID to change the timeout, the command must be issued prior to forming a network.”
- 272447 It has been observed that the sub-ghz network may see numerous timeouts and retries during certain periods of heavy traffic, such as CBKE or joining. It is possible that this may impact the a device’s ability to join successfully to the network.
- 274954 Resuming ZCL OTA downloads does not work when using a Gecko bootloader slot as the storage medium. Functionality for this configuration will be incorporated into a future release. Until functionality is added, if a device using a slot as the OTA storage medium resets during the download process, it will wipe the OTA storage area upon startup and begin the process from the beginning.
- 275407 EFR32 SPI NCP doesn’t sleep (in EM2) when expected
- 275718 A Z3 Sample Application does not automatically send a NWK Rejoin after reboot, which is required by Zigbee compliance testing. The application can manually initiate a NWK Rejoin as a workaround.
- 275809 XNCP Sample Applications for EFR32 UART NCPs don’t work properly when built with GCC. Workaround: Use IAR EWARM toolchain for now, or use pre-built binaries.
- 275885 Custom NCP builds with dual PHY settings don’t build properly with GCC toolchain. Workaround: Use IAR EWARM toolchain for now.
- 276009 nWAKE handshake on EFR32 SPI NCP causes NCP to wake but fall right back to sleep
- 276743 Z3Switch sample app may occasionally indicate Find And Bind Initiator error as ”Find and bind initiator complete: 0x01”. If this happens, enable Identify mode (initiate Find And Bind Target) on the target node and retry the Find And Bind process at the initiator.
- 277273 When opening a new xncp-led Sample App it will choose the wrong board by default for the EM3588. The board chosen is dev0680etm.h. The correct header is dev0680uart.h. This will prevent the image from compiling correctly for the CEL EM3588 USB stick. The workaround is to manually select the correct board header and to select the EZSP-UART plugin and set the option for ”Software Flow-control”.
- 277374 Pre-built NCP images are unable to properly launch the standalone bootloader for EFR32 and EM3xx parts. The workaround is to use the xNCP sample app to generate an image using Simplicity Studio.
- 277530 EFR32xG13 may encounter issues launching the standalone bootloader in an NCP uart configuration. This may be due to the NCP still processing UART packets and being unable to respond to the host. There is no workaround at this time.
- 277756 Host and NCP connections show signs of instability when forming a network. Occasionally, EZSP errors are printed on the host. Key establishment sometimes fails with the NCP application not responding to key establishment messages.

- 277799 Both IAR and GCC-related project files will be created for AFV2 (ZCL Framework) SOC applications, even if one one of those toolchains is selected in the project wizard. (Does not impact ability to build for either toolchain; active toolchain selection may still be changed via Edit Architecture button.)
- 277837 An assert was seen using the Gecko Bootloader with compression on the EFR32xG13 or EFR32xG12 when flashing an application and Gecko Bootloader onto the device.
- 279380 Secure EZSP plugin should not be selectable for EFR32 NCP builds, where it is not supported.

8 Open Issues

8.1 EmberZNet Open Issues:

- 108582 For the ZigBee Over-the-air Cluster the minBlockRequestPeriod is in milliseconds but compared to imageBlockRequestMinRequestPeriodSeconds as if they are both seconds. Unfortunately this causes problems for servers that make use of this value and try to throttle the rate of the client. Recommend that the client disables support for this feature to avoid problems.
- 277855 When the PTA feature ”Synch MAC to Grant” is enabled asserts have been observed during joining. The workaround is to disable this feature for now.
- 280633 Default plugin option values for ”OTA Simple Storage EEPROM Driver” plugin have changed: ”OTA Storage End Offset” has changed from 204800 to 262144, ”EEPROM Device Read-modify-write Support” has changed from TRUE to FALSE. If an older application uses this plugin, please verify these settings.
- 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
- 281283 We are seeing intermittent connectivity issues in the SubG bands for Zigbee. These appear mostly during joining, primarily on the 868mhz bands. We are currently working on trying to reproduce and troubleshoot this issue.

9 Intended Behavior

10 Documentation Changes

11 History